FACULTY OF ENGINEERING AND ARCHITECTURE

Spatiotemporal Event Detection for Real-Time Multi-Domain Applications

Jelle Vanhaeverbeke

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Electronics and ICT Engineering Technology

Supervisors
Prof. Steven Verstockt, PhD - Prof. Sofie Van Hoecke, PhD
Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

April 2025

GHENT UNIVERSITY

ISBN 978-94-6355-974-4
NUR 965
Wettelijk depot: D/2025/10.500/34

Members of the Examination Board

Chair

Prof. Hennie De Schepper, PhD, Ghent University

Other members entitled to vote

Prof. Hiep Luong, PhD, Ghent University
Prof. Glenn Van Wallendael, PhD, Ghent University
Florian Vandecasteele, PhD, Hulpverleningszone Fluvia
Prof. Tim Verdonck, PhD, Universiteit Antwerpen

Supervisors

Prof. Steven Verstockt, PhD, Ghent University
Prof. Sofie Van Hoecke, PhD, Ghent University

Acknowledgements

Looking back at my childhood, I remember learning programming languages such as Visual Basic and Flash, spending hours developing side projects simply for the joy of creating. While these technologies have since been replaced by more modern alternatives, they sparked my passion for technology and problem-solving, profoundly shaping the journey of my life. One of those important moments was deciding to pursue a master's degree in Electronics and ICT Engineering, which further deepened my fascination for technology, especially the possibilities of artificial intelligence. This ultimately led me to my master's thesis on predictive healthcare under the supervision of Prof. Sofie Van Hoecke. Though I hadn't initially considered research as a career path, Sofie encouraged me to join the research group, allowing me to work within both her team and that of Prof. Steven Verstockt.

First and foremost, I want to express my gratitude to my supervisors, Prof. Steven Verstockt and Prof. Sofie Van Hoecke. Thank you for giving me the opportunity to do what I love most, exploring and researching new technologies while solving real-world challenges. From working with X-rays in a concrete bunker to demonstrating our research on stage at the Nerdland Festival, you opened doors to experiences I could never have imagined. These projects not only provided unique opportunities but also allowed me to meet inspiring individuals and grow both personally and professionally. Every day brought something new, and for that, I am incredibly grateful.

Beyond my supervisors, I had the privilege of being part of two incredible teams: IMPACT and PreDiCT. On a daily basis, I worked closely within the IMPACT team, which initially included Dilawar, Jelle (yes, two Jelles!), Kenzo, and Krishna. Over time, the team expanded with Alec, Dieter, Joachim, Maarten, Robbe, Thomas, and Winter. Despite the growth, the strong bond among colleagues always remained intact. Thank you for the great conversations, the fun we shared, and the support through all the highs and lows of both research and life. I also want

to thank the larger PreDiCT team, where I was always welcomed with open arms for interesting meetings, team activities, and day-to-day discussions. While much of our research could be done from home, the warmth of these teams made me look forward to come to the office. It was a genuine pleasure to work alongside all of you.

Unexpectedly, my research journey led me to meet colleagues beyond my immediate teams. It all started when my childhood friend Thomas, also a researcher at UGent, invited me to join a "happy hour" at the university. There, I was introduced to the amazing group of PP10 (Department of Special Needs Education), who made me feel welcome despite being an outsider to their field. You all brought so much warmth, joy, and camaraderie to this journey, and I can't imagine what I would have missed without you. Thank you for all the genuine conversations, shared experiences, and incredible parties!

I am also deeply grateful for the phenomenal group of friends I've been part of for over a decade, known as "de zotte feestbende". Together, we've shared so many unforgettable moments, from attending our first festivals to celebrating the arrival of the first "bendebaby". Even after all these years, our friendship remains as strong as ever. The joy of being together, shared laughter, and heartfelt conversations mean the world to me. Thank you for all the memories we've created, and here's to many more years of friendship to come!

Finally, but by no means least, I want to thank my mom, sister, and brother. Over the years, you've seen me buried in work at my desk, often without fully knowing what I was working on or why it seemed endless. Yet, you always provided unwavering support, pulling me away from my computer when I needed a break and offering a calm and loving home environment that helped me persevere. Thank you for being my foundation and for standing by me every step of the way.

To all of you, my supervisors, colleagues, friends, and family, this journey would not have been possible without your incredible support, encouragement, and belief in me. You have all contributed to shaping not only my career but also the person I am today. For that, I will forever be grateful. Thank you!

Pittem, April 2025
Jelle Vanhaeverbeke

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

AI	Artificial Intelligence

B

BA	Balanced Accuracy
BIM	Building Information Model
BMS	Building Management System

C

CART	Classification and Regression Trees
CFD	Computational Fluid Dynamics
CLIP	Contrastive Language–Image Pre-training
CNN	Convolutional Neural Network
CO_2	Carbon Dioxide
COPD	Chronic Obstructive Pulmonary Disease

COVID-19 Coronavirus Disease 2019

D

DINO Self-Distillation with No Labels

DNN Deep Neural Network

E

ECG Electrocardiogram

EHR Electronic Health Records

F

FFT Fast Fourier Transform

FN False Negative

FP False Positive

FPS Frames per Second

G

GIGO Garbage In, Garbage Out

GMM Gaussian Mixture Model

GPS Global Positioning System

GRU Gated Recurrent Units

H

HVAC Heating, Ventilation, and Air Conditioning

I

ICA Independent Component Analysis

IoT Internet of Things

IoU Intersection over Union

K

KCF Kernelized Correlation Filters

L

LIME Local Interpretable Model-agnostic Explanations

LSTM Long Short-Term Memory

LTE Long-Term Evolution

M

MAE Mean Absolute Error

ML Machine Learning

MLE Maximum Likelihood Estimation

MPC Model Predictive Control

MSE Mean Squared Error

N

NDIR Nondispersive Infrared

NLP Natural Language Processing

NN Neural Network

NO_2 Nitrogen Dioxide

P

PCA principal Component Analysis

PCK Percentage of Correct Keypoints

PIR Passive Infrared

POI Point of Interest

PPM Parts per Million

R

R-CNN	Region-based Convolutional Neural Network
RAM	Random Access Memory
RG	Research Goal
RNN	Recurrent Neural Network

S

SAM	Segment Anything Model
SDI	Serial Digital Interface
SHAP	Shapley Additive Explanations
SIFT	Scale-Invariant Feature Transform
SO_2	Sulfur Dioxide
SPARQL	SPARQL Protocol And RDF Query Language
SSN	Semantic Sensor Network

T

TP	True Positive
TSF	Temporal Shift Features
TVOC	Total Volatile Organic Compounds

U

UCI Union Cycliste Internationale

UDP User Datagram Protocol

UNISAL Unified Image and Video Saliency Modeling

V

ViT Vision Transformer

VLS Viral Load Survival

Y

YOLO You Only Look Once

Samenvatting
– Summary in Dutch –

In de huidige datagedreven wereld is het detecteren van gebeurtenissen belangrijk in diverse sectoren, variërend van industriële productie en gezondheidszorg tot stedelijke planning en veiligheid. Dergelijke gebeurtenissen bieden waardevolle inzichten over veranderingen binnen een omgeving die mogelijk een directe reactie vereisen of relevant zijn voor verdere analyse. Het observeren van gebeurtenissen in de echte wereld, zoals het identificeren van een technische storing in een windturbine of het detecteren van een val tijdens een wielerwedstrijd, is echter vaak complex en is meestal niet te automatiseren met eenvoudige methoden. Hierdoor is het monitoren van gebeurtenissen traditioneel afhankelijk van menselijke supervisie, een aanpak die niet alleen tijdrovend en foutgevoelig is, maar ook steeds minder houdbaar gezien de groeiende vraag naar efficiëntie en schaalbaarheid.

Dankzij de vooruitgang in sensortechnologie en dataverwerking zijn er de laatste jaren nieuwe mogelijkheden ontstaan voor het automatiseren van gebeurtenisdetectie. Door het gebruik van gegevensbronnen zoals videobeelden en sensormetingen is het nu wel mogelijk om waardevolle informatie automatisch te extraheren en patronen te herkennen, zoals bosbranden in camerabeelden of machinestoringen via trillingsgegevens. Hoewel academisch onderzoek op dit gebied aanzienlijke vooruitgang heeft geboekt, zijn veel voorgestelde oplossingen gebaseerd op gecontroleerde, grootschalige datasets en gaan ze voorbij aan de praktische uitdagingen in de echte wereld. Dit proefschrift probeert deze kloof te overbruggen door zich te richten op zowel het onderzoek als de implementatie van methoden voor gebeurtenisdetectie in realistische omgevingen. Voor vier casestudies worden openstaande uitdagingen onderzocht, methodologieën ontwikkeld en geïllustreerd hoe deze methodologieën praktische problemen kunnen aanpakken, zoals beperkte data, de behoefte aan realtimeverwerking en wisse-

lende omgevingsfactoren. Tegelijkertijd wordt aangetoond hoe het integreren van ruimtelijke context bijdraagt aan diepgaandere inzichten en praktisch toepasbare resultaten.

De eerste casestudy, gepresenteerd in hoofdstuk twee, richt zich op gebeurtenisdetectie op machineniveau binnen de staalindustrie, uitgevoerd in samenwerking met een grote staalproducent. Een stabiele en gecontroleerde vlamproductie in staalovens is essentieel om de kwaliteit van het staal te waarborgen. Daarom is controle van de vlammen cruciaal om gebeurtenissen te detecteren die wijzen op een afwijkende toestand binnen de oven. Hiervoor is een computervisiegebaseerd systeem onderzocht dat gebruik maakt van een thermische camera die de binnenkant van de staaloven monitort. Het systeem hanteert een hybride aanpak, waarbij traditionele computervisiemethoden worden gecombineerd met *machine learning*, wat zorgt voor zowel nauwkeurigheid als uitlegbaarheid. Belangrijke functionaliteiten omvatten vlamdetectie, kwantificatie in branderzones en anomaliedetectie. De oplossing is gevalideerd met *real-world* data, verzameld onder uiteenlopende operationele omstandigheden, en behaalde een F1-score van meer dan 80% bij de detectie van anomalieën in de verschillende branderzones. De resultaten tonen aan dat deze aanpak effectief onregelmatigheden in de oven detecteert, wat tijdige interventie mogelijk maakt. Om de bruikbaarheid te vergroten, integreert het systeem een dashboard voor realtime monitoring van de ovenstatus. Dit stelt operatoren in staat om anomalieën op te volgen en weloverwogen beslissingen te nemen om de stabiele werking te behouden. De studie benadrukt het potentieel van geautomatiseerde gebeurtenisdetectiesystemen in de industrie, waarmee uitdagingen worden aangepakt zoals beperkte gelabelde data, realtime-vereisten en de behoefte aan uitlegbare oplossingen.

Hoofdstuk drie verlegt de focus van videogebaseerde naar sensorgebaseerde gebeurtenisdetectie, waarbij gebruik wordt gemaakt van CO_2-sensoren die vaak al in moderne gebouwen zijn geïnstalleerd. Deze sensordata wordt onderzocht om gebeurtenissen te detecteren die wijzen op menselijke aanwezigheid in ruimtes, wat waardevol is voor meerdere doeleinden. Op korte termijn maakt het de realtime aansturing van gebouwsystemen mogelijk, zoals het uitschakelen van verlichting of het aanpassen van de verwarming in ongebruikte ruimtes. Op langere termijn faciliteert het de creatie van bezettingsprofielen die inzicht geven in het gebruikspatroon van ruimtes. Deze profielen maken de overgang mogelijk van vaste tijdschema's voor gebouwsystemen naar adaptieve schema's die zich aanpassen aan veranderend gedrag van gebruikers, wat zowel de energie-efficiëntie als het gebruikerscomfort verbetert. Het voorgestelde systeem maakt

gebruik van een machine learning-gebaseerde aanpak om CO_2-metingen te analyseren en te voorspellen of een ruimte bezet is. Een belangrijke uitdaging hierbij is het verzamelen van gelabelde data uit diverse ruimten. Daarvoor introduceert de studie een methodologie die gebruik maakt van *sliding window* normalisatie, waardoor het model kan generaliseren naar nieuwe, onbekende ruimtes zonder extra trainingsdata. Door de toepasbaarheid van dergelijke modellen te verbeteren in ongeziene ruimtes legt dit werk de basis voor meer robuuste gebeurtenisdetectiesystemen in slimme gebouwen. Bovendien toont het hoe automatische gebeurtenisdetectie niet alleen gebruikt hoeft te worden voor onmiddelijke reacties, maar ook kan bijdragen aan langetermijnoptimalisatie.

Hoofdstuk vier vergroot de schaal van sensorgebaseerde gebeurtenisdetectie naar het gebouwniveau en introduceert een realtime *Internet of Things* softwarearchitectuur die het risico op transmissie van COVID-19 inschat. Gebeurtenissen van hoog risico bieden bruikbare inzichten door aanwezigen te waarschuwen, waardoor tijdige interventies mogelijk worden. Een unieke uitdaging is het volledig ontbreken van grondwaarheid met betrekking tot het transmissierisico. Om dit aan te pakken, integreert het systeem expertkennis over aerosoltransmissie met een semantische ontologie waarin relaties worden vastgelegd tussen omgevingsparameters (bv. CO_2-concentratie en temperatuur) en gebouwconcepten. Door SPARQL-querying op de semantische datastroom genereert het systeem interpreteerbare risicoscores en detecteert het automatisch momenten met verhoogd risico. De risicoschattingen worden gepresenteerd via een dynamisch dashboard dat visualisaties biedt op basis van hun semantische context, waaronder een plattegrond die het risico intuïtief weergeeft. Zo bevordert het systeem het bewustzijn onder aanwezigen en gebouwbeheerders, waardoor zij beter geïnformeerde beslissingen kunnen nemen. De aanpak is succesvol uitgerold in drie kantoorgebouwen en geëvalueerd tijdens studentexamens om de effectiviteit van COVID-19 veiligheidsmaatregelen te beoordelen. Zo toonden de resultaten van januari 2021 aan dat de geïmplementeerde veiligheidsmaatregelen het transmissierisico in een grote aula aanzienlijk verminderden. Deze studie illustreert hoe semantische technologieën gebruikt kunnen worden om interpreteerbare, modulaire oplossingen te ontwikkelen. Naast COVID-19-risicoanalyse kan deze aanpak worden ingezet voor andere toepassingen, zoals het monitoren van luchtkwaliteit binnenshuis of het beoordelen van het binnenklimaat.

De laatste studie in hoofdstuk vijf richt zich op de toepassing van visuele gebeurtenisdetectie op regionale schaal, specifiek binnen live-uitzendingen van wielerwedstrijden. Traditioneel voegen video-editors handmatig visualisaties toe

om Points of Interest (POI's), zoals historische gebouwen en monumenten, tijdens deze uitzendingen te markeren. Dit proces is echter repetitief en inefficiënt. Om deze werkwijze te optimaliseren, wordt een nieuw computervisiesysteem voorgesteld, ontworpen om POI's in realtime te herkennen en te volgen. Door gebeurtenissen te identificeren waarbij de uitzending de focus verschuift van de race naar een POI, vermindert het systeem de benodigde handmatige inspanning en biedt het tegelijkertijd een boeiendere kijkervaring. Het systeem gebruikt *saliency*-detectie en het *Segment Anything Model* om mogelijke POI-regio's te genereren. Deze regio's worden vervolgens geverifieerd met behulp van een *keypoint matching*-aanpak, die slechts een paar referentieafbeeldingen vereist. Dankzij deze *few-shot* capaciteit kan het systeem eenvoudig worden aangepast aan de unieke regionale context van verschillende wielerwedstrijden. Zodra een POI is herkend, wordt het gevolgd gedurende de volledige videostroom. Het systeem behaalt een precisie en recall van meer dan 75% bij POI-detectie en levert consistente trackingprestaties, wat het een waardevol hulpmiddel voor live-uitzendingen maakt. Dit onderzoek toont het potentieel van computervisie-technologieën om sportuitzendingen te moderniseren en te optimaliseren door middel van automatische gebeurtenisdetectiesystemen. Naast wielrennen kunnen nieuwe systemen voor gebeurtenisdetectie worden ontworpen voor andere sporten, wat een breed scala aan toepassingen mogelijk maakt, van het verbeteren van sportprestaties tot het verhogen van de veiligheid van atleten.

Het ontwikkelen van effectieve systemen voor gebeurtenisdetectie in praktische toepassingen brengt diverse uitdagingen met zich mee die in academisch onderzoek vaak onvoldoende aandacht krijgen. Mijn onderzoek gaat deze uitdaging niet uit de weg maar verkent net deze complexiteiten door het ontwerpen, implementeren en evalueren van op maat gemaakte gebeurtenisdetectiesystemen die geschikt zijn voor *real-world* omgevingen. Daarvoor zijn vier casestudies onderzocht, elk gebruikmakend van verschillende databronnen, gericht op diverse ruimtelijke schalen en navigerend door moeilijkheden zoals datakwaliteit, tijdsvereisten en veranderende contextuele omstandigheden. De bevindingen van deze studies benadrukken het aanzienlijke potentieel van gebeurtenisdetectiesystemen om de efficiëntie te verbeteren, besluitvorming te ondersteunen en de veiligheid in diverse domeinen te versterken. Het is dan ook de hoop dat de veelbelovende resultaten niet onbenut blijven, maar verder onderzoek in dit waardevolle veld stimuleren.

Summary

In today's data-driven landscape, the ability to detect real-world events is important across diverse sectors, from industrial manufacturing and healthcare to urban planning and safety. Such events provide insights into changes of state in an environment that might require immediate response or are interesting for further analysis. However, accurately observing and measuring these events, such as identifying a technical fault in a wind turbine or detecting a crash during a cycling race, can be highly complex and lacks straightforward automated methods. As a result, monitoring such events has traditionally relied on human supervision, an approach that is not only time-consuming and prone to errors but also increasingly unsustainable given the rising demand for efficiency and scalability.

Advancements in sensing technologies and data processing have opened new possibilities for automating event detection. By leveraging data sources, such as video feeds and sensor readings, it is now possible to automatically extract meaningful information and detect events like wildfires in camera streams or machine anomalies through vibration data. Despite significant progress in academic research, many proposed solutions rely on controlled, large-scale datasets and overlook the practical challenges of real-world applications. This dissertation seeks to bridge that gap by focusing on the research and implementation of event detection methods in real-world settings. Through four case studies, the research demonstrates how custom methodologies can overcome challenges, such as limited data, real-time processing requirements, and environmental variability, while also integrating spatial context to provide deeper insights and actionable outcomes.

The first case study, presented in chapter two, focuses on event detection at the machine level within the steel industry, conducted in partnership with a leading steel manufacturing company. Maintaining a stable and controlled flame production in steel reheating furnaces is essential to ensure the quality of the steel. Consequently, continuous monitoring of the flames is crucial to detect events that

signal transitions from a healthy to an anomalous state within the furnace. To address this need, a computer vision-based system is designed and implemented, leveraging video streams from a thermal camera installed to monitor the interior of the reheating furnace. The system employs a hybrid approach, combining traditional computer vision techniques with learning-based methods to achieve both accuracy and explainability. Its core functionalities include detecting and localizing flames within the furnace, quantifying flame activity in specific burner regions, and identifying combustion anomalies. The system is validated using real-world data collected under various operating conditions, achieving an F1 score exceeding 80% for anomaly detection across different burner zones. These results demonstrate the system's effectiveness in accurately identifying irregularities within the furnace, enabling timely intervention. To improve usability, the system integrates a dashboard that provides operators with a real-time view of the furnace's status, along with access to historical data. This enables efficient monitoring, anomaly detection, and informed decision-making to support stable and smooth furnace operations. This study highlights the potential of automated event detection systems in manufacturing, addressing the challenges of limited labeled data, real-time constraints, and the need for explainable solutions in industrial processes.

Chapter three shifts the focus from video-based event detection to a sensor-based case study, specifically leveraging cost-effective CO_2 sensors that are commonly integrated into modern buildings. This sensor data is explored to detect events of human presence in rooms, which are valuable for multiple purposes. In the short term, detecting human presence enables real-time control of building systems, such as turning off lights or adjusting heating in unoccupied rooms. In the long term, presence detection facilitates the creation of occupancy profiles that reveal room usage patterns. These profiles allow building systems to transition from fixed timing schedules to adaptive schemes that adapt to changing occupant behavior, enhancing both energy efficiency and user comfort. The proposed system employs a machine learning-based approach to analyze CO_2 measurements and predict whether a room is occupied. A major challenge in designing such models is obtaining sufficient labeled data across diverse spaces to ensure effective training. To overcome this, the study introduces a cross-room methodology featuring sliding window normalization, which enables the model to generalize effectively to new, unseen rooms without the need for fine-tuning. By improving the transferability of such models to unseen rooms, this work lays the foundation for more robust and generalized event detection systems in smart

buildings. Furthermore, the proposed system demonstrates the dual role of automatic event detection systems in enabling immediate responses and contributing to long-term optimization.

Chapter four expands the scope of sensor-driven event detection to the building level, introducing a real-time Internet of Things software architecture designed to estimate COVID-19 transmission risk in indoor environments. Events of high risk provide actionable insights by notifying occupants and building managers, enabling timely interventions. A unique challenge of this application lies in the absence of ground truth data for COVID-19 transmission risk. To address this, the system integrates expert knowledge on aerosol transmission with a semantic ontology that defines relationships between indoor environmental parameters (e.g., CO_2 and temperature) and building concepts. By using SPARQL querying on a semantic data stream, the system generates interpretable risk scores and automatically detects high-risk events. These risk estimations are presented through a dynamic dashboard that adapts visualizations based on the semantic context of the data. A key feature of the dashboard is a floor plan widget, which enhances accessibility and provides real-time awareness for both occupants and building managers. The proposed architecture was deployed across three office buildings and evaluated during student examination periods to assess the effectiveness of COVID-19 safety measures. For example, results from January 2021 demonstrated that implemented safety measures significantly reduced transmission risk in a large auditorium. This study demonstrates the potential of semantic technologies in creating interpretable and modular solutions which can extend to additional use cases, such as general indoor air quality monitoring and environmental comfort scoring, ensuring long-term relevance for building management.

The final study in chapter five investigates the application of vision-based event detection on a regional scale, specifically within live sports broadcasts of cycling races. Traditionally, video editors manually add overlays to highlight Points of Interest (POIs), such as historical buildings and monuments, during these broadcasts, which is a repetitive and inefficient process. To streamline this workflow, this study proposes a novel computer vision system designed to detect and track POIs in real-time. By identifying events where the broadcast shifts focus from the race to a POI, the system reduces the manual effort required while delivering a more engaging and seamless viewing experience. The system employs a combination of saliency detection and the Segment Anything Model to generate potential POI regions. These candidate regions are then verified using a keypoint matching approach, relying on just a few reference images. This few-shot learning

capability eliminates the need for extensive manually labeled datasets and enables the system to effectively adapt to the unique regional context of each race. Once a POI is recognized, the methodology initiates tracking to follow it continuously throughout the video stream. The system achieves a precision and recall of over 75% in POI detection, coupled with stable tracking performance, making it a valuable tool in live broadcast environments. This research shows the potential of computer vision technologies in modernizing and optimizing live sports broadcasting through automated event detection. Beyond cycling, novel event detection systems can be designed for other sports, enabling a wide range of applications, from enhancing sports performance to improving athlete safety.

In summary, designing effective event detection systems for real-world applications presents numerous challenges that are often overlooked in academic research. This dissertation explicitly explores these complexities by designing, implementing, and evaluating custom event detection systems suited to real-world environments. To achieve this, four case studies are conducted, each leveraging different data sources, addressing a range of spatial scales, and navigating issues such as data quality, timing constraints, and changing contextual conditions. The findings from these studies highlight the significant potential of event detection systems to improve efficiency, inform decision-making, and enhance safety across various domains. By presenting and validating their effectiveness through four diverse use cases, this dissertation aims to encourage further research and innovation in this valuable field.

1 Introduction

This chapter provides context for this dissertation, offering background information and highlighting the primary focus of the study. It also outlines the structure of this dissertation, and includes an overview of the scientific publications over the course of the research.

1.1 Context

Events are everywhere around us, whether it's a weld breaking in steel manufacturing, a leak appearing in the water distribution network, or a fire suddenly igniting in a kitchen. These events, though varied in nature, share a common characteristic: they represent a change of state in an environment that is significant for analysis or triggering automated responses. Especially in today's data-driven world, the ability to recognize when events occur allows us to respond swiftly and make well-informed decisions. For instance, identifying anomalies early in manufacturing can prevent costly equipment failures, while timely recognition of events in healthcare can safeguard patient well-being. By understanding events and their implications, we not only unlock the potential to enhance efficiency and safety but also gain better insights into patterns, enabling us to anticipate future events.

Over the past few decades, many events have become digital in nature due to the growing digitization of processes. For example, making an online reservation at a restaurant generates a digital event that can easily be captured, transmitted, and acted upon by automated systems. Such digital events are not the focus of this dissertation. Instead, this work centers on events occurring in the physical world. Given the importance of knowing when such events happen, humans are often tasked with monitoring and registering them. However, this is neither practical nor efficient in the long term. Human monitoring is labor-intensive, repetitive, and prone to error. As the volume and complexity of events increase, this approach becomes even more problematic. Consider, for example, an industrial process that requires 24/7 monitoring. Assigning human operators to track this process continuously would demand a substantial workload. Moreover, not all events are easily observable by humans. Subtle changes, such as slight variations in machine vibrations, can go unnoticed, yet may signal critical issues.

To overcome these challenges, there is a growing need for automated event detection systems [1]. Such systems can dramatically reduce the dependency on human labor while increasing the accuracy and efficiency of event monitoring. Achieving this requires the integration of multiple technologies. The first step involves identifying reliable data sources that capture relevant information about the event. Once the data is collected, it must be analyzed using appropriate techniques to detect events hidden within the data stream. These detection methods can range from simple rule-based systems to advanced algorithms, depending on the complexity of the task. With this integration of data and algorithms, we move toward a future where detecting real-world events is not only automated but also scalable and efficient.

1.2 Examples of Event Detection

As previously introduced, event detection plays an important role across a wide range of industries. Each domain has specific goals for event detection, which may include identifying anomalies, support decision-making, or enhancing user experience. To make its importance more tangible, a few examples are explored below.

With the growing need to decarbonize energy production, the number of wind turbines being installed both onshore and offshore has surged, as shown in Figure 1.1a. Ensuring the efficient and safe operation of these turbines is essential to maintain energy output and avoid damage. While human inspections can identify damage to turbine blades or assess engine performance periodically, it is impractical to monitor every turbine continuously. Automated data analysis is therefore necessary to detect events, such as mechanical anomalies or unexpected perfor-

mance deviations. For instance, if an issue arises with a turbine's motor, the system can send an alert to an operator to request inspection or automatically shut down the turbine to prevent further damage. IDLab tackled this use case during Janssens' Ph.D. research [2], employing sensor-based and video-based anomaly detection techniques, along with fleet-level visualization on an interactive dynamic dashboard. This approach allows for continuous monitoring at scale which facilitates safe and efficient operation.

(a) Monitoring wind turbine operation *(b) Detecting cycling crashes*

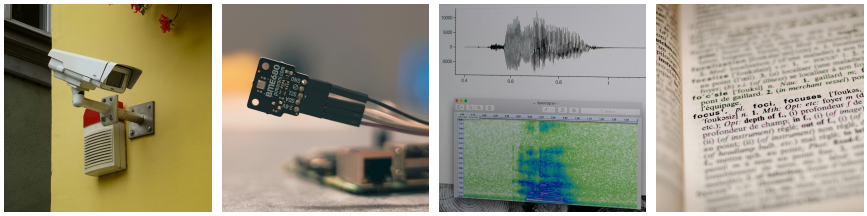
Figure 1.1: Example use-cases for event detection.

In the world of sports, event detection can also provide significant value. Consider a cycling race, as illustrated in Figure 1.1b, where crashes occur involving one or multiple riders. While human viewers can quickly interpret such incidents when watching the broadcast, computers require advanced detection algorithms to identify these events from pixel data in the video stream. However, detecting a crash in real-time can be useful for multiple stakeholders. For race organizers, immediate crash detection allows them to respond quickly, ensuring rider safety and making adjustments to the race if necessary. At the same time, governing bodies like the Union Cycliste Internationale (UCI) can use this data for post-race analysis, such as identifying road safety issues or improving future race conditions. Additionally, broadcasters and journalists benefit from receiving crash alerts, which allows them to provide timely updates and coverage to their audiences. This single event can trigger a chain of information sharing and decision-making that extends far beyond the moment it occurs. While conducting research at IDLab, De Bock devoted his entire Ph.D. to data-driven performance and safety analysis in cycling races, including event detection applications, such as road cycling crashes and madison handslings in track cycling [3].

These examples highlight the diverse applications and benefits of event detection in vastly different domains. Whether ensuring the smooth operation of wind turbines or improving safety in sports, event detection enables proactive responses and facilitates better decision-making.

1.3 Data Sources for Event Detection

Since this dissertation focuses on real-world events, it is essential to have relevant data sources that can be used to observe and detect these events. Commonly used data sources include video (image), audio, sensor, and textual data, as illustrated in Figure 1.2. While this dissertation mainly focuses on video and sensor data, a brief exploration of audio and textual data will also be performed at the end of this section.



(a) Video *(b) Sensor* *(c) Audio* *(d) Text*

Figure 1.2: Visualization of the different data sources.

1.3.1 Video Data

In many existing workflows, human operators are tasked with visually inspecting or monitoring processes or objects. For such cases, video data serves as an excellent source of information for event detection algorithms. A video is essentially a sequence of images, also known as frames, each of which has a specific spatial resolution. This spatial resolution refers to the number of pixels within each frame, with a pixel being the smallest unit of detail in an image. Common video resolutions include 1920x1080 pixels (Full HD) and 3840x2160 pixels (4K).

In addition to the spatial resolution, the frame rate of a video, measured in frames per second (FPS), determines its temporal resolution. This indicates how much time is between frames and determines how finely movements can be captured over time. Common frame rates are 25 FPS (one frame every 40 milliseconds) and 30 FPS (one frame every 33.3 milliseconds).

In the context of continuous monitoring, video streams are commonly used, where new frames are captured and transmitted in real-time. Therefore, it is important that the resolution and frame rate are compatible with the system's processing capabilities to avoid data overload. The appropriate resolution and frame rate depend on the specific use case and should be carefully considered during research. If high spatial or temporal accuracy is unnecessary, lower resolutions

and frame rates are often preferred to reduce data volume and lower processing requirements.

When working with video, the volume of raw data can be immense. Consider a live sports event broadcast where multiple cameras are positioned around the venue to capture various angles of the action. For such multi-camera setups, real-time video from each camera is transmitted to a central processing unit (e.g., a broadcast truck or control room). Transmitting uncompressed video would require an enormous amount of bandwidth. This is where video encoding becomes important. Video compression algorithms, like H.264 and H.265, reduce the size of video by efficiently encoding redundant or unnecessary data, lowering the amount of data that needs to be transmitted without sacrificing too much quality. This in turn reduces the risk of latency or dropped frames which could impact event detection performance.

1.3.2 Sensor (Time Series) Data

Another important data source used for event detection is sensor data. A sensor is a device that measures a physical property in the real world and converts it into a digital value. These measurements can be anything from environmental conditions (e.g., temperature and humidity) to biological signals (e.g., heart rate and skin conductance) or mechanical properties (e.g., acceleration and strain).

The advent of the Internet of Things (IoT) has significantly expanded the range of connected devices that continuously stream sensor data, creating networks of interconnected sensors. These IoT systems are widely used in fields such as healthcare, smart cities, environmental monitoring, and industrial automation, where they hold huge potential for real-time event detection and predictive analytics. For instance, in healthcare, wearable devices equipped with heart rate, temperature, and motion sensors can monitor a patient's health in real time, alerting medical professionals of any abnormal patterns that may signal an emergency, such as a heart attack or fall.

Sensors record measurements at a specific frequency, known as the sampling rate. For example, a sensor sampling at 50 Hz captures data 50 times per second (once every 20 milliseconds). The resulting data can be considered a time series, where each data point is accompanied by a timestamp indicating when the measurement was taken. If heart rate data is sampled once per second (1 Hz), a single day would produce 86,400 data points, each with a corresponding timestamp. This time-stamped data allows for the detection of patterns and anomalies over time.

While sensor data is typically less voluminous per sample compared to video, high sampling frequencies and the use of multiple sensors in combination can lead to the generation of substantial amounts of data. For example, the Whoop

wearable health monitoring device generates over 150 megabytes of data per day [4]. Multiply this by thousands of devices, and the need for efficient data processing becomes apparent. Moreover, while this thesis doesn't focus on data storage, many technologies exist that are specifically designed for the efficient storage and retrieval of high volumes of time series data, such as InfluxDB[1] and TimescaleDB[2].

1.3.3 Audio and Textual Data

Although audio and textual data are not the primary focus of this dissertation, they are valuable sources for detecting and analyzing events across different environments.

Audio can be considered a form of time series data, as sound waves are captured by a microphone, which functions as a sensor. The sound is sampled at a specific frequency, such as the common 44.1 kHz rate used in many audio recording settings. Each sample reflects the amplitude of the sound wave at a given moment, making it possible to analyze patterns over time.

For example, in sporting events like padel or tennis, a microphone can be set up to capture the sounds of the court. This audio data can then be processed to detect the sounds of the ball striking the racket or hitting the court. Beyond sports, audio data can be used in surveillance systems to detect abnormal events, such as the sound of breaking glass or gunshots, which may indicate security breaches.

Additionally, if the audio stream contains speech, it can be automatically transcribed into text using speech-to-text algorithms. This transcription opens up further possibilities for natural language processing (NLP). For example, during a football match, the commentary could be transcribed and analyzed to detect keywords like "goal," "penalty," or "red card," triggering automatic event logging.

In environments where written records are heavily relied upon, such as in healthcare, textual data plays an important role in documenting events. Electronic health records (EHRs), for instance, contain structured health record data and detailed unstructured medical notes that track every step of patient care, from diagnosis to treatment. These EHRs hold valuable information that can be processed to extract meaningful events, such as when medication is administered.

1.3.4 Key Takeaways

As discussed, a wide variety of data sources exist for event detection. Each data type, whether video, sensor, audio, or text, has its strengths and is suited to differ-

[1] https://www.influxdata.com

[2] https://www.timescale.com

ent applications, depending on the nature of the events being monitored and the specific requirements of the task at hand. It is therefore important to select the data source that offers the highest likelihood of accurately capturing the events of interest.

1.4 Processing Techniques for Event Detection

Within this thesis, a variety of data processing techniques will be leveraged to detect events. The choice of methodologies depends on various factors, including the data source, processing speed, data quality, and interpretability. This section provides a high-level overview of the key concepts and techniques employed to process different types of data, with a focus on video and sensor data.

1.4.1 Video Data

1.4.1.1 Traditional Computer Vision

Over the years, a variety of traditional computer vision techniques have been developed for video and image processing. For example, keypoint detection, which identifies distinctive points in an image, is often implemented using the scale-invariant feature transform (SIFT) algorithm [5]. This technique enables the matching of objects across different scenes, as demonstrated in Figure 1.3, which can be used for panorama stitching or 3D reconstruction. Another well-know technique is the Lucas-Kanade method for optical flow [6] that captures the motion of objects between consecutive video frames by calculating the flow of pixels.

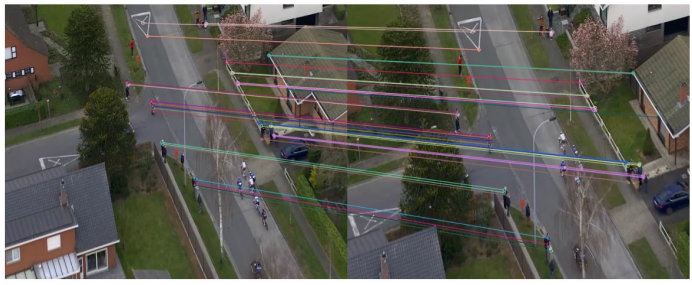


Figure 1.3: Keypoint matching between two different frames of the same video.

These classical computer vision algorithms have been extensively studied and successfully applied to a wide range of tasks. Therefore, many of these methods are implemented in popular libraries such as OpenCV, which remains a widely

used toolkit for computer vision engineers. Although these techniques are efficient and reliable for various applications, they often struggle with more complex tasks, such as understanding semantic information or handling varying lighting and occlusion conditions. These limitations highlight the need for more advanced approaches that can adapt to such challenges, like deep learning.

1.4.1.2 Deep Learning

The advent of artificial intelligence, and more specifically deep learning, has revolutionized the field of computer vision. These deep learning models have demonstrated remarkable improvements over traditional methods, particularly in tasks requiring robust pattern recognition and feature extraction. Convolutional neural networks (CNNs) have been a large driver of this revolution. A CNN is a type of deep learning architecture specifically designed for processing visual data. It consists of several layers, each responsible for different tasks. For instance, convolutional layers automatically extract features from input images by applying filters, pooling layers reduct the dimensionality of the data and fully connected layers perform the final prediction by combining the extracted features, as visualized in Figure 1.4. By stacking multiple layers, CNNs can learn to recognize patterns and features in images, making them ideal for many computer vision tasks [7].

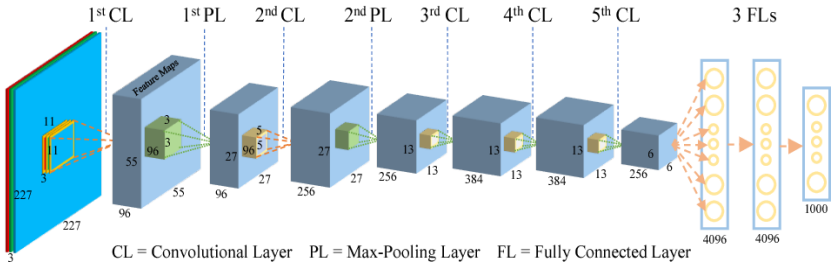


Figure 1.4: Visualization of a simple convolutional neural network architecture with convolutional, pooling and fully connected layers (image from [8]).

1.4.1.3 Image Classification

One of the simplest tasks handled by CNNs is image classification, where the goal is to assign a label to an entire image from a set of predefined classes. For example, determining whether an image contains a windmill or a solar panel, as show in Figure 1.5a. AlexNet [9], a pioneering CNN model, achieved state-of-the-art performance in the 2012 ImageNet competition, marking a breakthrough in

image classification. Since then, more advanced architectures such as ResNet [10] and EfficientNet [11] have been developed, further improving the accuracy and efficiency of image classification models. ResNet introduced the concept of residual learning, enabling the training of much deeper networks by addressing the vanishing gradient problem, while EfficientNet proposed a novel compound scaling method that optimally balances network depth, width, and resolution to achieve high performance with significantly reduced computational costs.

(a) Image classification *(b) Object detection*

Figure 1.5: Example model outputs from (a) image classification and (b) object detection.

1.4.1.4 Object Detection

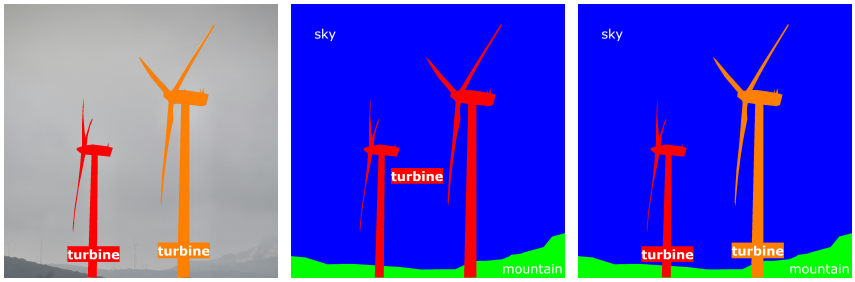
Unlike image classification, which assigns a single label to an entire image, object detection aims to identify multiple objects within a scene and localize them by predicting bounding boxes, such as in Figure 1.5b. This allows for the detection of several objects simultaneously. Notable architectures for object detection include Faster R-CNN [12] and the YOLO (You Only Look Once) [13] series, which have become widely used for real-time detection tasks.

1.4.1.5 Segmentation

Instance segmentation extends object detection by not only identifying and localizing objects but also predicting a pixel-level mask for each detected object, as shown in Figure 1.6a. This is useful in scenarios where understanding the exact boundaries of an object is important. Models such as Mask R-CNN [14] and variations of YOLO have been developed to address this task.

For cases where distinguishing individual object instances is not needed, but identifying areas (e.g., road, sky, or vegetation) within an image is required, semantic segmentation models like U-Net [15] and PSPNet [16] are employed. These

models essentially perform pixel-level classification, assigning a specific class label to each pixel in the image. This approach segments the image into meaningful regions without differentiating between separate instances of the same object class, as illustrated in Figure 1.6b.



(a) Instance segmentation *(b) Semantic segmentation* *(c) Panoptic segmentation*

Figure 1.6: Example model outputs from (a) instance segmentation, (b) semantic segmentation and (c) panoptic segmentation.

Panoptic segmentation is a task in computer vision that unifies the goals of instance segmentation and semantic segmentation, as illustrated in Figure 1.6c. This means that the model not only distinguishes between different categories (e.g., road, sky, vegetation, cars, pedestrians) like in semantic segmentation but also identifies individual instances of certain object categories (e.g., individual cars or pedestrians), similar to instance segmentation. By combining the strengths of both, panoptic segmentation offers a more richer understanding of the scene. Several state-of-the-art models have been developed to tackle panoptic segmentation by effectively merging the outputs of instance and semantic segmentation branches, such as Panoptic-DeepLab [17] and UPSNet [18].

1.4.1.6 Transfer Learning

Deep learning models have millions of parameters, so training them from scratch often requires large amounts of labeled data and computational resources. To mitigate these challenges, trained models are commonly reused. Once a model is trained on a large dataset, its backbone, which is the feature extraction component of a CNN, contains generic image features in the early layers. By combining the pretrained backbone with a custom prediction head, the model can be fine-tuned for specific tasks without having to retrain all layers from scratch. This approach, known as transfer learning, significantly reduces the amount of data and time required to achieve high performance on other tasks [19].

1.4.1.7 Vision Transformers

As the timeline of Figure 1.7 suggests, vision transformers (ViTs) have emerged in recent years as a powerful alternative to CNNs for various computer vision tasks. Originally developed for natural language processing, transformers have been adapted for image data [20], offering several advantages over CNNs, especially in capturing long-range dependencies within images.

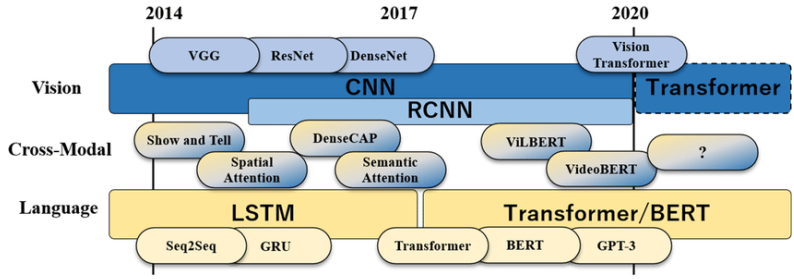


Figure 1.7: Timeline of the development of neural networks showing the advent of transformers in both the language and vision domain (image from [21]).

Unlike CNNs, which rely on convolutional layers to extract local features, vision transformers divide an image into patches and process them using self-attention mechanisms, as shown in Figure 1.8. This allows the model to focus on important regions of the image, making it highly effective for tasks requiring a global understanding of the scene. However, ViTs have some notable downsides. They rely on large datasets for effective training due to the lack of convolutional inductive biases, which makes them prone to overfitting on smaller datasets. Consequently, their training is often more resource-intensive than that of CNNs.

Vision transformers have demonstrated strong performance across a range of tasks, including classification, detection, and segmentation [22, 23], and are seen as a key component of foundational models. These foundational models are large, pre-trained neural networks designed to be versatile and adaptable across a wide range of downstream tasks and domains. Examples of such models include CLIP (Contrastive Language–Image Pre-training) [24] which combines vision and language understanding, DINO (Self-Distillation with No Labels) [25] for self-supervised image embeddings, and SAM (Segment Anything) [26] for prompt-based segmentation across unfamiliar objects and images.

1.4.1.8 Key Takeaways

The integration of traditional computer vision techniques with modern deep learning approaches has significantly expanded the range of possibilities for video-

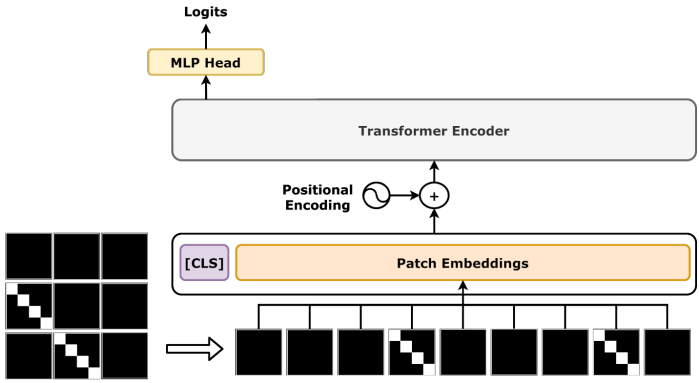


Figure 1.8: An image is divided into patches, embeddings are generated, and a positional encoding is applied, before being passed to the transformer (image by dvgodoy / CC BY).

based event detection. While traditional algorithms remain useful for simpler tasks, the introduction of CNNs, vision transformers, and foundational models has enabled more accurate and robust solutions for complex problems.

1.4.2 Sensor (Time Series) Data

Time series data, much like video data, can be analyzed using a variety of traditional signal processing techniques that remain widely used today, such as peak detection and the Fast Fourier Transform (FFT) [27]. While these methods are indispensable for certain fundamental operations, machine learning has become an invaluable tool for identifying more complex patterns and performing detection tasks. This can be achieved through either traditional machine learning models or deep neural networks.

1.4.2.1 Traditional Machine Learning

Traditional machine learning models are not designed to handle time series data efficiently since they would treat each observation as separate from others, ignoring the time-dependence among data points. To overcome this limitation, a common strategy is to calculate features that represent essential characteristics of the time series, such as magnitude, trends, or variability. These features can then be used by conventional machine learning techniques to understand the temporal relationships within the data. However, these features are not computed for the entire time series at once, as this would lose too much temporal details and would be impractical for continuous data streams. Instead, a sliding

window feature extraction approach is used, as illustrated in Figure 1.9 showing a time series of CO_2 data. For example, these CO_2 measurements can be used to predict room occupancy, a topic that will be explored in greater depth later in this dissertation. Since CO_2 data typically exhibits slower trends, features can be computed using a larger window size, such as 3 hours, with a stride of 1.5 hours. In contrast, tasks such as sleep staging based on polysomnography data (e.g., brain activity, eye movement, hearth rate, etc.) require capturing finer temporal details. Therefore, smaller window sizes and strides need to be used, such as 30 seconds each, to better retain the temporal dynamics of the data. The features derived from these windowed segments can then serve as input for traditional machine learning models, which often achieve very good predictive performance [28].

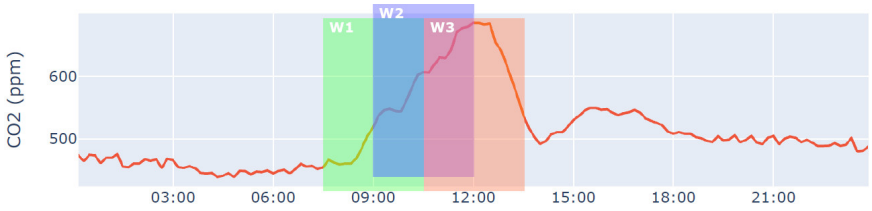


Figure 1.9: A sliding window with a certain size and stride is applied to the time series (e.g., W1, W2, etc.) from which features can be calculated.

In addition to training models to make predictions based on time series data, anomaly detection is another valuable task that focuses on identifying unusual patterns or behaviors, which may signify faults, fraud, or other critical events. Available techniques often leverage machine learning models trained on normal behavior patterns, enabling the detection of anomalies as deviations from expected trends. However, beyond model-based approaches, there exists a straightforward yet powerful methodology known as matrix profiling [29, 30]. This technique computes the matrix profile, a data structure that quantifies the distance between each subsequence of a time series and its closest matching subsequences. Peaks in the matrix profile act as flags for potential anomalies, as they indicate subsequences that are clearly dissimilar from the rest of the data. For instance, consider a time series representing power consumption in a manufacturing plant. Under normal conditions, power usage typically exhibits predictable cyclic patterns aligned with shifts or regular machinery operations. However, if a machine malfunctions and causes an unexpected power spike or if power usage drops unexpectedly during a production shift, these events will appear as peaks in the matrix profile, signaling anomalies. This combination of simplicity and effectiveness has made matrix profiling a widely adopted approach in time series analysis.

1.4.2.2 Deep Learning

In cases where the data is too complex for manual feature engineering, deep learning offers a powerful alternative. Recurrent neural networks (RNNs), particularly long short-term memory (LSTM) networks [31] and gated recurrent units (GRUs) [32], are employed for time series data due to their ability to capture temporal dependencies. Unlike traditional feedforward neural networks, RNNs have recurrent connections in their architecture that allow them to retain information from previous inputs, making them highly effective for sequential data. These architectures autonomously extract meaningful features from raw data in an end-to-end learning approach.

Beyond RNNs, CNNs and transformer models can also be applied for time series tasks. CNNs are particularly effective at detecting local temporal patterns and trends through their convolutional filters, enabling them to learn hierarchical representations. On the other hand, transformers use self-attention mechanisms to model long-range dependencies in the data. These transformers process data in parallel, making them computationally efficient for longer input sequences compared to traditional RNNs. However, deep learning models generally require way larger datasets to train effectively compared to traditional machine learning techniques. This can be a limitation in domains where collecting extensive time series data is challenging.

In finance, deep learning models are frequently applied for tasks such as stock price prediction and risk assessment. For instance, to forecasting stock prices, researchers have employed techniques like LSTMs or CNNs to learn complex temporal dependencies within historical financial data [33, 34]. These models can capture seasonality, trends, and even sudden market shifts, offering an advantage over simpler statistical tools. Deep learning is also used in healthcare for analyzing physiological signals such as heart rate, electrocardiogram (ECG) data, or blood glucose levels over time. For instance, transformer models have been successfully employed to detect irregular heartbeat patterns in ECG signals [35].

1.4.2.3 Knowledge-Based Systems

While machine learning algorithms excel at analyzing patterns and making predictions, another important approach in artificial intelligence is the use of knowledge-based systems. Unlike machine learning, which relies on statistical models and data-driven learning, knowledge-based systems operate on explicitly encoded knowledge, such as predefined rules, facts, and relationships. Therefore, these systems are often designed with the input of domain experts. Next to their usefulness in environments where data is scarce or noisy, a major advantage of knowledge-based systems is their transparency. Since every decision is based on

defined knowledge, it is easier to understand and explain how a decision was reached, an important consideration in fields like healthcare and finance, where the reasoning behind decisions must be clear and trustworthy.

An important concept within knowledge-based systems is the use of semantic data, which is data that is structured in a way that is meaningful and understandable to machines. This is achieved through ontologies [36], which define relationships between different entities and concepts, as exemplified in Figure 1.10. This structured representation allows the system to perform semantic reasoning, which goes beyond simple data matching and can infer new insights based on these relationships. With the right technologies, knowledge-based systems can also be applied to streamed time series data, enabling them to process queries and make reliable decisions based on the relationships between entities.

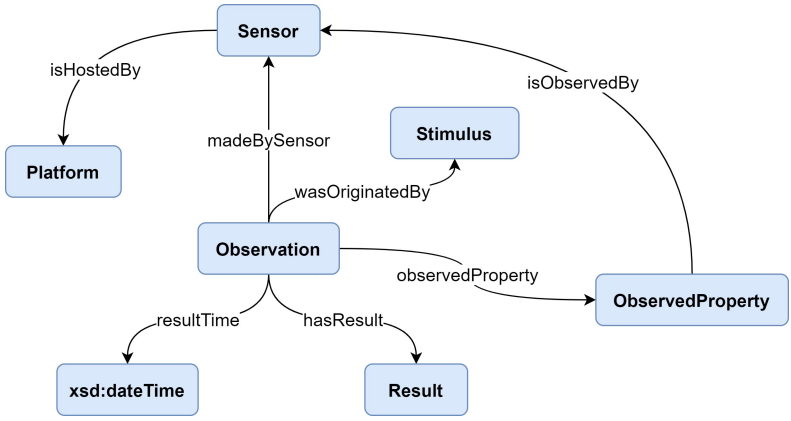


Figure 1.10: Main classes and properties from the Semantic Sensor Network (SSN) [37] ontology related to modelling an "observation".

An application of knowledge-based systems that uses semantic technologies is IBM Watson for healthcare [38]. Watson processes unstructured medical literature, structured patient data, and ontologies like SNOMED CT (a widely used clinical health terminology) to create a comprehensive understanding of a patient's condition. Semantic reasoning allows Watson to relate symptoms, test results, and historical data to potential diagnoses. For example, if a patient presents with symptoms like "chronic cough" and "shortness of breath," Watson could infer from its knowledge base that these symptoms might signify chronic obstructive pulmonary disease (COPD), even if the term "COPD" hasn't been explicitly mentioned. As an industrial example, Moens et al. presented a hybrid framework aimed at enhancing predictive maintenance [39]. The methodology combines knowledge-driven and data-driven approaches to detect anomalies (unexplained events) and faults (known issues) in sensor data streams. It employs predefined rules, based

on domain expertise, to identify specific faults while integrating semantic reasoning for event visualization and interpretation.

1.4.2.4 Key Takeaways

Both learning-based and knowledge-based systems offer powerful approaches for analyzing time series data and making decisions based on it. Traditional machine learning methods, using window-based feature extraction, remain effective for many applications. However, deep learning provides solutions for more complex datasets, which often show high dimensionality or complex temporal dependencies. In contrast, knowledge-based systems offer transparency and reliability, making them ideal for domains where understanding the decision-making process is critical.

1.5 Research Focus

This Ph.D. presents part of my research across multiple projects in collaboration with partners such as Aperam, ArcelorMittal, Artevelde UAS, EMG, Niko and UZ Gent. The diversity of these collaborations have enabled me to tackle event detection in various domains, including industry, healthcare, and sports. This involved leveraging a variety of data sources and employing a wide array of processing techniques.

While the application of fundamental research concepts to real-world scenarios may seem straightforward, the reality is far more complex. There is no one-size-fits-all solution, and significant challenges arise when adapting these techniques beyond controlled research environments [40, 41]. Each project presents its own set of challenges, requiring custom and optimized solutions tailored to the specific requirements and context of the problem at hand. The following sections will delve deeper into the challenges of designing real-world event detection systems. These are the core focus of this dissertation, which can be summarized as: **How can we design effective event detection systems that operate reliably in diverse real-world environments?**

1.5.1 Challenge 1: Handling Data Variety, Velocity and Availability

As discussed in Section 1.3, many types of data exist, each presenting their own challenges and requiring specific techniques for effective processing. Data sources,

whether they involve sensors, cameras, or other devices, introduce various complexities, such as sensor drift, environmental changes, calibration differences, positioning shifts, measurement noise, and data dropout. These factors can vary not only between different sensors but also within the same sensor over time, especially as conditions change during data collection or operational phases.

To design robust real-world solutions, it is important to address these variations during the research stage, and this is where the concept of "garbage in, garbage out" (GIGO) is relevant. GIGO underscores the principle that the quality of output is directly dependent on the quality of input data. If flawed data is used as input, the resulting analyses or decisions will likely be unreliable, regardless of how advanced the processing techniques are. Therefore, addressing issues such as sensor drift and calibration differences during the research phase is required to ensure reliable and accurate application of the solution.

Data velocity also introduces challenges in the context of this dissertation, as it focuses on streamed data from video and sensor systems. If the continuous flow of data is not processed quickly enough, it can lead to issues such as buffer overflow and system lag, ultimately causing the event detection to fall out of sync or freeze entirely. Therefore, it is essential to define and respect timing constraints tailored to the specific use case. In this context, it's important to distinguish between real-time and low-latency processing, as they address different priorities. Real-time systems prioritize predictability, ensuring that tasks are completed within strict, predefined timing constraints. In contrast, low-latency systems focus on speed, aiming to minimize the delay between input and output, but without necessarily guaranteeing when a specific task will finish. For example, in real-time applications such as autonomous vehicles, failing to meet the processing deadline can result in catastrophic failures. These systems demand predictable and consistent timing, even if it sometimes means sacrificing processing complexity or overall throughput. On the other hand, low-latency use cases, like sports video stream processing, aim to process data as quickly as possible but can tolerate occasional, small fluctuations in delay. While low-latency systems benefit from optimizations such as data batching or asynchronous processing, they don't necessarily guarantee fixed deadlines.

Another significant challenge is data availability, as many research projects face constraints due to limited data. In cases where there is no data at all, the development of a good data collection strategy is an important part of the research. Alternatively, projects may start with only a small dataset, requiring careful choices regarding the used methodology to ensure the solution can generalize to future data from different sensors and environments. In other instances, large quantities of data may be available, but they may lack labeled ground truth. As labeling large datasets is labor-intensive, it becomes important to maximize the value of limited labeled data. This often involves employing techniques such as

data augmentation, transfer learning, few-shot learning, and other approaches to extract meaningful insights without relying on extensive labeled datasets.

This thesis presents research projects that address these various data challenges to varying degrees and offers solutions that take into account the requirements of each use case.

1.5.2 Challenge 2: Detecting Spatial Event Context at Various Levels

Real-world events are inherently spatiotemporal, as they are always tied to both time and space. The temporal dimension specifies when an event occurs, while the spatial dimension provides context for where it takes place. This dissertation focuses primarily on the temporal aspect of events, as detecting events within data streams naturally yields this temporal information. However, spatial context is important in many applications for making data actionable. For example, in a wildfire detection system, identifying the geographic location of a fire outbreak is essential for ensuring timely resource deployment and minimizing damage. Recognizing the importance of this spatial dimension, this dissertation also explores the challenge of enriching detected events with such additional context. Spatial information can either be inferred from the placement of sensors or determined through more extensive data-analysis. Adding this spatial layer not only renders events more actionable but also enables a more comprehensive analysis. Events are often interdependent across both time and space, and understanding the patterns within these dimensions can uncover deeper insights. For instance, a traffic jam detected at a particular intersection at 8 AM could be linked to subsequent congestion at nearby intersections.

The spatial scale of event detection systems can vary widely, which has implications for system design and technical requirements, such as data storage and processing. Depending on the use case, some systems may need to detect events locally, such as within a specific room of a building, while others may operate across larger scales, such as a city, an entire country, or even globally. As the spatial scope increases, the volume of data that needs to be processed often grows accordingly. Moreover, these systems frequently require the integration of heterogeneous data sources, such as GPS signals, IoT sensors, or edge devices. Each of those may provide data at different frequencies and resolutions, which requires synchronization and registration to ensure compatibility. Therefore, this dissertation aims to explore the detection of events at various spatial scales, investigating the associated challenges and proposing methodologies to overcome them.

1.5.3 Challenge 3: Addressing Real-World Event Detection Cases

Over the past years, AI has made significant advancements in research environments. However, translating these advancements into real-world applications presents numerous challenges. One of these challenges, as already discussed in Section 1.5.1, is the availability and quality of data. While fundamental research relies on large and well-curated datasets, applied research often has to deal with data that is limited, lacks labels, or varies greatly in quality. Despite these constraints, methodologies must be designed that generalize and adapt to new contexts such as seasonal changes, evolving user behavior, or fluctuating market conditions. This requires a careful approach to data handling, technique selection, and methodology development to ensure the robustness and adaptability of the system.

Another challenge is that real-world problems often differ from the standard tasks for which many algorithms and methods are originally designed. In industry, partners frequently have specific requirements that must be met for solutions to be usable and integrable within their existing systems or workflows. This could involve timing constraints, such as the need to process data streams in real-time, or resource constraints to have the system operate under limited processing power, memory or battery life. This often requires methodologies that balance accuracy with the applicable constraints. While existing research can provide a foundation, it typically requires adaptation or extension to meet the specific needs of real-world applications.

Moreover, a common requirement within projects is the ability to provide explainable outputs. This ensures that users can understand how decisions are made by AI systems, which is important for building trust and facilitating smoother user acceptance. In some cases, this can be achieved by using inherently interpretable models like linear regression or decision trees, where the internal mechanisms are transparent enough for humans to easily comprehend the decisions. However, these simpler models may not always be adequate for all tasks, requiring the use of more complex or "black box" models such as neural networks. While these models lack inherent interpretability, external explanations can be provided to clarify how they arrive at specific outcomes. Techniques like LIME (Local Interpretable Model-agnostic Explanations) [42] and SHAP (SHapley Additive exPlanations) [43] are valuable for this purpose, as they offer post-hoc explanations of model predictions. Additionally, presenting intuitive visualizations to users can enhance their understanding of the model's decision-making process.

The work presented in this thesis places a strong emphasis on making research applicable to real-world environments. This involves addressing various challenges, including designing systems that are both robust and adaptable to

dynamic contexts through custom methodologies, while also ensuring it offers explainability where required.

1.5.4 Research Goals

Building on the three main research challenges outlined in the previous sections, a set of research goals (RGs) is defined to summarize the high-level objectives of the case studies presented in this dissertation.

- **Handle inconsistent data quality (RG 1.1):** Real-world data often comes with inconsistencies and imperfections, requiring effective preprocessing and modeling techniques to address such suboptimal data. This ensures that the designed solutions remain robust and reliable, even when confronted with varying data quality conditions.

- **Comply with time constraints (RG 1.2):** Event detection in continuous data streams demands efficient processing within the required timeframe. Therefore, the speed of processing should align with the requirements of the use case to ensure timely insights and smooth operation.

- **Exploit no or limited labeled data (RG 1.3):** In scenarios with scarce or absent labeled data, it is important to carefully design solutions using innovative techniques that make best use of the limited data resources.

- **Provide spatial event context (RG 2.1):** While knowing when an event occurs is valuable, understanding where it occurs provides even more insights. The designed event detection systems should output spatial context, allowing for richer, more actionable outcomes.

- **Demonstrate event detection across scales (RG 2.2):** Events can occur at any spatial scale, ranging from microscopic cellular processes to universe-wide phenomena. This dissertation seeks to validate the possibilities of event detection methods across a broad range of scales, illustrating their significance in varying spatial settings.

- **Ensure robustness to changing contexts (RG 3.1):** Real-world deployment of an event detection application often involves operating in contexts different from those during the design phase, as well as adapting to evolving conditions over time. To address this, the proposed methodologies should be robust, flexible, and generalizable to ensure reliable performance despite the changing environments and context.

- **Achieve explainable solutions (RG 3.2):** The ability to explain decision-making processes is important in applications where trust and transparency are priorities. The designed research should incorporate explainable methods when required, ensuring that all stakeholders can confidently understand and act on the results.

1.6 Chapter Outline

The previous sections of this dissertation have provided the necessary background context (Section 1.1), some event detection examples (Section 1.2), an overview of potential data sources (Section 1.3), and a discussion on processing techniques (Section 1.4) for event detection. Furthermore, the focus of this dissertation was clarified in Section 1.5. After this outline, the introduction will close with a summary of my research publications.

The next four chapters delve into specific case studies that design event detection methodologies across different domains, including industry, building management, and sports entertainment. Moreover, these case studies are situated at different spatial levels, ranging from the machine level to the regional level, which is also reflected in the structure of this dissertation, as can be seen in the schematic overview of the chapters in Figure 1.11. Below is a brief explanation of the case studies covered in each chapter.

At the **machine level**, Chapter 2 addresses event detection in the steel manufacturing sector, focusing specifically on reheating furnaces. The system leverages thermal video streams to identify anomalies in burner flame production, signaling a transition from a healthy to an anomalous furnace state. This event detection system improves operational efficiency and ensures steel quality by rapidly identifying combustion issues. To achieve this, a hybrid AI-driven computer vision approach is proposed to monitor flame stability and detect anomalies. Despite the availability of only a limited labeled dataset, the methodology is designed to be robust against the wide range of variations within the furnace, while also ensuring real-time video stream processing.

As illustrated in Figure 1.11, event detection at the **room level** is explored in Chapter 3 by introducing a CO_2-based presence detection methodology. A machine learning model is designed using a limited labeled dataset of CO_2 sensor readings, which is capable of generalizing across various rooms in an unsupervised manner. The detected presence events are used to generate occupancy profiles for each room, aiding in the optimization of heating, cooling, and ventilation control systems. This approach not only enhances energy efficiency and occupant comfort but also avoids reliance on privacy-invasive technologies, such as cameras.

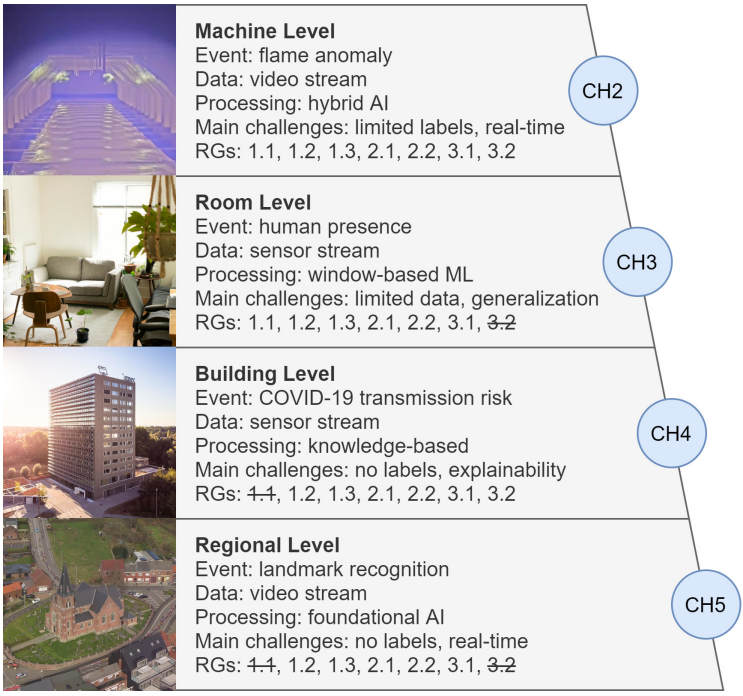


Figure 1.11: Visual outline of the following chapters.
(iGent image © UGent, Jonas Vandecasteele)

Chapter 4 moves to the **building level** and presents an event detection system designed to estimate and monitor the risk of COVID-19 aerosol transmission in indoor environments. This system proved valuable during the pandemic by offering real-time insights into air quality and alerting employees and building managers to events of increased transmission risks. In the absence of labeled data, the study proposes a real-time, knowledge-based system that utilizes IoT sensor data, such as CO_2 and temperature, to calculate risk levels. These risk assessments are dynamically visualized on an interactive dashboard.

At the **regional level**, Chapter 5 introduces a novel method for detecting and tracking points of interest (POIs) during live road cycling broadcasts. By detecting events where the broadcast shifts focus from the race to a landmark, the system reduces manual effort of video editors, enhances scalability, and improves the viewer experience. As shown in Figure 1.11, an AI-based foundational system is proposed to address the lack of labeled data, enabling POI recognition without requiring extensive datasets for fine-tuning. This adaptability ensures the system can be applied across different races and locations.

Each of these chapters underscores the versatility of event detection techniques across different spatial levels and domains, demonstrating the potential of these methods to improve operational efficiency, safety, and user experience. The final chapter of this dissertation, Chapter 6, will summarize the main findings and revisit the core focus identified earlier, while also outlining potential directions for future research.

1.7 Publications

The findings of this dissertation have been disseminated through publications in peer-reviewed scientific journals and presentations at international conferences. Below is a summary of the publications that resulted from my PhD research.

1.7.1 Publications in International Journals (Listed in the Science Citation Index[3])

1. Decorte, R., **Vanhaeverbeke, J.**, VanDen Berghe, S., Slembrouck, M., & Verstockt, S. (2025). *Continuous Monitoring of Recruits during Military Basic Training to Mitigate Attrition.* SENSORS, 25(6).

2. **Vanhaeverbeke, J.**, Deprost, E., Verstockt, S. & Van Hoecke, S. (2024). *Cross-Room CO2-based Presence Detection for Occupancy Profiling.* Manuscript under review at IEEE Access.

3. **Vanhaeverbeke, J.**, Verstockt, S. & Van Hoecke, S. (2024). *Flame Monitoring and Anomaly Detection in Steel Reheating Furnaces Based on Thermal Video Using a Hybrid AI Computer Vision System.* Manuscript under review at Scientific Reports.

4. **Vanhaeverbeke, J.**, Decorte, R., Slembrouck, M., Van Hoecke, S., & Verstockt, S. (2024). *Point of Interest Recognition and Tracking in Aerial Video during Live Cycling Broadcasts.* APPLIED SCIENCES-BASEL, 14(20).

5. **Vanhaeverbeke, J.**, Deprost, E., Bonte, P., Strobbe, M., Nelis, J., Volckaert, B., Ongenae, F., Verstockt, S., Van Hoecke, S. (2023). *Real-Time Estimation*

[3]The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: Articles included in one of the Web of Science databases 'Science Citation Index', 'Social Science Citation Index' or 'Arts and Humanities Citation Index'. Limited to the publications document type article, review, letter, note, proceedings paper.

and Monitoring of COVID-19 Aerosol Transmission Risk in Office Buildings. SENSORS, 23(5).

1.7.2 Publications in International Conferences (Listed in the Science Citation Index [4])

1. Decorte, R., Paré, M., **Vanhaeverbeke, J.**, Taelman, J., Slembrouck, M., & Verstockt, S. (2024). *Multi-modal hit detection and positional analysis in padel competitions*. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 3306–3314.

1.7.3 Publications in International Conferences

1. Le Sommer, A., **Vanhaeverbeke, J.**, Decorte, R., Slembrouck, M., & Verstockt, S. (2024). *Video-based detection of hurdle crossings by athletes to gather intermediate race timings*. Proceedings from the 15th International Conference on the Engineering of Sport (ISEA 2024).

2. Verstockt, S., De Bock, T., De Bock, J., Decorte, R., **Vanhaeverbeke, J.**, & Slembrouck, M. (2023). *Sensor-driven recording and annotation of dry slope jumps*. Abstract Book of the 9th International Congress on Science and Skiing, 63–64.

1.7.4 Publications in National Conferences

1. **Vanhaeverbeke, J.**, Slembrouck, M., & Verstockt, S. (2023). *Helicopter video geolocalization for cycling races*. Faculty of Engineering and Architecture Research Symposium (FEARS).

[4] The publications listed are recognized as 'P1 publications', according to the following definition used by Ghent University: P1: Proceedings included in one of these Web of Science indexes: 'Conference Proceedings Citation Index - Science' of 'Conference Proceedings Citation Index - Social Science and Humanities'. Limited to publications document type: article, review, letter, note, proceedings paper, with exception of publications classified A1.

References

[1] M. Yu, M. Bambacus, G. Cervone, K. Clarke, D. Duffy, Q. Huang, J. Li, W. Li, Z. Li, Q. Liu, B. Resch, J. Yang, and C. Yang. *Spatiotemporal event detection: a review*. 13:1339–1365, December 2020. doi:10.1080/17538947.2020.1738569.

[2] O. Janssens. *Data-driven performance monitoring, fault detection and dynamic dashboards for offshore wind farms*. dissertation, Ghent University, 2017. Available from: http://hdl.handle.net/1854/LU-8525426.

[3] J. De Bock. *Data-driven performance and safety analysis in cycling races*. dissertation, Ghent University, 2023. Available from: http://hdl.handle.net/1854/LU-01HD65KBVMGX1K8G1RHSX6JBYG.

[4] Whoop. *WHOOP Closes $12M, Announces Availability of its Performance Optimization System Exclusively Designed for Elite Athletes and Teams*, September 2015. Available from: https://www.whoop.com/us/en/press-center/whoop-closes-12m-announces-availability-of-its-performance-optimization-system-exclusively-designed-for-elite-athletes-and-teams.

[5] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, 60(2):91–110, November 2004. doi:10.1023/B:VISI.0000029664.99615.94.

[6] S. Baker and I. Matthews. *Lucas-Kanade 20 Years On: A Unifying Framework*. International Journal of Computer Vision, 56(3):221–255, February 2004. doi:10.1023/B:VISI.0000011205.11775.fd.

[7] M. M. Taye. *Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions*. Computation, 11(33):52, March 2023. doi:10.3390/computation11030052.

[8] Z. Qin, F. Yu, C. Liu, and X. Chen. *How convolutional neural networks see the world — A survey of convolutional neural network visualization methods*. Mathematical Foundations of Computing, 1(2):149–180, May 2018. doi:10.3934/mfc.2018008.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *ImageNet classification with deep convolutional neural networks*. Communications of the ACM, 60(6):84–90, May 2017. doi:10.1145/3065386.

[10] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition*. In 2016 IEEE Conference on Computer Vision and Pattern Recognition

(CVPR), page 770–778, Las Vegas, NV, USA, June 2016. IEEE. Available from: http://ieeexplore.ieee.org/document/7780459/, doi:10.1109/CVPR.2016.90.

[11] M. Tan and Q. V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. (arXiv:1905.11946), September 2020. arXiv:1905.11946. Available from: http://arxiv.org/abs/1905.11946, doi:10.48550/arXiv.1905.11946.

[12] S. Ren, K. He, R. Girshick, and J. Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6):1137–1149, June 2017. doi:10.1109/TPAMI.2016.2577031.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), page 779–788, Las Vegas, NV, USA, June 2016. IEEE. Available from: http://ieeexplore.ieee.org/document/7780460/, doi:10.1109/CVPR.2016.91.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. *Mask R-CNN*. (arXiv:1703.06870), January 2018. arXiv:1703.06870. Available from: http://arxiv.org/abs/1703.06870, doi:10.48550/arXiv.1703.06870.

[15] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, page 234–241, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-24574-4_28.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. *Pyramid Scene Parsing Network*. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), page 6230–6239, July 2017. Available from: https://ieeexplore.ieee.org/document/8100143, doi:10.1109/CVPR.2017.660.

[17] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen. *Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation*. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), page 12472–12482, Seattle, WA, USA, June 2020. IEEE. Available from: https://ieeexplore.ieee.org/document/9156495/, doi:10.1109/CVPR42600.2020.01249.

[18] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. *UPSNet: A Unified Panoptic Segmentation Network*. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), page 8810–8818, Long Beach,

CA, USA, June 2019. IEEE. Available from: https://ieeexplore.ieee.org/document/8953750/, doi:10.1109/CVPR.2019.00902.

[19] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. *A Comprehensive Survey on Transfer Learning*. Proceedings of the IEEE, 109(1):43–76, January 2021. doi:10.1109/JPROC.2020.3004555.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. (arXiv:2010.11929), June 2021. arXiv:2010.11929. Available from: http://arxiv.org/abs/2010.11929, doi:10.48550/arXiv.2010.11929.

[21] A. Shin, M. Ishii, and T. Narihira. *Perspectives and Prospects on Transformer Architecture for Cross-Modal Tasks with Language and Vision*. International Journal of Computer Vision, 130(2):435–454, February 2022. doi:10.1007/s11263-021-01547-8.

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), page 9992–10002, Montreal, QC, Canada, October 2021. IEEE. Available from: https://ieeexplore.ieee.org/document/9710580/, doi:10.1109/ICCV48922.2021.00986.

[23] B. Cheng, A. G. Schwing, and A. Kirillov. *Per-Pixel Classification is Not All You Need for Semantic Segmentation*. (arXiv:2107.06278), October 2021. arXiv:2107.06278 [cs]. Available from: http://arxiv.org/abs/2107.06278, doi:10.48550/arXiv.2107.06278.

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. *Learning Transferable Visual Models From Natural Language Supervision*. (arXiv:2103.00020), February 2021. arXiv:2103.00020. Available from: http://arxiv.org/abs/2103.00020, doi:10.48550/arXiv.2103.00020.

[25] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin. *Emerging Properties in Self-Supervised Vision Transformers*. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), page 9630–9640, Montreal, QC, Canada, October 2021. IEEE. Available from: https://ieeexplore.ieee.org/document/9709990/, doi:10.1109/ICCV48922.2021.00951.

[26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. *Segment Anything*.

(arXiv:2304.02643), April 2023. arXiv:2304.02643 [cs]. Available from: http://arxiv.org/abs/2304.02643, doi:10.48550/arXiv.2304.02643.

[27] J. W. Cooley and J. W. Tukey. *An algorithm for the machine calculation of complex Fourier series*. Mathematics of computation, 19(90):297–301, 1965.

[28] J. Van Der Donckt, J. Van Der Donckt, E. Deprost, N. Vandenbussche, M. Rademaker, G. Vandewiele, and S. Van Hoecke. *Do not sleep on traditional machine learning: Simple and interpretable techniques are competitive to deep learning for sleep scoring*. Biomedical Signal Processing and Control, 81:104429, March 2023. doi:10.1016/j.bspc.2022.104429.

[29] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh. *Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets*. In 2016 IEEE 16th International Conference on Data Mining (ICDM), page 1317–1322, December 2016. Available from: https://ieeexplore.ieee.org/abstract/document/7837992?casa_token=fSuyyYrF1jIAAAAA:Gd7ydUhmdP_5AnERPDGBwa04Ava4T5EIFhvExFmkRHwGHd1RNksHV__Z8GqZUM8l1XD4e4WwEQDy, doi:10.1109/ICDM.2016.0179.

[30] D. De Paepe, S. Vanden Hautte, B. Steenwinckel, F. De Turck, F. Ongenae, O. Janssens, and S. Van Hoecke. *A generalized matrix profile framework with support for contextual series analysis*. Engineering Applications of Artificial Intelligence, 90:103487, April 2020. doi:10.1016/j.engappai.2020.103487.

[31] S. Hochreiter and J. Schmidhuber. *Long Short-Term Memory*. Neural Comput., 9(8):1735–1780, November 1997. doi:10.1162/neco.1997.9.8.1735.

[32] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. (arXiv:1412.3555), December 2014. arXiv:1412.3555. Available from: http://arxiv.org/abs/1412.3555, doi:10.48550/arXiv.1412.3555.

[33] M. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi. *Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model*. In 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), page 87–92, October 2020. Available from: https://ieeexplore.ieee.org/abstract/document/9257950, doi:10.1109/NILES50944.2020.9257950.

[34] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang. *A CNN-LSTM-Based Model to Forecast Stock Prices*. Complexity, 2020(1):6622927, January 2020. doi:10.1155/2020/6622927.

[35] C. Che, P. Zhang, M. Zhu, Y. Qu, and B. Jin. *Constrained transformer network for ECG signal processing and arrhythmia classification*. BMC Medical Informatics and Decision Making, 21(1):184, June 2021. doi:10.1186/s12911-021-01546-2.

[36] T. R. Gruber. *A translation approach to portable ontology specifications*. Knowledge Acquisition, 5(2):199–220, June 1993. doi:10.1006/knac.1993.1008.

[37] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor. *The SSN ontology of the W3C semantic sensor network incubator group*. 17:25–32, December 2012. doi:10.1016/j.websem.2012.05.003.

[38] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller. *Watson: beyond jeopardy!* Artif. Intell., 199–200(1):93–105, June 2013. doi:10.1016/j.artint.2012.06.009.

[39] P. Moens, S. Vanden Hautte, D. De Paepe, B. Steenwinckel, S. Verstichel, S. Vandekerckhove, F. Ongenae, and S. Van Hoecke. *Event-Driven Dashboarding and Feedback for Improved Event Detection in Predictive Maintenance Applications*. Applied Sciences, 11(2121):10371, January 2021. doi:10.3390/app112110371.

[40] Z. O'Leary. *Researching Real-World Problems: A Guide to Methods of Inquiry*. SAGE, 2005. Available from: https://us.sagepub.com/en-us/nam/researching-real-world-problems/book226862.

[41] A. L. Valentino and J. F. Juanico. *Overcoming Barriers to Applied Research: A Guide for Practitioners*. Behavior Analysis in Practice, 13(4):894–904, December 2020. doi:10.1007/s40617-020-00479-y.

[42] M. T. Ribeiro, S. Singh, and C. Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, page 1135–1144, New York, NY, USA, August 2016. Association for Computing Machinery. Available from: https://dl.acm.org/doi/10.1145/2939672.2939778, doi:10.1145/2939672.2939778.

[43] S. M. Lundberg and S.-I. Lee. *A unified approach to interpreting model predictions*. In Proceedings of the 31st International Conference on Neural In-

formation Processing Systems, NIPS'17, page 4768–4777, Red Hook, NY, USA, December 2017. Curran Associates Inc.

2

Event Detection at Machine Level: Flame Anomaly Detection in Steel Reheating Furnaces

As introduced in Chapter 1, events occur in virtually any domain, and the manufacturing sector is no exception. Manufacturing presents significant opportunities for automated event detection, with countless events stemming from the many processes involved. While some of these events, such as how many units were produced, may offer useful insights into productivity, other events like anomalies or faults are far more critical. These anomalous events should not remain undetected as identifying them early can prevent major issues, and ensure safe and smooth operations.

This chapter focuses on a case study of event detection at machine level (RG 2.2), conducted in collaboration with a major steel manufacturing company. In steel reheating furnaces, maintaining a controlled and stable flame production is essential to ensure the quality of the steel. Therefore, 24/7 monitoring is desired to allow for timely intervention when needed. Human supervision, however, is inefficient, lacks scalability, and may lead to overlooked anomalies.

To address this, a thermal camera was installed to capture a continuous video stream of the furnace interior. My proposed system accurately analyzes this stream in real-time (RG 1.2), not only to detect when an anomaly occurs but also to de-

termine the specific burner zone, providing valuable spatial context (RG 2.1). The limited labeled data (RG 1.3) and the need for explainability (RG 3.2), ruled out a purely black-box approach. Instead, a hybrid AI computer vision system was designed and developed, blending the strengths of black-box models with traditional techniques.

This chapter is a slightly adapted version of the following publication:

Vanhaeverbeke, J., Verstockt, S. & Van Hoecke, S. (2024). **Flame Monitoring and Anomaly Detection in Steel Reheating Furnaces Based on Thermal Video Using a Hybrid AI Computer Vision System**. Manuscript under review at Scientific Reports.

Abstract: Reheating furnaces are essential in steel manufacturing, ensuring steel reaches the optimal temperature for hot-rolling. Burners within these furnaces produce flames to maintain the necessary thermal conditions. However, inconsistent burner performance can result in irregular or extreme flames, compromising steel quality and production safety. Traditionally, flame monitoring has relied on human supervision, which is inefficient and prone to errors. To overcome these limitations, we propose a computer vision-based system for automated flame monitoring and anomaly detection. The system analyzes the video stream from a thermal camera that continuously monitors the furnace interior. Our methodology involves three steps: (1) detecting flames and furnace keypoints using a deep learning model, (2) quantifying flames across burner regions with traditional computer vision techniques, and (3) identifying anomalies using an interpretable machine learning model. Validation with real-world data from a large steel manufacturing facility demonstrates that the system achieves an F1 score above 80% in detecting anomalies across various burner zones. To support operators, the results are presented in a dashboard that provides both real-time and historical insights into furnace performance. This enables timely anomaly detection and intervention, ensuring safe, efficient, and high-quality steel production.

2.1 Introduction

The steel industry is a cornerstone of the manufacturing sector, producing essential materials such as steel sheets, bars, beams, and wires. These materials are important for a wide range of applications, from constructing buildings to manufacturing automobiles. The production of these steel base products, such as sheets and beams, involves a process known as rolling [1], which shapes the steel

into various forms with specific mechanical properties. Initially, steel is hot rolled by heating it above its recrystallization temperature and then passing it through a series of rollers to achieve the desired shape and dimensions. In some cases, the hot-rolled steel undergoes additional cold rolling to further refine its dimensions, surface quality, and strength.

The heating process prior to hot rolling is conducted in reheating furnaces equipped with multiple burners across various zones. Given the substantial heat generation involved, precise control is important, as excessive and unstable flame production can compromise steel quality and pose safety risks. Consequently, early detection of such events is essential to enable timely operator intervention. Traditionally, burner anomalies are identified through observing significant deviations in temperature measurements or conducting visual inspections of the furnace. However, this approach requires constant human supervision, which can be resource-intensive and may lead to missed anomalous events. To overcome these limitations, a thermal camera is installed inside the furnace which facilitates direct and automated monitoring of flames.

Several approaches can be employed to design such solution. Traditional computer vision algorithms can detect flames in the furnace, but practical implementation may encounter issues due to data variability, such as noise, illumination changes, and flame reflections on steel surfaces [2, 3]. A more robust approach involves learning-based methods, like autoencoders, which detect anomalies from raw video footage [4, 5]. Although they have shown to be effective, autoencoders offer limited insight into their decision-making process which is required for smoother adoption and acceptance by the operators. Alternatively, a semantic segmentation model can identify flames, allowing for anomaly detection based on the flame masks [6], which has the benefit of providing visually explainable outputs.

In this work, we not only predict flame masks using a segmentation model but we introduce a three-step computer vision methodology for flame monitoring and anomaly detection. This approach integrates the segmentation model with traditional computer vision techniques to further refine flame predictions, assigning each flame to its respective burner region for more detailed spatial analysis. Additionally, we employ an interpretable machine learning model to detect anomalies within each burner region, providing operators actionable insights and increasing trust in the system's decisions. By combining machine learning with traditional computer vision, this hybrid methodology leverages the strengths of both techniques.

The entire framework is integrated into a real-time monitoring and anomaly detection system designed for steel reheating furnaces. This system provides operators with a dashboard to monitor historical and real-time flame behavior across three zones, and receive alerts about anomalous flame production. As a

result, operators can timely investigate and address potential issues, reducing the risk of damaged steel and improving production quality and consistency. The main contributions of this work include:

- a joint flame semantic segmentation and furnace keypoint detection model;
- a traditional computer vision pipeline to process flames and furnace keypoints, determining the flame quantity per burner region;
- decision stumps for fast and interpretable anomaly detection;
- a monitoring dashboard, indicating if, when, and where anomalies occur.

The following section provides an overview of related work on vision-based and sensor-based process monitoring within the steel industry. Next, Section 2.3 introduces the dataset of thermal videos captured from the furnace, along with the corresponding ground-truth data used in this study. Building on this, Section 2.4 details our proposed hybrid methodology for flame monitoring and anomaly detection. Afterwards, the results of both individual components and the integrated solution are presented and discussed in Section 2.5. Finally, Section 2.6 concludes the work and outlines potential directions for future research.

2.2 Related Work

2.2.1 Vision-Based Monitoring in the Steel Industry

The steel industry involves numerous processes that require close monitoring to ensure product quality, production efficiency, and workplace safety. Among these, reheating furnaces play a key role in the processing of steel. Despite their importance, to the best of our knowledge, no prior research has proposed vision-based solutions specifically designed for monitoring the flame production in steel reheating furnaces. However, there has been considerable progress in developing vision-based systems for other types of furnaces used in steel manufacturing.

Zhang et al. [7] introduced a vision-based system for analyzing flames in blast furnaces. Their work focuses on processing flame images to study temperature distributions and the flicker frequency of flames. The findings revealed significant temperature variations within raceways, which helped explain fluctuations in production quality and efficiency. This demonstrates the potential of flame analysis for improving furnace operations.

Similarly, Compais et al. [8] developed a visual monitoring system for blast furnaces, but their work focused on estimating oxygen concentration in flue gases.

Using images captured inside the furnace, they extracted features such as intensity and texture to train machine learning models, including logistic regression, support vector machines, and artificial neural networks. Their system enables early detection of abnormal combustion events, allowing operators to identify and address issues before they escalate, which aligns with the objective of our work.

In another study, Zhu et al. [9] proposed a vision-based approach for monitoring the operational status of blast furnaces using burden surface video footage. Their system extracts and integrates multilevel features, handcrafted features from high-temperature gas images, and sequential video features into a monitoring network. The approach is effective in detecting abnormal conditions such as hanging, collapsing, and irregular gas flow, ensuring smooth and stable furnace operation.

Patra et al. [10] developed a vision-based system for monitoring basic oxygen furnaces, specifically designed to detect slag in the tapping stream. Their approach leverages infrared imaging to distinguish slag from steel by using differences in emissivity between the two materials. The system generates alerts to minimize slag carry-over into ladles, which is important for preserving steel quality. Although their study focuses on slag detection rather than flame monitoring, their vision-based methodology also aims to ensure production quality.

Further, Selim et al. [11] concentrated on monitoring ladles in steel facilities. Their system uses thermal cameras to track ladle surface temperatures and identify ladle numbers. They combined traditional computer vision techniques with deep learning models such as Faster RCNN to provide a real-time monitoring solution capable of early anomaly detection. This combination of conventional and modern approaches is similar to our methodology, emphasizing practical and efficient monitoring systems.

2.2.2 Sensor-Based Monitoring in the Steel Industry

In addition to visual methods for monitoring and anomaly detection, sensor data can also be used as input for processing algorithms and machine learning models. A closely related work by Thai et al. [12] developed a flame monitoring system for steel reheating furnaces, focusing on optimizing burner performance. The researchers employed fiber-optic sensors to capture flame radiation characteristics over a wide range of wavelengths. They then used signal processing techniques in combination with a neural network to estimate two key burner performance indicators: excess air and nitrogen oxide emissions. Similarly, Bao et al. [13] worked on optimizing reheating furnaces by developing a multivariate linear regression model aimed at predicting and regulating furnace temperature.

Beyond reheating furnaces, research has been conducted on monitoring blast furnaces. Zhou et al. [14] employed principal component analysis (PCA) and independent component analysis (ICA) to monitor and diagnose abnormal furnace conditions. Meanwhile, Zhu et al. [15] developed a fault monitoring algorithm that combines PCA with a Gaussian mixture model (GMM). Agrawal et al. [16] contributed by estimating and visualizing the hearth liquid level in real-time, enabling operators to maintain better control and stability of the blast furnace.

2.2.3 Conclusion

Many related studies have been conducted on vision-based and sensor-based analyses for monitoring various processes within the steel industry. While some sensor-based solutions for reheating furnaces share a similar research focus with our work, they primarily concentrate on optimizing the combustion process. In contrast, our research is dedicated to flame monitoring specifically for anomaly detection. To the best of our knowledge, there currently is no vision-based system that uses thermal video to monitor and detect anomalies of flames in steel reheating furnaces. This gap highlights the novelty and potential impact of our approach in enhancing safety and efficiency in steel production.

2.3 Data

The required heat for hot-rolling steel is generated through a combustion process that produces flames. To monitor these flames, the partnering company installed an AXIS thermal camera positioned at the entrance of the furnace. Our access was limited to a continuous stream of video footage, which is stored in 5-minute segments as H.264 encoded videos for easier handling. Consequently, there is no accompanying metadata regarding the camera's exposure settings or other configuration parameters.

2.3.1 Data Exploration

Examining the data reveals that a color palette has been applied to the thermal video footage, making it easier for humans to interpret. The palette used is similar to those employed by other cameras. Therefore, if a methodology can reliably process this common visualization of thermal images, it is adaptable to other thermal cameras as well.

Additionally, the collected data shows significant variation in flame, furnace, and camera characteristics, resulting in diverse visual appearances, as illustrated

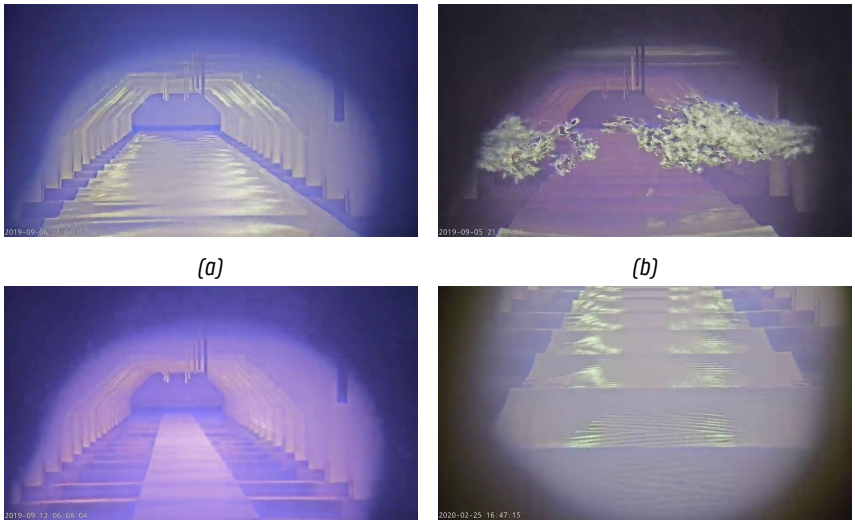


Figure 2.1: Example video frames illustrating the data variety, including (a) normal furnace operation, (b) high-intensity flames, (c) cold furnace, and (d) bad camera position.

in Figure 2.1. For example, it appears that the thermal camera automatically adjusts its exposure and/or tone mapping based on the intensity of the thermal radiation within the image to prevent oversaturation of high-intensity flames. While this adjustment preserves details in the flames, it simultaneously reduces the contrast in other areas of the furnace, as shown in Figure 2.1b. Images can also exhibit low contrast in situations where the furnace is relatively cold, such as Figure 2.1c.

Moreover, the camera's position introduces further variability. The thermal camera is not entirely stable, exhibiting both short-term and long-term movement. Short-term variations include a constant, subtle up-and-down stuttering motion due to vibrations. Long-term movement involves changes in the camera angle over time, eventually requiring manual re-adjustment, as seen in Figure 2.1d. Consequently, the camera's position can also be changed by operator interventions.

These variations highlight the need for the designed methodology to be robust and capable of handling diverse conditions within the furnace environment.

2.3.2 Labeled Dataset

To create a dataset for training and validation of the various algorithms, a balanced and representative selection of videos is made. These videos show flames of different sizes under various furnace conditions. For each video, a subset of frames (e.g., Figure 2.2a) is annotated with three types of labels per frame. First, a pixel-level flame mask is created (Figure 2.2b). Second, 12 furnace keypoints are marked at important locations along the front and back edges of the furnace (Figure 2.2c). Keypoints occluded by flames are excluded from the annotations. Since the camera position shifts over time, this keypoint labeling process is repeated for each frame. Finally, anomaly labels are added, indicating whether an anomaly is present in one of the three burner regions, i.e., front, middle, or back (Figure 2.2d).

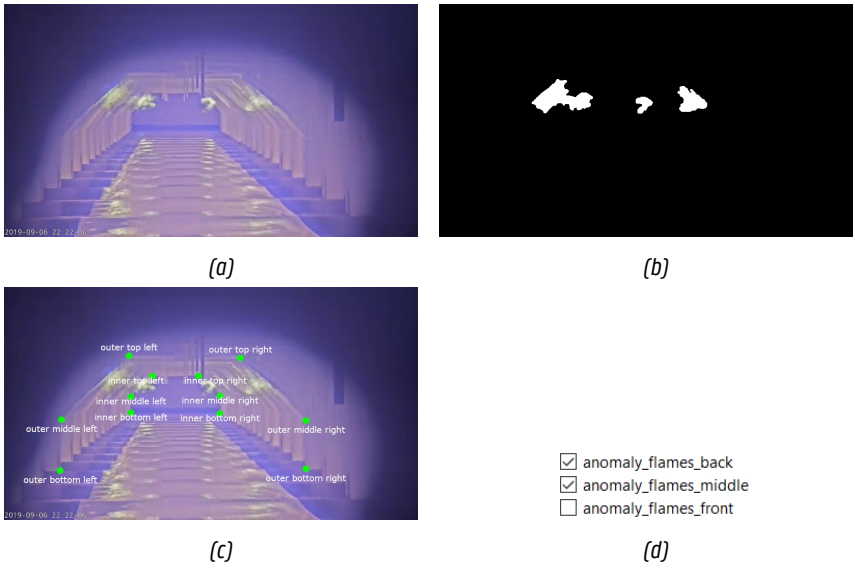


Figure 2.2: Example of the dataset illustrating (a) the thermal video frame, alongside (b) the corresponding flame mask, (c) furnace keypoints, and (d) flame anomaly status per burner region.

The labeling of the flames and keypoints is performed with high precision. To further ensure quality, all labels are reviewed by domain experts from the partnering company, who corrected inaccuracies when needed. Due to this rigorous labeling quality, the process required a significant effort per frame. Consequently, the final dataset consisted of 212 annotated frames sourced from 21 videos. Although the dataset size is relatively small, the frames encompass a wide range of

conditions, making them highly representative of potential scenarios within the furnace. A robust methodology should be able to leverage this limited yet diverse dataset to its fullest potential.

The dataset is divided into a training and validation set, containing 137 and 75 frames, respectively. This split is carefully chosen to ensure both subsets are representative of the data's diversity. To prevent bias in validation results, frames from the same video are not shared between the training and validation sets.

2.4 Methodology

The proposed anomaly detection methodology combines machine learning with traditional computer vision techniques, creating a hybrid approach that consists of three processing phases, as illustrated in Figure 2.3.

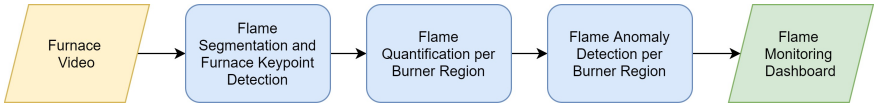


Figure 2.3: High-level overview of the furnace monitoring and anomaly detection pipeline.

In the first phase, a deep learning model is used to process thermal camera images, segmenting the flames within the furnace and detecting multiple furnace keypoints that serve as input for further analysis. In the second phase, the detected keypoints are processed using a traditional computer vision algorithm to construct three furnace regions, i.e., front, middle, and back. Subsequently, the segmented flames are assigned to their respective region and quantified, enabling more spatially fine-grained monitoring. In the final phase, anomaly detection is performed using a simple, fully interpretable machine learning model that evaluates the flame quantity in each region to identify anomalies. Dividing the methodology into multiple distinct phases ensures that the most suitable technique is applied to each step, resulting in an approach that is both accurate and capable of providing control and explainability for the generated results.

2.4.1 Joint Flame Semantic Segmentation and Furnace Keypoint Detection

Using traditional computer vision algorithms to detect the flames is not feasible due to the challenging and varying data, including flame reflections, noise, and

exposure changes. Hence, a deep learning model is designed and trained to perform two primary tasks: detecting all flames and identifying 12 furnace keypoints.

2.4.1.1 Model

The model architecture consists of a pretrained ResNet [17] encoder combined with a U-Net-style [18] decoder featuring two output heads, as shown in Figure 2.4. ResNet18, the smallest variant of the ResNet family, was selected as the backbone due to its balance of performance and efficiency. Larger configurations, such as ResNet34 and ResNet50, were also evaluated but showed comparable or slightly worse results. Additionally, ResNeSt50 [19], a ResNet-style network with improved layer structure, demonstrated marginally improved performance. Alternative backbone architectures, including VGG11 [20] and EfficientNetV2S [21], were tested as well, yielding slightly better results than ResNet18. Despite these findings, ResNet18 was ultimately selected as the backbone due to its minimal performance differences compared to other architectures, which did not result in any significant qualitative impact. Additionally, ResNet18 provides faster inference speeds and a considerably lower parameter count, effectively reducing computational demands and reducing the risk of overfitting. It should be noted, however, that many of the other evaluated backbones demonstrated strong performance and could be viable alternatives depending on the specific requirements and priorities of the use case. For a full comparison of backbone results, refer to Appendix 2.A.

The decoder used for the proposed model closely resembles the one used within the U-Net architecture, using skip connections from encoder to decoder in order to give it access to high-resolution information. The decoding blocks consist of an interpolation of the output of the previous block, concatenation with the output of the encoding block of the same resolution, a 3x3 convolution with batch normalization and ReLu, and a 1x1 convolution with batch normalization and ReLu.

To jointly predict the flame segmentation and furnace keypoints, the decoder is adapted to have two prediction heads, one for each task. The flame segmentation head is added at the top of the decoder and consists of an interpolation layer, to match the resolution of the input image, followed by a 1x1 convolution to predict the mask. The keypoint head does not regress the coordinates directly but predicts heatmaps containing a Gaussian peak since this is know to yield a higher accuracy and also gives extra insights in the prediction process of the model [22, 23]. Therefore, the keypoint head does a 1x1 convolution outputting 12 channels, one for each keypoint location, based on the decoder features with a four times lower resolution. Afterwards, postprocessing of the heatmaps is needed to extract the coordinates of the keypoints.

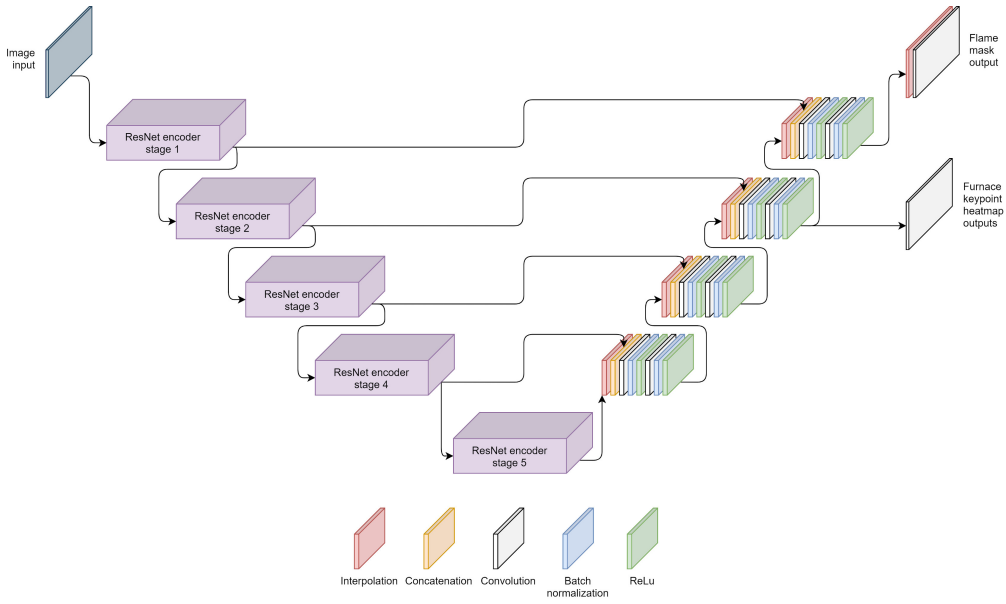


Figure 2.4: Architecture of the proposed joint semantic segmentation and keypoint detection model.

2.4.1.2 Multitask Loss Function

The proposed model predicts both the flame segmentation and furnace keypoints in one forward pass. To learn this multitask prediction, a custom loss function is used that quantifies the error for both tasks.

The loss for the semantic segmentation task consists of two metrics, namely the cross-entropy and Jaccard loss. The cross-entropy loss optimizes for pixel-wise confidence, while the Jaccard loss aims for an optimal intersection of the predicted and target mask. We found that combining both losses made training more stable and better generalized for the use case at hand.

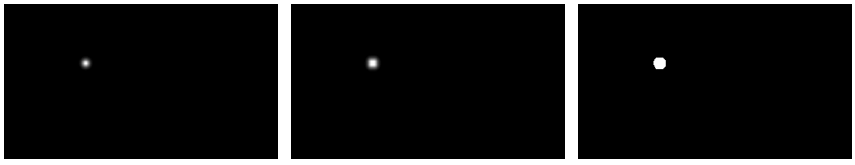
Training the model end-to-end on the error of the keypoint coordinates is not possible since they are predicted as heatmaps. Hence, the loss for the keypoint detection task is determined by calculating the mean squared error (MSE) between the predicted and target heatmaps. Using a regular MSE leads to slow and sub-optimal convergence since the size of the Gaussian peak is much smaller than the background area, resulting in a significant imbalance. This is solved by calculating a weighted MSE according to Equation (2.1) [24].

$$WMSE = \frac{1}{n}\sum_{i=1}^{n}((y_i - \hat{y}_i)^2 * (s * m_i + 1)) \tag{2.1}$$

where:

$WMSE =$ the weighted mean squared error (scalar)
y and $\hat{y} =$ the true and predicted values, respectively (tensor)
$m =$ the binary mask of keypoint regions (tensor)
$s =$ a scaling constant for the keypoint regions (scalar)
$n =$ the number of data points (scalar)

The weight mask is automatically generated by applying a small dilation on the target heatmap followed by a binary threshold, as visualized in Figure 2.5. The weight multiplier is a hyperparameter telling how much attention must be given to the peak location. For this case, a value of 10 was found to give the best results. This weighted MSE loss does not ignore the background region completely but gives the Gaussian peaks a higher importance which helps the model to learn better and faster.



(a) *(b)* *(c)*

Figure 2.5: The weighting mask for the mean squared error is automatically generated by (a) obtaining the ground truth Gaussian peak, (b) applying a dilation, and (c) thresholding the result to create a binary mask of keypoint regions.

The losses for both tasks are then combined via a weighted sum, as shown in Equation (2.2). The cross-entropy and Jaccard loss are given an equal weight. However, since the keypoint MSE has a much lower range of values, it is assigned a higher weight of 20 to equalize the importance of both tasks. These weights are determined experimentally to optimize performance across both tasks.

$$ML = 0.5 * CE_{seg} + 0.5 * J_{seg} + 20 * WMSE_{kp} \quad (2.2)$$

where:

ML = the multitask loss (scalar)
CE_{seg} = the cross-entropy loss of the flame segmentation masks (scalar)
J_{seg} = the Jaccard loss of the flame segmentation masks (scalar)
$WMSE_{kp}$ = the weighted MSE of the furnace keypoint heatmaps (scalar)

2.4.1.3 Data Preprocessing and Augmentations

Several transformations and augmentations are applied to the data before feeding it to the model, as illustrated in Figure 2.6. This process includes conventional transformations, such as resizing and normalizing the images, to ensure compatibility with an ImageNet-based pretrained backbone.

Additionally, the images are converted to grayscale. While the color palette applied to the original video frames enhances human interpretation, it introduces irrelevant and redundant information for deep learning models. This added complexity can cause the model to rely on color associations rather than focusing on the underlying patterns within the data. By converting the images to grayscale, this unnecessary complexity is removed, reducing the dimensionality of the input data and enabling the network to extract meaningful features more effectively. The benefits are evident in Appendix 2.B, showing that applying grayscaling significantly improved validation results for both tasks, demonstrating that the model learned more relevant and robust features.

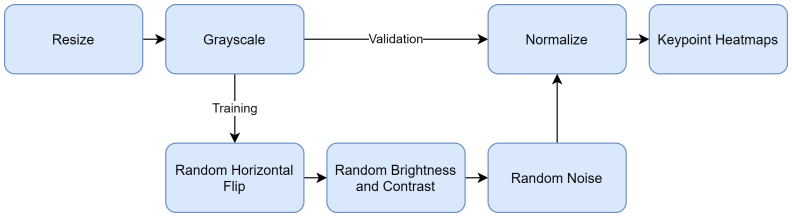


Figure 2.6: Data preprocessing and augmentation pipeline.

The keypoints require some preprocessing as well. Each keypoint is transformed into a heatmap featuring a Gaussian peak centered at the keypoint location, as depicted in Figure 2.5a. For this study, 12 furnace keypoints are present, resulting in 12 heatmaps per image. Following previous research [25, 26], the

Gaussian peak's sigma is set to 2, providing a good balance between location accuracy and smoothness.

To increase data variation and regularization during training, runtime image augmentations [27] are employed. These include random adjustments to brightness, contrast, and noise levels. Additionally, a custom random horizontal flip augmentation is used, which maintains the correct meaning of left and right. This is achieved by flipping the image along with its segmentation mask and keypoint coordinates while remapping keypoint labels according to their new positions. Although these augmentations do not further increase model performance, they play an important role in reducing overfitting and narrowing the generalization gap, as can be seen in Appendix 2.B.

2.4.1.4 Training

The final model is trained for 100 epochs, which provides sufficient time for the model to converge. The training is executed in two stages. For the first stage, the ResNet18 backbone is kept frozen, and only the decoder is trained. After 75 epochs, the 3rd, 4th, and 5th ResNet encoder blocks are unfrozen and fine-tuned which boosts the performance of the model a little more. Throughout the entire training process, the Adam optimizer is employed with a learning rate of 0.001. Other learning rates were tested but resulted in a slower or less stable convergence.

2.4.2 Flame Quantification per Burner Region

The previous section introduced a machine learning model designed to detect flames and furnace keypoints. In this section, the output of that model is further processed through a series of algorithmic and traditional computer vision steps, as summarized in Figure 2.7. This processing pipeline is responsible for constructing the three furnace burner regions (i.e., front, middle, and back), assigning the flames to their respective regions, and quantifying the number of flames in each burner region. The details of each step in the pipeline will be explained in the following sections.

2.4.2.1 Inference of Missing Keypoints

Bad image contrast or occlusion by flames can lead to incorrect keypoint predictions. In order to recover some of those missing keypoints, a series of simple rule-based checks is applied relying on the rectangular relation of the *middle* and *bottom* keypoints. This relation allows us to make assumptions on the position of missing keypoints based on other known ones.

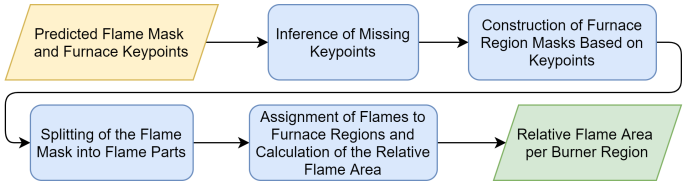
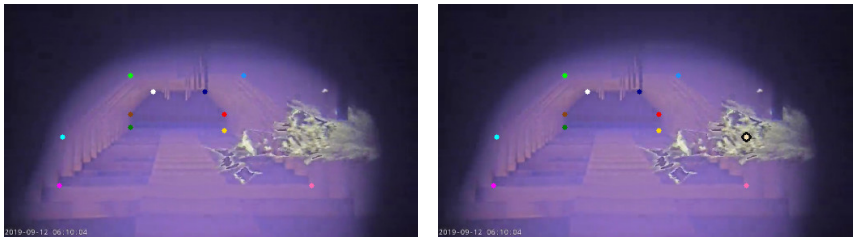


Figure 2.7: Overview of the computer vision pipeline for constructing the three burner regions and quantifying the flames per burner zone.

For example, when the *outer middle right* keypoint is missing and the *outer bottom right* and *outer middle left* keypoints are available, the *outer middle right* point can easily be inferred by combining the horizontal and vertical positions of the known keypoints, as shown in Figure 2.8. However, this technique will not be able to recover keypoints if too many are missing.



(a) *(b)*

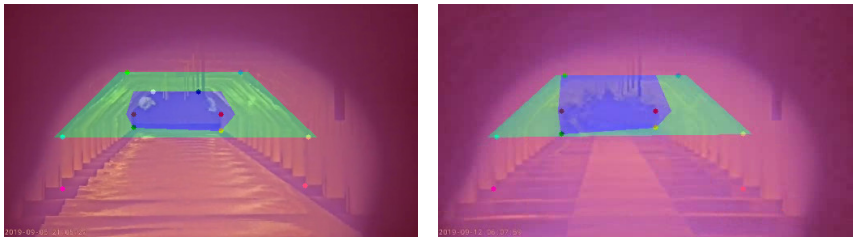
Figure 2.8: The outer middle right keypoint is (a) initially not predicted by the model but (b) successfully recovered using inference rules.

2.4.2.2 Construction of Furnace Region Masks Based on Keypoints

Ultimately, the flames in the front, middle and back region of the furnace must be quantified. Therefore, three region masks are generated based on the known furnace keypoints.

The back furnace region mask is primarily based on the information of the *inner* keypoints. If all of them are available, the region is built as the blue mask in Figure 2.9a. The bottom of the region is exactly equal to the *inner bottom* keypoints, and the middle and top are slightly moved outward to better cover the flames in the back region.

The green mask in Figure 2.9a covers the middle region of the furnace and is based on the four *top* and *middle outer* keypoints. The *outer bottom* keypoints are



(a) *(b)*

Figure 2.9: Examples of furnace region construction illustrating (a) the back (blue), middle (green), and front (red) burner regions when all keypoints are available, and (b) the fallback configuration for the back (blue) region in cases where the inner top keypoints are missing.

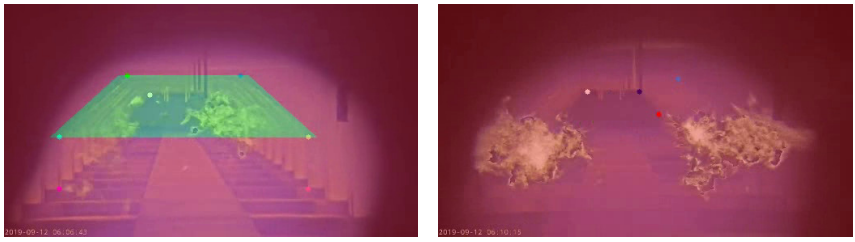
not included to build the mask since middle flames are not produced that close to the steel sheet. The complete region mask is however moved slightly outward to also cover flames that go just beyond the edge.

The front region mask (red in Figure 2.9a) is the easiest to build. This mask covers the entire frame and will capture all flames that are not assigned to any of the other region masks.

The procedure described above is executed when all keypoints are available, but this is not always the case. Even after the inference of missing keypoints, some might still be unavailable. To handle this, multiple fallback options are implemented for the different situations of missing keypoints that might occur. For example, when one of the *inner top* keypoints is missing, the back region is created based on the vertical position of the other known *inner top* keypoint and the horizontal positions of the *inner middle* and *bottom* keypoints. When both *inner top* keypoints are missing, the back region extends to the height of the *outer top* keypoints, as show in Figure 2.9b. Similar fallbacks are implemented for the *outer top* keypoints. When too much keypoints are missing for the construction of a region mask, it is discarded completely. This is often not an issue since it only happens when large parts of a region are occluded with flames from regions in front of it. In Figure 2.10a, the back region is not available because it is largely occluded by flames of the middle region, whereas in Figure 2.10b, only the front region is available.

2.4.2.3 Splitting of the Flame Mask into Flame Parts

The model introduced in Section 2.4.1 segments all flames in the image as one mask. As a consequence, the blobs of different flames can be merged if they are

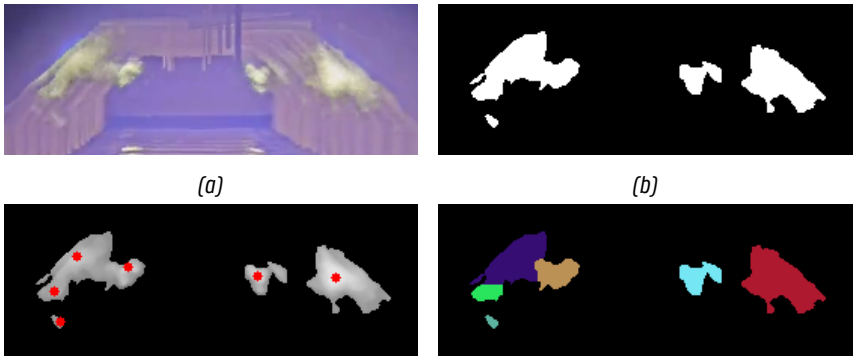


(a) *(b)*

Figure 2.10: Examples of furnace region construction when too many keypoints are missing, resulting in (a) the absence of the back (blue) region and (b) the absence of both the back (blue) and middle (green) regions.

large or close together. Assigning those merged flames as a whole to one region would lead to incorrect results. Alternatively, assigning the flames pixel-by-pixel to their intersecting region would be inaccurate as well since some of the flames have offshoots into other regions. Therefore, this step in the pipeline aims to split the flame mask into multiple smaller flames that can be correctly assigned to their region afterwards.

Based on our inspection of the available thermal video footage, we observed that the core of the flame emits the highest levels of infrared radiation. This intensity gradually diminishes toward the flame's edges, as illustrated in Figure 2.11a. Similar findings have been reported in other research involving the imaging of flames in the infrared spectrum [28, 29].



(a) *(b)*

(c) *(d)*

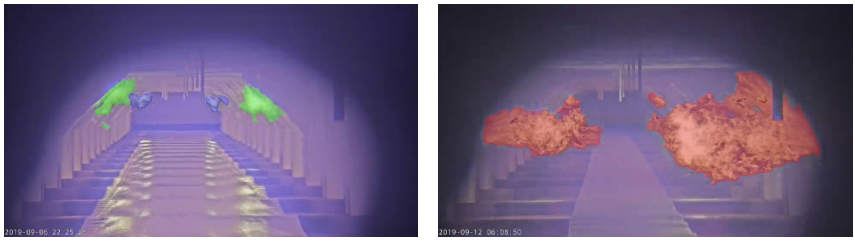
Figure 2.11: Visualizations of the flame splitting procedure, showing (a) the thermal image, (b) the flame mask, (c) the local peaks of the gray image, and (d) the flame mask split into regions using the Watershed algorithm.

This observation suggests that each high-intensity region corresponds to a different flame. Consequently, processing the intensity values by converting them to grayscale and applying a light median blur produces an ideal input for the Watershed transform [30]. This algorithm works by interpreting intensity values as a topographic landscape that is progressively "flooded" until the boundaries of different basins meet. These boundaries then define the segmentation regions. The seeds required for the Watershed transform are also determined from the grayscale image by identifying the local peaks.

An example of this procedure can be seen in Figure 2.11. The mask of two flames on the left are undesirably merged into one chunk (Figure 2.11b). After finding the local peaks (Figure 2.11c) and performing the Watershed transform as described above, the flame is spit into multiple flame parts, separating the overlapping flames (Figure 2.11d). Sometimes, too many splits are made, such as the bright green part in the example. Yet, this is not an issue since they will be assigned to the proper region in the next stage of the pipeline.

2.4.2.4 Assignment of Flames to Furnace Regions and Calculation of the Relative Flame Area

Once the furnace regions are constructed and the flame masks are split, the flames can be assigned to their corresponding furnace region. This assignment begins by calculating the intersection area between each flame and the various burner regions. A flame is then allocated to the region with which it shares the largest intersection. This procedure is repeated until all flames have been assigned to a specific region, as illustrated in Figure 2.12a.



(a) *(b)*

Figure 2.12: Examples of flame assignments, showing (a) flames allocated to the middle and back regions and (b) flames allocated to the front region.

To easily monitor the behavior of flames over time, the relative flame area for each burner region is calculated. This is done by dividing the flame area by the area of its respective burner region. Using relative rather than absolute flame

areas provides a more robust measurement that is less sensitive to variations in camera placement.

Additionally, an extra rule is implemented to improve the accuracy. If the relative flame area of the front burners exceeds a certain threshold, occlusion may prevent accurate assignment of flames to the middle and back regions. In such cases, all flames are reallocated to the front region, as shown in Figure 2.12b. The flame areas are then summed, and the relative flame area is recalculated. Although this adjustment may decrease the precision of the algorithm, it enhances its overall stability in practical applications.

2.4.3 Flame Anomaly Detection per Burner Region

The methodology outlined in the previous section quantified the flames in each burner region. Building on this, the final phase of the proposed methodology leverages this information to perform automatic anomaly detection using a lightweight and interpretable machine learning model. This automated anomaly detection plays a crucial role in alerting operators to any irregularities without requiring them to continuously monitor the furnace. These alerts enable operators to promptly investigate and address potential issues, ensuring smooth furnace operation and preventing damage to the steel.

2.4.3.1 Model

Figure 2.13 illustrates the relative flame area of frames across the complete dataset, plotted against their anomaly ground truth for the three burner regions. These plots demonstrate a clear distinction in flame area between normal and anomalous samples, indicating that the relative flame area is a highly informative feature for detecting anomalies. Notably, some samples labeled as anomalous have a relative flame area of zero. In such cases, the corresponding burner region could not be defined based on the available keypoints, leading the flames to be assigned to the region in front and contributing to the anomalous state over there.

Given the high discriminatory power of the relative flame area, a simple model suffices for anomaly classification. For this task, decision stumps are selected, which are a type of decision tree [31, 32] characterized by a single split or decision node. Decision stumps classify data into two groups based on a single feature and threshold, making them lightweight, fast, and, most importantly, fully interpretable.

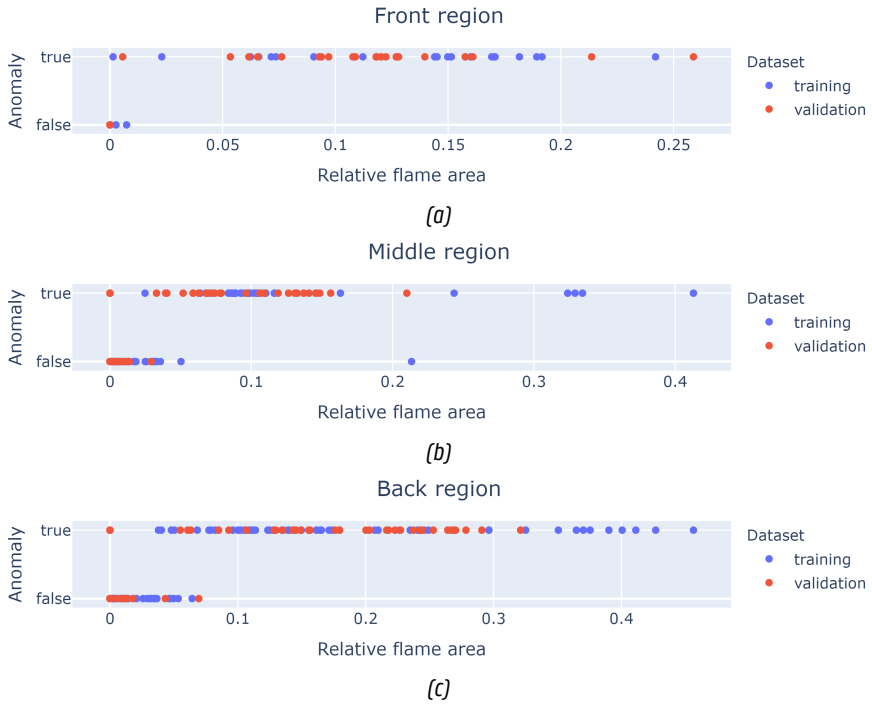


Figure 2.13: Scatter plots depicting the relationship between relative flame area and anomaly labels across the entire dataset for (a) the front region, (b) the middle region, and (c) the back region.

2.4.3.2 Training

As described in Section 2.3, the thermal camera frames in the dataset are labeled with three anomaly states, one for each burner region. Each state is binary, indicating whether an anomaly is present in that region or not. These labels serve as the basis for constructing the decision stumps.

A separate decision stump is created for each region, as a single stump would fail to generalize effectively across all three zones, as can be seen in Figure 2.13. The resulting decision stumps, designed to predict the anomaly status per region, are shown in Figure 2.14.

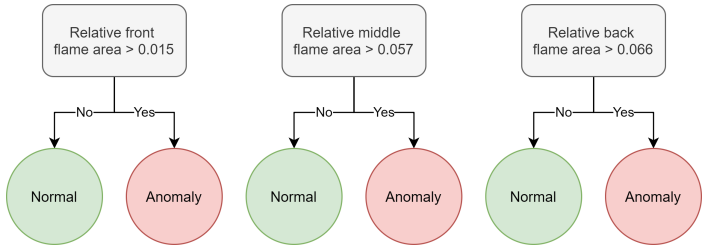


Figure 2.14: Graphical visualization of the three trained decision stumps, one for each burner region (front, middle and back).

2.5 Results and Discussion

This section presents and discusses the results in three parts. First, the performance of the flame segmentation and furnace keypoint detection model is evaluated and analyzed. Second, the anomaly detection results are presented, providing insights into the performance of the complete methodology. Finally, the integration of the methodology into a dashboard is described, including an overview of its visualizations and processing speed on standard hardware.

2.5.1 Joint Flame Semantic Segmentation and Furnace Keypoint Detection

Table 2.1 presents the performance of the final flame segmentation and furnace keypoint detection model evaluated using two metrics. The Jaccard index, which is the complement of the Jaccard loss used during training, validates the flame segmentation task. For keypoint detection, the percentage of correct keypoints (PCK) metric is used, as it offers a more intuitive validation compared to mean squared error (MSE). The PCK metric assesses whether the predicted keypoints lie within a specified distance from the target keypoints. In this study, this distance threshold is set to 1% of the image width. For both metrics, higher scores indicate better performance.

The model demonstrates solid performance with validation results exceeding 80%. The generalization gap between training and validation scores is limited, suggesting that the model is well-fitted and robust despite being trained on a small dataset. This is attributed to the use of a low-parameter model and relevant augmentations. To illustrate how these metrics translate into practice, several real-world examples are discussed below.

In Figure 2.15a, an example of normal furnace operation is presented, characterized by controlled and limited burner flames. The model accurately predicts

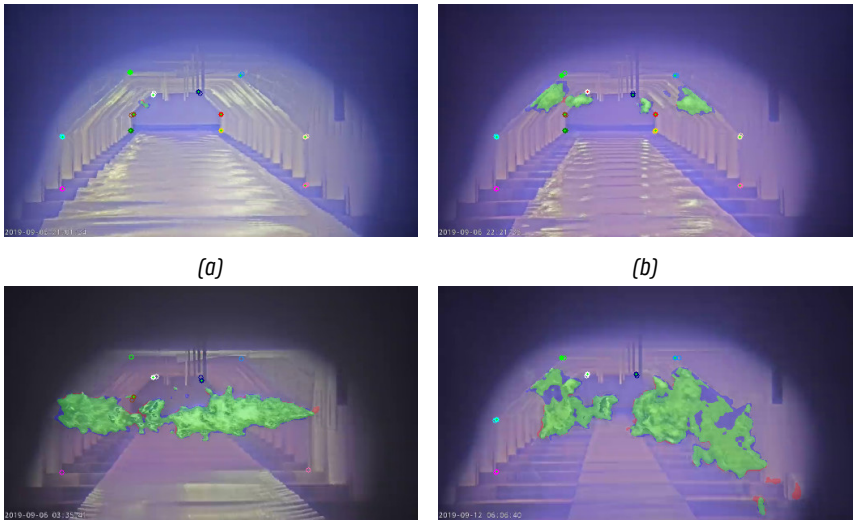
Table 2.1: Training and validation metrics of the final joint flame segmentation and furnace keypoint detection model.

Jaccard index (flame segmentation)		Percentage of correct keypoints (furnace keypoint detection)	
Training	**Validation**	**Training**	**Validation**
87.6%	82.8%	94.3%	90.7%

the flame mask, even in cases where the flames in the back are small, and successfully identifies all keypoints. Figure 2.15b depicts a scenario where the burners are operating more intensively, which may suggest potential instability in furnace operation. Despite this, the model effectively predicts both the flame mask and keypoints. However, one keypoint, the *inner top left*, is considered incorrect by the PCK metric since it was not annotated due to occlusion caused by the flames. Nonetheless, the prediction demonstrates the model's strong generalization capabilities.

Figure 2.15c showcases a situation where the front burners are active, which is undesirable as it could damage the steel. While the flame segmentation remains accurate, the limited visibility and reduced contrast result in missing keypoints. In the final example, shown in Figure 2.15d, there is excessive flame production across all burners. In this case, the flame segmentation accuracy is slightly lower because parts of the flames are highly transparent. Additionally, many keypoints are occluded, although the visible ones are correctly predicted.

In summary, the model performs almost perfectly under normal furnace operating conditions. However, excessive flame production can reduce visibility and contrast in thermal images, leading to less accurate keypoint predictions. Although augmentations such as brightness and contrast adjustments improve the model's robustness, they cannot fully overcome the challenges, as even human interpretation would be difficult under these conditions. Nevertheless, even when predictions are imperfect, such as in Figures 2.15c and 2.15d, subsequent pipeline steps can extract meaningful insights, as the presence of an anomalous event is evident. Missing keypoints are not necessarily problematic since their absence also conveys useful information, indicating that flames in preceding regions are causing occlusion, and it is more accurate to avoid constructing regions behind these flames altogether. Finally, an important observation is that reflections on the steel sheets are not mistakenly classified as flames, further demonstrating the reliability of the model.



(a) *(b)*

(c) *(d)*

Figure 2.15: Example model predictions on varying furnace conditions. The segmentation mask is visualized using three colors: green represents a true positive (TP) prediction, blue represents a false negative (FN) and red is a false positive (FP). The correctness of the 12 furnace keypoints is visualized using two circles each. The outlined circle shows the target location whereas the solid circle visualizes the predicted keypoint. The small red or green dot within this circle tells whether the prediction is seen as correct or not according to the PCK metric.

2.5.2 Flame Anomaly Detection per Burner Region

Building on the results of the flame segmentation and furnace keypoint detection model, this section assesses the anomaly detection performance of the proposed methodology. Table 2.2 provides a comparison of the training and validation F1 scores for the decision stumps applied to the three burner regions. These scores are calculated using two different strategies. The first approach leverages the ground truth flame mask and furnace keypoints, offering a more focused assessment of the anomaly detection performance. In contrast, the second approach employs predicted flame masks and furnace keypoints to evaluate the end-to-end performance of the proposed system.

When using the ground truth data, the front region flames are classified with high accuracy, achieving a validation F1 score of 98%. However, the middle and back regions exhibit slightly lower performances, with validation F1 scores of 88%

Table 2.2: Anomaly detection performance per burner region evaluated in two ways. First, using the ground truth flame mask and furnace keypoints giving a more isolated view on the performance of the anomaly detection. Second, using the trained model to predict the flame mask and furnace keypoints giving an end-to-end view on the performance of the proposed system.

	F1 score			
	Using ground truth flame mask and furnace keypoints		Using predicted flame mask and furnace keypoints	
Burner region	Training	Validation	Training	Validation
Front	97.4%	97.7%	97.4%	97.7%
Middle	90.0%	87.9%	93.5%	80.6%
Back	90.9%	88.8%	92.7%	83.1%

and 89%, respectively. Overall, this shows a robust classification capability of the anomaly detection model based on the relative flame area.

In the end-to-end evaluation scenario, where predicted data is used, the front region maintains its high validation F1 score of 98%. However, there is a noticeable decline in performance for the middle and back regions, with scores dropping to 81% and 83%, respectively. This decrease can be attributed to the dependence on keypoints that are not consistently available when predicted by the model. The front region does not experience this drop in performance because its classification does not rely on keypoints for construction of its region.

In all cases, the anomaly detection F1 score for the front region consistently surpasses those of the middle and back regions. This can be explained by the larger size of front flames, which creates a more pronounced distinction between normal and anomalous operations, thereby simplifying the classification task. Despite some challenges in the middle and back regions, these results show the effectiveness of using decision stumps to detect flame anomalies.

2.5.3 Flame Monitoring Dashboard

The presented methodology is integrated as a system, going from video footage to anomaly detection, which visualizes the results through a dashboard for operators. 2 frames per second are processed to reduce the computational needs and allow for real-time execution on CPU. This is however sufficient to produce an insightful data stream to monitor the furnace. Some examples of the furnace

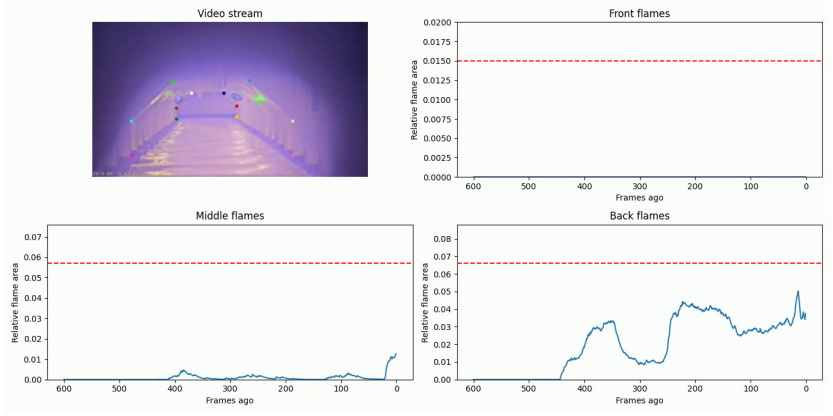
monitoring dashboard can be seen in Figure 2.16. The interface is divided into four sections visualizing different pieces of information. The top left corner shows the current processed frame of the video stream together with the predicted furnace keypoints and flames assigned to their respective region. The other three plots present the relative flame area over time for the front, middle and back burner regions. The data of these plots, in blue, are smoothed exponentially to make the signal more stable while the horizontal red line denotes the anomaly threshold learned by the decision stumps of Section 2.4.3. This allows the operator to easily see if, when and how much the flame area exceeds the threshold.

A normally operating blast furnace usually has some small and controlled flames in the middle and back to keep the furnace at temperature, as seen in Figure 2.16a. This is also reflected by the time series plots of the middle and back relative flame area which have rather low values, far below the anomaly threshold. Sudden blasts of flames and burners that work too hard are clearly noticeable in the data, as is the case for the middle flames in Figure 2.16b. Large flames in the front of the furnace are never desired. Figure 2.16c shows such an event where the data clearly shows the intervals and amount of flames that were detected in the front of the furnace.

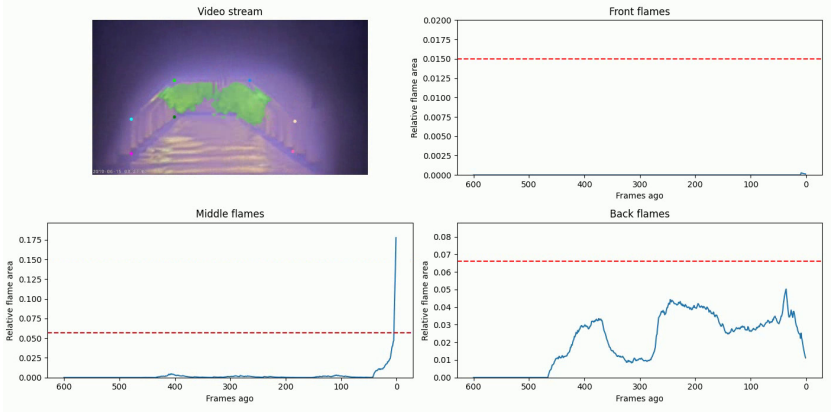
2.6 Conclusions and Future Work

The production of flames in steel reheating furnaces often deviates from the expected combustion process, which can impact efficiency and quality. To address this issue, a computer vision system has been developed to monitor flames and detect anomalies in three distinct burner zones using thermal camera footage. This system comprises three main components. First, a joint semantic segmentation and keypoint detection model is employed to extract a flame mask and identify 12 keypoints within the furnace. Second, a traditional computer vision pipeline quantifies the flames in each burner region. Third, decision stumps are utilized to detect anomalies in flame production. These components are integrated and displayed on a monitoring dashboard for real-time analysis.

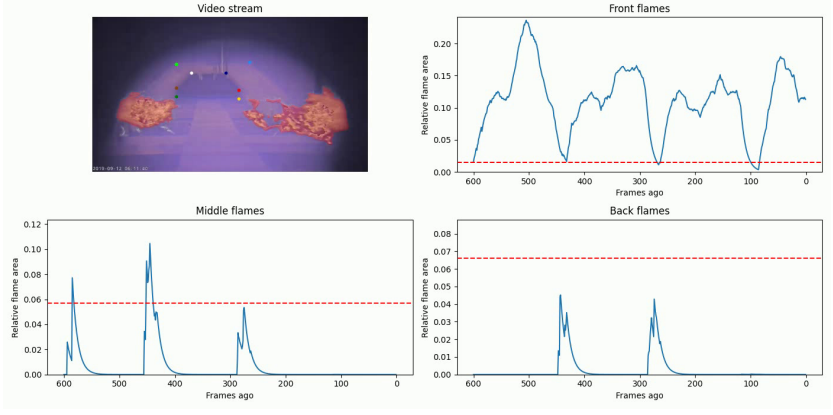
The system's performance has been validated with promising results, demonstrating its potential application in an industrial setting. The flame segmentation model achieved a Jaccard index of 82.8%, while the furnace keypoints were predicted 90.7% correct (PCK). The end-to-end anomaly detection evaluation yielded F1 scores of 97.7%, 80.6%, and 83.1% for the front, middle, and back regions, respectively. Despite these strong results, the model occasionally struggles with low-contrast thermal images. However, this challenge do not significantly affect the overall effectiveness of the monitoring and anomaly detection outputs.



(a)



(b)



(c)

Figure 2.16: Examples of the monitoring dashboard during different furnace conditions, including (a) normal furnace operation, (b) sudden overactivity of middle burners, and (c) uncontrolled front flames.

Several possibilities for future research exist. Currently, only one furnace is equipped with a thermal camera. While every aspect of the proposed methodology was designed to be as generic and robust as possible, expanding the dataset to include footage from additional furnaces would allow for validation across different environments. Additionally, exploring deep learning architectures that exploit the temporal nature of the video stream could further enhance model performance.

In conclusion, this work enables the automatic monitoring of steel reheating furnaces. The promising results demonstrate the system's potential in the steel industry, contributing to more efficient manufacturing and consistent product quality.

2.A Appendix: Flame Semantic Segmentation and Furnace Keypoint Detection Results using Different Backbones

	Jaccard index (flame segmentation)		Percentage of correct keypoints (furnace keypoint detection)		Inference speed	Number of parameters
Backbone	**Training**	**Validation**	**Training**	**Validation**	**Intel Core i7-8650U**	
ResNet18	87.6%	82.8%	94.3%	90.7%	**3.2 FPS**	**14M**
ResNet34	87.5%	83.4%	93.8%	89.1%	2.5 FPS	24M
ResNet50	87.2%	81.1%	93.7%	90.3%	0.7 FPS	73M
ResNeSt50	84.0%	**84.3%**	93.6%	91.1%	0.4 FPS	75M
VGG11	88.4%	83.6%	95.7%	**92.0%**	1.5 FPS	19M
EffNetV2S	88.9%	83.2%	96.5%	91.1%	1.5 FPS	32M

2.B Appendix: Flame Semantic Segmentation and Furnace Keypoint Detection Results using ResNet18 Backbone with Different Preprocessing and Augmentations

	Jaccard index (flame segmentation)		Percentage of correct keypoints (furnace keypoint detection)	
Preprocessing	**Training**	**Validation**	**Training**	**Validation**
Resize and normalize	91.1%	79.3%	97.5%	81.7%
+ Grayscale	91.5%	**83.1%**	97.3%	89.1%
+ Random horizontal flip, brightness, contrast and Gaussian noise	87.6%	82.8%	94.3%	**90.7%**

References

[1] grobdev. *Hot Rolled Steel vs. Cold Rolled Steel - What's the Difference?*, Jul 2020. Available from: https://www.grobinc.com/blog/hot-rolled-steel-vs-cold-rolled-steel/.

[2] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboğlu, B. U. Töreyin, and S. Verstockt. *Video fire detection – Review*. Digital Signal Processing, 23(6):1827–1843, Dec 2013. doi:10.1016/j.dsp.2013.07.003.

[3] S. Geetha, C. S. Abhishek, and C. S. Akshayanat. *Machine Vision Based Fire Detection Techniques: A Survey*. Fire Technology, 57(2):591–623, Mar 2021. doi:10.1007/s10694-020-01064-z.

[4] T. Qiu, M. Liu, G. Zhou, L. Wang, and K. Gao. *An Unsupervised Classification Method for Flame Image of Pulverized Coal Combustion Based on Convolutional Auto-Encoder and Hidden Markov Model*. Energies, 12(1313):2585, January 2019. doi:10.3390/en12132585.

[5] Z. Xu, Y. Guo, and J. H. Saleh. *Advances Toward the Next Generation Fire Detection: Deep LSTM Variational Autoencoder for Improved Sensitivity and Reliability*. IEEE Access, 9:30636–30653, 2021. doi:10.1109/ACCESS.2021.3060338.

[6] J. Großkopf, J. Matthes, M. Vogelbacher, and P. Waibel. *Evaluation of Deep Learning-Based Segmentation Methods for Industrial Burner Flames*. Energies, 14(66):1716, Jan 2021. doi:10.3390/en14061716.

[7] R. Zhang, Y. Cheng, Y. Li, D. Zhou, and S. Cheng. *Image-Based Flame Detection and Combustion Analysis for Blast Furnace Raceway*. IEEE Transactions on Instrumentation and Measurement, 68(4):1120–1131, April 2019. doi:10.1109/TIM.2017.2757100.

[8] P. Compais, J. Arroyo, F. Tovar, V. Cuervo-Piñera, and A. Gil. *Promoting the valorization of blast furnace gas in the steel industry with the visual monitoring of combustion and artificial intelligence*. Fuel, 362:130770, April 2024. doi:10.1016/j.fuel.2023.130770.

[9] J. Zhu, W. Gui, Z. Chen, and Z. Jiang. *Monitoring Multiple Operational Statuses of Blast Furnace via Multi-Feature Fusion from Burden Surface Video Images*. IEEE Transactions on Instrumentation and Measurement, page 1–1, 2025. doi:10.1109/TIM.2025.3541708.

[10] P. Patra, A. Sarkar, and A. Tiwari. *Infrared-based slag monitoring and detection system based on computer vision for basic oxygen furnace*. Ironmaking & Steelmaking, 46(7):692–697, Aug 2019. doi:10.1080/03019233.2018.1460909.

[11] M. Selim, P. L. de Uralde, J. Mata, E. Gorostegui-Colinas, B. Chicote, A. Pagani, and D. Stricker. *Vision-Based Ladle Monitoring System for Steel Factories*. In A. Wagner, K. Alexopoulos, and S. Makris, editors, Advances in Artificial Intelligence in Manufacturing, page 185–194, Cham, 2024. Springer Nature Switzerland. doi:10.1007/978-3-031-57496-2_19.

[12] S. M. Thai, S. J. Wilcox, C. K. Tan, J. Ward, and G. Andrews. *Development of an intelligent flame monitoring system for steel reheating burners*. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy, 226:1014–1031, Dec 2012. doi:10.1177/0957650912458859.

[13] Q. Bao, S. Zhang, J. Guo, Z. Li, and Z. Zhang. *Multivariate linear-regression variable parameter spatio-temporal zoning model for temperature prediction in steel rolling reheating furnace*. Journal of Process Control, 123:108–122, Mar 2023. doi:10.1016/j.jprocont.2023.01.013.

[14] P. Zhou, R. Zhang, J. Xie, J. Liu, H. Wang, and T. Chai. *Data-Driven Monitoring and Diagnosing of Abnormal Furnace Conditions in Blast Furnace Ironmaking: An Integrated PCA-ICA Method*. IEEE Transactions on Industrial Electronics, 68(1):622–631, Jan 2021. doi:10.1109/TIE.2020.2967708.

[15] X. Zhu, D. Gao, C. Yang, and C. Yang. *A blast furnace fault monitoring algorithm with low false alarm rate: Ensemble of greedy dynamic principal component analysis-Gaussian mixture model.* Chinese Journal of Chemical Engineering, Oct 2022. doi:10.1016/j.cjche.2022.09.012.

[16] A. Agrawal, S. C. Kor, U. Nandy, A. R. Choudhary, and V. R. Tripathi. *Real-time blast furnace hearth liquid level monitoring system.* Ironmaking & Steelmaking, 43(7):550–558, Aug 2016. doi:10.1080/03019233.2015.1127451.

[17] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition.* In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), page 770–778, Jun 2016. doi:10.1109/CVPR.2016.90.

[18] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Lecture Notes in Computer Science, page 234–241, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-24574-4_28.

[19] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola. *ResNeSt: Split-Attention Networks.* In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), page 2735–2745, Jun 2022. doi:10.1109/CVPRW56347.2022.00309.

[20] K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition.* In 3rd International Conference on Learning Representations (ICLR 2015), page 1–14. Computational and Biological Learning Society, 2015. Available from: https://ora.ox.ac.uk/objects/uuid:60713f18-a6d1-4d97-8f45-b60ad8aebbce, doi:10.48550/arXiv.1409.1556.

[21] M. Tan and Q. Le. *EfficientNetV2: Smaller Models and Faster Training.* In Proceedings of the 38th International Conference on Machine Learning, page 10096–10106. PMLR, July 2021. Available from: https://proceedings.mlr.press/v139/tan21a.html.

[22] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. *Integral Human Pose Regression*, volume 11210 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2018. doi:10.1007/978-3-030-01231-1_33.

[23] Q. Dang, J. Yin, B. Wang, and W. Zheng. *Deep learning based 2D human pose estimation: A survey.* Tsinghua Science and Technology, 24(6):663–676, Dec 2019. doi:10.26599/TST.2018.9010100.

[24] X. Wang, L. Bo, and L. Fuxin. *Adaptive Wing Loss for Robust Face Alignment via Heatmap Regression*. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), page 6970–6980, Oct 2019. doi:10.1109/ICCV.2019.00707.

[25] B. Xiao, H. Wu, and Y. Wei. *Simple Baselines for Human Pose Estimation and Tracking*. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, Computer Vision – ECCV 2018, Lecture Notes in Computer Science, page 472–487, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-030-01231-1_29.

[26] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu. *AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[27] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. *Albumentations: Fast and Flexible Image Augmentations*. Information, 11(2), 2020. doi:10.3390/info11020125.

[28] G. Parent, Z. Acem, S. Lechêne, and P. Boulet. *Measurement of infrared radiation emitted by the flame of a vegetation fire*. International Journal of Thermal Sciences, 49(3):555–562, Mar 2010. doi:10.1016/j.ijthermalsci.2009.08.006.

[29] A. Huot, M.-A. Gagnon, K.-A. Jahjah, P. Tremblay, S. Savary, V. Farley, P. Lagueux, E. Guyot, M. Chamberland, and F. Marcotte. *Time-resolved multispectral imaging of combustion reaction*. In S.-J. T. Hsieh and J. N. Zalameda, editors, Thermosense: Thermal Infrared Applications XXXVII, volume 9485, page 94851C. SPIE, 2015. Backup Publisher: International Society for Optics and Photonics. doi:10.1117/12.2177258.

[30] L. Vincent and P. Soille. *Watersheds in digital spaces: an efficient algorithm based on immersion simulations*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(6):583–598, Jun 1991. doi:10.1109/34.87344.

[31] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression Trees*. Routledge, 1 edition, 1984. doi:10.1201/9781315139470.

[32] J. R. Quinlan. *Induction of decision trees*. Machine Learning, 1(1):81–106, Mar 1986. doi:10.1007/BF00116251.

3

Event Detection at Room Level: Cross-Room CO_2-based Presence Detection

While the previous chapter explored event detection using video streams, this chapter shifts the attention to environmental sensors as a data source. These sensors, which are commonly integrated into modern buildings, measure parameters such as temperature, humidity, and CO_2 levels. They are cost-effective, widely available, and provide valuable data for various event detection applications within the building management domain.

One application of such data is the estimation of human presence in a room (RG 2.1 and 2.2), a capability that offers multiple practical benefits. For instance, presence detection can be used to directly control room appliances (RG 1.2), such as turning off lights or adjusting heating when a space is unoccupied for a certain period. However, in the context presented here, the presence events are used to enable long-term occupancy profiling. By analyzing patterns of room usage over time, insightful scheduling recommendations can be made to improve the efficiency of building control systems. This shows that automatic event detection is not always used to immediately act upon, but can also contribute to long-term optimization.

The research methodology proposed in this chapter utilizes a window-based, learning-driven approach to infer human presence from CO_2 sensor data. While other environmental and building-related measurements may also contribute to presence detection, CO_2 is widely monitored in buildings, making this methodology broadly applicable by focusing solely on CO_2 data. Nevertheless, similar to the case study presented in Chapter 2, obtaining accurate ground truth data for this task poses significant challenges. From a research perspective, this constraint requires working with a limited dataset (RG 1.3). In real-world deployments, it highlights the need to avoid reliance on labeled data for fine-tuning the model for each individual room within a building (RG 3.1). This is where our work stands apart from other similar existing research. We emphasize robust, real-world presence detection capable of adapting to changing environments. To address these challenges, this work introduces a cross-room methodology with optimized normalization, allowing the model to better generalize to new, unseen rooms in an unsupervised manner.

This chapter is a slightly adapted version of the following publication:

Vanhaeverbeke, J., Deprost, E., Verstockt, S. & Van Hoecke, S. (2024). **Cross-Room CO2-based Presence Detection for Occupancy Profiling**. Manuscript under review at IEEE Access.

Abstract: Control systems for building services, such as heating and cooling, often rely on fixed timing schemes. While such approaches are convenient, they make strong assumptions about room usage, often leading to inadequate comfort and energy efficiency. This study addresses this limitation by presenting two contributions aimed at automating the configuration of building control systems. The first contribution involves the development of a presence detection model based on CO_2 data, which is easy to measure and non-privacy intrusive. Unlike existing literature, which typically focuses on single-room applications, this work introduces a dataset and machine learning methodology demonstrating the generalizability of a presence detection model across various real-world rooms, even among different building types. Sliding window normalization of the sensor data is the key to achieve this unsupervised cross-room adaptability. As second contribution, we propose an occupancy profiling technique that relies on the predicted presence information. This approach facilitates the automated configuration of building control systems by using historical presence probabilities to anticipate future occupancy. In contrast to fixed timing schemes, these occupancy profiles dynamically adapt over time, accommodating changes in occupant behavior. As such, this work improves the configuration of building control systems, leading to a more comfortable and energy-efficient environment.

3.1 Introduction

For building control systems, the integration of presence information plays an essential role in achieving both comfort and energy efficiency, especially for the regulation of heating and cooling. However, actual presence information per room is often not available, leading to the use of estimated occupancy schedules or necessitating manual interventions, such as adjusting thermostatic valves. While manual adjustments are straightforward, they are easily overlooked, and fixed programmed schedules often fail to account for the unique occupancy patterns of individual rooms [1].

These shortcomings can be mitigated through more advanced approaches, such as those proposed by Baldi et al. [2], who demonstrated that incorporating dynamic occupancy behavior into HVAC (Heating, Ventilation, and Air Conditioning) control strategies allows systems to adapt to actual needs. This not only minimizes energy waste but also maintains optimal environmental conditions. However, such strategies depend on the availability of accurate room-level occupancy information, which is the focus of this work.

Various techniques exist for detecting human presence in indoor environments, including passive infrared (PIR) sensors, camera-based systems, and radar-based solutions. Each option has its unique characteristics and applicability in different contexts, as detailed in Section 3.2.1.

Within this work, we opt for a different approach, investigating a methodology centered around CO_2-based presence detection. This approach proves viable as humans naturally exhale CO_2 during breathing, which causes a gradual increase in the CO_2 levels of the room. This can be easily measured with cost-effective and commercially available CO_2 sensors, which are often already installed in modern buildings or can be easily retrofitted in older properties. In contrast to motion-dependent sensors like PIR, the CO_2-based approach effectively detects presence even when individuals are sleeping or at rest, as it does not rely on motion. Unlike camera-based systems, CO_2-sensing is not affected by line of sight. Moreover, this proves a non-intrusive approach that does not raise privacy concerns as it only measures the CO_2 levels in the air. Despite these advantages, it is important to note that CO_2-based presence detection has its own set of limitations, including a slower response time and susceptibility to external factors, as discussed in Section 3.6.

Several prior studies have employed measured CO_2 levels, sometimes in combination with other sensor data, to identify the presence of persons in a room [3–5]. These works present physical or machine learning models and assess their efficacy in single or multiple (simulated) rooms, demonstrating promising outcomes. However, the ability of these models to generalize to new, unseen spaces is either unexplored or requires fine-tuning and adaptation with labeled data from

the new environment to achieve optimal performance. This is significant given the challenges in obtaining accurate presence ground truth data. Consequently, the need to retrain and reconfigure a model for each distinct room poses scalability issues when applied to entire buildings.

This chapter advances existing research by extending the applicability of a binary presence detection model from one room to unobserved rooms in a fully unsupervised fashion. Notably, we are the first to use sliding window normalization for CO_2-based presence detection, demonstrating its efficacy in helping the generalization across diverse rooms. Importantly, our evaluation is conducted using real-world data collected from multiple rooms, spanning both office and residential buildings.

In addition, we introduce a novel methodology for room occupancy profiling using the designed presence detection model. This approach aims to characterize the average usage patterns of rooms over time, which can then be used, for instance, to automatically configure the heating schedule. This approach offers the advantage of a dynamically adaptive schedule that aligns with users' evolving behavior, which is particularly beneficial during periods of remote work and flexible hours. Alternatively, the occupancy profile can serve to propose optimized timing schedules to end users. This helps users in establishing an effective schedule that aligns with their preferences and routines.

The relevant data and code of this research can be accessed on GitHub at https://github.com/predict-idlab/cross-room-CO2-presence-occupancy for reproducibility and to support future research in this domain.

The main contributions of this work are:

- a CO_2-based presence detection methodology that generalizes to unseen rooms in a fully unsupervised manner, evaluated on real-world data;
- an approach for room occupancy profiling, leveraging CO_2-based presence detection to characterize room usage patterns over time;
- an open-source repository that includes both the dataset and the implementation of our research, ensuring transparency and reproducibility.

The remainder of this chapter is structured as follows. First, related work on presence detection and occupancy profiling is summarized and discussed in Section 3.2. Next, Section 3.3 gives information on the used data, and how it is collected and preprocessed. Afterwards, the key aspects of the proposed methodology for CO_2-based presence detection and occupancy profiling are introduced in Section 3.4. This is followed by a validation in Section 3.5 of which the results are discussed in three parts. First, the proposed model and features are leveraged for single-room presence detection. Second, the model is validated on unseen

rooms and the cross-room presence detection performance is presented. Lastly, two example occupancy profiling approaches are discussed and evaluated. After the results, Section 3.6 takes a deeper look at the limitations of CO_2-based presence detection. Finally, Section 3.7 concludes the work and presents future research possibilities.

3.2 Related Work

3.2.1 Presence Detection Sensors

Various solutions exist for detecting human presence in indoor environments, with one common, widely adopted, method being the passive infrared sensor [6]. PIR sensors are compact, cost-effective electronic devices that identify presence by detecting variations in infrared radiation in their immediate surroundings. Being passive, they do not emit infrared light but solely monitor these variations in radiated heat. PIR sensors are frequently employed as motion-activated light switches. Despite PIR sensors are affordable and easy-to-use, a limitation of these is their reliance on changes in radiated heat, requiring active movement of a heat source nearby [7]. Consequently, PIR sensors may not reliably detect individuals who are stationary, making them less suitable for environments such as offices or living rooms where people often sit still for extended periods.

An alternative category of presence detection solutions involves camera-based systems, encompassing thermal vision [8, 9] or conventional visible-light spectrum cameras [10, 11]. In these systems, a computer vision model processes the camera stream to identify humans in the captured images. This detection yields both presence detection as well as person count output. To safeguard occupant privacy, image processing should be performed on-device, and images should never be transmitted externally. While camera-based solutions offer high accuracy, they are comparatively expensive due to the need for powerful hardware for capturing and processing image data, as opposed to the simpler PIR sensor. Furthermore, acceptance suffers if images are not purely processed on the edge.

Additional techniques for presence detection include radar-based systems, where radio frequency signals are emitted, received, and analyzed [12, 13]. These systems offer several advantages, such as their ability to detect movement through obstacles and in various environmental conditions, but require sophisticated signal processing techniques and are costly to implement.

Acoustic signals can also be utilized, capturing and processing sounds of human origin, such as speech and footsteps [14, 15]. These are generally considered less intrusive than camera's, but still pose privacy issues if the data is not handled

carefully. Additionally, acoustic systems might face challenges in noisy environments, leading to potential false detections.

Monitoring energy consumption is another approach, as human presence often correlates with increased energy usage in a room [16, 17]. This method provides indirect and non-intrusive sensing, yet might require infrastructure modifications to measure the energy usage accurately.

Lastly, wireless networks can be leveraged for presence detection, either by relying on the presence of connected smart devices like smartphones or laptops [18, 19] or by directly analyzing the interference of the wireless signal without relying on device connections [20, 21]. Monitoring connected smart devices offers the advantage of utilizing existing infrastructure and can achieve high accuracy. However, it raises privacy concerns as it involves tracking of personal devices. Analyzing the interference of the wireless network closely resembles the radar-based approach, but offers the advantage of using existing hardware infrastructure.

3.2.2 CO_2-based Presence Detection

Numerous studies have explored the detection of presence using CO_2 sensor values, either independently or in combination with other environmental sensor data. A brief overview of select works is discussed below. For a comprehensive overview, readers are directed to existing review studies [22–24].

White-box models leverage physical equations to capture the relationship between the environment and occupants. Typically, the literature within this category employs the mass balance equation which considers the various factors influencing CO_2 levels [25]. Calì et al. propose a dynamic algorithm that optimizes the parameters based on actual CO_2 levels and ground truth presence data [26]. Occupancy is subsequently detected by rewriting the mass balance equation and using the actual CO_2 level as input, yielding promising results across their dataset of five rooms. Nienaber et al. enhance this by refining air exchange rate estimations and expanding the dataset's quality and size [27].

Gray-box models extend white-box approaches by incorporating data-driven parameter estimation, enhancing robustness against measurement and model uncertainties. Ebadat et al. propose a nonlinear gray-box model, utilizing Maximum Likelihood Estimation (MLE) to approximate the parameters based on CO_2 and ventilation data [28]. Their work also focuses on adapting the model to other rooms without additional ground truth data, showcasing promising results albeit in a simulated environment. Wolf et al. employ stochastic differential equations to model the mass balance, also using MLE for parameter estimation [29].

Despite their accuracy and interpretability, physical models rely on the accuracy of the underlying physical equations and assumptions, which might not al-

ways hold true in real-world scenarios. Therefore, literature also explores learning-based approaches, leveraging their capacity to capture complex relationships in a data-driven manner. Candanedo et al. incorporate CO_2, temperature, humidity, and brightness, and employ various model techniques such as random forest, and classification and regression trees (CART) [30]. They discover that including the time of day as a feature enhances model performance, although it can potentially introduce bias towards specific times of day. Arief-Ang et al. focus on enhancing model scalability by adapting their DA-HOC++ model from one room to another [31]. However, it still requires limited labeled data of the target room for fine-tuning.

Our research also focuses on a scalable learning-based method, yet it employs an unsupervised approach to tackle this challenge. This eliminates the need for additional data for fine-tuning and minimizes the risk of biasing the model toward a particular room. Furthermore, the evaluation is performed on real-world data.

3.2.3 Occupancy Profiling

Occupancy profiling can help to enhance the precision of building HVAC and energy consumption simulations or optimize building control system schedules.

Diraco et al. proposed an occupancy profiling methodology utilizing 3D depth cameras to measure occupant count, trajectory, and density [32]. Subsequently, they model the occupancy profiles using inhomogeneous Markov chains which allow the transition probabilities between states to vary with time. However, the deployment of 3D depth cameras is costly, and may face challenges by occlusions and lighting conditions.

In contrast, Kang et al. leveraged mobile positioning data sourced from social media platforms, employing the k-means clustering technique to derive typical weekly occupancy profiles for non-residential buildings [33]. While this approach uses readily available data, it suffers from limitations such as privacy concerns and biases in the social media data, which may not accurately represent the actual occupancy patterns of the environment.

Yang et al.'s work shares similarities with ours, particularly in the use of environmental sensors [34]. They designed a custom sensor box measuring eight room parameters, including CO_2, sound, temperature, and PIR signals. These sensors were mounted at the entrances of single-person offices, accompanied by a camera for ground truth collection. Utilizing this data, pruned decision tree models are trained to predict occupancy status. The resulting occupancy patterns were then analyzed to generate personalized occupancy profiles for both weekdays and weekends. Furthermore, they assessed model generalization by testing it on different offices, demonstrating promising results. However, relying solely

on single-person offices for data collection may limit the generalizability of the model to larger spaces with varying occupancy dynamics.

In our study, we focus on a single sensor, specifically CO_2, to offer a cost-effective solution while also preserving privacy. Additionally, we assess the generalizability across various building types within a less controlled, multi-person environment.

3.2.4 Conclusion

Numerous studies have presented learning-based CO_2-based presence detection methods, with some addressing the challenge of transferring models to unseen rooms. However, to the best of our knowledge, none of these studies have introduced an approach that can generalize to unseen, real-world rooms in an unsupervised manner. Additionally, our work goes beyond existing research by extending the presence detection approach with an occupancy profiling methodology, aiming to make the configuration of building control systems more accurate and dynamic.

3.3 Dataset

This research uses two datasets comprising environmental and occupancy data from two different building types. The first dataset was collected in an office setting, while the second dataset was acquired in a residential context. All data is captured in real-world environments, wherein occupants could perform their work and daily activities without any prescribed instructions or constraints. The subsequent sections describe the methodology employed in the collection and processing of these datasets.

3.3.1 Office Data

The office dataset consists of three rooms situated on the tenth floor of a high-rise office building. Two of these rooms, referred to as office L1 and L2 throughout this chapter, are large and adjacent offices with a capacity of up to 15 persons each. The third office, designated as office S3, is smaller and designed for individual use.

For environmental monitoring, commercially available sensors from Netatmo[1] are employed to collect data in each office. Given the limited size of the offices, one sensor is installed per office in a central position at desk height. These sensors

[1] https://www.netatmo.com/smart-indoor-air-quality-monitor

are configured to record temperature, humidity, CO_2 and loudness at 10-minute intervals. In the context of this study, only the CO_2 measurement will be used.

The dataset annotations were generated using a Steinel HPD2 sensor[2], which is a camera-based detection system. This sensor employs computer vision to count the number of persons within its field of view. Person detection occurs on-device, ensuring that images are not transmitted to safeguard the privacy of occupants. In offices L1 and L2, two Steinel sensors each are deployed to eliminate blind spots, while the smaller office S3 only requires one sensor. For compatibility with the environmental data, the occupancy data is resampled to a 10-minute interval by taking the mode. As we want to perform presence detection and do not need the exact amount of people in the room, the people count is transformed into a binary presence signal.

The data collection within the three offices occurred concurrently, yielding an equal quantity of samples, as illustrated in Table 3.1.

Table 3.1: Number of samples and presence distribution per room in the dataset.

Room	Sample count	No presence / presence distribution
Office L1	30234 ($\pm$ 210 days)	82.2% / 17.8%
Office L2	30234 ($\pm$ 210 days)	81.9% / 18.1%
Office S3	30234 ($\pm$ 210 days)	92.8% / 7.2%
Home 1	25917 ($\pm$ 180 days)	82.2% / 17.8%

3.3.2 Residential Data

The residential dataset comprises data collected from a single room within a house, specifically, the living room featuring a large window leading to an outdoor terrace. The environmental parameters are captured using a proprietary sensor capable of measuring temperature, humidity, CO_2, total volatile organic compounds (TVOC) and brightness. For our analysis, only the CO_2 data from this dataset is retained. The sensor is installed against the wall, away from the sliding window.

In addition to the environmental data, presence annotations are obtained using a radar sensor. This sensor is also fixed to the wall and has a small blind spot in the sitting area.

[2] https://www.steinel.de/en/sensors-professional/products/series-sensors-professional/hpd/hpd2-033200.html

To ensure compatibility with the office dataset, all data is resampled at 10-minute intervals using the mean of the CO_2 measurements and the mode of the presence. However, it is worth mentioning that the logging for this particular room started at a different time and had a slightly shorter runtime, as shown in Table 3.1.

3.3.3 Data Imbalance

In the final column of Table 3.1, the distribution of the *presence* and *no presence* classes for each room in the dataset is presented. Notably, the *no presence* class occurs on average seven times more than the *presence* class. This distribution aligns with our expectations.

For the two large offices, a standard 40-hour work week would suggest a presence rate of approximately 24%. However, in reality, this estimation is expected to be lower due to factors such as flexible working hours and holiday periods. Therefore, the observed presence distribution of about 18% appears reasonable. In contrast, the small office is utilized less frequently, typically by a single occupant with a schedule filled with numerous meetings, resulting in less time spent within the office room. Regarding the living room, assuming an average occupancy of approximately 4 hours per day, the expected presence rate comes to around 17%, closely corresponding with the measured presence rate.

Given the imbalanced nature of the data, balanced accuracy (BA) is chosen as the metric for validating the methodology in this work. Since this is the average between sensitivity and specificity, it offers a more insightful evaluation compared to metrics such as accuracy.

3.3.4 Data Availability

The office dataset is accessible for academic research purposes through the GitHub repository provided at https://github.com/predict-idlab/cross-room-CO2-presence-occupancy. However, the residential data, collected through collaboration with a private company, is restricted from publication.

3.4 Methodology

The methodology proposed in this study comprises of two components. Firstly, a presence detection pipeline based on CO_2 is introduced. This approach is further expanded to incorporate additional temporal information and improve its

applicability across unseen rooms. Secondly, the developed presence detection approach is used to obtain occupancy profiles for a given space.

3.4.1 CO_2-based Presence Detection

3.4.1.1 Machine Learning Pipeline

Given the limited size of the dataset, our approach focuses on the usage of traditional machine learning techniques. Moreover, it has been shown that for time series classification tasks, deep learning models do not necessarily perform better than traditional approaches when good features are employed [35]. Therefore, the first part of the pipeline extracts meaningful features from the time series sensor data. These extracted features aim to capture certain characteristics and temporal aspects of the data, with the objective to inform the model to make a good prediction.

In particular, a strided window feature extraction methodology is employed, wherein features for each window of interest are derived. This procedure is performed using the tsflex Python library, enabling convenient definition of features, window sizes, and strides [36].

Given that the whole dataset is resampled at 10-minute intervals, the stride of this feature extraction is set to 10 minutes as well. Figure 3.1 illustrates the extraction of three features for every 10-minute prediction window. Firstly, the CO_2 value of the current window is used. Subsequently, the mean and slope are computed for a 1-hour window based on historical data. These three base features provide the model with information regarding the current CO_2 level, as well as the longer-term CO_2 level and trend. To maintain simplicity, no additional features are calculated, ensuring a minimal feature set to limit overfitting and enhance model interpretability. Lastly, presence annotations per 10 minutes are retrieved and serve as the target variable during both the training and evaluation processes.

CatBoost is used as classification model, known for its robust performance with minimal parameter tuning [37].

3.4.1.2 Temporal Shift Features

To further augment the model's understanding of temporal aspects related to CO_2 levels and trends, the mean and slope of preceding and/or subsequent hours are incorporated as well. We refer to these features as temporal shift features, which can be used to, for example, include data from 2 hours before the prediction window.

When only historical temporal shift features are integrated, the model can perform real-time presence detection, which we define as prospective analysis in

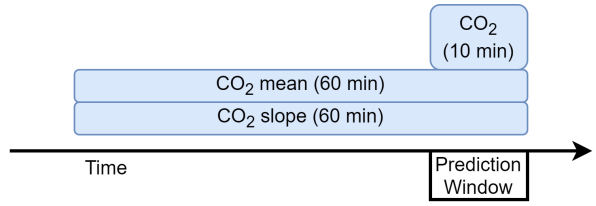


Figure 3.1: Visualization showing the base features to predict the presence of a 10-minute prediction window. The current CO_2 value is included, and both the CO_2 mean and slope of a 1-hour window are calculated.

this chapter. However, if future shift features are included, a delay in inference is required equivalent to the duration of future data used for the prediction. Consequently, we term this methodology as retrospective. Figure 3.2 visually illustrates this concept. The prediction window signifies the 10-minute interval for which presence is predicted. Historical temporal shift features are incorporated to the left of it, while temporal shift features based on future data after the prediction window are calculated to the right.

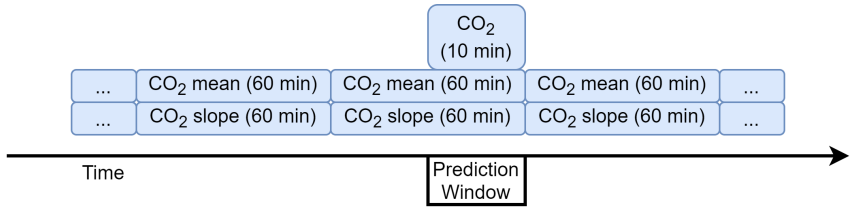


Figure 3.2: Visual representation showing both prospective (using historical data) and retrospective (using "future" data) temporal shift features. The prediction is delayed in order to calculate and incorporate the retrospective shift features.

We have chosen to only shift by complete hours, avoiding any overlap between the windows, since more frequent features are not needed to capture the temporal trends effectively. The number of historical and/or future temporal shift features can be seen as a hyperparameter. The decision to use historical and/or future data depends entirely on the use case, considering factors such as the need for real-time prediction and the acceptable level of delay.

3.4.1.3 Sliding Window Normalization

To achieve cross-room generalization of the CO_2-based presence detection model, we introduce a different normalization methodology known as sliding window

normalization. With this, the common practice of computing normalization parameters once on the training set and reusing them during evaluation or inference is not followed. Sliding window normalization dynamically calculates the normalization parameters, i.e., mean and standard deviation, for each sample based on a sliding window of historical data. This window adjusts with every new data sample, as depicted in Figure 3.3. While sliding window normalization is not a new technique and has been successfully employed in various studies [38, 39], our work is the first to implement it for CO_2-based presence detection.

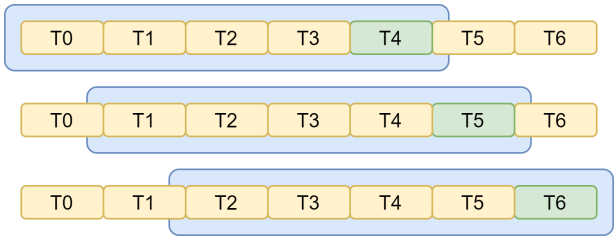


Figure 3.3: Visualization showing the concept of sliding window normalization. The green box denotes the sample undergoing normalization, which is the window for prediction. Meanwhile, the blue box represents the historical data window employed to compute the normalization parameters. The normalization window slides in tandem with the sample being predicted.

Our approach offers several benefits for presence detection. By basing normalization on a window of historical room data, it automatically adapts to different environments, user behavior changes, seasonal differences and sensor drift. In our implementation, the window size is set to 30 days, striking a balance between adaptability (short enough) and stability (long enough).

Moreover, our method ensures that the normalization parameters are based on the room where the model is applied on. This helps to reduce some differences between rooms by aligning the characteristics of the data which should improve generalization. While this effect can be partially achieved by transforming based on the normalization parameters of the target room, it lacks the temporal self-adaptation and is impractical for real-world deployment, requiring the capturing of a historical dataset for parameter calculation.

In contrast, our approach is easily deployed. After installing the CO_2 sensor in a room, the normalization gradually adapts to the current environment over a few weeks, enhancing model performance. Although there is an initial cold-start period, this is resolved after a few weeks of operation.

3.4.2 Occupancy Profiling

Our cross-room presence detection methodology can be practically applied in real-world scenarios, without the need for retraining the model for each new room. One valuable application of this model is occupancy profiling, where the focus is beyond real-time presence detection by analyzing average room usage patterns over an extended period. This information is helpful for configuring building control systems, such as heating and cooling, in a smart, automated, and self-adapting manner.

For this occupancy profiling, the retrospective model is leveraged since real-time prediction is not required and a higher performance is achievable. To generate the occupancy profile, the presence probability is calculated at each 10-minute interval by grouping corresponding intervals from multiple days and averaging the presence, as illustrated in Figure 3.4. For instance, all presence at 8:10 AM is averaged over the past months to acquire the presence probability of that time window. The presence profile is obtained by doing this for every other interval as well, and is then resampled to a more practical 30-minute window by calculating the mean.

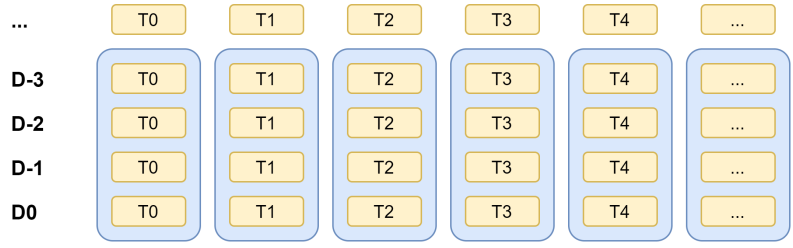


Figure 3.4: Visualization demonstrating the computation process for occupancy profiling. The yellow boxes depict distinct 10-minute intervals throughout a day, while the blue boxes indicate the window used to group and average occupancy, determining the presence probability for each interval. In this example, the window size ranges from the current day D0 to three days prior D-3. This base grouping method can be customized to distinguish between workdays, weekends, and other variations, as detailed in Section 3.5.3.1 and 3.5.3.2.

The window size can be freely chosen, allowing to balance between stability and adaptability. A larger window size yields a more stable occupancy profile, while a smaller window size increases adaptability. In this chapter, a window size of four months (a quarter) is selected, providing a stable profile which yet responds fairly quickly to changes in occupancy patterns.

The grouping methodology is also adjustable based on the specific needs. At a minimum, grouping is done per 10-minute interval, as shown in Figure 3.4, but

additional factors like day of the week or working/weekend days can be considered. For instance, grouping based on working and weekend days provides insight into average occupancy during a working day, without distinguishing between the different weekdays. For more fine-grained insights, grouping can be performed separately for each weekday, allowing investigation into the individual occupancy profiles of weekdays. Moreover, the grouping can extend across multiple rooms and floors to create a more general occupancy profile for an entire floor or building.

To effectively use these occupancy profiles for configuring schedules in building control systems, several approaches can be considered. A simple yet effective method is to apply a threshold, such as 10%, on the occupancy probabilities to determine appropriate activation and deactivation times of building systems. Alternatively, more sophisticated strategies can be employed, such as occupancy-driven Model Predictive Control (MPC), which dynamically adjusts control schedules based on occupancy data to better align with actual needs [2].

3.5 Results

3.5.1 Single-Room Presence Detection

This section investigates the outcomes of the proposed model and techniques employed for single-room presence detection. The features are calculated in accordance with the methodology detailed in Section 3.4.1.1, and subsequently used to train and assess a CatBoost model, employing data from the same room. Furthermore, the study explores the impact of incorporating temporal shift features.

For each room, the first two-thirds (2/3) of the data are allocated for training, while the last one-third (1/3) is reserved for evaluation. Note that the data is chronologically split rather than randomly sampled, aiming to reduce bias of the evaluation process. During this phase, sliding window normalization is not introduced yet. However, the features are z-normalized, with normalization parameters determined on the training set solely and subsequently applied to both the training and evaluation datasets. Every CatBoost model is trained for 100 iterations since higher numbers do not result in better learning performance, as shown in Appendix 3.A.

This approach is adopted to assess the performance of our CO_2-based presence detection methodology in a single-room configuration. It is important to note that this analysis does not offer insights into the cross-room generalization capabilities of the model, which is later covered in Section 3.5.2.

3.5.1.1 Baseline

Table 3.2 highlights the single-room performance of some key configurations. The baseline results only use three features: the current 10-minute CO_2 level, and the 1-hour CO_2 mean and slope.

Among the baseline results, office L1 and home 1 show the highest performance, achieving respective evaluation balanced accuracy (BA) scores of 67.6% and 65.3%. Office L2, which is closely related to office L1, lags behind with a BA score of 59.6%. The one-person office, office S3, demonstrates the lowest performance with a BA score of 52.3%, indicating that the model failed to learn a relevant correlation to detect presence based on CO_2 data.

The low performance of office S3 can be attributed to the frequent opening of windows, allowing fresh air into the room. This disturbs the anticipated rise in CO_2 levels that typically occurs in a closed environment. Given that the presence detection model relies on such a rise in CO_2, the data becomes more challenging for the model to interpret and discern presence accurately.

Table 3.2: Training (TBA) and validation (VBA) balanced accuracy scores across all rooms employing different feature configurations. The baseline model solely comprises the current 10-minute CO_2, along with the 1-hour CO_2 mean and slope. The subsequent results incorporate temporal shift features (TSF) in prospective, retrospective, and feature-reduced configurations.

	Office L1		Office L2		Office S3		Home 1	
Configuration	**TBA**	**VBA**	**TBA**	**VBA**	**TBA**	**VBA**	**TBA**	**VBA**
Baseline, no TSF	79.3%	67.6%	82.0%	59.6%	57.3%	52.3%	83.9%	65.3%
TSF up to 8 hours ago	91.1%	77.4%	91.5%	**69.9%**	72.4%	56.3%	90.0%	66.1%
TSF up to 8 hours ago+later	97.7%	79.0%	99.0%	69.4%	92.7%	60.0%	96.8%	69.4%
TSF 1/2/4/8 hours ago+later	94.6%	**80.7%**	95.9%	69.5%	78.8%	**60.6%**	93.2%	**70.4%**

3.5.1.2 Temporal Shift Features

To assess the impact of historical temporal shift features on model performance, we progressively introduce an increasing number of shift features, as depicted in Figure 3.5. Overall, an increase in BA score is noticeable when more temporal shift features are provided to the model, showing the benefits of the additional temporal information. However, there is no optimal quantity of shift features across all rooms. Offices L1 and L2 show improved performance with the inclusion of more temporal information, whereas office S3 and Home 1 reach a plateau early

when shift features are incorporated. Consequently, we empirically evaluated that including up to 8 hours of temporal shift features is a good trade-off between performance enhancement and feature vector length.

Figure 3.5: Validation BA scores across all rooms, highlighting the influence of incorporating an increasing number of historical temporal shift features. At $x = 0$, no historical shift features are incorporated, aligning with the baseline model outlined in Table 3.2. As x values increase, historical temporal shift features are cumulatively included, extending up to 12 hours ago.

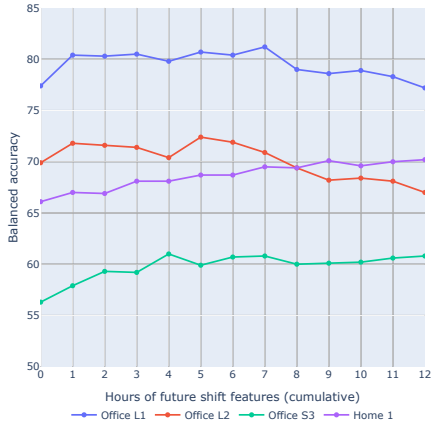


Figure 3.6: Validation BA scores across all rooms, showcasing the impact of increasing the number of future temporal shift features. At $x = 0$, no future shift features are incorporated, aligning with the model containing historical shift features up to 8 hours ago. As x values increase, future temporal shift features are cumulatively integrated, extending up to 12 hours later.

When allowing for prediction delays, the integration of future shift features yields additional improvement in performance, as illustrated in Figure 3.6. The BA scores for all rooms increase when future temporal information is incorporated. The most substantial improvement is observed with the inclusion of the first hour which is due to the delayed response time of a CO_2 signal. To elaborate, while a person's presence in a room is immediate, CO_2 levels increase gradually over time which is why the first future hour is so valuable. Similar to the earlier results, no single configuration is optimal across all rooms. Consequently, we opt for the 8-hour variant once again, as it consistently yields a notable improvement across all rooms on average.

3.5.1.3 Feature Reduction

The previous section introduced a set of temporal shift features aimed at enhancing the model's performance. While this objective is achieved, the many new features also increase the potential for overfitting. Consequently, several configurations with a reduced set of features are assessed, as outlined in Table 3.3.

Table 3.3: Training (TBA) and validation (VBA) balanced accuracy scores across all rooms with a reduced set of temporal shift features. The first row represents the complete set of shift features up to 8 hours ago and later, as specified in Section 3.4.1.2. Subsequent rows eliminate combinations of shift features.

	Office L1		Office L2		Office S3		Home 1	
Configuration	**TBA**	**VBA**	**TBA**	**VBA**	**TBA**	**VBA**	**TBA**	**VBA**
TSF up to 8 hours ago+later	97.7%	79.0%	99.0%	69.4%	92.7%	60.0%	96.8%	69.4%
TSF 1/2/3/4/5/7/8 hours ago+later	97.8%	80.2%	98.8%	**69.8%**	91.0%	**61.5%**	96.8%	69.7%
TSF 1/2/4/5/7/8 hours ago+later	97.6%	78.9%	98.9%	69.0%	90.1%	60.8%	96.8%	69.0%
TSF 1/2/4/7/8 hours ago+later	97.1%	80.6%	98.8%	69.5%	89.7%	61.0%	96.9%	69.5%
TSF 1/2/4/8 hours ago+later	94.6%	**80.7%**	95.9%	69.5%	78.8%	60.6%	93.2%	**70.4%**
TSF 1/4/8 hours ago+later	94.6%	79.4%	95.6%	67.6%	76.8%	59.8%	93.1%	70.1%

Analysis of the BA scores reveals that a substantial number of temporal shift features can be omitted without compromising performance compared to the initial feature set. This improvement in performance can be attributed to the reduced overfitting, which is shown by the smaller gap between the training and validation BA scores.

Limiting the usage of temporal shift features to 1, 2, 4 and 8 hours ago and later strikes the best balance between feature reduction and overall performance, motivating the use of this configuration throughout the remainder of this study. Moreover, by using this reduced set of features, both data preprocessing and model training times are minimized, with the entire process now completing in approximately 3.3 seconds on an Intel Core i7-8650 CPU.

Table 3.2 presents the BA scores for the final feature selection, alongside the most noteworthy prospective and retrospective configuration, as well as the baseline. The results demonstrate that incorporating past and future temporal shift features yield notable improvements, ranging from 4.1% to 11.4% over the baseline. Moreover, refining the feature set to include only the most important feature shifts result in an additional improvement of up to 1.7%.

3.5.2 Cross-Room Presence Detection

In the previous section, we demonstrated the detection capability of presence through CO_2 sensors. However, training a model for each individual room is highly impractical for an entire building, as this would require the collection of labeled data for every room.

To address this constraint, it is essential to develop a model capable of cross-room generalization. Such a model should effectively predict presence in rooms beyond the training set without requiring specific labeled data for each room.

To investigate the cross-room generalization of the CO_2-based presence detection methodology, we evaluate a model trained on one room across other rooms in the dataset. The following sections present the results of this cross-room evaluation, using two distinct data normalization approaches.

3.5.2.1 Traditional Normalization Approach

A first and straightforward strategy involves directly assessing the single-room model's performance on an unseen room. In this approach, the normalization parameters from the training data are applied to the data of the unseen room.

The results obtained from this traditional method, as presented in Table 3.4, are in line with our expectations. In the majority of cases, the model demonstrates poor performance when applied on unseen rooms, with most of the BA scores around or below 60%. Such scores reflect the model's proficiency in identifying absence instances while facing challenges in accurately predicting presence.

3.5.2.2 Sliding Window Normalization Approach

To reduce variations in characteristics, sensor operation, and user behavior across different rooms, sliding window normalization is introduced. Table 3.4 displays the cross-room results achieved by applying this sliding window normalization approach to various rooms.

Overall, a substantial improvement in BA scores is evident compared to the traditional normalization discussed in the previous section. The best result with the traditional normalization approach is a BA score of 81.4% when assessing the model from office L1 on office L2. In contrast, the sliding window approach here yields a higher performance, achieving a BA score of 84.6%. Similarly, evaluating the model of office L2 on office L1 results in a BA score of 83.4%, demonstrating robust generalization between these related offices irrespective of the training room.

Another interesting observation is the models' ability to generalize across different types of rooms. The traditional approach yields a BA score of 65.4% when

Table 3.4: Comparison of cross-room validation BA scores employing both the traditional and sliding window normalization approaches. With the traditional method, normalization parameters are calculated for the training room and subsequently applied to the validation room. The sliding window normalization approach dynamically adjusts the normalization parameters based on a window of historic data. Each row specifies the training room and presents the validation results for other rooms. Additionally, the performance of the single-room model is provided as an indication of achievable results. However, they should not be directly compared with due to differences in the size of the validation set.

Training on	**Validation BA on**			
	Office L1	**Office L2**	**Office S3**	**Home 1**
Single-room with traditional normalization approach				
Same room	80.7%	69.5%	60.6%	70.4%
Cross-room with traditional normalization approach				
Office L1	X	81.4%	55.7%	65.4%
Office L2	63.0%	X	51.1%	75.7%
Office S3	62.0%	69.8%	X	70.6%
Home 1	50.0%	53.5%	50.0%	X
Cross-room with sliding window normalization approach				
Office L1	X	84.6%	66.8%	80.6%
Office L2	83.4%	X	64.1%	78.0%
Office S3	60.1%	62.2%	X	65.7%
Home 1	71.4%	70.5%	64.5%	X

evaluating the model from office L1 on home 1. However, with the introduction of sliding window normalization, this performance increases to 80.6%. The reverse evaluation, using the model of home 1 on the offices, shows a substantial improvement as well from 50%, 53.5%, and 50% to 71.4%, 70.5%, and 64.5% respectively, with office S3 being the least improved due to CO_2 data influenced by the window.

Remarkably, most of the evaluation performances using sliding window normalization approximate or even exceed the indicative single-room performance, except when office S3 serves as the source room. Although these results are not exactly comparable, this demonstrates the significance of sliding window normalization not only in generalizing between rooms but also in adapting to changing conditions within a room over time.

3.5.3 Occupancy Profiling

Using the occupancy profiling methodology from Section 3.4.2, the CO_2-based presence detection approach is employed to characterise room usage patterns over a defined period of time. In the following sections, we will explore two example configurations. The first configuration involves grouping data into working and weekend days, while the second configuration creates separate profiles for each day of the week.

These examples use the retrospective model of office L1 to predict the presence of office L2, representing a realistic usage scenario. The occupancy profiles are computed with a window size of four months. This implies that a quarter of historical presence data is considered when constructing the average room usage profile.

3.5.3.1 Grouping per Working or Weekend Day

In this first approach, the data is grouped into workdays (Monday to Friday) and weekend days (Saturday and Sunday). The presence profile for a specific day is thus based on the presence of all work or weekend days over the preceding four months. This grouping strategy is beneficial when configuring control systems that require different settings for weekdays and weekends. Furthermore, it remains stable across the various days within each category, making it predictable and reliable for occupants.

An example of the presence profile obtained through this method is presented in Figure 3.7. The profile shows the presence probability for the last Friday in the dataset of office L2 using predicted presence data over the past four months. As anticipated, the presence probability starts to gradually increase around 8:30 AM, followed by a decline approaching noon. Subsequently, a prolonged period of presence is visible in the afternoon, suggesting that employees often remain in the office until 6:00 PM or later. This insight is important for configuring a heating system, as a conventional 9-to-5 schedule would leave employees working late in cold conditions.

While Figure 3.7 shows the presence profile based on predicted presence data, reflecting real-world application, Figure 3.8 displays the profile based on the ground truth presence data of office L2. Although both profiles exhibit similar trends, the predicted presence profile generally shows lower presence probabilities, indicating that the model did not predict all instances of presence accurately. Nevertheless, the advantage of occupancy profiling lies in its temporal averaging, ensuring that overall trends remain visible and useful for configuring building control systems.

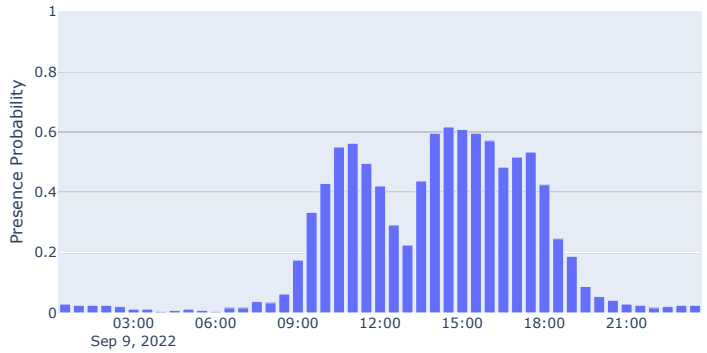


Figure 3.7: Occupancy profile of the last Friday in the dataset for office L2 when grouping in working and weekend days. This profile is generated based on the predicted presence data using the cross-room model of office L1. The occupancy profile illustrates presence before and after noon, with a noticeable decrease during the lunch break.

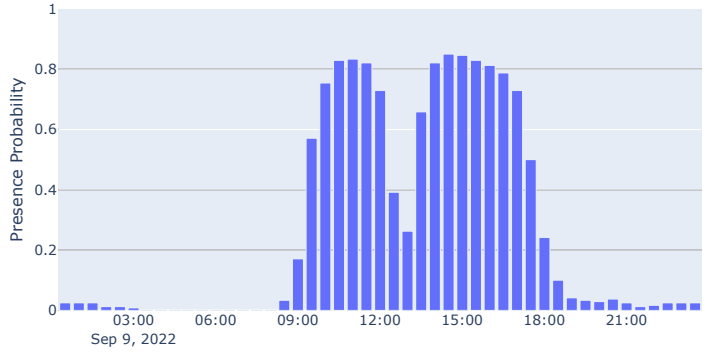


Figure 3.8: Occupancy profile of the last Friday in the dataset for office L2 when grouping in working and weekend days. This profile is constructed using the ground truth presence data.

When comparing the ground truth and predicted occupancy profiles numerically across all data for office L2, the mean absolute error (MAE) of the presence probabilities is 0.043, with a standard deviation of 0.079. While the MAE is low, the relatively large standard deviation suggests inconsistencies in the model's predictions. However, as mentioned earlier, the trends remain visible given the prolonged averaging period.

3.5.3.2 Grouping per Weekday

For a more fine-grained approach, the data can also be grouped per weekday, resulting in distinct occupancy profiles for each day of the week (Monday through Sunday). This approach is valuable in scenarios where presence patterns vary significantly across weekdays. For instance, if an office remains unused on certain weekdays due to remote work, this method is capable of identifying days with lower presence probabilities and building control systems can be configured accordingly.

In the case of office L2 in our dataset, while there are no designated remote working days, clear variations exist in the 4-month average occupancy profiles across the different weekdays, as illustrated in Figure 3.9. Mondays and Tuesdays exhibit the highest presence probabilities, with a noticeable decline towards the end of the workweek, indicating a greater variability and uncertainty in occupancy. However, there are no workdays with near-zero presence probability, showing the need for building control systems to be prepared if employees do arrive. In contrast, the occupancy profile of the weekend shows that the office is then unused.

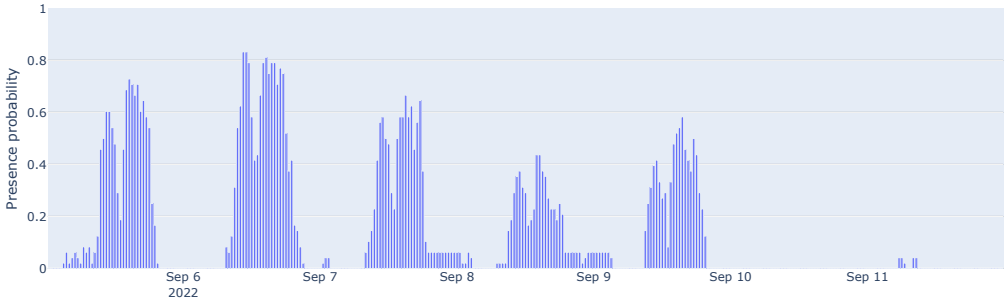


Figure 3.9: Occupancy profile showing last complete week (Monday to Sunday) in the dataset for office L2 when grouping per weekday. This profile is generated based on the predicted presence data using the cross-room model of office L1. Each workday exhibits a distinct occupancy profile, clearly showing variations in presence between the beginning and end of the workweek. The profile of the weekend indicates virtually no presence in the office.

When employing grouping per weekday, there is a slight increase in both the mean absolute error and the standard deviation, reaching 0.052 and 0.103 respectively. This increase can be attributed to the more fine-grained profiling methodology.

3.6 Limitations of CO_2-based Presence Detection

This work demonstrated the feasibility of CO_2-based presence detection in both office and residential settings, with the ability to apply the model to unseen rooms. However, notable performance differences can be seen in the evaluation results. Office L1 shows the highest performance, followed by office L2 and home 1, while office S3 consistently shows the lowest performance. This poor performance for office S3 is attributed to the disturbed CO_2 data in spaces where windows are frequently open, which significantly influences the expected CO_2 trends within a room. An example of this phenomenon is shown in Figure 3.10. Before noon, presence in the room with the windows closed results in clearly rising CO_2 levels, which is ideal for presence detection. However, in the afternoon, despite the continued presence, the open windows disturb the natural rise of the CO_2 levels, making presence detection more challenging.

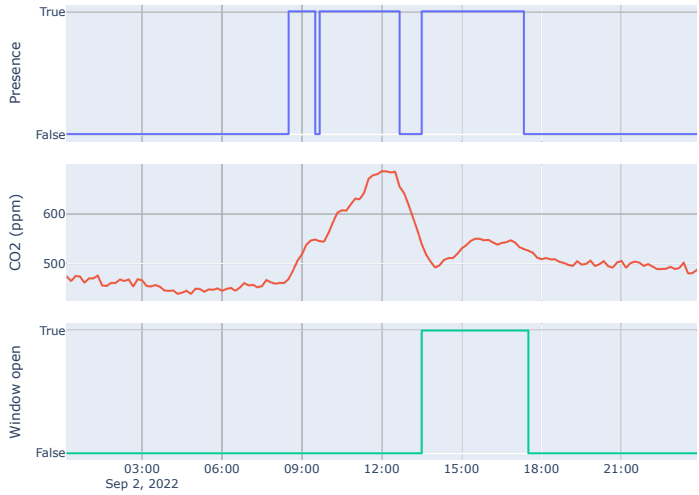


Figure 3.10: Example situation in office L1 demonstrating the influence of an open window on the CO_2 concentration. Notably, the presence in the afternoon with an open window leads to significantly lower CO_2 levels. The status of the window is monitored through a contact sensor.

In addition to open windows, ventilation directly influences the CO_2 levels in a room for the same reasons. Moreover, open doors can impact CO_2 levels by allowing it to disperse into adjacent rooms, making it more difficult to detect the rising trend. Similarly, performing CO_2-based presence detection is more challenging in large rooms, where CO_2 can diffuse more extensively, and the presence of a single individual has a minimal effect on the overall CO_2 concentration in the room.

In an ideal case for CO_2-based presence detection, the room would be a sealed environment with no external influences. However, in reality, this is neither feasible nor desirable, as individuals require a constant supply of fresh air for comfort and well-being. The impact of windows and ventilation on CO_2 levels is often not a binary condition. Some degree of influence is acceptable, provided it does not significantly disrupt the CO_2 signal. For instance, if a window is open and the CO_2 level rises gradually, the model will take a longer time to detect presence, but it should eventually succeed. However, when a window is open and the CO_2 level begins to lower, the model's outcome depends on the speed and extent of the CO_2 decrease. If the decline is too rapid and/or the CO_2 level becomes too low, distinguishing between an open window and people leaving the room becomes challenging for the model.

Next to the impact of external factors on the CO_2 level, another drawback is the delayed response time associated with CO_2 sensors. These sensors have a latency in detecting occupancy as they rely on the accumulation of CO_2 in the air, which occurs after individuals have been present for a certain duration. The impact of this response delay is noticeable in Figure 3.6, where the introduction of temporal shift features of one hour later significantly improved performance.

3.6.1 Addressing the Limitations

To address the limitations of relying solely on CO_2 data, future work may explore incorporating additional sources of information. For scenarios where instant presence detection is required, the slower response time of CO_2 signals may not be sufficient. In such cases, the presence detection approach could be extended with alternative sensor technologies, such as PIR or radar sensors, which offer real-time detection capabilities. However, this also increases hardware requirements and associated costs.

In instances where a delay in detection is acceptable, or when real-time presence information is not required, the methodology could also be further refined. For example, to account for the impact of open windows on CO_2 levels, window state information could be captured using contact sensors. Such data could be integrated as a feature into the proposed machine learning approach, enabling the model to make more informed decisions. Additionally, window size is another significant factor that can be considered. This information could be extracted from a building information model (BIM) and incorporated into the detection model by specifying the area (e.g., in square meters) that is open.

To better handle the influence of ventilation on CO_2 dynamics, integrating ventilation system data, such as airflow rates, would be beneficial. Similar to CO_2 data, ventilation data is a time series from which relevant window-based features, such as levels and trends, can be extracted. Additionally, temporal shift features

could be employed to capture long-term trends. These features can easily be incorporated into the proposed machine learning framework.

Many modern ventilation systems already measure CO_2 concentrations to regulate ventilation rates. These systems present significant potential for implementing automated presence detection, as they have access to both CO_2 and ventilation rate data. A well-functioning ventilation system can also reduce the need for window ventilation, thereby minimizing its impact on presence detection performance and improving reliability. Furthermore, since such systems are often part of an HVAC solution, the presence information could be shared with, for example, the heating system to optimize timing schedules, aligning with the objectives of the proposed work.

Although these potential improvements offer promising directions for future research, they fall outside the scope of this study. The focus of this work remains on providing a methodology that is broadly applicable, requiring only CO_2 data.

3.7 Conclusion and Future Work

Fixed configuration schemes for building control systems, such as heating, are still a common practice and lead to energy inefficiency and/or suboptimal comfort. This work proposes an approach to automatically configure building control systems based on a room's occupancy profile generated using CO_2-based presence information.

To achieve this objective, the primary focus was on developing a CO_2-based presence detection methodology with good generalization across diverse rooms. We showed that traditional models achieve good predictive capabilities using simple features, namely the current CO_2, 1-hour CO_2 mean and 1-hour CO_2 slope, augmented with temporal shift features to incorporate past and/or future information into the model's prediction. Additionally, the importance of a robust normalization strategy was identified. Employing sliding window normalization significantly improved the model's ability to generalize to unseen rooms and makes practical deployment more convenient, as the model autonomously adapts to room-specific conditions over time.

The generalization of the model depends on the similarity between rooms. For instance, applying the model developed for office L1 to office L2 yields a high BA score of 84.6%. Performance on different room types is slightly lower but remains acceptable, as illustrated by applying the model of office L1 to home 1, resulting in an BA score of 80.6%. However, CO_2-based presence detection has its limitations, such as in cases like office S3, where open windows affect the performance.

The CO_2-based presence predictions were shown to be valuable for the generation of occupancy profiles, indicating the presence probability during the day. Various calculation approaches are available, allowing flexibility in determining whether the resulting profiles should be more coarse-grained or fine-grained. Based on the predicted presence profiles for office L2 using the model of office L1, an average error of about 5% with a standard deviation of about 10% is observed. These profiles align closely with the trends of actual room usage, making them a reliable basis for automatically configuring building control systems based on historical presence probability. As the occupancy profiles evolve over time with changing occupant behavior, there is no need for manual reconfiguration of building control systems.

Future research could explore the performance of neural networks operating on raw time series data to validate if these are more effective than the proposed methodology based on feature extraction. Furthermore, to overcome the limitations associated with CO_2 for presence detection, the integration of additional sensors and building-related information can be considered. Moreover, the integration of calendar metadata, such as holidays and vacation periods, can provide valuable support to the occupancy profiling methodology.

We believe that the insights from this study on generalizable CO_2-based presence detection and occupancy profiling will contribute to the shift from fixed timing schemes to more dynamic and smart configuration of building control systems.

3.A Appendix: Impact of Varying CatBoost Iterations

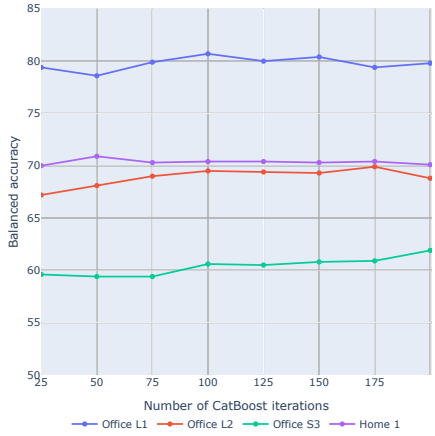


Figure 3.11: Validation BA scores across all rooms using the final single-room model described in Section 3.5.1.3. The results demonstrate that increasing the number of CatBoost iterations does not lead to substantial improvements in learning performance. For all other experiments, 100 iterations are used as it provides a good balance between learning capacity and model complexity.

References

[1] T. Peffer, M. Pritoni, A. Meier, C. Aragon, and D. Perry. *How people use thermostats in homes: A review*. Building and Environment, 46(12):2529–2541, December 2011. doi:10.1016/j.buildenv.2011.06.002.

[2] S. Baldi, C. D. Korkas, M. Lv, and E. B. Kosmatopoulos. *Automating occupant-building interaction via smart zoning of thermostatic loads: A switched self-tuning approach*. Applied Energy, 231:1246–1258, December 2018. doi:10.1016/j.apenergy.2018.09.188.

[3] C. Karasoulas, C. Keroglou, E. Katsiri, and G. C. Sirakoulis. *Hidden Markov models for presence detection based on CO2 fluctuations*. Frontiers in Robotics and AI, 10, October 2023. Available from: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2023.1280745/full, doi:10.3389/frobt.2023.1280745.

[4] J. Kim, J. Bang, A. Choi, H. J. Moon, and M. Sung. *Estimation of Occupancy Using IoT Sensors and a Carbon Dioxide-Based Machine Learning Model with Ven-*

tilation System and Differential Pressure Data. Sensors, 23(22):585, January 2023. doi:10.3390/s23020585.

[5] Q. Huang, M. Syndicus, J. Frisch, and C. van Treeck. *Spatial features of CO2 for occupancy detection in a naturally ventilated school building*. Indoor Environments, 1(3):100018, October 2024. doi:10.1016/j.indenv.2024.100018.

[6] J. Caniou. *Passive Infrared Detection: Theory and Applications*. Springer Science & Business Media, March 2013.

[7] T. Teixeira, G. Dublon, and A. Savvides. *A Survey of Human-Sensing: Methods for Detecting Presence, Count, Location, Track, and Identity*. ACM Computing Surveys, 5:59, January 2010.

[8] A. A. Trofimova, A. Masciadri, F. Veronese, and F. Salice. *Indoor Human Detection Based on Thermal Array Sensor Data and Adaptive Background Estimation*. Journal of Computer and Communications, 5(44):16–28, March 2017. doi:10.4236/jcc.2017.54002.

[9] S. Singh and B. Aksanli. *Non-Intrusive Presence Detection and Position Tracking for Multiple People Using Low-Resolution Thermal Sensors*. Journal of Sensor and Actuator Networks, 8(33):40, September 2019. doi:10.3390/jsan8030040.

[10] D. Lefloch, F. A. Cheikh, J. Y. Hardeberg, P. Gouton, and R. Picot-Clemente. *Real-time people counting system using a single video camera*. In Real-Time Image Processing 2008, volume 6811, page 71–82. SPIE, February 2008. doi:10.1117/12.766499.

[11] J. Zou, Q. Zhao, W. Yang, and F. Wang. *Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation*. Energy and Buildings, 152:385–398, October 2017. doi:10.1016/j.enbuild.2017.07.064.

[12] J. W. Choi, D. H. Yim, and S. H. Cho. *People Counting Based on an IR-UWB Radar Sensor*. IEEE Sensors Journal, 17(17):5717–5727, September 2017. doi:10.1109/JSEN.2017.2723766.

[13] J.-H. Choi, J.-E. Kim, and K.-T. Kim. *Deep Learning Approach for Radar-Based People Counting*. IEEE Internet of Things Journal, 9(10):7715–7730, May 2022. doi:10.1109/JIOT.2021.3113671.

[14] Q. Huang. *Occupancy-Driven Energy-Efficient Buildings Using Audio Processing with Background Sound Cancellation*. Buildings, 8(66):78, June 2018. doi:10.3390/buildings8060078.

[15] S. Uziel, T. Elste, W. Kattanek, D. Hollosi, S. Gerlach, and S. Goetze. *Networked embedded acoustic processing system for smart building applications*. In 2013 Conference on Design and Architectures for Signal and Image Processing, page 349–350, October 2013. Available from: https://ieeexplore.ieee.org/abstract/document/6661570.

[16] W. Kleiminger, C. Beckel, T. Staake, and S. Santini. *Occupancy Detection from Electricity Consumption Data*. In Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings, BuildSys '13, page 1–8, New York, NY, USA, November 2013. Association for Computing Machinery. doi:10.1145/2528282.2528295.

[17] A. Akbar, M. Nati, F. Carrez, and K. Moessner. *Contextual occupancy detection for smart office by pattern recognition of electricity consumption data*. In 2015 IEEE International Conference on Communications (ICC), page 561–566, June 2015. doi:10.1109/ICC.2015.7248381.

[18] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal. *Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings*. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, SenSys '13, page 1–14, New York, NY, USA, November 2013. Association for Computing Machinery. doi:10.1145/2517351.2517370.

[19] X. Lu, H. Wen, H. Zou, H. Jiang, L. Xie, and N. Trigoni. *Robust occupancy inference with commodity WiFi*. In 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), page 1–8, October 2016. doi:10.1109/WiMOB.2016.7763228.

[20] S. Depatla, A. Muralidharan, and Y. Mostofi. *Occupancy Estimation Using Only WiFi Power Measurements*. IEEE Journal on Selected Areas in Communications, 33(7):1381–1393, July 2015. doi:10.1109/JSAC.2015.2430272.

[21] Y. Zhang, X. Wang, J. Wen, and X. Zhu. *WiFi-based non-contact human presence detection technology*. Scientific Reports, 14(1):3605, February 2024. doi:10.1038/s41598-024-54077-x.

[22] L. Rueda, K. Agbossou, A. Cardenas, N. Henao, and S. Kelouwani. *A comprehensive review of approaches to building occupancy detection*. Building and Environment, 180:106966, August 2020. doi:10.1016/j.buildenv.2020.106966.

[23] D. Trivedi and V. Badarla. *Occupancy detection systems for indoor environments: A survey of approaches and methods*. Indoor and Built Environment, 29(8):1053–1069, October 2020. doi:10.1177/1420326X19875621.

[24] T. Li, X. Liu, G. Li, X. Wang, J. Ma, C. Xu, and Q. Mao. *A systematic review and comprehensive analysis of building occupancy prediction*. Renewable and Sustainable Energy Reviews, 193:114284, April 2024. doi:10.1016/j.rser.2024.114284.

[25] S. Wang, J. Burnett, and H. Chong. *Experimental Validation of CO2-Based Occupancy Detection for Demand-Controlled Ventilation*. Indoor and Built Environment, 8(6):377–391, June 2000. doi:10.1159/000057493.

[26] D. Calì, P. Matthes, K. Huchtemann, R. Streblow, and D. Müller. *CO2 based occupancy detection algorithm: Experimental analysis and validation for office and residential buildings*. Building and Environment, 86:39–49, April 2015. doi:10.1016/j.buildenv.2014.12.011.

[27] F. Nienaber, M. Wesseling, D. Calì, and D. Mueller. *Validation and optimization of air quality sensor based occupancy detection algorithms*. In Proceedings of Roomvent & Ventilation 2018, pages 109–114, June 2018.

[28] A. Ebadat, G. Bottegal, M. Molinari, D. Varagnolo, B. Wahlberg, H. Hjalmarsson, and K. H. Johansson. *Multi-room occupancy estimation through adaptive gray-box models*. In 2015 54th IEEE Conference on Decision and Control (CDC), page 3705–3711, December 2015. doi:10.1109/CDC.2015.7402794.

[29] S. Wolf, D. Calì, J. Krogstie, and H. Madsen. *Carbon dioxide-based occupancy estimation using stochastic differential equations*. Applied Energy, 236:32–41, February 2019. doi:10.1016/j.apenergy.2018.11.078.

[30] L. M. Candanedo and V. Feldheim. *Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models*. Energy and Buildings, 112:28–39, January 2016. doi:10.1016/j.enbuild.2015.11.071.

[31] I. B. Arief-Ang, M. Hamilton, and F. D. Salim. *A Scalable Room Occupancy Prediction with Transferable Time Series Decomposition of CO 2 Sensor Data*. ACM Transactions on Sensor Networks, 14(3–4):1–28, November 2018. doi:10.1145/3217214.

[32] G. Diraco, A. Leone, and P. Siciliano. *People occupancy detection and profiling with 3D depth sensors for building energy management*. Energy and Buildings, 92:246–266, April 2015. doi:10.1016/j.enbuild.2015.01.043.

[33] X. Kang, D. Yan, J. An, Y. Jin, and H. Sun. *Typical weekly occupancy profiles in non-residential buildings based on mobile positioning data*. Energy and Buildings, 250:111264, November 2021. doi:10.1016/j.enbuild.2021.111264.

[34] Z. Yang and B. Becerik-Gerber. *Modeling personalized occupancy profiles for representing long term patterns by using ambient context*. Building and Environment, 78:23–35, August 2014. doi:10.1016/j.buildenv.2014.04.003.

[35] J. Van Der Donckt, J. Van Der Donckt, E. Deprost, N. Vandenbussche, M. Rademaker, G. Vandewiele, and S. Van Hoecke. *Do not sleep on traditional machine learning: Simple and interpretable techniques are competitive to deep learning for sleep scoring*. Biomedical Signal Processing and Control, 81:104429, 2023.

[36] J. Van Der Donckt, J. Van Der Donckt, E. Deprost, and S. Van Hoecke. *tsflex: flexible time series processing & feature extraction*. SoftwareX, 2021. Available from: https://github.com/predict-idlab/tsflex.

[37] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. *CatBoost: unbiased boosting with categorical features*. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, page 6639–6649, Montréal, Canada, December 2018. Curran Associates Inc. doi:10.5555/3327757.3327770.

[38] T. Tanaka, I. Nambu, Y. Maruyama, and Y. Wada. *Sliding-Window Normalization to Improve the Performance of Machine-Learning Models for Real-Time Motion Prediction Using Electromyography*. Sensors, 22(13):5005, July 2022. doi:10.3390/s22135005.

[39] S. Simtharakao and D. Sutivong. *Exploring Normalization Techniques in Neural Networks for Bitcoin Candlestick Price Prediction*. In 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), page 483–488, February 2023. doi:10.1109/ICAIIC57133.2023.10067086.

4

Event Detection at Building Level: COVID-19 Transmission Risk Estimation in Office Buildings

This chapter continues to explore the use of indoor environmental sensors, shifting focus from building management to occupant wellbeing. Initially, the research aimed to assess indoor environmental comfort in office buildings. However, the onset of the COVID-19 pandemic and the resulting absence of employees in office spaces led to a reorientation of the research objective towards evaluating the risk of COVID-19 transmission in indoor environments.

COVID-19, or coronavirus disease 2019, is a highly contagious illness caused by the SARS-CoV-2 virus. First identified in December 2019 in Wuhan, China, the disease quickly spread worldwide, prompting the World Health Organization (WHO) to declare a global pandemic in March 2020. The virus is primarily transmitted through respiratory droplets and aerosols, with aerosols capable of lingering in the air and traveling longer distances, especially in poorly ventilated spaces.

Unlike previous discussed cases, assessing COVID-19 transmission risk based on indoor environmental measurements poses a unique challenge since there is no ground truth data at all (RG 1.3). However, as the pandemic progressed, experts contributed their research on how to model transmission risk. These experts have the knowledge and credibility to propose such virological models, which form the

foundation of our proposed knowledge-based system. This system uses a semantic ontology to define relationships between sensors and building concepts, and employs SPARQL querying to integrate expert knowledge. By analyzing semantic data, the system generates COVID-19 transmission risk scores per room (RG 2.1) and detects events where the risk is high. This approach enabled the design and development of a fully interpretable methodology (RG 3.2) capable of delivering automated risk assessments using commonly available environmental measurements.

During the pandemic, the proposed model was deployed on a building level (RG 2.2) in three offices (RG 3.1), producing a substantial volume of sensor data. This data stream was efficiently managed and exploited using several IDLab technologies, including DYAMAND[1] for data handling, and the dynamic dashboard[2] to provide employees with real-time visualizations (RG 1.2). The dashboard leverages the semantic properties of my risk assessments to automatically suggest appropriate visualizations and create a custom dashboard for each high-risk event.

This chapter is a slightly adapted version of the following publication:

Vanhaeverbeke, J., Deprost, E., Bonte, P., Strobbe, M., Nelis, J., Volckaert, B., Ongenae, F., Verstockt, S., Van Hoecke, S. (2023). **Real-Time Estimation and Monitoring of COVID-19 Aerosol Transmission Risk in Office Buildings**. SENSORS, 23(5).

Abstract: A healthy and safe indoor environment is an important part of containing the coronavirus disease 2019 (COVID-19) pandemic. Therefore, this work presents a real-time internet of things (IoT) software architecture to automatically calculate and visualize a COVID-19 aerosol transmission risk estimation. This risk estimation is based on indoor climate sensor data, such as carbon dioxide (CO_2) and temperature, which is fed into Streaming MASSIF, a semantic stream processing platform, to perform the computations. The results are visualized on a dynamic dashboard that automatically suggests appropriate visualizations based on the semantics of the data. To evaluate the complete architecture, the indoor climate during the student examination periods of January 2020 (pre-COVID) and January 2021 (mid-COVID) was analyzed. When compared to each other, we observe that the COVID-19 measures in 2021 resulted in a safer indoor environment.

[1] J. Nelis, T. Verschueren, D. Verslype, and C. Develder. DYAMAND: dynamic, adaptive management of networks and devices. In T. Pfeifer, A. Jayasumana, and D. Turgut, editors, Conference on Local Computer Networks, pages 192–195. IEEE, 2012.

[2] S. Vanden Hautte, P. Moens, J. Van Herwegen, D. De Paepe, B. Steenwinckel, S. Verstichel, F. Ongenae, and S. Van Hoecke. A Dynamic Dashboarding Application for Fleet Monitoring Using Semantic Web of Things Technologies. Sensors, 20(4):1152, Feb 2020. doi:10.3390/s20041152.

4.1 Introduction

Because of the coronavirus disease 2019 (COVID-19) pandemic, countries have required employees to work from home and imposed strict safety measures. Now, employees are allowed to partially or completely return to the office and may have to follow certain guidelines, such as increasing ventilation or limiting occupancy, to help reduce the indoor transmission of COVID-19 [1].

Monitoring the indoor carbon dioxide (CO_2) concentration has become a widespread preventive strategy. Multiple governments have suggested keeping the CO_2 level below, for example, 900 ppm [2, 3]. Should this threshold be exceeded, action must be taken to increase ventilation or reduce occupancy. While monitoring this threshold is straightforward in practice with the help of CO_2 sensors, it is too generic to be true in every situation. Researchers have addressed this issue and proposed more advanced models that estimates a safe CO_2 concentration based on the indoor and epidemiological situation [4, 5]. However, this research remains largely unknown and unused by the general public since commercially available CO_2 monitors do not provide such a functionality. These only display the measured CO_2 on a screen, and at best show a color coding according to some fixed thresholds.

Furthermore, while these CO_2 devices are very convenient to use, they only show the results locally. This becomes especially problematic for large buildings where building managers require a good overview of every room for which an internet of things (IoT) system is needed to collect and store the data centrally. The application of IoT systems during the COVID-19 pandemic has also been discussed for many other domains, ranging from personal health tracking to contact tracing [6, 7]. Next to IoT data collection, software tools need to be provided to make monitoring and analysis as easy as possible for users and building managers. This is often performed through mobile or web applications that visualize the data in a dashboard. Many real-time air quality monitoring software architectures were presented in the past that include all necessary components to monitor a complete building [8–10]. However, in order to implement more advanced COVID-19 risk estimation models, a powerful and flexible data stream processing platform is needed, which is missing in existing works. Furthermore, dashboards are often tailor-made for one use case, but users cannot easily adapt them to other applications or needs.

The purpose of our work is two-fold. Firstly, these recent COVID-19 aerosol transmission risk estimation models are applied into practice so occupants can benefit from their research. Secondly, a complete software architecture that collects, processes and visualizes the sensor data stream is proposed.

To achieve this first goal, two existing models [5, 11] were combined and translated into a methodology for the real-time estimation of the COVID-19 aerosol

transmission risk. Since the focus is on office buildings, some assumptions can be made on, for example, the activity level of the occupants, helping make the model more practical without being too general. This methodology is then also converted into a declarative SPARQL query which can be executed by the second part of the research, i.e., the real-time monitoring software architecture.

The proposed software architecture is built to be scalable and flexible in every aspect. First, the sensor data are ingested and stored centrally. Subsequently, the stream processing is performed by microservices that first semantify the data, after which semantic technologies, i.e., SPARQL, can be used to query the data stream and link it with other static semantic building data. As a result, these stream processing services are not only limited to the use case of this chapter, but can handle a large range of other applications, such as general air quality monitoring and comfort scoring. Lastly, the visualization is performed by a dynamic dashboard that suggests visualizations based on the semantics of the data. Additionally, the dynamic dashboard is also capable of receiving events, e.g., a high-COVID-19 transmission risk, and notify the users of this.

This work will allow occupants and building managers to monitor the estimated COVID-19 aerosol transmission risk so that they can take action when needed. Furthermore, it will give the tools to implement and monitor a large variety of other building management applications.

The main contributions of this work are:

- Combination of existing models into a methodology for COVID-19 aerosol transmission risk estimation;
- Creation of a flexible and scalable software architecture to collect, process and visualize large data streams;
- Implementation of a real-time stream processing microservice to execute the defined COVID-19 aerosol transmission risk estimation on incoming semantic sensor data linked with static semantic building data;
- Visualization in a dynamic dashboard which suggests appropriate visualizations based on the semantics of the selected data;
- Automatic notification of occupants and building managers when the estimated COVID-19 risk is too high.

The next section will briefly discuss the existing related research concerning COVID-19 risk estimation and mitigation, and real-time monitoring systems. Some of this literature is then used to establish the risk estimation methodology discussed in Section 4.3. Afterwards, Section 4.4 will go over all elements in our software architecture, from sensors to visualization, which make it possible to employ

the methodology in large buildings. Section 4.5 then discusses the strengths and weaknesses of the proposed work. In Section 4.6, the research is applied in a practical setting in order to compare the impact of preventive on-campus COVID-19 measures. Finally, the conclusions and future research directions are discussed in Section 4.7.

4.2 Related Work

4.2.1 COVID-19 Safety

Some models that estimate the risk of COVID-19 aerosol transmission, the probability of infection and/or safe CO_2 concentration already exist. Peng et al. [4, 12] proposed a model and spreadsheet [3] with multiple examples for common indoor environments, such as a classroom, supermarket or stadium. The parameters can also be adapted for other custom room and viral conditions. Similarly, Bazant et al. [5, 13] created a model and online web application [4] that uses parameters such as room dimensions, human activity, etc., to derive the maximum allowed exposure time or safe CO_2 concentration for an occupant to that room before exceeding a specific transmission risk. Both tools provide dynamic guidelines depending on the indoor environment but lack the ability to provide real-time feedback.

On the other hand, Pang et al. [14] trained an artificial neural network based on the results of computational fluid dynamics (CFD) simulations to predict the COVID-19 infection risk with respect to the CO_2 concentration. A smart ventilation control system was developed using this model in order to reduce the infection risk in the building. While this system dynamically optimizes the indoor environment, it does not provide feedback to the occupants for them to take action.

Others have worked on tools to provide the real-time validation of COVID-19 safety and check whether certain guidelines have been respected. Petrović and Kocić [15] created an IoT architecture that checks social distancing and the wearing of masks with a camera-based system, while also using semantic technologies to link all data. Numerous other literature for social distance validation is available, for example [16, 17]. However, these works do not investigate the risk of aerosol transmission.

[3] https://tinyurl.com/covid-estimator (accessed on 25 July 2022)

[4] https://indoor-covid-safety.herokuapp.com (accessed on 27 July 2022)

4.2.2 Real-time Monitoring Systems

Many real-time monitoring software architectures exist with different focuses on, for example, indoor air quality, indoor comfort and energy consumption. The literature study by Saini et al. reviewed 40 air quality monitoring systems, finding answers to questions such as the used sensors and wireless technologies [18]. Below, some interesting and closely related works are briefly discussed.

Marques et al. proposed the iAirCO_2 system for which they developed their own air quality sensor based on the Arduino hardware platform [8]. As the name of their system suggests, the main focus lies on CO_2 analysis. The captured CO_2 data are transmitted using a Wi-Fi connection to their own software architecture. Users can access and analyze the data via a web and smartphone application which are built using open source technologies. Next to that, they also give the option to set CO_2 thresholds, which produce a notification when exceeded. Although the system is advertised as modular, there is currently no possibility of easily adding additional microservices to perform more advanced processing on the data stream.

Similarly, Benammar et al. proposed a modular IoT platform for real-time indoor air quality monitoring [9]. Instead of focusing on just CO_2, they added a range of additional air quality sensors that measure, for example, sulfur dioxide (SO_2) and nitrogen dioxide (NO_2). This gives a more complete view of the indoor air quality than Marques et al. [8]. Furthermore, the sensor network is designed differently. Instead of directly transmitting the data through Wi-Fi, the sensors are part of a Zigbee mesh network and send their data to a gateway which allows the sensors to be more energy-efficient. The gateway then takes care of the data transmission over Wi-Fi to the webserver. One of the main focuses of this work was ensuring a reliable communication between the sensors, gateway and webserver. Hence, a data logging and retry mechanism is present on the gateway in case of an Internet outage. For data processing and visualization, the open source Emoncms platform was used, which allows for a lot of flexibility but does not take advantage of semantic technologies to, for example, allow the linking of the sensor data to other sources.

Another interesting work is the SAMBA system proposed by Parkinson et al. [10]. It consists of a neat and comprehensive hardware device that monitors the indoor environmental parameters at a desk space and transmits its data to the gateway using a Zigbee mesh network topology. The gateway itself is equipped with cellular technologies, e.g., long-term evolution (LTE), to transmit the data to a centralized web service called IEQAnalytics. There, the data are processed, effectively checking the compliance of the environment with different indoor environment quality standards. An online dashboard visualizes the measured parameters and environment quality indices. The complete SAMBA system is an accurate and user-

friendly means of monitoring the indoor environment quality. However, the visualizations in the dashboard are predefined and cannot be adapted by users with different needs.

4.2.3 Conclusions

Previous research has focused on creating models to estimate the aerosol transmission risk of a certain environment, controlling the building's ventilation system, creating IoT systems that assess social distance or mask wearing, or building software systems for indoor environmental monitoring. However, to the best of our knowledge, no work has presented a flexible and scalable software architecture that uses semantic technologies to estimate and visualize the COVID-19 aerosol transmission risk based on real-time sensor measurements in office buildings.

4.3 COVID-19 Aerosol Transmission Risk Estimation

The COVID-19 risk estimation methodology provides an automatic scoring mechanism of the indoor office environment conditions. Although this estimation does not represent the real virological risk, it is a valuable tool to easily assess whether an indoor climate is favorable for limiting the transmission of COVID-19. Not only does this help increase the awareness of employees regarding the impact of the indoor climate on COVID-19 transmission but it also allows them to take action if the estimated risk is high.

This methodology combines the proposed work of two literature sources. The first and main part of this risk estimation relies on the work of Bazant et al. [5] who proposed a model to estimate the safe CO_2 concentration in a room. To further take the thermal properties of the room into account, the research of Spena et al. [11] was applied to estimate the viral load survival which is then fed into the model of Bazant et al. to improve its estimation.

Figure 4.1 gives an overview of the complete methodology with all its inputs (sensors, building information model (BIM), etc.) at the bottom and the resulting risk estimation at the top. The executed calculation steps for safe CO_2 concentration, viral load survival and risk estimation will be discussed in more detail below.

4.3.1 Safe CO_2 Concentration

Researchers agree that aerosols are one of the main ways of COVID-19 transmission [19, 20]. Whereas respiratory droplets caused by coughing and sneezing are

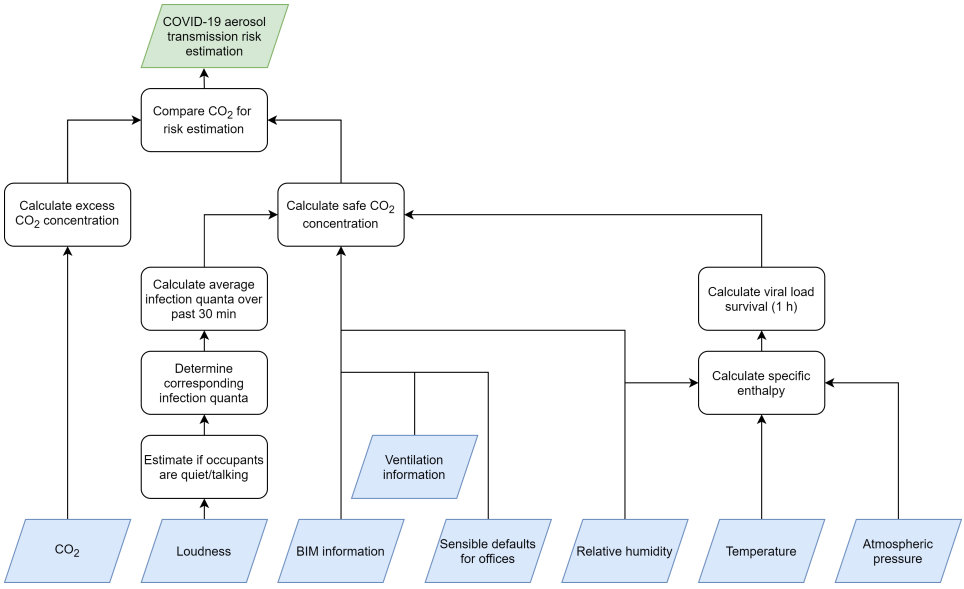


Figure 4.1: Overview of the coronavirus disease 2019 (COVID-19) aerosol transmission risk estimation methodology.

one form of airborne transmission, aerosols are small virus particles produced by exhalation that can float longer in the air than respiratory droplets. This results in a risk of transmission to people in the same room, even when no close contact took place. Therefore, proper ventilation has become one of the key means of reducing the risk of indoor COVID-19 transmission [1].

Because of the importance of ventilation, this risk estimation methodology validates the air quality of a room based on CO_2 measurements. Although the CO_2 concentration is not a perfect reflection of ventilation and air quality, it is often used as a proxy [4, 5, 21] as it can be measured using CO_2 sensors which are cheap, widely available and easily retrofittable in existing buildings. However, this is only valid when people are the main source of CO_2 in a room. Then, increasing CO_2 values indicate that stale air builds up quicker than the room is ventilated. This poses a risk when an infected person is in the same room since higher CO_2 levels correlate with more rebreathed air and thus a higher chance of inhaling virus particles.

The safe CO_2 concentration of the risk estimator is dynamically determined based on the work of Bazant et al. [5] who proposed a model to calculate this for a given environment. The calculated safe CO_2 threshold can then easily be compared to real CO_2 sensor readings. We refer to the paper of Bazant et al. [5] for the details of their model; however, in the following paragraphs, a discussion

of a few important parameters and how they could be configured in order to apply the model in a practical environment is conducted. Most of the suggested default values were obtained from the accompanying web application [5].

A first and important parameter is the prevalence of COVID-19, specified by the percentage of the population which is infected. It is best to determine this based on the effective local COVID-19 infection statistics, but in case these are not available, some literature states that a fixed prevalence value of 0.1% is a possible base case to work with [4]. Equally, for the immunity percentage, it is possible to look at the real local immunity and vaccination rates, but it can also be set to, for example, zero for a more conservative guideline. Every now and then, a new variant emerges which is more infective than the previous. Therefore, it is also important to take the current dominant viral strain into account. Currently, the omicron strain is the most widespread [22], to which this parameter should be set accordingly. The risk tolerance is a parameter that can be freely chosen based on the use case and the desired safety level. The web application suggests a default of 10%.

Another important parameter is the exposure time in the room. The longer occupants are in a room, the longer they are exposed to the potential risk of transmission. The duration of a working day varies from country to country, but according to 2021 statistics, the average working week ranged from 32 to 40 h in Europe [23], resulting in an average workday of approximately 8 h. We use this value as the exposure time in the model of Bazant et al. [5].

The model also requires information on the size of the room. Since a lot of modern office buildings, including ours, have a BIM model available, that can be used to extract the size information per room. Moreover, the ventilation of a room is also of high importance. If some of this information, such as the ventilation rate, is available in the building management system (BMS), it can be directly used from there. Otherwise, the ventilation control scheme and technical installation files can provide the information needed to make a good estimation of these parameters, i.e., the ventilation rate, filtration and recirculation. The relative humidity in the room can be provided in real-time by installed environmental sensors.

Not only are the properties of the pandemic and room included, but the filtration efficiency of facial masks is also considered. This parameter could be dynamically set based on mask detection using a computer vision system, but an easier and more privacy-friendly way would be to define it according to the corporate mask policy. For example, if masks are not enforced, the filtration efficiency can be set to zero.

How much CO_2 and virus aerosols are produced is influenced by the occupant's activity. Since this methodology applies the work of Bazant et al. [5] specifically

[5] https://indoor-covid-safety.herokuapp.com (accessed on 27 July 2022)

in an office environment, it is assumed that the occupants will be mostly sitting and not performing heavy physical activities. On the other hand, whether the occupants are talking varies. Therefore, a coarse-grained differentiation is made between two activities, i.e., breathing and talking, based on decibel level sensor data, as shown in Figure 4.1. When the measured decibel level is lower or equal than 50 dB, we assume that everyone is working quietly in the room and that the production of CO_2 and virus particles is at the rate of breathing. A decibel level higher than 50 dB could indicate that some of the occupants are talking which allows adjusting of the model's parameters accordingly. The 50 dB threshold was empirically determined based on the historical sensor data of our offices but can be chosen differently for other environments. This elementary estimation has its limitations (e.g., sound intensity decreases with distance, other sources of noise can be present, etc.), but the simplicity and fact that it is non-privacy intrusive are a clear advantage.

Table 4.1 shows the examples of the predicted safe CO_2 concentration by the model of Bazant et al. [5] for the two different activities, breathing and talking. These calculations were performed using the web application, leaving most of the parameters at their default, to show the significantly different result between the two activities. As can be seen, the safe excess CO_2 concentration for talking is approximately 4 times smaller than for breathing, which is due to the infection quanta of talking being much larger. Note that these are just examples, and the effective safe CO_2 concentration will be dynamically calculated by the system described in Section 4.4 based on the real parameters of the room.

Table 4.1: Example of safe carbon dioxide (CO_2) concentrations for the occupant activities: breathing (working quietly) and talking.

Parameter	Breathing ($\leq$50 dB)	Talking ($>$50 dB)
Prevalence	0.001	0.001
Risk tolerance	0.1	0.1
Viral strain	Omicron BA.2	Omicron BA.2
Exposure time	8 h	8 h
Floor area	200 m²	200 m²
Respiratory activity	Breathing (4.2 q/m³)	Talking (72 q/m³)
Safe excess CO_2	850 ppm	209 ppm
+ background CO_2 [24]	1265 ppm	624 ppm

The large difference in the safe CO_2 threshold of breathing and talking would also lead to large fluctuations in the risk estimation if occupants go from one ac-

tivity to the other. This could confuse the occupants and ultimately demotivate them to further monitor it. Therefore, not only could the current decibel measurement be used to decide which infection quanta to apply, but all decibel readings of the past 30 min were used to calculate the average infection quanta for that period, as shown in Equation (4.1). This allows for a more smooth transition of the quanta, and thus risk estimation, between these two activities.

$$q_{avg} = \frac{n_{breath} * q_{breath} + n_{talk} * q_{talk}}{n_{breath} + n_{talk}} \tag{4.1}$$

where:

q_{avg} = the average infection quanta (q/m³)

n_{breath} = the number of decibel measurements considered breathing ($\leq$50 dB)

n_{speak} = the number of decibel measurements considered speaking (>50 dB)

q_{breath} = the infection quanta for breathing (q/m³)

q_{talk} = the infection quanta for speaking (q/m³)

4.3.2 Viral Load Survival

The thermal conditions of the environment also have an influence on the risk of transmission. While these are integrated in the model of Bazant et al. [5] through the effect of the "viral decay rate" parameter, other literature more extensively estimates this, incorporating both the temperature and humidity conditions. Although research is not consistent and shows a different relation and significance, the common trend is that COVID-19 spreads more in a cold and dry indoor climate [11, 25]. Therefore, it is recommended to optimize the indoor temperature and humidity in buildings to reduce the survival time of COVID-19 aerosols. An additional benefit of a higher humidity is that aerosol droplets will grow due to hygroscopy and fall down quicker. This reduces the time that virus particles are airborne and thus reduces their ability to travel long distances [26].

For a more extensive estimation of the viral survival time, the COVID-19 research performed by Spena et al. [11] was incorporated. Temperature, relative humidity and atmospheric pressure were used to calculate the specific enthalpy, which is a measure of the energy in a thermodynamical system. Based on the literature data of COVID-19 and other related viruses, Spena et al. studied the relation between the specific enthalpy and the viral load survival after 1 h (VLS_{1h}) for different thermal conditions, which revealed a quadratic relation, as shown by Equation (4.2). This quadratic relation is used by our methodology to estimate the

viral load survival based on the real-time indoor conditions, and is then converted into the viral decay rate in order to pass it to the model of Bazant et al. [5].

$$VLS_{1h} = \begin{cases} 1 & \text{if } h \leq 38 \text{ or } h \geq 67 \\ C_1 h^2 + C_2 h + C_3 & \text{otherwise} \end{cases} \tag{4.2}$$

where:

$$\begin{aligned} VLS_{1h} &= \text{viral load survival after 1 h (\%)} \\ h &= \text{specific enthalpy } (kJ/kg_{dry\ air}) \\ C_1 &= 0.0047562426 \\ C_2 &= -0.4994054697 \\ C_3 &= 13.1093935791 \end{aligned}$$

4.3.3 Risk Estimation

With the safe CO_2 concentration calculated based on the room's conditions, it is possible to take the last step, namely estimating the COVID-19 aerosol transmission risk. Therefore, the current excess CO_2 concentration needs to be calculated based on the last real-time CO_2 measurement. There are multiple options to tackle this. The easiest but incomplete option is to use the global average CO_2 level, which is currently about 415 ppm [24]. A second and better option is to find the minimum sensor reading over a period of time (e.g., 1 week) and use the found minimum as the background CO_2 level. This assumes that the room will be completely ventilated at least once during the chosen period. Once the background CO_2 level is known, the excess CO_2 concentration can be easily calculated by subtracting the background CO_2 from the measured CO_2 concentration. Afterwards, the measured excess CO_2 can be compared to the estimated safe excess CO_2 concentration by taking the ratio between both values as shown in Equation (4.3). This results in a unitless number specifying the estimated risk. As a final step, this number is clipped between 0 and 1 which gives us the final COVID-19 aerosol transmission risk estimation.

$$RE = CO_{2,measured} / CO_{2,safe} \tag{4.3}$$

where:

$$\begin{aligned} RE &= \text{risk estimation} \\ CO_{2,measured} &= \text{the measured excess } CO_2 \text{ concentration (ppm)} \\ CO_{2,safe} &= \text{the safe excess } CO_2 \text{ concentration (ppm)} \end{aligned}$$

4.4 Real-Time Monitoring System

The real-time monitoring of our COVID-19 aerosol transmission risk is handled by a modern software architecture which is a combination of microservices fulfilling different tasks, such as data ingestion, persistence, risk estimation, and visualization in a dashboard. Using a microservice architecture provides fault isolation, eliminates long-term commitment to a single technology stack and is easily extendable by adding services later on. These microservices can also be easily scaled up or down to handle different load scenarios. The overview of the microservice architecture is given in Figure 4.2, of which the individual components will be detailed in the following subsections.

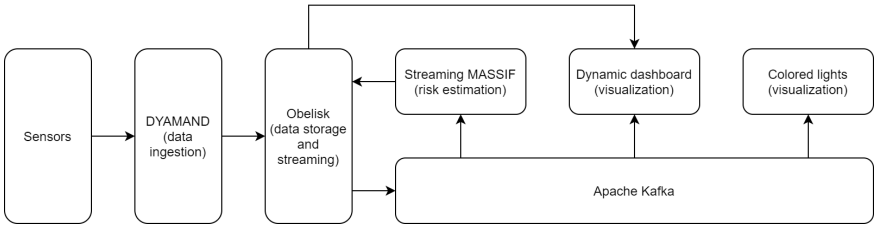


Figure 4.2: Overview of the microservice architecture.

4.4.1 Sensors

Slightly over a hundred Netatmo Smart Home Weather stations [6] were installed to measure the environmental conditions in every room of our office building. These are off-the-shelf commercially available sensors that measure the environmental parameters for the COVID-19 aerosol transmission risk estimation presented in Section 4.3. These sensors communicate through Wi-Fi which makes the integration in existing buildings easy as those usually already have a Wi-Fi infrastructure. The Netatmo stations are placed at desk height and in the center of the room, away from ventilation exhausts or inlets and windows, which can have a significant impact on the measurements. The nondispersive infrared (NDIR) CO_2 sensor auto-calibrates itself once a week by setting the minimum measured value to 400 ppm. Every 10 min, the environmental parameters are measured and transmitted.

Nevertheless, our monitoring system is not limited to a specific type or brand of sensor. It can process environmental data from any source since the DYAMAND [27] middleware is used in the system architecture which serves as an interoperability layer that abstracts and standardizes the communication between

[6] https://www.netatmo.com/en-row/weather/weatherstation (accessed on 28 January 2022)

smart devices and applications. This standardization ensures that no hardware-specific adaptations need to be made further down the processing pipeline.

4.4.2 Semantic Sensor Metadata

All sensors are accompanied by metadata which are stored as a semantic graph. For this, the standard semantic sensor network (SSN) ontology [28] is used which is specifically designed for the description of sensor networks and their measurements. These metadata allow us to easily query the data, for example, selecting all sensors that measure CO_2. Additionally, this also facilitates the linking of different parts of information, such as the location, which is, on its turn, also semantically described. Each room in the building is available in the ontology with the information extracted from the BIM model, such as the room's name, floor, area, etc.

Similarly to how the physical sensors and their measurements are described semantically, the COVID-19 risk estimator is also defined as an, albeit virtual, semantic sensor, as shown in Figure 4.3. By doing this, the results are linked to the room and available for further usage just like any sensor.

```
igent:COVID-19-Risk-Estimator-200.020
    a sosa:Sensor ;
    rdfs:label "COVID-19 risk estimator in room 200.020" ;
    brick:hasLocation igent:room-200.020 ;
    sosa:observes <Kantoor%20200.020/risk_estimation::number> .

<Kantoor%20200.020/risk_estimation::number>
    a sosa:ObservableProperty ;
    rdfs:label "risk estimation" ;
    dashb:produces metrics:quantity .

igent:room-200.020 a brick:Room .

igent:floor-200 a brick:Floor ;
    brick:hasPart igent:room-200.020 .
```

Figure 4.3: Semantic annotation in Turtle format of a COVID-19 risk estimator, describing it as a (virtual) sensor with a location and the COVID-19 aerosol transmission risk estimation as observed property.

4.4.3 Ingestion, Persistence and Messaging

As mentioned in Section 4.4.1, the Netatmo sensors transmit their measurements through Wi-Fi to the DYAMAND middleware which standardizes the data. After-

wards, the result is transmitted to Obelisk [29], a scalable IoT integration platform, which both stores the data and streams them onto Apache Kafka [7]. As shown in Figure 4.2, this Kafka bus is the central element in our architecture and allows for the easy addition of new services while still decoupling them all.

4.4.4 Streaming MASSIF

With the sensor data streamed onto the Kafka bus, different applications can now make use of it. In our software architecture, Streaming MASSIF [30] is used to process these sensor data. Streaming MASSIF is an in-house designed stream processing platform that solves data velocity and variety by exploiting stream reasoning techniques. Targeting velocity is necessary for our scenario as the sensors continuously produce data. Variety needs to be targeted as the sensor observations need to be combined with static data, such as the semantic sensor ontologies. The Streaming MASSIF web interface [31] is used to set up the complete processing pipeline of the COVID-19 aerosol transmission risk estimation, as shown in Figure 4.4. An overview of the components will be given in the following paragraph.

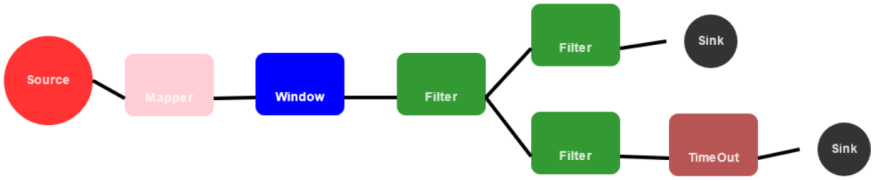


Figure 4.4: Streaming MASSIF pipeline to calculate the COVID-19 aerosol transmission risk estimation.

The first component, left in Figure 4.4, is a “source” which is the point where the sensor data come in. Multiple source types are available, but here the source was configured to read all data of a certain topic from the Kafka bus. When data come in, these are sent to the next component which is a “mapper”. As the name implies, this component can map the incoming messages to another format. In this case, it is used to transform the sensor data from Kafka into a semantic structure which allows us to use semantic technologies, i.e., SPARQL, in the later steps. Then, the data are passed to the “window” component. This collects and groups all data over a given time period (window size) and outputs them with a certain interval (stride). For this use case, the window size is set to 30 min which allows us

[7] https://kafka.apache.org (accessed on 16 March 2022)

to calculate statistics and perform aggregations over a longer period of time. The stride is set to 10 min since the sensors also transmit their data at this rate. With each stride, the window data are passed to the next block which is a "filter". This versatile and powerful component allows us to write any SPARQL query to act on the semantic messages. Next to that, ontologies can also be included in order to link additional semantic data, such as the sensor descriptions and locations. This filter block provides the implementation and execution of our COVID-19 aerosol transmission risk estimation using a declarative SPARQL query which consists of three major steps: firstly, all required data are queried, secondly, the calculations are performed, and lastly, the results are outputted semantically. After this filter, the pipeline branches into two (see Figure 4.4): one branch for feeding the calculated risk estimations back into the software architecture (upper); and one to throw events when the risk estimation in a room is too high (lower). The upper branch consists of another "filter", containing a simple SPARQL query to select the risk estimations of the previous block, and a "sink" that sends the output to the configured destination. Just like the "source" component, multiple destinations are available, but Obelisk is chosen in this case. The lower branch also starts with another "filter" that selects the risk estimations and checks whether they are higher than a set threshold. If that is the case, an event graph is constructed. The following "time out" block makes sure that the event is not sent to the "sink" more than once in a certain period, e.g., 1 h.

4.4.5 Dynamic Dashboard

The last step in the COVID-19 risk monitoring architecture was the dynamic dashboard where users can monitor the building status and be alerted of events and anomalies. Dashboards need to balance between flexibility and ease of use. Classical dashboards require the user to specify and configure the widgets of each desired visualization. To solve this shortcoming, our in-house designed semantic dashboard [32] dynamically suggests suitable visualizations by reasoning over the sensors' semantic descriptions and supported visual widgets.

For the use case at hand, a new visualization was added to dynamically visualize the sensor values (e.g., CO_2, COVID-19 risk estimation, etc.) as a heatmap. The visualization is based on floor plans acquired from the BIM model of the building. Every room is colored based on the requested sensor value which gives an easy overview of the different rooms. Figure 4.5 depicts such floor plans, visualizing the CO_2 concentration or risk estimation.

Figure 4.5: Heatmap visualization of the CO_2 concentration and COVID-19 aerosol transmission risk estimation of the 10th floor of our office building. Green colors indicate low, good values while red colors highlight areas with high, bad environmental parameters.

4.4.5.1 Visualization Suggestion

The dashboard dynamically suggests visualizations based on the selected sensor(s). This suggestion is made by semantic reasoning over the metadata of sensors and visualizations. Reasoning allows our approach to be loosely coupled, facilitating the reuse of the same functionality on sensors described with another ontology. The visualizations are annotated using our own dashboard ontology. Figure 4.6 shows the annotation of a heatmap visualization. The most important fields in the semantic metadata are "dashb:accepts" which describes the supported datatypes, and "dashb:locationScope" which describes the location that this visualization is relevant for. Thus, in the example of Figure 4.6, one can see that the heatmap accepts any quantitative data (xsd:double) and is only relevant for sensors located on the 10th floor. When the user selects one or multiple sensors, the reasoning engine will infer which visualizations are compatible by ensuring that the sensor data type and location match those accepted by the visualization.

```
<heatmap-10th-floor>
    a dashb:HeatMap, dashb:RealtimeDataVisualization ;
    rdfs:label "Heatmap 10th floor" ;
    dashb:component [
        dashb:accepts [ dashb:datatype xsd:double ] ;
        dashb:locationScope igent:floor-200] .
```

Figure 4.6: Semantic description in Turtle format of the real-time heatmap visualization for the 10th floor of our office building.

4.4.5.2 Event View

The dynamic dashboard is not only capable of visualizing data upon request by the user, but it can also react to events. Whenever the COVID-19 risk estimation exceeds a defined threshold, Streaming MASSIF sends out an event to Obelisk which is then again pushed to Kafka (see Figure 4.2). As the dashboard is listening to the Kafka, it automatically and dynamically constructs a new tab based on this event containing information about the signal that triggered the event, a textual description, the occurrence time, and optionally multiple stimuli. A stimulus is a data signal that influenced the event and is added to the event view tab as a visualization widget. The dashboard reasoner is used on all data signals to dynamically infer which visualization is best suited.

Figure 4.7 describes an example of a "High-COVID-19 risk estimation" event produced by Streaming MASSIF. In this case, the stimuli are the sensor measurements used to calculate the risk estimation. An example of the resulting dynamically created dashboard tab is shown in Figure 4.8.

4.4.6 Colored Lights

While the dashboard is available for all employees, probably not everyone will check it continuously during the day. Next to that, our offices are often visited by external partners and students who do not have access to the dashboard. Therefore, a second and more direct visualization is provided by means of colored lights. Smart Yeelight [8] light bulbs were installed in free-standing lamps in every office and meeting room, as can be seen in Figure 4.9. The lights are controlled by a microservice that listens to the Kafka bus (see Figure 4.2). Three possible colors are set according to the risk estimation: green when the risk estimation is ≤ 0.6, orange when it is > 0.6 and < 0.9, and red when it is ≥ 0.9.

4.5 Strengths and Weaknesses

4.5.1 COVID-19 Aerosol Risk Estimation

The risk estimation methodology heavily relies on CO_2 data which has both benefits and limitations. Its main advantage is the ease of use since CO_2 is easily measured using sensors while still being a good proxy for indoor air quality [4, 5, 21]. This comes with the important assumption, and possible limitation, that people must be the main source of CO_2 in the room. Only then will CO_2 levels correlate

[8] https://en.yeelight.com) (accessed on 10 August 2022)

```
:event_0
    ns0:observedProperty  <Kantoor%20200.020/metrics/risk_estimation::number> ;
    dc:description "High COVID-19 risk estimation" ;
    sosa:resultTime "2022-11-13T21:13:04.432000+01:00" ;
    ssn:wasOriginatedBy
        :stimulus_0 ,
        # More stimuli...

:stimulus_0
    a ssn:Stimulus ;
    ns0:observedProperty
        <Kantoor%20200.020/metrics/CO2::number> ;
    ns0:fromObservation
        [sosa:resultTime "2022-11-13T21:38:04.432000+01:00"] ;
    ns0:toObservation
        [sosa:resultTime "2022-11-13T22:08:04.432000+01:00"] .

# More stimulus definitions...
```

Figure 4.7: Semantic annotation in Turtle format of a "High COVID-19 risk estimation" event produced by Streaming MASSIF. The event is linked to the COVID-19 aerosol transmission risk estimation metric of a room, the occurrence time and multiple stimuli.

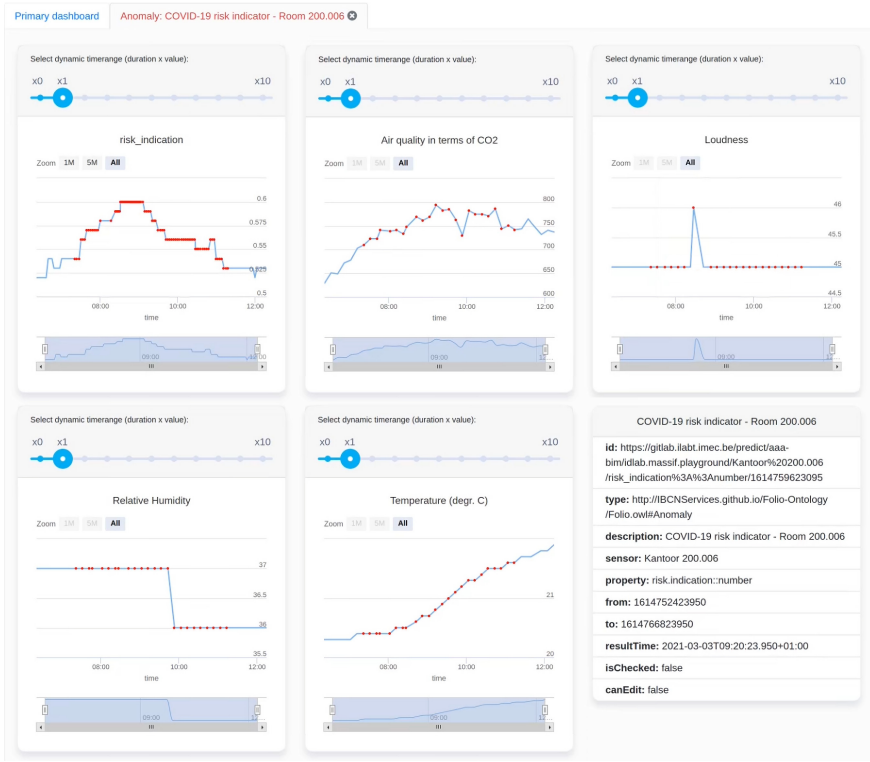


Figure 4.8: Dynamically constructed event view when a "High COVID-19 risk estimation" event occurs, visualizing the COVID-19 aerosol transmission risk estimation signal and linked stimuli.

Figure 4.9: A colored light showing the real-time COVID-19 aerosol transmission risk estimation status in a meeting room.

with the respiratory activity of occupants in the room. When other sources are present, e.g., gas furnaces, this assumption is no longer true, and the analysis of CO_2 data will give misleading results. Another possible weakness is the importance of the placement of the CO_2 sensors. On the one hand, these need to be located close to the used desk space, but on the other hand, they should not be too close to persons, windows, doors, and ventilation.

Besides CO_2, our risk estimation methodology also takes advantage of other environmental parameters, such as temperature and humidity. These are equally practical to measure since they are often incorporated in one measurement device. By using these additional sensor values, our methodology can give a more complete view of the COVID-19 aerosol transmission risk than checking a fixed CO_2 threshold. On the downside, the methodology also needs a lot of additional parameters on the building. Some are easy to set according to preference or policy, but others can be more difficult to provide and may require the help of a building manager.

The estimation of the occupant's activity is another strength of this methodology. While the approach for this is basic, it still brings an important improvement since breathing and speaking have vastly different infection quanta. Nevertheless, it can also be incorrectly influenced by, for example, loud background noise in which case a more advanced sound analysis would be beneficial.

Lastly, this methodology only focuses on giving information to occupants and building managers. The responsibility of acting when needed is up to them. Depending on the use case, it might be more interesting to control the ventilation system directly as in [14].

4.5.2 Real-Time Monitoring System

The use of commercial environmental sensors can be seen as a benefit and limitation at the same time. On the one hand, these sensors can be bought in large numbers and are guaranteed to work if from a trustworthy vendor. On the other hand, the available functionality is provided by a third party and can sometimes be a limiting factor. Related works building their own sensor device are more flexible in that respect.

The complete architecture is designed to be performant, flexible and modular. Obviously, this comes with some complexity as well. This type of system is not meant to be set up by a non-technical user, such as a CO_2 monitor, but requires experience in the deployment and maintenance of the Kubernetes cluster containing all the components. Another downside is the dependence on Internet access of our system. Currently, there is no fallback mechanism in the case of an Internet outage as in the research by [9].

The Kafka bus in our software system enables a microservice architecture that brings a multitude of benefits, such as flexibility, scalability, fault isolation, etc. Different services can easily plug into the bus to bring additional functionalities. One of the services in our case is Streaming MASSIF, which allows us to use semantic technologies to process the sensor data stream. This brings the benefit that all data are semantically described and linked, from observation to room, allowing the implementation of a large variety of use cases besides this COVID-19 risk estimation. A downside is that knowledge of semantic technologies is required.

Similar strengths are true for our dynamic dashboard, which uses the power of semantic data to achieve a flexible and user-friendly interface. The users can add visualizations according to their preference and receive suggestions for these based on the selected data. The dashboard also receives events and automatically visualizes all included information, which helps in finding and analyzing problems and risks in the building.

4.6 Functional Evaluation: Impact of On-Campus COVID-19 Measures

The COVID-19 risk estimation was evaluated by comparing the statistics of two auditoriums of our university building during a pre- and mid-COVID-19 period. As the pre-COVID-19 period, the examination session of January 2020 was used. COVID-19 was not present in Belgium back then, so exams were made on-campus without any increased ventilation or occupancy restriction. As the mid-COVID-19 period, the examination session of the following year, January 2021, was used.

Then, COVID-19 was a worldwide pandemic and restrictions applied in Belgium. On-campus exams were only allowed with a limited number of students and increased ventilation was required. Comparing the risk estimations of these two periods allows us to see whether the enforced guidelines and rules effectively resulted in a safer indoor environment.

Normally, for the calculation of the risk estimation, the prevalence of COVID-19 should be dynamically set based on the local epidemiological statistics, but since none are available for January 2020 (pre-COVID-19), that would lead to unusable results for comparison. Therefore, the prevalence was set to a fixed value of 0.1% for both examination periods to allow us to compare the effect of the additional guidelines on the risk estimations. Both examination periods were equally long and thus have an equal amount of data samples to process.

Figure 4.10 shows the box plots of the risk estimations of a large and small auditorium during the examination period of January 2020 and 2021. For the large auditorium, it is clear that the maximum risk estimation is much lower in 2021 than 2020 thanks to the applied guidelines and restrictions. Additionally, the 25% quantile, 75% quantile, and mean risk estimations are also lower in 2021. For the small auditorium, the maximum risk estimation is also much lower in 2021 compared to 2020, again showing the positive impact of the safety rules. However, the other statistics are higher because this small auditorium was used more frequently in 2021. Students had to be spread across multiple rooms to give everyone the opportunity to take the exam at the same time, resulting in higher room occupation than normal.

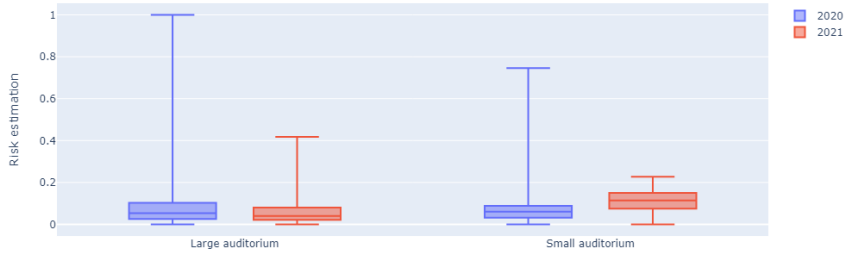


Figure 4.10: COVID-19 aerosol transmission risk estimation box plots of the examination periods of January 2020 and January 2021 for two auditoriums.

To statistically test whether the risk estimations in 2021 were lower than in 2020, an alternative hypothesis was defined accordingly and the p-values were

calculated based on the data of the large and small auditoriums. Since the data were not normally distributed, the Mann–Whitney U test was used. As shown in Table 4.2, the *p*-value for the large auditorium was far below $p < 0.05$, and is thus statistically significant, suggesting that our alternative hypothesis is true and that the taken COVID-19 measures had a positive impact on the estimated COVID-19 aerosol transmission risk. The *p*-value for the small auditorium on the other hand is far above the significance level of $\alpha = 0.05$, thus rejecting the alternative hypothesis. This is in line with our expectation, given the fact that this small auditorium was more frequently used in 2021 as mentioned earlier.

Table 4.2: p-*values calculated with the Mann–Whitney U test checking whether the COVID-19 aerosol transmission risk estimations in 2021 are lower than in 2020 for the large and small auditorium.*

Room	p-Value
Large auditorium	1.13×10^{-16}
Small auditorium	1

4.7 Conclusions and Future Research

Given the importance of aerosol transmission of COVID-19, we designed and implemented a real-time software architecture to estimate and monitor the COVID-19 transmission risk in office buildings, helping to increase the awareness of occupants and safety inside. The platform is able to ingest large volumes of environmental sensor data in real time with the help of Obelisk. Then, the Kafka bus enables a versatile and scalable microservice architecture, allowing processing and visualization services to be easily added. The COVID-19 aerosol transmission risk estimation was implemented as an intuitive pipeline in Streaming MASSIF which lets us combine the power of semantic and stream processing technologies. The resulting risk estimations can be visualized on the floor plan widget of our dynamic dashboard to enable occupants and building managers to grasp the COVID-19 risk estimations with one glance. Finally, the colored lights bring an even more accessible visualization inside our building. The proposed software system is not only limited to the COVID-19 aerosol transmission risk estimation, but is flexible and modular, allowing the addition of other building management use cases, such as general indoor air quality testing and comfort scoring. Next to the practical deployment of this system in our university building, a comparison

of the estimated COVID-19 aerosol transmission risk was performed in two auditoriums during a pre- and a mid-COVID-19 examination period. The estimated risk in January 2021 for the large auditorium was statistically significantly lower than in January 2020, showing the positive impact of the COVID-19 safety measures taken in 2021. The same conclusion could not be drawn for the small auditorium. However, this small auditorium was also used more often during the examination period of January 2021 since students had to be spread across rooms.

Multiple aspects of this work can still be improved in future research. Currently, the position of the installed sensors is only known at the room level. To make our risk estimation more spatially fine-grained, the location in the room should be further refined by linking the exact sensor position to the BIM model. Next to that, the elementary activity estimation based on decibel-level sensor measurements could be improved by analyzing the complete sound signal instead. This would increase the hardware and processing demands of our system, but should allow for a better and more fine-grained estimation of the different speaking levels, i.e., whispering, normal talking, loud talking, etc. Currently, making the floor plan views is a manual task that needs to be performed once when setting up a new building. Since the BIM model is already available, that will be used in future work to automatically generate the floor plan visualizations and link the sensors to their respective locations on the map. Another step that could be taken in future research is the implementation of edge processing capabilities. Now, all the processing for the COVID-19 risk estimation is executed in the cloud by Streaming MASSIF, but large parts of the computations can be offloaded to an edge device, benefitting the scalability of Streaming MASSIF. Finally, COVID-19 research is still heavily ongoing, so the methodology of the risk estimation should be updated to reflect future insights into the virus and its transmission.

References

[1] L. Morawska and D. K. Milton. *It Is Time to Address Airborne Transmission of Coronavirus Disease 2019 (COVID-19)*. Clinical Infectious Diseases, 71(9):2311–2313, Nov 2020. doi:10.1093/cid/ciaa939.

[2] FPS Public Health (Belgium). *Ventilation | Coronavirus COVID-19*. Available from: https://www.info-coronavirus.be/en/ventilation.

[3] High Council of Public Health (France). *Covid-19 : aeration, ventilation and CO2 measurement in Public Accesses Buildings (PAB)*. Available from: https://www.hcsp.fr/explore.cgi/avisrapportsdomaine?clefr=1114.

[4] Z. Peng and J. L. Jimenez. *Exhaled CO2 as a COVID-19 Infection Risk Proxy for Different Indoor Environments and Activities*. Environmental Science & Technology Letters, 8(5):392–397, May 2021. doi:10.1021/acs.estlett.1c00183.

[5] M. Z. Bazant, O. Kodio, A. E. Cohen, K. Khan, Z. Gu, and J. W. M. Bush. *Monitoring carbon dioxide to quantify the risk of indoor airborne transmission of COVID-19*. medRxiv, page 2021.04.04.21254903, Apr 2021. doi:10.1101/2021.04.04.21254903.

[6] A. A. Al-Atawi, F. Khan, and C. G. Kim. *Application and Challenges of IoT Healthcare System in COVID-19*. Sensors, 22(1919):7304, Jan 2022. doi:10.3390/s22197304.

[7] M. S. Al-kahtani, F. Khan, and W. Taekeun. *Application of Internet of Things and Sensors in Healthcare*. Sensors, 22(1515):5738, Jan 2022. doi:10.3390/s22155738.

[8] G. Marques, C. R. Ferreira, and R. Pitarma. *Indoor Air Quality Assessment Using a CO2 Monitoring System Based on Internet of Things*. Journal of Medical Systems, 43(3):67, Feb 2019. doi:10.1007/s10916-019-1184-x.

[9] M. Benammar, A. Abdaoui, S. H. M. Ahmad, F. Touati, and A. Kadri. *A Modular IoT Platform for Real-Time Indoor Air Quality Monitoring*. Sensors, 18(22):581, Feb 2018. doi:10.3390/s18020581.

[10] T. Parkinson, A. Parkinson, and R. de Dear. *Continuous IEQ monitoring system: Context and development*. Building and Environment, 149:15–25, Feb 2019. doi:10.1016/j.buildenv.2018.12.010.

[11] A. Spena, L. Palombi, M. Corcione, M. Carestia, and V. A. Spena. *On the Optimal Indoor Air Conditions for SARS-CoV-2 Inactivation. An Enthalpy-Based Approach*. International Journal of Environmental Research and Public Health, 17(1717):6083, Jan 2020. doi:10.3390/ijerph17176083.

[12] Z. Peng, A. P. Rojas, E. Kropff, W. Bahnfleth, G. Buonanno, S. Dancer, J. Kurnitski, Y. Li, M. Loomans, L. Marr, L. Morawska, W. Nazaroff, C. Noakes, X. Querol, C. Sekhar, R. Tellier, T. Greenhalgh, L. Bourouiba, A. Boerstra, J. Tang, S. Miller, and J. Jimenez. *Practical Indicators for Risk of Airborne Transmission in Shared Indoor Environments and Their Application to COVID-19 Outbreaks*. 56:1125–1137, Jan 2022. doi:10.1021/acs.est.1c06531.

[13] M. Z. Bazant and J. W. M. Bush. *Beyond Six Feet: A Guideline to Limit Indoor Airborne Transmission of COVID-19*. Sep 2020. Available from:

http://medrxiv.org/lookup/doi/10.1101/2020.08.26.20182824, doi:10.1101/2020.08.26.20182824.

[14] Z. Pang, P. Hu, X. lu, Q. Wang, and Z. O'Neill. *A Smart CO2-Based Ventilation Control Framework to Minimize the Infection Risk of COVID-19 In Public Buildings*. Feb 2021.

[15] N. Petrović and D. Kocić. *IoT-based System for COVID-19 Indoor Safety Monitoring*. Sep 2020.

[16] M. Rezaei and M. Azarmi. *DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic*. Applied Sciences, 10(2121):7514, Jan 2020. doi:10.3390/app10217514.

[17] I. Ahmed, M. Ahmad, J. J. P. C. Rodrigues, G. Jeon, and S. Din. *A deep learning-based social distance monitoring framework for COVID-19*. Sustainable Cities and Society, 65:102571, Feb 2021. doi:10.1016/j.scs.2020.102571.

[18] J. Saini, M. Dutta, and G. Marques. *Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review*. International Journal of Environmental Research and Public Health, 17(1414):4942, Jan 2020. doi:10.3390/ijerph17144942.

[19] S. L. Miller, W. W. Nazaroff, J. L. Jimenez, A. Boerstra, G. Buonanno, S. J. Dancer, J. Kurnitski, L. C. Marr, L. Morawska, and C. Noakes. *Transmission of SARS-CoV-2 by inhalation of respiratory aerosol in the Skagit Valley Chorale superspreading event*. Indoor Air, 31(2):314–323, Mar 2021. doi:10.1111/ina.12751.

[20] T. Greenhalgh, J. L. Jimenez, K. A. Prather, Z. Tufekci, D. Fisman, and R. Schooley. *Ten scientific reasons in support of airborne transmission of SARS-CoV-2*. The Lancet, 397(10285):1603–1605, May 2021. doi:10.1016/S0140-6736(21)00869-2.

[21] A. Hartmann and M. Kriegel. *Risk assessment of aerosols loaded with virus based on CO2-concentration*. Jul 2020. Accepted: 2020-08-19T10:10:44Z. Available from: https://depositonce.tu-berlin.de/handle/11303/11478.3, doi:10.14279/depositonce-10362.3.

[22] World Health Organization. *Tracking SARS-CoV-2 variants*. Available from: https://www.who.int/activities/tracking-SARS-CoV-2-variants.

[23] *Hours of work - annual statistics - Europe*. Available from: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Hours_of_work_-_annual_statistics.

[24] P. Tans and R. Keeling. *Global Monitoring Laboratory - Carbon Cycle Greenhouse Gases*, 2022. Available from: https://gml.noaa.gov/ccgg/trends/data.html.

[25] P. Dabisch, M. Schuit, A. Herzog, K. Beck, S. Wood, M. Krause, D. Miller, W. Weaver, D. Freeburger, I. Hooper, and et al. *The influence of temperature, humidity, and simulated sunlight on the infectivity of SARS-CoV-2 in aerosols*. Aerosol Science and Technology, 0(0):1–12, Oct 2020. doi:10.1080/02786826.2020.1829536.

[26] A. Ahlawat, A. Wiedensohler, and S. K. Mishra. *An Overview on the Role of Relative Humidity in Airborne Transmission of SARS-CoV-2 in Indoor Environments*. Aerosol and Air Quality Research, 20(9):1856–1861, 2020. doi:10.4209/aaqr.2020.06.0302.

[27] J. Nelis, T. Verschueren, D. Verslype, and C. Develder. *DYAMAND: dynamic, adaptive management of networks and devices*. In T. Pfeifer, A. Jayasumana, and D. Turgut, editors, Conference on Local Computer Networks, pages 192–195. IEEE, 2012.

[28] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, and et al. *The SSN ontology of the W3C semantic sensor network incubator group*. Journal of Web Semantics, 17:25–32, Dec 2012. doi:10.1016/j.websem.2012.05.003.

[29] V. Bracke, M. Sebrechts, B. Moons, J. Hoebeke, F. De Turck, and B. Volckaert. *Design and evaluation of a scalable Internet of Things backend for smart ports*. Software: Practice and Experience, 51, Apr 2021. doi:10.1002/spe.2973.

[30] P. Bonte, R. Tommasini, E. Della Valle, F. De Turck, and F. Ongenae. *Streaming MASSIF: cascading reasoning for efficient processing of iot data streams*. Sensors, 18(11):3832, 2018. doi:10.3390/s18113832.

[31] P. Bonte and F. Ongenae. *RDF Stream processing prototyping with streaming MASSIF*. In ISWC2020, the International Semantic Web Conference, pages 1–4, 2020.

[32] S. Vanden Hautte, P. Moens, J. Van Herwegen, D. De Paepe, B. Steenwinckel, S. Verstichel, F. Ongenae, and S. Van Hoecke. *A Dynamic Dashboarding Application for Fleet Monitoring Using Semantic Web of Things Technologies*. Sensors, 20(4):1152, Feb 2020. doi:10.3390/s20041152.

5

Event Detection at Regional Level: Point of Interest Recognition in Aerial Video

Live sports broadcasts, whether delivered via television or online streaming platforms, generate large volumes of visual data, offering great potential for automated event detection applications. Among these sports, cycling holds a special place in Belgian culture, captivating audiences through its intense races and the scenic landscapes showcased along the routes. This chapter explores an event detection use case within the sports entertainment domain, using live video data spanning large regions.

Although the primary focus of cycling broadcasts is the competition itself, these broadcasts frequently feature regional landmarks. Currently, video editors manually identify and annotate these landmarks with visual overlays, which is not efficient. To address this challenge, this chapter introduces a computer vision pipeline designed to automate the recognition and tracking of landmarks showcased during live broadcasts (RG 2.1). Once a landmark is identified, the system communicates recognition and tracking events to the live stream's visualization engine, enabling real-time updates. Moreover, with the increasing adoption of drones in the broadcasting sector, this automated approach offers significant scalability advantages, facilitating efficient monitoring of multiple video streams.

Given the unique regional context of each race (RG 2.2 and 3.1), manually creating a labeled dataset for every landmark is impractical. Therefore, the proposed research solution employs foundational models, which enable few-shot recognition using a limited set of reference images (RG 1.3). Additionally, the pipeline is designed to meet the speed requirements of live broadcasting, ensuring real-time performance (RG 1.2) without disrupting the seamless viewing experience.

This chapter is a slightly adapted version of the following publication:

Vanhaeverbeke, J., Decorte, R., Slembrouck, M., Van Hoecke, S., & Verstockt, S. (2024). **Point of Interest Recognition and Tracking in Aerial Video during Live Cycling Broadcasts**. APPLIED SCIENCES-BASEL, 14(20).

Abstract: Road cycling races, such as the Tour de France, captivate millions of viewers globally, combining competitive sportsmanship with the promotion of regional landmarks. Traditionally, points of interest (POIs) are highlighted during broadcasts using manually created static overlays, a process that is both outdated and labor-intensive. This chapter presents a novel, fully automated methodology for detecting and tracking POIs in live helicopter video streams, aiming to streamline the visualization workflow and enhance viewer engagement. Our approach integrates a saliency and Segment Anything-based technique to propose potential POI regions, which are then recognized using a keypoint matching method that requires only a few reference images. This system supports both automatic and semi-automatic operations, allowing video editors to intervene when necessary, thereby balancing automation with manual control. The proposed pipeline demonstrated high effectiveness, achieving over 75% precision and recall in POI detection, and offers two tracking solutions: a traditional MedianFlow tracker and an advanced SAM 2 tracker. While the former provides speed and simplicity, the latter delivers superior segmentation tracking, albeit with higher computational demands. Our findings suggest that this methodology significantly reduces manual workload and opens new possibilities for interactive visualizations, enhancing the live viewing experience of cycling races.

5.1 Introduction

Road cycling races have long been a captivating spectator sport, with major events like the Tour de France, Giro d'Italia, and Vuelta a España drawing in millions of viewers worldwide [1, 2]. These races showcase the competition among professional cycling teams, while also offering a unique platform to promote regional landmarks and attractions. This blend of sportsmanship and scenery has been a

tried and trusted formula for decades. Whenever a landmark, or so called point of interest (POI), is highlighted by the helicopter view, it is commonly accompanied by a static overlay presenting its name. However, not only has this static overlay become obsolete in an age where viewers seek more engaging experiences, it also involves manual work of video editors to present these overlays during the live broadcast. Therefore, there is a need to improve the visualization workflow of points of interest during cycling races.

Our work tackles these problems by introducing a fully automatic methodology to recognize and track POIs. Automatically recognizing POIs in the helicopter stream during the live broadcast reduces the manual workload, while tracking allows for more interactive visualizations, for example, by anchoring the POI information to the tracked location instead of displaying a static overlay.

Achieving this outcome involves addressing several challenges. First, to ensure smooth adoption, the proposed system needs to be flexible and extensible. Therefore, the methodology recognizes POIs by matching them with a few reference images. This approach allows for the easy addition of new POIs in different races without requiring a retraining of the model, making it practical for real-world usage. Additionally, video editors still need the freedom to override the automatic pipeline. Hence, the system offers a semi-automatic workflow with manual input that can be used in case the automatic pipeline fails or when an unknown POI appears. This combination of both automatic and semi-automatic operation results in a good balance of reducing the manual workload while still providing the control and confidence to handle unexpected situations that happen during live broadcasts. Furthermore, the tracking of POIs across the video stream should be accurate and smooth to benefit the viewing experience. Finally, the live broadcast must remain uninterrupted by the processing pipeline. Therefore, the methodology is tailored for live usage, ensuring that every component works in real time or in a non-blocking manner.

The contributions of this work can be summarized as follows:

- A saliency and Segment Anything-based methodology to propose potential POI region masks;
- A keypoint matching-based POI recognition only requiring a few reference images per POI;
- A comparison of traditional and deep learning approaches for the tracking of POIs;
- A complete POI recognition and tracking system to process live helicopter video streams.

The remainder of this chapter is organized as follows: First, the related literature regarding landmark recognition and object tracking is discussed in Section 5.2. Next, Section 5.3 elaborates our methodology for point of interest recognition and tracking during live cycling broadcasts. Then, the results of the proposed approach are presented and discussed in Section 5.4, providing both quantitative and qualitative evaluations. Before concluding in Section 5.6, Section 5.5 explores additional applications of the methodology beyond the live usage scenario, highlighting its broader potential and versatility.

5.2 Related Work

While no directly related research was found that addresses point of interest recognition and tracking during live sports broadcasts, there are several related applications that perform landmark recognition to reduce manual work or enhance user engagement. For example, software services exist that process video footage to generate metadata for media content, thereby improving searchability and facilitating the retrieval of appropriate clips for storytelling purposes [3]. Additionally, smartphone applications designed for tourists can recognize points of interest in a city and provide relevant background information [4, 5]. To provide a broader perspective on existing literature, an overview is separately given of the relevant methods proposed for the two main components of our work: landmark recognition and object tracking.

5.2.1 Landmark Recognition

Our research aims to detect points of interest, also known as landmarks, such as buildings and monuments. This problem has been extensively studied, focusing on landmark retrieval and recognition, where either similar images to a query image are retrieved, or the depicted landmark is labeled with its name.

Early approaches relied on handcrafted global image descriptors, as noted by Smeulders et al. [6]. However, these methods lack robustness against variations, such as illumination and perspective changes. Local feature descriptors, such as the scale-invariant feature transform (SIFT) [7], later became the prominent method due to their ability to better address these challenges. These features can be used for local feature matching [7, 8] or combined into a single representation for fast, large-scale retrieval [9, 10]. Other approaches use a combination of both methods, combining the advantages of fast matching and geometric verification [11, 12].

In recent years, convolutional neural networks (CNNs) have become the dominant approach in landmark recognition. Embeddings of CNNs trained with classification or similarity losses are used as performant global feature descriptors [13, 14], while CNNs are also employed to produce local features and match images based on them [15, 16]. Recognizing the strengths and weaknesses of both global and local features, Cao et al. have employed a combination of both to achieve state-of-the-art performance [17].

Although there is extensive related work on landmark recognition and retrieval, the main focus has been on recognizing the image as a whole, without detecting the exact location of the landmark. However, for certain use cases, knowing the precise location is important. One potential solution is to utilize the spatial information of local feature descriptors to extract an estimated position of the landmark, assuming that the most important features will be located on the landmark itself. Another option is to reuse the regional information that some of the related works incorporate. For example, Teichmann et al. [18] employ a two-stage approach where a landmark object detector first proposes regions of interest. These regions are then used to determine local features, which are aggregated to improve image representation. While these regions are currently not used to provide the location of the landmark, they could be employed for this purpose. Similarly, Kumar et al. [19] use a retrained BING objectness detector [20] to propose salient regions, which are used to enhance landmark recognition but not to find the exact location of the landmark in the image.

Our research also integrates saliency information in combination with Segment Anything [21] to propose potential landmark regions. However, this is performed not only to improve the recognition but also to provide an accurate location of the landmark within the image.

5.2.2 Object Tracking

A second component of our research involves fast and accurate object tracking. As the term suggests, object tracking aims to follow one or more objects across different frames in a video. This longstanding research area has seen the development of various approaches, leveraging either traditional or modern deep learning techniques.

Traditional techniques, despite their age, remain relevant due to their simplicity and speed. Two well-known categories rely on either features or correlation. Feature-based techniques use characteristics like color, textures, and edges to track objects from one frame to the next [22, 23], while correlation-based techniques learn a filter to discriminate the tracked object from the background [24, 25]. While these traditional methods are effective for basic tracking tasks, they face limitations in complex scenarios involving occlusions and illumination changes.

To address these challenges, many deep learning-based trackers have been developed, leveraging their ability to learn more efficient and robust feature embeddings. These have been popularized with the advent of Siamese CNN-based trackers, introduced by Bertinetto et al. with SiamFC [26]. This approach consists of two branches that extract features from the template and the search region, which are then used to find the most similar object to the template in the search region using similarity matching. Researchers have continued to improve upon this method to enhance tracking performance and robustness [27, 28]. However, Siamese CNN-based trackers still struggle with global context, similar objects, and occlusion [29].

The success of transformers in natural language processing has led to their application in computer vision tasks, including object tracking. Some approaches employ a hybrid CNN–transformer architecture since CNNs excel at representing local features, while transformers capture global features more effectively. By combining them, these hybrid approaches aim to achieve the best of both worlds [30, 31]. Other approaches opt for a completely transformer-based solution, relying solely on the attention mechanism to achieve state-of-the-art tracking performance [32, 33].

Next to video object tracking, related work has extended the focus to video object segmentation, where a segmentation of the object is tracked across a video instead of only the bounding box [34, 35]. This segmentation information enables more fine-grained analysis but comes at the cost of increased computational complexity. Several software solutions [36, 37] have been developed using video object segmentation trackers in combination with Segment Anything [21]. These tools allow user input to extract a mask, which is then tracked across the video to facilitate data labeling, keying, and inpainting. While these solutions share similarities with parts of our proposed solution, they do not focus on real-time tracking for live broadcasts.

5.2.3 Conclusions

Numerous studies have explored landmark recognition and object tracking as separate areas of research. Next to that, several user interfaces exist that employ segmentation techniques to facilitate object tracking in videos with minimal user input. However, most existing landmark recognition approaches do not locate the landmark within the image or through a video sequence. Therefore, to the best of our knowledge, no comprehensive solutions have been proposed to recognize and track points of interest in real time during cycling race broadcasts.

5.3 Methodology

The global architecture of our solution can be conceptually divided into three main stages: input, processing, and output, as depicted in Figure 5.1.

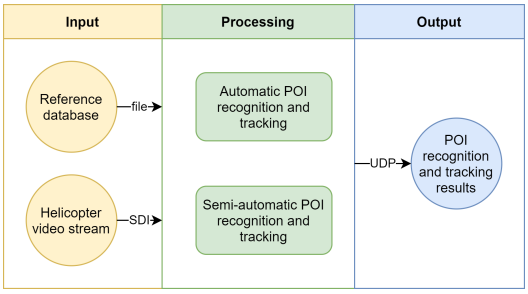


Figure 5.1: High-level overview of the complete point of interest recognition and tracking methodology.

The input consists of two sources. First, a live video stream is ingested through a serial digital interface (SDI) connection, which is a standard in professional video production environments. Second, a POI reference database is available, providing necessary information for the processing stage. The details of this POI database will be further explained in Section 5.3.1.

The processing component comprises two main parts: a fully automatic POI recognition and tracking pipeline and a semi-automatic user interface. The automatic pipeline employs computer vision techniques to detect, recognize, and track POIs in the live video stream. Additionally, the semi-automatic user interface allows human operators to intervene and make adjustments when needed, ensuring flexibility and reliability. These processing components will be discussed in depth in Sections 5.3.2 and 5.3.3, respectively.

In the output stage, the results generated by the processing component are transmitted to a visualization engine, such as Vizrt[1], using the User Datagram Protocol (UDP). When both the processing and visualization servers are located within the same local network, communication between these is fast and reliable. The use of UDP allows for a decoupled architecture, where the hardware for processing and visualization can be separate.

[1] https://www.vizrt.com

5.3.1 Point of Interest Reference Database

For each race, a database is constructed by the broadcaster or broadcast service provider, which serves as a central resource for the POI recognition system and includes several key pieces of information of all POIs. First, the name of each POI is listed, which is required to transmit the correct identifier to the visualization engine when a POI is successfully recognized. Second, the world coordinates of the POI are included, enabling the system to determine which POIs are near the helicopter at any given point in time. Lastly, a small selection of reference images (up to four) for each POI is added to the database. These images play an important role in the POI recognition process, as they are used to match the contents of the current helicopter frame with the reference images.

The reference images can be sourced from various origins. A quick and low-effort method is to use high-quality images from the internet. Alternatively, pictures can be specifically captured for this purpose, ensuring a more tailored and controlled set of reference images. However, another good approach is to extract reference images directly from the helicopter footage of previous race editions. Using images from past footage results in representative references, as they will closely match the expected visual characteristics and perspective of the POIs within the new broadcast.

The process of constructing the reference database can also be automated using race course data. Online services, such as OpenStreetMap[2], can be queried to find POIs along the race course, of which a few reference images can be automatically retrieved from the internet. Although this saves preparation time before the race, this does not guarantee that all required POIs are available and are accompanied by high-quality reference images. Therefore, it is recommended to check the quality and completeness of the automatically retrieved data to ensure a smooth operation of the recognition pipeline during the live broadcast.

5.3.2 Automatic Point of Interest Recognition and Tracking

The automatic POI recognition and tracking pipeline forms the core of our methodology. As illustrated in Figure 5.2, the pipeline consists of several distinct components, each fulfilling a specific role within the overall process. The following sections will provide a detailed discussion of the pipeline and its components.

[2] https://www.openstreetmap.org

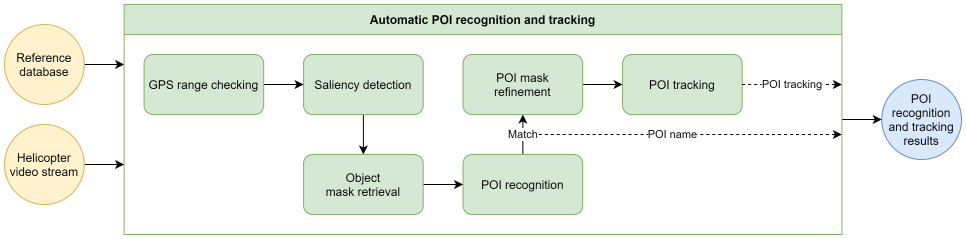


Figure 5.2: Overview of the automatic point of interest recognition and tracking methodology.

5.3.2.1 GPS Range Checking

The automatic approach to recognize POIs begins with a check of the helicopter's GPS position. This simple and efficient step can significantly reduce computation time and resource usage by pausing the pipeline when no POIs are in close proximity.

The haversine formula is employed to calculate the distance between the helicopter's location and the POIs in the reference database. When no POI is detected within a 500 m range, the POI recognition pipeline is stopped and reattempted after a 1 s wait. The 500 m threshold is an adjustable parameter that can be set based on specific requirements. A shorter range will reduce computing resource utilization but increase the likelihood of missing distant POIs, for example when the helicopter's camera is zoomed in. Conversely, a larger radius will minimize the chances of missing far-away POIs but will lead to increased resource consumption due to a more frequent execution of the detection pipeline.

5.3.2.2 Saliency Detection

When one or more POIs are within range of the helicopter, the next step in the pipeline, being saliency detection, is initiated. The objective of saliency detection is to identify the most visually interesting areas within an image or video. Since POIs are supposed to attract attention, they should be highlighted by the saliency detection algorithm. This approach enables the pipeline to focus on these specific areas when searching for POIs in later stages, eliminating the need to scan the entire image.

After empirically testing multiple saliency detectors, the UNISAL model developed by Droste et al. [38] has been found to provide the best and most consistent results. One of the key advantages of the UNISAL model is its optimization for both image and video data, making it well suited for our use case. The model's focus on video data ensures temporally consistent saliency results, which is important

for maintaining coherence across frames. As the model is trained on eye-tracking data, its output closely resembles the visual attention patterns of a human observer, generating a heatmap that highlights the most visually engaging regions, as illustrated in Figure 5.3. Consequently, the UNISAL model was selected for integration into our system.

Figure 5.3: Example visualization of the saliency heatmap predicted by the UNISAL model.

5.3.2.3 Object Mask Retrieval

The heatmap generated in the previous step serves as the input for this stage. Here, the focus shifts to identifying the most salient object within the image. To achieve this, the most and least salient points are first located within the heatmap, which are then utilized to prompt a Segment Anything Model (SAM).

SAM is a versatile model capable of generating segmentation masks for any object in an image based on various prompts, such as bounding boxes or points. Its promptable design enables zero-shot performance on object types not encountered during training. By providing the most salient point as a positive prompt and the least salient point as a negative prompt, SAM is guided to focus on the object of interest while avoiding uninteresting areas.

The original SAM proposed by Kirillov et al. has been improved in its second version, known as SAM 2, as detailed by Ravi et al. [39]. This updated version introduces multiple encoder sizes, which allows users to balance the model's size with its segmentation performance. Empirical evaluations have demonstrated that SAM 2, using the largest Hiera encoder, achieves superior results in segmenting points of interest.

Figure 5.4 illustrates an example of this process. The green and red dots represent the most and least salient points, respectively. The blue mask visualizes the output of the large SAM 2, showing a well-segmented object of interest based on the provided prompt points. This demonstrates the effectiveness of the approach in accurately identifying and isolating the most salient object within the image.

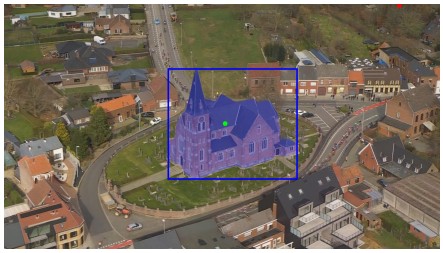


Figure 5.4: Example visualization of the predicted mask and derived bounding box by the Segment Anything Model (SAM) when prompted with the most and least salient point.

5.3.2.4 Point of Interest Recognition

After acquiring the most salient object in the previous step, the next task is to determine whether this object is a POI and, if so, recognize the POI in question. To achieve this, the bounding box derived from the segmentation mask is used to crop the image, focusing on the most salient object. This cropped image serves as the input for the POI recognition process.

As discussed in Section 5.3.1, a POI reference database is constructed, containing one or more reference images for each POI. These images are not used for training or fine-tuning a machine learning model, as the goal is to maintain a flexible and easily extensible methodology that can handle new POIs and races without requiring retraining. Instead, a keypoint matching approach is employed, enabling training-free recognition of POIs. This matching procedure consists of several steps.

First, the reference images of POIs within a 500 m radius of the helicopter's GPS position are retrieved. This saves computational resources and ensures that the matching process focuses on relevant nearby POIs. Both the cropped image of the most salient object and the retrieved reference images are then resized to a width of 300 pixels. This size strikes a balance between maintaining a high-enough resolution for accurate feature point calculation and achieving efficient processing speeds.

Next, the keypoints and descriptors of the resized images are determined using the SuperPoint algorithm [40]. SuperPoint is a fully convolutional neural network that extracts points and descriptors in a single forward pass, offering both speed and superior performance compared with traditional keypoint detectors and descriptors, such as SIFT. To optimize future iterations of the pipeline, the keypoint information of the reference images is automatically cached and loaded.

Finally, the SuperPoint descriptors of the salient object's crop and reference images are matched using SuperGlue [16], a graph neural network designed to find correspondences between two sets of sparse image features. Figure 5.5 il-

lustrates the keypoints and matches between the salient object crop and a reference image. Each point match is assigned a confidence score; these scores are summed to obtain a total match score per reference image. The reference image with the highest total confidence, exceeding a threshold of 10, is selected as the matched POI. If the highest total confidence falls below 10, no match is found, and the pipeline is exited, waiting for the next iteration, as shown in Figure 5.2.

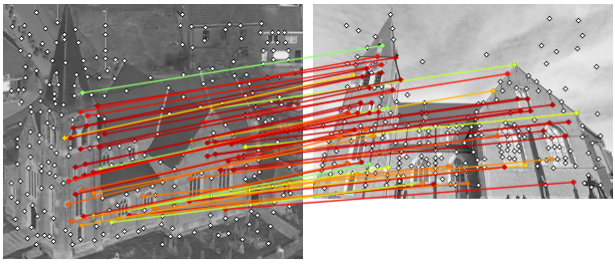


Figure 5.5: Example visualization of the keypoints and matches generated by the SuperGlue model. The left image is the helicopter video frame cropped to the proposed region by the saliency and SAM model, whereas the right image shows the reference. The color of the line indicates the confidence of the match with red being the strongest.

5.3.2.5 Point of Interest Mask Refinement

When a POI is recognized, it is important to ensure that the region to be tracked is as accurate as possible. As the initial POI mask and bounding box are derived from two points of the saliency detection, there are instances where the SAM output is suboptimal for tracking. To address this issue, the mask and bounding box of the POI are further refined using the available POI recognition information, thereby enhancing the subsequent tracking process.

The POI recognition step yields multiple keypoint matches between the salient object crop and the reference POI image. These keypoints are now used to re-prompt SAM, resulting in an improved segmentation mask. To optimize the refinement process, not all keypoint matches are used. First, only keypoints that fall within the non-refined SAM mask are selected, eliminating keypoints of the environment, such as trees or the road, which can negatively impact the refinement. Second, five keypoints that are maximally apart from each other are chosen. This approach ensures a limited set of points that still provides good coverage over the entire POI area, as using too many prompt points can reduce the segmentation accuracy of the SAM model.

Figure 5.6 illustrates an example of this refinement procedure. The left image shows the SAM segmentation based solely on the saliency information, while the

right image displays the refined mask incorporating the keypoint matching information. Initially, the mask incorrectly includes part of the road, which is resolved in the refinement step. By employing this refinement technique, the tracking process can be initiated with a more reliable and representative area of the POI.

(a) *(b)*

Figure 5.6: Example visualization of (a) the predicted SAM mask before refinement and (b) the predicted SAM mask after refinement using the POI recognition keypoint information.

5.3.2.6 Point of Interest Tracking

To create more visually appealing visualizations around POIs, it is important to track the location of the POI across consecutive frames. However, rerunning the entire POI recognition pipeline for every frame is not feasible due to the computational resources required, which would result in unacceptable delays for a live race broadcast. To overcome this challenge, an optical tracking algorithm is initialized after a POI is recognized, which follows the region of interest through subsequent frames. The tracking algorithm should ideally be faster than real-time to ensure smooth tracking while minimizing the delay in the broadcast chain.

Several traditional and machine learning-based trackers were compared empirically to assess their tracking quality and speed. Of those, two are chosen to integrate in the methodology. Traditional techniques, such as KCF and CSRT, are simple and offer fast tracking performance. Despite their simplicity, they have limitations in handling occlusion and large orientation changes. However, given that buildings are not frequently occluded and camera movement is often linear, they are a valid tracking possibility. Among the tested traditional algorithms, MedianFlow [41] demonstrates accurate, smooth, and high-speed tracking results on buildings, which is why it is included in this work. Machine learning-based techniques such as Stark and Cutie perform well when tracking objects like humans and cars but struggle with tracking buildings. In contrast, the large SAM 2,

which also has tracking capabilities, demonstrates very high segmentation tracking accuracy, even when the camera rotates around the POI. Therefore, next to MedianFlow, SAM 2 is provided as an option within the pipeline. Which tracker to use can be decided based on the tracking accuracy needed and the computational resources available.

To further improve the accuracy of the MedianFlow tracker, the largest inner bounding box of the POI mask is determined and tracked. By using the inner bounding box, the tracker can better focus on the POI itself, reducing the chances of being distracted by the background. Figure 5.7a shows the initial inner bounding box, whereas Figure 5.7b presents the tracked bounding box 20 s later.

(a) *(b)*

Figure 5.7: Example visualization of the tracked inner bounding box using the MedianFlow tracker with (a) and (b) being 20 s apart from each other.

5.3.3 Semi-Automatic Point of Interest Recognition and Tracking

While the fully automatic pipeline for recognizing and tracking POIs is effective in many situations, there are instances where it may not be sufficient. For example, the automatic pipeline may fail to detect a POI or recognize it too late. Additionally, video operators may want to track an object that is not included in the reference database. To address these limitations, the proposed software provides an interface that allows for a manual selection of objects of interest. This interface displays a live helicopter video stream, allowing operators to interact directly with the footage. By clicking or drawing on any part of the stream, operators can manually initiate the POI recognition and tracking pipeline, as illustrated in Figure 5.8.

The manual input from the user is used to replace the automatic detection of salient regions and points. Most of the subsequent steps in the pipeline are reused to maintain consistency, as depicted in Figure 5.9. In contrast to the automatic pipeline, the semi-automatic approach reorders the POI recognition and

Figure 5.8: Visualization of the manual input (yellow) for the recognition and tracking pipeline by drawing on the live broadcast.

tracking steps. After the user input is processed by the Segment Anything Model to generate a mask of the object, the tracking process is immediately initiated. This allows the broadcast operator to track objects that may not be present in the reference database. Concurrently, the POI recognition methodology is executed on a background thread. If the POI is successfully recognized, the information is passed to the visualization engine. In case of failure, the recognition process is reattempted every second, up to a maximum of 10 tries.

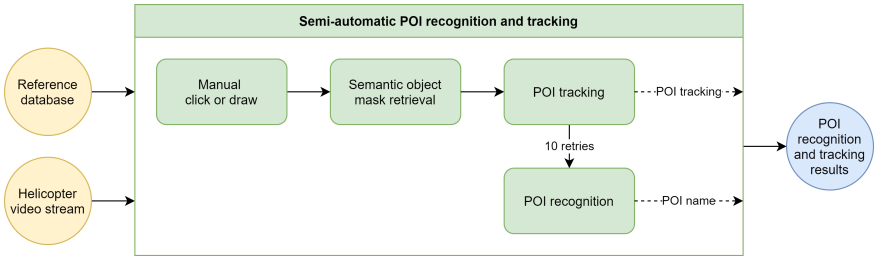


Figure 5.9: Overview of the semi-automatic point of interest recognition and tracking methodology.

Note that the segmentation mask refinement step, as detailed in Section 5.3.2.5, is omitted in the semi-automatic methodology. On the one hand, this decision was made to provide the user with full control over the input prompt of the SAM stage. On the other hand, it is uncertain if the selected object is in the reference database, which makes waiting for the recognition result unnecessary.

Since the semi-automatic approach reuses POI recognition and tracking components, this leads to performance levels that are comparable with, and potentially exceed, those of the fully automatic pipeline. By omitting the saliency detection component, the computational overhead and potential errors associated with it are reduced.

By incorporating a semi-automatic pipeline alongside the fully automatic one, the proposed software offers flexibility to various scenarios encountered during live broadcasts.

5.4 Results and Discussion

5.4.1 Dataset

To validate our automatic point of interest recognition and tracking pipeline, we have labeled one race, specifically the 2023 edition of *Omloop Het Nieuwsblad*. This Flemish race, situated around the Flemish Ardennes, includes several well-known segments such as the *Muur van Geraardsbergen*. Throughout the race, numerous other POIs, such as churches and windmills, are highlighted, making it a representative example for many other Belgian and European races.

For this race, a total of 10 POIs have been annotated with their names, coordinates, reference images, start times, and end times. The reference images are internet-sourced, aiming to simulate a realistic usage scenario. However, as discussed in Section 5.3.1, using reference images from previous race editions might yield better recognition results. The ground truth start and end times are based on the broadcasted times of the POI in the final television broadcast, i.e., when the director switches to and from the helicopter view. Relying on the director's broadcast decisions ensures an objective way to validate the performance of the recognition pipeline in detecting POIs in advance and ensuring that the tracking does not fail prematurely. Note that we only use the final broadcast to determine these ground truth times; the recognition and tracking itself is still performed on the raw helicopter stream. This approach was preferred to manual labeling based on the helicopter stream, which would require subjective rules for determining when a POI is in view. Additionally, the inner bounding boxes of the POIs have been annotated at the start, middle, and end of their ground truth broadcast times. These annotations allow the evaluation of the used tracking algorithm, ensuring that it remains accurate throughout the broadcast duration. The inner bounding box is used instead of the outer to allow for a more relevant evaluation of the MedianFlow tracker, which relies on the inner bounding box.

5.4.2 Metrics

To validate the performance of the automatic point of interest recognition and tracking, we define several custom metrics that provide an individual assessment

of each POI regarding the recognition correctness and timing. The metrics include the following:

- **Predicted**: Determines whether a POI is predicted during the ground truth timeframe;
- **Correct**: Assesses whether the POI prediction is correct;
- **Start offset**: Measures how early or late the initial recognition occurs;
- **End offset**: Evaluates how early or late the tracking stops;
- **Completeness**: Quantifies the extent to which the ground truth POI shot is covered by tracking.

Figure 5.10 illustrates these metrics by presenting a ground truth POI time window (in green) that the automatic methodology aims to predict. In this example, both the "predicted" and "correct" metrics are true because of a correct POI prediction (in blue) within the timeframe. The "start offset" is a positive number, indicating that the prediction was delayed. Ideally, the offset should be negative, indicating that recognition occurred before the director switched to the helicopter view. Conversely, the "end offset" is positive, meaning that the tracking continued beyond the director's switch, which is desirable in this context. The "Completeness" metric is approximately 75%, as the entire helicopter shot was not fully encompassed by POI recognition and tracking information.

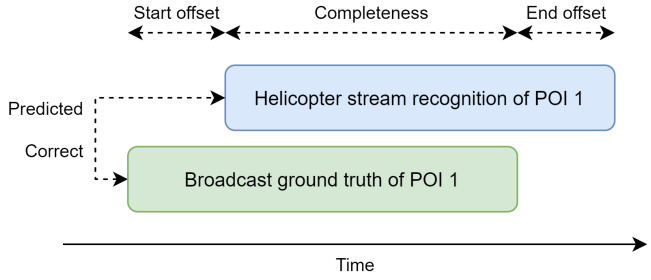


Figure 5.10: Graphical overview of the calculated metrics per POI.

Subsequently, these individual validation results are aggregated into metrics such as precision, recall, mean completeness, mean start offset, and mean end offset to provide a global overview of the system's performance.

The validation of the tracking is conducted using the intersection over union (IoU) metric, comparing the predictions against the annotated bounding boxes

at the start, middle, and end of the POI time window. By calculating the IoU at these specific points, the accuracy and consistency of the tracking can be assessed throughout the duration of the POI.

5.4.3 POI Recognition

The results for each point of interest, as detailed in Appendix 5.A, are summarized in the scores presented in Table 5.1. The recall rate shows that 80% of all broadcasted POIs are correctly detected, while the precision of about 79% indicates that few misdetections are made. To elaborate on this number, all broadcasted POI detections were correct, but a few false positives were generated at times where the helicopter view was not being broadcasted. On average, each POI is recognized 15 s before the director switches to the helicopter view, demonstrating an effective and timely recognition process. When using the MedianFlow tracker, POIs are followed for approximately 6 s after the director switches away from the helicopter view. This results in a completeness of 95%, ensuring that POIs are monitored nearly throughout the entire shot. The SAM 2 tracker improves this with a mean end offset of almost 6 s longer, showing the effectiveness of this tracker. The mean completeness stays the same since the missing 5% is due to a late detection of a POI.

Table 5.1: POI recognition and tracking results when using either the MedianFlow or SAM 2 tracker. The precision, recall, and mean start offset are purely based on the recognition phase, so these results cannot differ between trackers.

Metric	Using MedianFlow Tracking	Using SAM 2 Tracking
Precision	78.6%	
Recall	80%	
Mean start offset	−15.3 s	
Mean end offset	5.8 s	11.1 s
Mean completeness	95.3%	95.3%

Examining the failure cases more closely, we observe that the *Sint-Sebastiaankerk* in *Michelbeke* was not recognized because the saliency model did not allocate sufficient attention to it, thereby impeding the subsequent stages of the recognition pipeline. In another instance, the first appearance of the chapel on top of the *Muur van Geraardsbergen* fails at an even earlier stage of the pipeline. This

failure occurred because the POI was filmed from a considerable distance, well beyond the 500 m range. As discussed in Section 5.3.2.1, increasing the range would resolve this issue, but it would also result in a less resource-efficient system.

5.4.4 POI Tracking

The tracking performance of the system is validated at three moments during the broadcasted time window: the start, middle, and end. Table 5.2 presents the mean IoU scores across all points of interest at these moments using both trackers. Additionally, the impact of the object refinement step is presented.

Table 5.2: POI tracking results when using either the MedianFlow or SAM 2 tracker. For MedianFlow, the IoU between the ground truth and tracked inner bounding box is calculated. For SAM 2, the inner bounding box is first derived based on the tracked mask in order to provide IoU results that are comparable with the other tracker. Next, IoU scores for the tracking based on both the initial and refined mask are given to evaluate the effectiveness of this processing step.

	Mean IoU Using MedianFlow Tracking			Mean IoU Using SAM 2 Tracking		
SAM Mask	**Start**	**Middle**	**End**	**Start**	**Middle**	**End**
Initial	48.5%	39.9%	39.4%	62.9%	61.5%	61.0%
Refined	55.4%	46.8%	41.8%	66.8%	65.7%	65.6%

When using the MedianFlow tracker, the results demonstrate that the refinement improves the tracking IoU on average, with increases of 8%, 17%, and 0.6% for the start, middle, and end, respectively. However, it is also noted that tracking performance tends to decline towards the end of the POI broadcast time. The refined bounding boxes maintain better tracking performance for a longer duration compared with the non-refined boxes, as indicated by the middle scores, but ultimately, both approaches see a drop to approximately 40% by the end of the tracking period. This is due to this traditional tracker not being able to handle rotational movements well.

This limitation is clearly solved by using SAM 2 tracking. The IoU scores are not only higher overall compared with MedianFlow, they are also more consistent from the start to the end, with a refined IoU of about 65% across the board. The performance gain by the refinement step is less steep, but still decent as it helps to reduce some segmentation mistakes. Note that these are IoU scores for the inner bounding box of the tracked mask to make them comparable with

the traditional tracker. This makes their results appear worse, while in practice, the tracked masks are often nearly perfect.

Appendix 5.B provides insights in the individual tracking results for each POI, of which some of the results will be discussed in detail. For instance, the *Sint-Martinuskerk* achieves high tracking scores after refinement, with most of the IoU scores over 75% for both trackers. Figure 5.11a illustrates that the tracked region using the MedianFlow approach is indeed good. One outlier is the SAM 2 tracker that starts with an IoU of 61%. However, this is due to the calculation of the inner bounding box, whereas the underlying tracked mask is nearly perfect, as can be seen in Figure 5.11b.

(a) *(b)*

Figure 5.11: Visualizations of the tracking results of the Sint-Martinuskerk at the end of the POI time window using (a) the MedianFlow tracker (green box = ground truth, blue box = prediction) and (b) the SAM 2 tracker (blue mask = prediction). Both trackers perform very well on this POI.

However, lower IoU scores do not necessarily indicate unusable tracking. For example, the MedianFlow tracking of the *Kartuizerpriorij* starts with an IoU of 67% and drops to a score of 30%. Figure 5.12a reveals that this is due to not poor tracking but rather a change in camera angle. The tracking begins when the camera is facing the POI diagonally, resulting in a smaller inner bounding box. As the helicopter turns to face the front of the POI, a larger inner bounding box becomes possible, leading to a lower IoU score, but the tracking remains relevant. Interestingly, this is one of the few failure cases of the SAM 2 tracker. It starts out perfect, but about halfway, the tracker only starts to focus on the left half of the building, leading to low IoU scores. However, the remaining segmentation mask is still decent, but not complete, as shown in Figure 5.12b.

The MedianFlow tracker is not always accurate, as illustrated by the second appearance of the chapel on the *Muur van Geraardsbergen*, where the tracking IoU is 0% throughout the entire live broadcast time window. This is because the POI was detected 52 s in advance, and the helicopter circled around it during this time, causing the tracker to lose the POI entirely. A similar behavior is observed with

(a) *(b)*

Figure 5.12: Visualization of the tracking results of the Kartuizerpriorij at the middle of the broadcast time window using (a) the MedianFlow tracker (green box = ground truth, blue box = prediction) and (b) the SAM 2 tracker (blue mask = prediction). The IoU of the MedianFlow tracker seems low, but the resulting tracking is good and stable. SAM 2 loses track of the right side of the building, which also leads to lower IoU scores.

the *Vinkemolen*, as shown in Figure 5.13a, where the helicopter circled around the POI for 12 s before the director switched to the helicopter view. This highlights a limitation of the current tracking solution. The MedianFlow tracker performs well with linear or limited rotational movement but struggles when the helicopter flies around the POI, often losing track once the original surface disappears. In contrast, this is where the SAM 2 tracker really excels. Figure 5.13b also shows the *Vinkemolen* at the end of the POI timeframe. Despite the large orientation and zoom level difference, SAM 2 is still able to track the mill perfectly.

(a) *(b)*

Figure 5.13: Visualization of the tracking results of the Vinkemolen at the middle of the broadcast time window using (a) the MedianFlow tracker (green box = ground truth, blue box = prediction) and (b) the SAM 2 tracker (blue mask = prediction). MedianFlow lost track of the windmill because the helicopter was rotating around it. In contrast, SAM 2 handles this rotation perfectly.

Although SAM 2 produces more accurate and robust tracking results than MedianFlow, the output of this traditional tracker is sufficient and stable for most

POIs. When used live, the director could give instructions to the helicopter camera crew on which types of movement to avoid. Moreover, the semi-automatic approach allows for manual input to reset the tracking to the correct POI location, which can be performed shortly before the director switches to the helicopter view. However, if enough computational power is available, the SAM 2 tracker can be used for its superior tracking performance.

5.4.5 Speed Benchmark

Next to the accuracy of the POI recognition and tracking, the speed of the pipeline is equally important, especially since it is intended for use during live broadcasts. To assess the pipeline's efficiency, the speed is benchmarked on a virtual machine with one Tesla V100 GPU, four 2.7 GHz vCPU cores, and 32 GB RAM. Live tests have shown that faster performance is achievable on dedicated hardware over this virtual machine. The average speed of the main actions in the pipeline was measured over a period of 1 h and 30 min of helicopter video footage, resulting in the timings presented in Table 5.3.

Table 5.3: Speed benchmark results when using either the MedianFlow or SAM 2 tracker. The first 5 steps are equal for both since they are independent of the used tracker.

Functionality	Using MedianFlow Tracking	Using SAM 2 Tracking
Validation phase		
GPS range checking	0.11 ms $\pm$ 0.02 $\approx$ 9090 FPS	
Recognition phase		
Saliency detection	91 ms $\pm$ 60 $\approx$ 11 FPS	
Object mask retrieval	61 ms $\pm$ 5 $\approx$ 16 FPS	
POI recognition	122 ms $\pm$ 41 $\approx$ 8 FPS	
Subtotal	*274 ms $\pm$ 106 $\approx$ 4 FPS*	
Tracking prep phase		
POI mask refinement	136 ms $\pm$ 87 $\approx$ 7 FPS	
Tracker initialization	441 ms $\pm$ 418 $\approx$ 2 FPS	311 ms $\pm$ 33 $\approx$ 3 FPS
Subtotal	*577 ms $\pm$ 505 $\approx$ 2 FPS*	*447 ms $\pm$ 120 $\approx$ 2 FPS*
Tracking phase		
Tracking step	13 ms $\pm$ 1 $\approx$ 77 FPS	364 ms $\pm$ 36 $\approx$ 3 FPS

As anticipated, verifying the distance between the helicopter and the points of interest to determine if any POIs are nearby is a fast operation that can signifi-

cantly reduce unnecessary computations. In contrast, the POI recognition pipeline is considerably slower, requiring an average of 274 ms per check. However, since our pipeline only performs this recognition phase once per second, the achievable 4 FPS is more than sufficient. If a POI is identified, an additional 136 ms is needed to refine the POI's region. Initializing the tracker also comes with a computational cost, which is different based on the chosen tracker. Here, it is shown that the traditional MedianFlow tracker has a higher initialization cost over the more advanced SAM 2 tracker.

Consequently, after a successful POI recognition and initialization of the tracker, the processing will, on average, lag behind the live broadcast with 851 or 721 ms for the MedianFlow or SAM 2 modus, respectively. However, with the MedianFlow tracker only taking about 13 ms per frame, i.e., processing 77 frames per second, this buffer is quickly eliminated. However, our benchmark shows that the SAM 2 tracking is not fast enough to run real-time. Still, the authors of this model report a speed of 30.2 frames per second (FPS), indicating that it is possible to run this model real-time on our video stream of 25 FPS with more powerful, dedicated hardware.

The system is designed to maintain consistent speed, even if it scales to more races and POIs. On the one hand, this is because the GPS location check is efficient, regardless of the number of POIs that need to be verified. This initial step effectively filters out unnecessary computations when no POIs are within range. On the other hand, the majority of the methodology is independent of the number of nearby POIs. The recognition step is the only component directly influenced by this. However, this rarely poses a challenge as most scenarios involve only one nearby POI, with occasional instances of up to three. The system handles these situations effectively through efficient keypoint caching.

5.4.6 Visualization Examples

As one of the main goals of this research is to improve the viewing experience of POI overlays, we also demonstrate that the recognition and tracking data can effectively be used to generate more dynamic visualizations. Therefore, an integration is made with the Vizrt visualization engine in order to test our proposed solution end to end. Figure 5.14 shows the result of this integration with two visualized examples where the name is anchored to the POI itself.

Figure 5.14: Example visualizations generated by Vizrt based on the POI recognition and tracking data, demonstrating more dynamic POI overlays.

5.5 Additional Use Cases

Throughout the chapter, a computer vision system has been introduced with the primary aim of reducing the manual workload for video editors during live broadcasts, while simultaneously making the point of interest visualizations more dynamic for viewers. Although this is the main application of the system, it also has potential for other uses, such as improving user engagement, automating reporting, and generating metadata.

Next to dynamic live visualizations, the system can also be used to generate clips of POIs for sharing through various other mediums, such as social media. This could be particularly interesting for cities looking to promote tourism to a broader audience. These clips can also incorporate tracking information for visualization purposes and can be augmented with text overlays or voice-overs using automatically generated descriptions by a large language model. Additionally, all POI clips could be compiled into a comprehensive touristic summary video, including footage that did not make the final broadcast.

Moreover, the race organization and the cities through which the race passes agree on the specific POIs to be showcased and the duration. However, currently, there is a lack of proof on whether the actual broadcasting time aligns with the agreed-upon duration. Therefore, this system could automatically generate reports detailing the POIs featured in the broadcast and their corresponding airtime, serving as evidence for the cities.

Finally, the POI detection and tracking information can be stored as metadata alongside the broadcast. This facilitates the querying and retrieval of images and video clips of these POIs in the future, which can be useful for other programs or series in need of video footage of certain POIs.

5.6 Conclusions and Future Work

Broadcasting a cycling race is a complex process that requires the coordinated efforts of numerous people, including cameramen, directors, and video editors. Their collective aim is to deliver a seamless live viewing experience, which requires minimal delay in processing and presenting the footage. This constraint, however, limits the complexity of visualizations, such as overlays that highlight points of interest during the race. To address these challenges and improve visualizations while reducing manual workload, a computer vision pipeline has been proposed to automatically recognize and track POIs in the helicopter stream of live race broadcasts.

The POI recognition process comprises five stages, integrating a saliency detection model with SAM to propose potential objects of interest. These are then verified using a keypoint matching approach to determine which POI they represent. This methodology has demonstrated high effectiveness, achieving a precision and recall of over 75%. Once a POI is recognized, a tracker is started to follow it through the video stream. Two tracking solutions are discussed and provided within the solution to choose from. A traditional tracker, MedianFlow, is provided due to its speed and simplicity, maintaining a mean tracking completeness of approximately 95%. Although the intersection over union scores drop from the beginning to the end of the broadcasted POI time window, the tracking remains stable and usable for live visualization. A discussed limitation of this approach is the potential loss of the POI when the camera rotates around it. To improve this, the more advanced SAM 2 tracking is also available, which achieves nearly perfect segmentation tracking results for almost all POIs, even when the camera is circling around the POI. However, this tracker requires more computing power in order to work real-time. Both tracking approaches benefit from our proposed POI region refinement step that uses the POI recognition information in order to refine the mask of the object.

Future research opportunities are available to refine this work. The current saliency-based method for proposing interesting regions could be replaced with a custom model specifically designed to detect objects like churches, windmills, and monuments, thereby increasing detection rates. For POI recognition, an embedding-based approach could enhance system scalability. Additionally, developing a custom tracking model tailored to POIs could improve robustness against rotational challenges at faster inference speeds.

In summary, this research highlights the potential of computer vision technologies to reduce human workload in live cycling race broadcasts. Additionally, it also paves the way for more engaging visualization options, enhancing the overall viewing experience of cycling races.

5.A Appendix: Individual POI Recognition Results

POI	Broadcasted	Predicted	Correct	Start Offset	End Offset		Completeness
					Using MedianFlow Tracking	Using SAM 2 Tracking	
Sint-Martinuskerk	True	True	True	−16 s	1.4 s	1.6 s	100%
Vinkemolen	True	True	True	−12 s	3.3 s	5.1 s	100%
Kartuizerpriorij	True	True	True	−6 s	8.7 s	8.7 s	100%
Sint-Ursmaruskerk	True	True	True	−17 s	1.8 s	2.3 s	100%
Sint-Bartholomeuskerk	True	True	True	−4 s	17 s	22 s	100%
Kapel O-L-V op de Oudenberg	True	True	True	−52 s	7.2 s	7.1 s	100%
Sint-Jan-Baptistkerk	True	True	True	−20 s	1 s	2.3 s	100%
Sint-Pieterskerk	True	True	True	5 s	61 s	40 s	62%
Sint-Sebastiaankerk	True	False	False	/	/	/	/
Kapel O-L-V op de Oudenberg	True	False	False	/	/	/	/
Sint-Martinuskerk	False	True	False	/	/	/	/
Vinkemolen	False	True	True	/	/	/	/
Sint-Bartholomeuskerk	False	True	True	/	/	/	/
Sint-Bartholomeuskerk	False	True	False	/	/	/	/
Kapel O-L-V op de Oudenberg	False	True	True	/	/	/	/
Sint-Jan-Baptistkerk	False	True	False	/	/	/	/

/ indicates that either the POI was not recognized or has no matching ground truth.

5.B Appendix: Individual POI Tracking Results

POI	SAM Mask	Mean IoU Using MedianFlow Tracking			Mean IoU Using SAM 2 Tracking		
		Start	Middle	End	Start	Middle	End
Sint-Martinuskerk	Initial	43.3%	44.6%	47.6%	52.2%	94.9%	91.2%
	Refined	80.0%	78.5%	74.1%	61.2%	95.3%	91.2%
Vinkemolen	Initial	26.5%	15.9%	14.0%	60.0%	97.7%	91.3%
	Refined	25.8%	19.4%	3.5%	57.9%	97.4%	91.3%
Sint-Sebastiaankerk	Initial	/	/	/	/	/	/
	Refined	/	/	/	/	/	/
Kartuizerpriorij	Initial	68.4%	30.1%	60.1%	42.2%	28.3%	0%
	Refined	67.3%	29.8%	60.2%	41.5%	28.2%	0%
Sint-Ursmaruskerk	Initial	72.6%	70.1%	57.1%	74.4%	71.0%	92.7%
	Refined	74.0%	77.0%	63.9%	74.2%	76.9%	95.5%
Kapel O-L-V op de Oudenberg	Initial	/	/	/	/	/	/
	Refined	/	/	/	/	/	/
Sint-Bartholomeuskerk	Initial	57.1%	55.8%	54.5%	59.6%	58.7%	56.5%
	Refined	66.3%	64.6%	59.8%	81.9%	78.7%	91.4%
Kapel O-L-V op de Oudenberg	Initial	0%	0%	0%	62.8%	56.1%	60.0%
	Refined	0%	0%	0%	62.8%	56.1%	60.2%
Sint-Jan-Baptistkerk	Initial	71.4%	57.4%	58.3%	89.0%	85.0%	96.1%
	Refined	74.1%	60.9%	63.6%	88.0%	93.0%	95.6%
Sint-Pieterskerk	Initial	/	45.5%	23.2%	/	0%	0%
	Refined	/	44.3%	9.2%	/	0%	0%

/ indicates that the POI was not recognized.

References

[1] ASO. *EBU members rack up highest numbers of hours viewed for Tour de France since 2015*, July 2022. Available from: https://www.letour.fr/en/news/2022/ebu-members-rack-up-highest-numbers-of-hours-viewed-for-tour-de-france-since-2015/1308860.

[2] UCI. *Spectacular TV and digital audiences for 2023 UCI Cycling World Championships in Glasgow and across Scotland*, October 2023. Available from: https://www.uci.org/pressrelease/spectacular-tv-and-digital-audiences-for-2023-uci-cycling-world/3KSV2mdsYiRRoPBupy1tDT.

[3] Sports Video Group. *Newsbridge Conquers the Limitations of AI Landmark Detection*, March 2023. Available from: https://www.sportsvideo.org/2023/03/21/newsbridge-conquers-the-limitations-of-ai-landmark-detection/.

[4] A. S. Timmaraju and A. Chatterjee. *Monulens: Real-time mobile-based Landmark Recognition*.

[5] M. N. Razali, E. O. N. Tony, A. A. A. Ibrahim, R. Hanapi, and Z. Iswandono. *Landmark Recognition Model for Smart Tourism using Lightweight Deep Learning and Linear Discriminant Analysis*. International Journal of Advanced Computer Science and Applications, 14, 2023. Available from: http://thesai.org/Publications/ViewPaper?Volume=14&Issue=2&Code=IJACSA&SerialNo=25, doi:10.14569/IJACSA.2023.0140225.

[6] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. *Content-based image retrieval at the end of the early years*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12):1349–1380, 2000. doi:10.1109/34.895972.

[7] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, 60(2):91–110, 2004. doi:10.1023/B:VISI.0000029664.99615.94.

[8] K. Mikolajczyk and C. Schmid. *An Affine Invariant Interest Point Detector*. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, Computer Vision — ECCV 2002, page 128–142, Berlin, Heidelberg, 2002. Springer. doi:10.1007/3-540-47969-4_9.

[9] J. Sivic and A. Zisserman. *Video Google: a text retrieval approach to object matching in videos*. In Proceedings Ninth IEEE International Conference on Computer Vision, page 1470–1477 vol.2, 2003. Available from: https://ieeexplore.ieee.org/document/1238663, doi:10.1109/ICCV.2003.1238663.

[10] H. Jégou, M. Douze, C. Schmid, and P. Pérez. *Aggregating local descriptors into a compact image representation*. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, page 3304–3311, 2010. Available from: https://ieeexplore.ieee.org/document/5540039, doi:10.1109/CVPR.2010.5540039.

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. *Object retrieval with large vocabularies and fast spatial matching*. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, page 1–8, 2007. Available from: ht tps://ieeexplore.ieee.org/document/4270197, doi:10.1109/CVPR.2007.383172.

[12] H. Jegou, M. Douze, and C. Schmid. *Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search*. In D. Forsyth, P. Torr, and A. Zisserman, editors, Computer Vision – ECCV 2008, page 304–317, Berlin, Heidelberg, 2008. Springer. doi:10.1007/978-3-540-88682-2_24.

[13] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. *End-to-End Learning of Deep Visual Representations for Image Retrieval*. International Journal of Computer Vision, 124(2):237–254, 2017. doi:10.1007/s11263-017-1016-8.

[14] F. Radenović, G. Tolias, and O. Chum. *Fine-Tuning CNN Image Retrieval with No Human Annotation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(7):1655–1668, 2019. doi:10.1109/TPAMI.2018.2846566.

[15] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. *Large-Scale Image Retrieval with Attentive Deep Local Features*. In 2017 IEEE International Conference on Computer Vision (ICCV), page 3476–3485, 2017. Available from: https: //ieeexplore.ieee.org/document/8237636, doi:10.1109/ICCV.2017.374.

[16] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. *SuperGlue: Learning Feature Matching With Graph Neural Networks*. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), page 4937–4946, 2020. Available from: https://ieeexplore.ieee.org/document/9157489, doi:10.1109/CVPR42600.2020.00499.

[17] B. Cao, A. Araujo, and J. Sim. *Unifying Deep Local and Global Features for Image Search*. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, Computer Vision – ECCV 2020, page 726–743, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-58565-5_43.

[18] M. Teichmann, A. Araujo, M. Zhu, and J. Sim. *Detect-To-Retrieve: Efficient Regional Aggregation for Image Search*. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), page 5104–5113. IEEE Computer

Society, 2019. Available from: https://www.computer.org/csdl/proceedings-article/cvpr/2019/329300f104/1gyr0dsE4gM, doi:10.1109/CVPR.2019.00525.

[19] A. Kumar, S. Bhowmick, N. Jayanthi, and S. Indu. *Improving Landmark Recognition Using Saliency Detection and Feature Classification*, page 157–175. Springer International Publishing, 2021. Available from: https://doi.org/10.1007/978-3-030-57907-4_9, doi:10.1007/978-3-030-57907-4_9.

[20] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. *BING: Binarized Normed Gradients for Objectness Estimation at 300fps*. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, page 3286–3293, 2014. Available from: https://ieeexplore.ieee.org/document/6909816, doi:10.1109/CVPR.2014.414.

[21] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. *Segment Anything*. 2023. Available from: http://arxiv.org/abs/2304.02643, doi:10.48550/arXiv.2304.02643.

[22] S. Baker and I. Matthews. *Lucas-Kanade 20 Years On: A Unifying Framework*. International Journal of Computer Vision, 56(3):221–255, 2004. doi:10.1023/B:VISI.0000011205.11775.fd.

[23] K. Fukunaga and L. Hostetler. *The estimation of the gradient of a density function, with applications in pattern recognition*. IEEE Transactions on Information Theory, 21:32–40, 1975. doi:10.1109/TIT.1975.1055330.

[24] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. *High-Speed Tracking with Kernelized Correlation Filters*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(3):583–596, 2015. arXiv:1404.7584 [cs]. doi:10.1109/TPAMI.2014.2345390.

[25] A. Lukežič, T. Vojíř, L. Čehovin Zajc, J. Matas, and M. Kristan. *Discriminative Correlation Filter Tracker with Channel and Spatial Reliability*. International Journal of Computer Vision, 126(7):671–688, 2018. doi:10.1007/s11263-017-1061-3.

[26] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. *Fully-Convolutional Siamese Networks for Object Tracking*. In Computer Vision – ECCV 2016 Workshops, page 850–865, Cham, 2016. Springer International Publishing. doi:10.1007/978-3-319-48881-3_56.

[27] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. *High Performance Visual Tracking with Siamese Region Proposal Network*. In 2018 IEEE/CVF Conference on Computer

Vision and Pattern Recognition, page 8971–8980, Salt Lake City, UT, 2018. IEEE. Available from: https://ieeexplore.ieee.org/document/8579033/, doi:10.1109/CVPR.2018.00935.

[28] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu. *SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines*. In AAAI Conference on Artificial Intelligence, volume 34, page 12549–12556, 2020. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/6944, doi:10.1609/aaai.v34i07.6944.

[29] M. Ondrašovič and P. Tarábek. *Siamese Visual Object Tracking: A Survey*. IEEE Access, 9:110149–110172, 2021. doi:10.1109/ACCESS.2021.3101988.

[30] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu. *Transformer Tracking*. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), page 8122–8131, 2021. Available from: https://ieeexplore.ieee.org/document/9578609, doi:10.1109/CVPR46437.2021.00803.

[31] J. Wang, Y. Song, C. Song, H. Tian, S. Zhang, and J. Sun. *CVTrack: Combined Convolutional Neural Network and Vision Transformer Fusion Model for Visual Tracking*. Sensors, 24(11):274, 2024. doi:10.3390/s24010274.

[32] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu. *Learning Spatio-Temporal Transformer for Visual Tracking*. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), page 10428–10437, Montreal, QC, Canada, 2021. IEEE. Available from: https://ieeexplore.ieee.org/document/9710846/, doi:10.1109/ICCV48922.2021.01028.

[33] Y. Cui, C. Jiang, G. Wu, and L. Wang. *MixFormer: End-to-End Tracking With Iterative Mixed Attention*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(6):4129–4146, 2024. doi:10.1109/TPAMI.2024.3349519.

[34] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. *Fast Online Object Tracking and Segmentation: A Unifying Approach*. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), page 1328–1338, 2019. Available from: https://ieeexplore.ieee.org/document/8953931, doi:10.1109/CVPR.2019.00142.

[35] H. K. Cheng, S. W. Oh, B. Price, J.-Y. Lee, and A. Schwing. *Putting the Object Back into Video Object Segmentation*. 2024. arXiv:2310.12982 [cs]. Available from: http://arxiv.org/abs/2310.12982, doi:10.48550/arXiv.2310.12982.

[36] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng. *Track Anything: Segment Anything Meets Videos*. 2023. arXiv:2304.11968 [cs]. Available from: http://arxiv.org/abs/2304.11968, doi:10.48550/arXiv.2304.11968.

[37] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang. *Segment and Track Anything*. 2023. arXiv:2305.06558 [cs]. Available from: http://arxiv.org/abs/2305.06558, doi:10.48550/arXiv.2305.06558.

[38] R. Droste, J. Jiao, and J. A. Noble. *Unified Image and Video Saliency Modeling*. In 16th European Conference on Computer Vision (ECCV), volume 12350, 2020. Available from: https://link.springer.com/10.1007/978-3-030-58558-7_25, doi:10.1007/978-3-030-58558-7_25.

[39] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. *SAM 2: Segment Anything in Images and Videos*. 2024. Available from: http://arxiv.org/abs/2408.00714, doi:10.48550/arXiv.2408.00714.

[40] D. DeTone, T. Malisiewicz, and A. Rabinovich. *SuperPoint: Self-Supervised Interest Point Detection and Description*. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), page 337–33712, 2018. Available from: https://ieeexplore.ieee.org/document/8575521/?arnumber=8575521, doi:10.1109/CVPRW.2018.00060.

[41] Z. Kalal, K. Mikolajczyk, and J. Matas. *Tracking-Learning-Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(7):1409–1422, 2012. doi:10.1109/TPAMI.2011.239.

6
Conclusion

Historically, event detection has relied heavily on human monitoring, but this manual approach is becoming increasingly impractical due to the sheer volume of events. Advancements in artificial intelligence have significantly expanded the scope and accuracy of event detection systems, making them a viable alternative to reduce dependency on human oversight. However, the development of robust, real-world event detection systems come with challenges that are often overlooked in fundamental academic research.

To function effectively, event detection systems must handle inputs from diverse sources such as sensors, cameras, and other data streams. These sources introduce challenges that compromise data quality and consistency, such as noise, sensor drift, and differences across devices. Additionally, the continuous flow of streamed data requires timely and efficient processing to avoid lag or system failures, particularly in time-sensitive applications like live monitoring. Another common hurdle is data availability. Many scenarios involve limited or unlabeled datasets, calling for innovative approaches to maximize the use of such data. Beyond these, systems must also integrate additional context like spatial information and adapt to changing real-world conditions, all while ensuring explainability to build user trust and support informed decision-making.

This dissertation therefore focused on designing and evaluating effective event detection systems that are capable of operating reliably in diverse, real-world environments. To achieve this, four case studies were presented, each incorpo-

rating different data sources (e.g., video and sensor data), exploring varying spatial scales (ranging from machine-level to regional-level applications), and addressing practical challenges such as data quality, timing limitations, and context changes. The case studies explored the detection of a broad spectrum of events, including flame anomaly events in steel reheating furnaces (Chapter 2), human presence events in rooms (Chapter 3), high COVID-19 risk events in buildings (Chapter 4), and landmark recognition events across regions (Chapter 5). In this chapter, the methodologies designed and developed for each case study are reviewed in relation to the research focus outlined in Chapter 1, followed by an in-depth reflection of the research. Additionally, the chapter proposes a framework for future event detection studies and discusses recommendations for future research.

6.1 Review of the Research Focus

6.1.1 Challenge 1: Handling Data Variety, Velocity and Availability

Each of the presented case studies highlighted a variety of data issues. Consequently, a key focus of this dissertation was to explore how the challenges posed by real-world data can be effectively addressed.

Chapter 2 highlighted challenges arising from limited availability of labeled data and high variability in the video footage. Therefore, we successfully designed a transfer learning methodology, which allowed the model to leverage an existing image encoder for better generalization with limited labeled frames (RG 1.3). Image augmentations and the conversion to grayscale were employed to improve model robustness against the significant variations in the thermal camera footage (RG 1.1). Furthermore, the furnace keypoint detection model can not only be used for determining burner zones, but can also be employed to signal when the camera's position is shifted due to vibrations, helping to manage positional variations. A lightweight CNN model architecture was designed, along with a strategy of selectively skipping frames, to maintain smooth and efficient processing (RG 1.2). This approach allowed for the system's deployment on standard CPU hardware, achieving an acceptable frame rate without the need for expensive GPU systems.

Chapter 3 delved into the processing of environmental sensor data, which presented its own set of challenges due to variability in sensor sensitivity, installation locations, and external factors like seasonal effects. To cope with these issues, we introduced a sliding window normalization technique that adapts the normalization to each sensor's local history (RG 1.1). The limited dataset led to

the adoption of a non-deep learning approach using CatBoost (RG 1.3). This model used window-based features, keeping feature dimensionality low to prevent overfitting (RG 1.3) while delivering a light, fast solution (RG 1.2).

In Chapter 4, the absence of ground truth for COVID-19 transmission risk required an alternative approach. By reviewing literature and translating expert knowledge into semantic concepts, we designed a knowledge-based system that bypassed the need for labeled data (RG 1.3) yet delivered results aligned with expert findings. Our system's architecture, incorporating components like a Kafka bus and Streaming MASSIF, efficiently processed large volumes of streamed sensor data, providing timely updates in a dynamic dashboard (RG 1.2).

Finally, Chapter 5 focused on a live broadcasting application, where unlabeled datasets and changing race regions required a generalizable solution. We demonstrated how foundational models, when used in tandem, can effectively propose, recognize, and track regions with minimal reference images (RG 1.3). The designed system met strict speed requirements by processing some operations in the background, syncing results back to the live feed when necessary (RG 1.2).

As a result, this dissertation designed, developed, and evaluated a variety of strategies to manage the inherent complexities of real-world data, offering insights and solutions adaptable to different domains.

6.1.2 Challenge 2: Detecting Spatial Event Context at Various Levels

The four case studies presented in this dissertation illustrate the potential of event detection across different spatial levels: machine level, room level, building level, and regional level. This underscores the widespread nature of events, which can be detected at any spatial scale, given the appropriate data sources and processing techniques.

More importantly, this dissertation demonstrated that event detection can extend beyond the temporal dimension. In addition to identifying when an event occurs, each case study linked the detected events to a specific spatial context. The machine level (RG 2.2) methodology designed in Chapter 2 not only identifies anomalies but also pinpoints their location within the furnace, whether at the front, middle, or back (RG 2.1). This spatial information provides important context for operators, enabling more effective responses. Chapter 3 focused on predicting presence at room level (RG 2.2). The approach presented in this chapter can be scaled to entire buildings, providing insights into which spaces are occupied and which are available (RG 2.1). This spatial information has a wide range of applications, from controlling building systems to improving evacuation planning during emergencies. Chapter 4 expanded the scope to entire buildings (RG 2.2),

including multiple offices. It introduced a methodology capable of identifying rooms with a high risk of COVID-19 transmission (RG 2.1), thereby enabling targeted, data-driven interventions to enhance safety measures and mitigate risks. At an even larger scale, Chapter 5 explored landmark recognition at the regional level (RG 2.2). By leveraging a reference database, this approach assigns global coordinates to identified landmarks. Additionally, it allows for the localization of landmarks within video frames, providing both accurate geographical and visual spatial context (RG 2.1).

These case studies demonstrated that, using the presented research methodologies, events can be identified across all spatial levels. Moreover, the findings highlighted the value of linking events to their spatial contexts, which not only increases understanding but also enhances the actionability of the data.

6.1.3 Challenge 3: Addressing Real-World Event Detection Cases

Each use case explored in this dissertation was undertaken in collaboration with industry partners and characterized by unique requirements, constraints, and challenges. The aim was to design custom solutions that could be deployed and integrated effectively in practical scenarios.

The first case study in Chapter 2 was designed and developed in partnership with a major steel manufacturing company. A hybrid approach was employed to improve control and explainability (RG 3.2). Deep learning was leveraged for tasks requiring strong scene understanding and feature extraction, while traditional computer vision methods processed the results further. A custom-built segmentation and keypoint detection model enabled joint prediction, thereby improving the efficiency of the solution. Image augmentations were implemented, improving the model's robustness and adaptability to changes in the captured footage (RG 3.1). While the deep learning model offers less inherent explainability, its outputs are visualized on a dashboard to facilitate human interpretation. In contrast, the anomaly detection component used decision trees, which are completely interpretable. The entire methodology is easy to integrate, as it operates on standard CPU hardware.

The study detailed in Chapter 3 explored presence detection for a home automation company, with the goal of enhancing smart home functionalities. A priority was to design a scalable and easily deployable methodology that effectively adapts to varying contexts across different rooms. The research successfully used a sliding window normalization technique, which enabled robust unsupervised generalization applicable to diverse rooms and building types (RG 3.1). The model requires no fine-tuning or configuration for deployment, it automatically adapts to new environments by leveraging historical data. While interpretability was not the main focus, the model was intentionally designed to be lightweight, facilitat-

ing easy implementation and scalability. Additionally, a custom-developed occupancy profiling layer was incorporated to analyze temporal events, allowing for the extraction of occupancy patterns. These insights can be integrated into smart building control systems.

Chapter 4 explored a COVID-19 healthcare application where interpretability was a key requirement. To address this, a knowledge-based system was designed that leveraged expert information to support decision-making (RG 3.2). The system integrated semantically annotated data from three real buildings, employing SPARQL to perform calculations. This approach ensured compatibility even when processing data from environmental sensors produced by different manufacturers (RG 3.1), highlighting the advantages of using semantic data. The software architecture was designed to meet the specific demands of the project. Key components included a Kafka bus for handling high data volumes, Streaming MASSIF for processing semantic sensor data streams, and a dynamic dashboard that visually presented results to end users.

The final project described in Chapter 5 was carried out in collaboration with a large broadcasting company. We designed a methodology for robust recognition using only a few reference images, designed to effectively handle the varying contexts encountered in different races (RG 3.1). Our approach significantly reduces the preparation required to recognize new landmarks in different events, improving its practical application. Furthermore, the introduction of automated tracking unlocked new visualization options that were unavailable before. The system was integrated with the company's existing visualization engine and successfully demonstrated its ability to process race footage in real time.

In summary, this dissertation successfully addressed diverse event detection challenges by designing custom, robust, and explainable methodologies. Furthermore, we validated them using real-world data and implemented them in practice.

6.2 Reflection on the Case Studies

6.2.1 Flame Anomaly Detection in Steel Furnaces

The proposed hybrid approach effectively monitors the status of flames across three burner zones and automatically detects anomalous events. This capability is highly valuable for steel manufacturing companies, such as the one we partnered with. Traditionally, operators had to monitor temperature readings, periodically visit the furnace to visually inspect the flames, or manually review the thermal camera stream to ensure proper functioning. The novel approach not only reduces the workload of these operators but also enables faster detection of anomalies and prevents episodes of unstable burner performance from going unnoticed. This

helps maintain the quality of steel production, minimizes resource wastage, and ultimately reduces production costs.

For this research, the partner company only equipped one reheating furnace with a thermal camera. Despite this, the collected data showed substantial variability over the data collection period. To address this, the model was optimized for robustness to varying environments through preprocessing and augmentations, including grayscaling, adjustments to brightness and contrast, and blurring. Removing color information from thermal images proved particularly beneficial, as the color is intended solely for human visualization and introduces unnecessary information that can lead to the model learning non-generalizable patterns. Runtime image augmentations further reduced the generalization gap. The model demonstrated strong robustness to unseen variations, indicating its potential to perform reliably with future furnace setups.

If additional furnaces were to be equipped with this monitoring system, the required time and budget would be rather limited. Setting up a new installation involves installing another thermal camera and hardware to process the video stream. Thermal cameras typically cost a few thousand euros, depending on the model, and are relatively easy to install since they are mounted externally to the furnace. The processing pipeline is lightweight and does not require high-end hardware for achieving an acceptable frame rate. Companies can choose to process the video streams from new installations on an existing server if it has unused resources, or on dedicated hardware for workload distribution. As more furnaces are equipped with thermal cameras, the scalability benefits of this automatic event detection system will become even more apparent, enabling monitoring without the need for constant human supervision.

There are several directions for future research that could further enhance the methodology's performance. In this study, video frames were processed independently to keep computational complexity low. This approach already delivered strong segmentation and keypoint detection performance. However, incorporating the temporal dimension of video data could yield even better results. The model architecture could be adapted to include recurrent layers, such as LSTMs or GRUs, to process embeddings across multiple consecutive frames. Additionally, the flame segmentation model could be extended to replace the traditional computer vision pipeline. Although this was not implemented in this study to give the industrial partner greater control over processing, allowing the model to perform flame segmentation per burner region could further improve accuracy.

Beyond methodological enhancements, the installation of additional thermal cameras per furnace could also bring improvements. For example, using multiple cameras could address inaccuracies caused by flame occlusion since one camera could focus on the left side of the furnace while another covers the right side. Furthermore, using multiple cameras could provide depth information, enabling

more precise identification of the flames' burner zone. Other sensors, such as those measuring temperature or gas composition, could be integrated into the system as well to complement the visual data from the thermal camera and enhance anomaly detection performance. Nonetheless, as the primary goal of this study was to make visual inspection more efficient, focusing on processing thermal camera footage has proven to be an effective solution.

6.2.2 Cross-Room CO_2-based Presence Detection

Indoor human presence information is essential for numerous smart home applications, one of which is optimizing the timing schedules of building systems. Manually defining such schedules often results in generic configurations that are not tailored to actual room usage, leading to suboptimal comfort and unnecessary energy consumption. The proposed approach addresses this issue by automatically detecting human presence and generating occupancy profiles. CO_2 sensors were selected as the primary source of data due to their strong correlation with human presence, while also being cost-effective, easy to install, and less intrusive with regards to privacy.

Although CO_2-based presence detection algorithms are not new, this research was conducted in collaboration with a smart home company, aiming to create a solution that is robust and practical for real-world deployment. Unlike many existing studies, our approach generalizes across unseen rooms without requiring fine-tuning and has been evaluated on real-world data rather than easy or simulated environments. This is important because room characteristics often evolve over time, affected by factors such as seasonal changes, room-specific characteristics (e.g., size, ventilation, and window configurations) and usage patterns (e.g., the number of occupants, and types of activities). To account for these dynamics, our method employs sliding window normalization, which adjusts CO_2 readings based on a local window of data specific to the room, ensuring the model adapts in real time to environmental conditions.

This sliding window normalization also makes the approach highly practical for deployment in new rooms. When installed in a new location, the system initially undergoes a cold start period during which the model learns the characteristics of the CO_2 data as the room is used normally, and adapts automatically without requiring manual calibration. This methodology has demonstrated good robustness and generalization across unseen environments, including different building types. Extending the dataset to include a broader range of rooms, such as offices, residential spaces, or school/university classrooms, would enhance the evaluation of the system and further validate its effectiveness.

Despite its benefits, CO_2-based presence detection has limitations due to external factors that influence CO_2 levels, such as ventilation systems and open win-

dows. For example, periods of intense ventilation may cause CO_2 levels to drop steeply, potentially leading to false negatives in presence detection. Incorporating ventilation data into the methodology could mitigate this issue. Ventilation data, such as airflow rate, is time series information that can be processed similar to CO_2 data. Moreover, CO_2-induced ventilation systems, which are often integrated within HVAC systems, present a significant opportunity. These systems already monitor CO_2 and ventilation rates, making them ideal candidates for implementing the presence detection model which could then provide insights to other smart systems. In addition, effective ventilation systems reduce the frequency of window opening, further improving the reliability of CO_2-based presence detection. That is because open windows dilute indoor CO_2 concentrations with fresh outdoor air and thus also influence CO_2 levels. Integrating information from window contact sensors into the presence detection model could also help address this issue by providing real-time data on window states.

Beyond CO_2, other parameters such as temperature, humidity, energy consumption, or radar information could also enhance the presence detection system. However, incorporating additional sensors increases the requirements and reduces the universal applicability of the model, which is why this study chose to focus solely on CO_2 data. If additional labeled data was available, it could also enable other smart building applications, such as open window detection without contact sensors. Much of the proposed data processing and modeling framework could be reused for such use cases, further demonstrating the versatility of the methodology.

6.2.3 COVID-19 Transmission Risk Estimation in Office Buildings

The most common method of assessing indoor air quality during the pandemic involved monitoring CO_2 levels. Public health authorities and governments frequently recommended maintaining CO_2 concentrations below 900 ppm to reduce transmission risk. If levels exceeded this threshold, occupants were advised to ventilate or evacuate the room. This strategy was both simple and effective, underscoring the importance of ventilation while providing actionable guidance. However, for a more comprehensive assessment of safety, alternatives like the physical models presented by Bazant et al. offered a deeper understanding by factoring in additional parameters beyond CO_2 concentrations. Moreover, the pandemic initiated a widespread deployment of CO_2 sensors in buildings to monitor real-time indoor air quality. While these sensors were effective at providing instantaneous measurements of CO_2 levels, many lacked centralized data storage capabilities, making retrospective analysis impossible. This limitation prevents

building managers from gaining insights into long-term air quality or implementing more informed preventative measures.

Our research showed to be valuable by addressing both challenges. Therefore, it integrated two state-of-the-art COVID-19 transmission models developed by experts in the field. These models were refined to work effectively in real-world conditions using real-time sensor data, incorporating dynamic adjustments based on room-specific parameters such as size and loudness. Additionally, we translated the models into a knowledge-based system using semantic technologies and integrated them into a scalable IoT architecture. This approach not only enabled real-time monitoring but also facilitated centralized data analysis for improved decision-making.

The scalability of our system was successfully demonstrated in three buildings, covering approximately 150 rooms, some of which having multiple sensors. This deployment showed the system's ability to handle large volumes of sensor data efficiently and reliably.

Setting up the system from scratch does require expertise since multiple components must be deployed and integrated. While the system is sensor-agnostic due to its use of semantic information, the installation of CO_2 sensors involves some considerations to ensure accurate readings. Sensors should be mounted at a height corresponding to the breathing zone of occupants. However, direct alignment with human exhalation should be avoided, as well as placement near doors, windows, or air vents to prevent skewed readings by airflow or fresh air. For large spaces or rooms with complex layouts, multiple sensors should be installed to ensure adequate coverage.

Once the system and sensors are in place, the methodology can be extended to monitor other airborne viruses, such as influenza or RSV, provided sufficient research exists on the specific parameters for these diseases. For example, estimating safe CO_2 concentration using the methodology of Bazant et al. requires knowledge of the infection quanta for the target disease. Similarly, the viral load survival by Spena et al. could be adapted if researchers model the specific enthalpy for other pathogens.

Beyond viral transmission monitoring, the system can easily be expanded to support other smart building applications. For instance, automated comfort scoring can be performed by designing new queries within streaming MASSIF and visualizing the results on the dynamic dashboard.

6.2.4 Point of Interest Recognition in Aerial Video

The value of this research, conducted in collaboration with a large broadcaster, is twofold. First, when a POI is recognized within the helicopter stream, an event is

sent to their visualization engine, where the accompanying overlay is automatically prepared. This reduces the workload of video editors as they no longer need to actively monitor the stream in manually set up the overlays. Additionally, this automated approach scales far better than human supervision, especially as the use of drones continues to grow in the broadcasting industry. Second, the automated POI tracking opens the door to more dynamic visualizations, which enhance the viewing experience. This functionality not only improves broadcasts but also attracts new race organizers and clients by offering cutting-edge features.

At the core of the automated POI recognition system lies a keypoint matching approach using reference images. Currently, providing these reference images is a manual process, but once created, they can be reused for future races. Typically, two or three reference images from different angles are sufficient for most POIs. To ensure optimal results, the images should be sharp, well-lit, and captured from angles that highlight distinguishing features. In the future, this process could be automated by leveraging recordings of previous broadcasts. Optical character recognition (OCR) could be used to detect and extract POI names displayed in overlays of past footage. Detected POI names could then be stored with the corresponding video frames. These frames could be further refined by applying our saliency and SAM methodology to crop the POI from the background. Comparing the result with nearby frames would help validate the image crops.

Although the methodology was validated on only one race, it successfully recognized POIs that are common in many other popular races in Belgium, France, Italy, and beyond. Since the keypoint detection model is based on foundational techniques, it is not tied to specific landmark types or styles but simply requires visually distinctive features to perform matching. For instance, landmarks such as churches, which tend to have complex and unique visual features, are easier to identify than more uniform objects like windmills. The keypoint matching model is optimized for factors such as changes in illumination, rotation, and perspective, ensuring robust performance. Additionally, the methodology demonstrated its reliability by processing interlaced video footage without issues.

The automated tracking capabilities offer directors greater flexibility to enhance the viewing experience, while still allowing them to decide on whether to use these features or not. For many POI shots, such as flyovers or pass-bys, the MedianFlow tracker performs very well and is ready for use. In contrast, during our benchmarks, the SAM 2.0 tracker did not achieve the speed necessary for live broadcasts. However, the SAM tracker is expected to be viable for real-time processing. According to its authors, the large SAM 2.0 model can achieve 39.7 FPS, and the more recently released SAM 2.1 model features a faster "base plus" variant with tracking performance comparable to the large 2.0 model. With improved hardware and a switch to the SAM 2.1 "base plus" model, tracking speeds could reach up to 64.1 FPS, as reported by the researchers.

While the methodology was developed with live cycling races in mind, it is also applicable in other contexts, such as news broadcasts or travel programs where landmarks are often featured. Beyond that, the system can enrich video metadata, enabling more efficient querying of clips featuring specific landmarks. Additionally, the individual components of this methodology are broadly usable across various domains. For example, the keypoint matching approach could be applied to recognize products in retail stores or vehicles on roads. The saliency model, meanwhile, could be used to identify salient regions in advertisements or security camera footage.

6.3 Framework for Future Use Cases

Like the proposed case studies, other real-world use cases are unique will typically require a tailored approach to address their specific requirements and challenges. Consequently, the likelihood of directly transferring one of the proposed solutions to a different task is low, unless the tasks are closely related, such as applying comfort scoring for indoor environments instead of estimating the risk of COVID-19 transmission. However, individual components of a solution, such as feature extraction methods or model architectures, can often be adapted, retrained, and reused in new applications.

Beyond reusing certain components, the knowledge and experience gained from handling the data sources, techniques, and challenges in the presented case studies can help inspire solutions for new use cases. To apply these insights effectively, it is important to first gain a clear understanding of the event detection problem being addressed. This includes identifying which events need to be detected, considering the data sources available, and determining whether detection should occur retrospectively or in real time. It is also helpful to assess whether the solution requires explainability and additional spatial context to be detected.

If data collection is still in progress and there is an opportunity to influence the process, it is beneficial to optimize the quality and consistency during that phase as much as possible. Collecting data from various environments will also help to evaluate the future methodology under realistic conditions. On the other hand, if the data has already been captured or there is no control over the collection process, the available dataset must be thoroughly explored. Aspects such as data variety, consistency, and quality should be assessed to identify factors that might affect later technical decisions.

Once the resources and requirements of the task are fully defined, it becomes possible to select the most suitable processing techniques within the given constraints. Many of the choices depend on the type of data being used. Therefore,

the following two sections outline a framework for potential techniques tailored to the data types discussed within this dissertation, being video and sensor data.

6.3.1 Video Data

Once the data type is known, video in this section, the choice of processing techniques is largely influenced by the availability of labeled data, timing constraints, and the need for explainability, as illustrated in Figure 6.1.

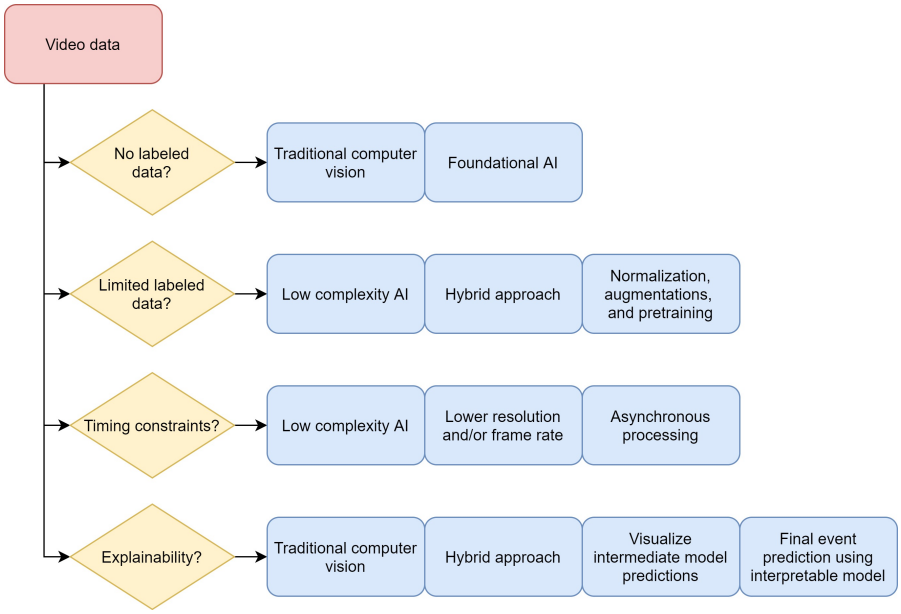


Figure 6.1: Schematic representation of the framework when working with a video data source.

In cases where a large dataset of high-quality labeled video data is available, the data itself does not pose a restriction and the possibilities for computer vision and machine learning are extensive. However, many real-world event detection tasks involve working with limited data. For example, the complete lack of labeled data can severely constrain the design of effective solutions. For simple tasks, traditional computer vision techniques can suffice for developing a processing algorithm. However, these approaches often fall short when dealing with more complex patterns, objects, or behaviors. Fortunately, over recent years, a variety of high-performance foundational models have become publicly available. Techniques such as CLIP for zero-shot classification or Microsoft's Florence-2 for

prompt-based object detection can be particularly valuable. Even, if these models cannot meet the accuracy or speed requirements of the application, they can still assist in the development of a labeled dataset, which can then be manually refined to improve quality.

When a labeled dataset is available but limited, different strategies become viable for designing and training custom machine learning models. In such cases, keeping the model size small is often necessary to avoid overfitting. Certain preprocessing strategies, such as converting thermal images to grayscale as demonstrated in Chapter 2, can help simplify and standardize the data, making the task easier to learn for the model. Image augmentations and starting with a pretrained model are also recommended, as they can make the model more robust to variations and ensure a more smooth training process. Additionally, a hybrid approach may be considered, combining traditional computer vision techniques, custom AI models, and foundational AI models as required to achieve the desired results.

If timing constraints are a factor, optimizing AI models for reduced complexity can significantly improve inference speeds. Two other straightforward techniques to reduce computational load are lowering the video resolution and frame rate. While these reductions might impact accuracy, many scenarios do not require high resolution or frame rates, making moderate adjustments feasible. Asynchronous processing is another useful strategy, allowing data streams to remain unblocked during processing and enabling the workload to be distributed, helping to maintain smooth operation.

For cases where explainability is a requirement, the first step should be to assess whether the problem can be addressed using traditional computer vision techniques. These methods offer greater transparency, as each step of the pipeline is controlled and interpretable. When artificial intelligence is necessary, it is valuable to ensure that models produce visual outputs, such as bounding boxes or segmentations, which provide users with insights into the model's decision-making process. Breaking a complex task into smaller steps with intermediate visual outputs can also enhance explainability, allowing subsequent processing through traditional or interpretable methods. If the model directly predicts event states, visualization techniques like Grad-CAM can help identify the regions of the input data that influenced the prediction, enhancing user trust and understanding.

6.3.2 Sensor Data

As with video data, several suggestions and strategies can be considered when addressing specific requirements and challenges associated with sensor data, as illustrated in Figure 6.2.

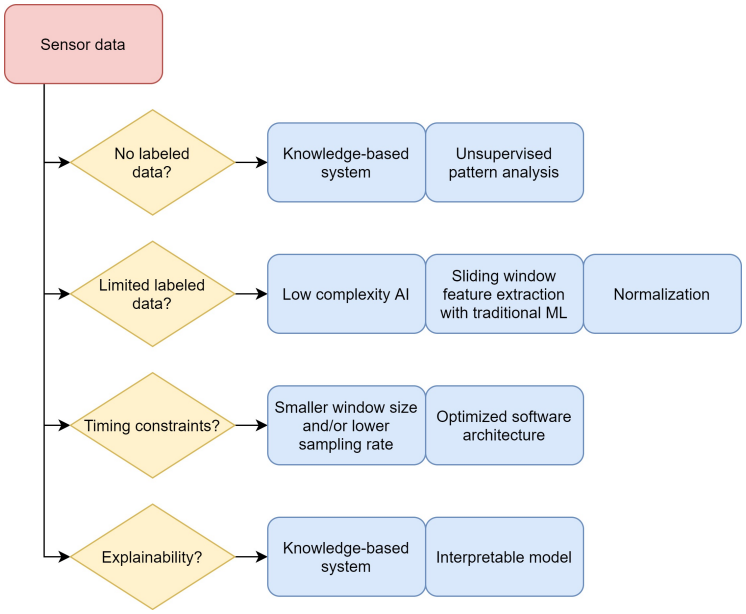


Figure 6.2: Schematic representation of the framework when working with a video sensor source.

When labeled data is unavailable, the options for processing sensor data are more limited. If expert knowledge is available regarding how sensor measurements can be analyzed or modeled, this information can be incorporated into a knowledge-based system. Such systems may range from simple rule-based or algorithmic methods to more sophisticated semantic systems, which enable advanced reasoning capabilities. Alternatively, events can be detected in an unsupervised manner using techniques like matrix profiling, which performs anomaly and motif detection by analyzing the similarity between subsequences within a dataset. This approach is very valuable for detecting anomalies without labeled data. However, if a classification of the detected anomalies is required, additional processing steps will be necessary.

The availability of labeled data significantly expands the options for processing sensor data. Modern deep learning architectures such as LSTMs, GRUs, and transformers are highly effective for modeling time series data. However, these architectures can be complex, with many trainable parameters, which may pose challenges when working with limited datasets. This is especially notable for time series data, where augmentation techniques are less diverse compared to image data. In cases where even smaller deep learning models are unsuitable, a valuable and strong alternative is to perform manual feature engineering on the

time series data and use these features as input for traditional machine learning models. Features should be extracted from smaller sliding windows of the data stream, capturing relevant patterns within the segments. Well-executed feature engineering allows for the extraction of meaningful features that support high-performing models while maintaining low dimensionality, avoiding overfitting, and improving robustness. As with most machine learning tasks, proper normalization is critical to address variability and inconsistencies across different environments and time periods, such as the sliding window normalization proposed in Chapter 3.

Processing time series data is generally less resource-intensive than processing video data. As a result, many of the techniques discussed thus far can be employed to achieve low-latency responses and high throughput rates, provided the size of the data windows is not excessive. If large windows are being used, experimenting with shorter window sizes or reducing sampling rates may help lower computational demands. For systems handling high volumes of sensor data, optimizing the software architecture for scalability is important. This can include leveraging distributed processing or edge computing approaches to manage data effectively.

When explainability is required, two practical options are available. If expert knowledge is accessible, a knowledge-based approach that directly incorporates this information is often the best choice. This ensures the outcomes are reliable, trustworthy, and interpretable. Alternatively, if a data-driven approach is preferred, sliding window feature extraction combined with a traditional machine learning model, such as a linear model or decision tree, is an excellent choice. These models provide fully interpretable results, allowing users to understand the reasoning behind the predictions.

6.3.3 Example New Use Case

The toolbox of techniques discussed earlier can be applied to new event detection use cases. For example, consider the design of a system to detect events of football supporters using or throwing pyrotechnics, such as smoke or fireworks, during a football match. This type of information is valuable for improving security by identifying individuals or groups that require closer monitoring, as illustrated in Figure 6.3.

Even in situations where labeled data is unavailable, several promising approaches can be employed. For instance, a foundational model like CLIP can be leveraged for zero-shot classification by comparing the embeddings of video frames to textual embeddings such as "smoke" or "fire". If explainability is required, the activations of the network can be analyzed using Grad-CAM to highlight to model's areas of focus, such as smoke in the video frame. These activation maps could

Figure 6.3: Example of the use of pyrotechnics by the crowd during a football game.

also serve as input for object extraction and tracking using a foundational model like SAM 2, similar to the implementation in the POI recognition pipeline of Chapter 5. By employing this strategy, it becomes possible to track pyrotechnic objects, determine if they remain within the stadium or are thrown onto the pitch, and thereby provide additional spatial event information.

Crowd detection can be accomplished using pretrained object detection models to identify individuals, while the pitch location can be estimated through traditional computer vision techniques, such as detecting large green areas in the video frames. Custom detection models, however, would offer improved accuracy and reliability for these tasks.

The monitoring solution likely does not require real-time processing of every frame in the video stream, making it possible to reduce the computational load by skipping frames. This allows for more processing time per frame without blocking the video stream. Also the frame resolution should be reduced as much as possible to lower the computational load. Given the large number of cameras involved in such an event detection system, distributing the processing demands across multiple servers could also be necessary.

6.4 Future Research Directions

Beyond the four event detection cases of this dissertation, numerous opportunities remain to explore in various domains, such as security and smart cities. Moreover, expanding the range of spatial scales is possible, from detecting events on a microscopic level, such as in plankton, to monitoring global phenomena like weather patterns.

While the current work mainly focuses on video and sensor data, future research should delve into integrating audio and textual data into event detection systems. Audio signals can offer valuable context in environments where visual data might be insufficient, such as identifying machinery noises to predict main-

tenance needs and prevent anomalies. Additionally, textual data sourced from platforms like social media, news reports, or transcriptions can be investigated to extract events hidden in them. For instance, monitoring real-time social media posts during natural disasters could help detect and map critical events.

The field of artificial intelligence is rapidly advancing, with a continuous flow of new technologies. Future studies should consider experimenting with newer model techniques, such as transformers, which have demonstrated strong performance across various tasks, spanning both sequential and image data. Thanks to their self-attention mechanism, transformers excel at focusing on the most relevant parts of the input, enabling them to effectively capture long-range dependencies and global contextual relationships. This could make them well-suited for event detection, where identifying events often relies on understanding the broader context and connections between data points that may be far apart. Foundational models, which are typically built using the transformer architecture, are also becoming increasingly more powerful and versatile. Their pre-training on massive datasets allows them to generalize well, even when applied to out-of-domain data. This makes them an ideal choice when designing event detection systems in scenarios where data is limited. Moreover, multi-modal foundational models are emerging as valuable tools, as they enable the integration of diverse data sources. Lastly, edge computing offers promising opportunities to process and analyze data closer to its source, reducing latency. This approach is particularly relevant for sensor-based applications where an immediate response could be important.

6.5 Closing Statement

This dissertation highlighted the importance and feasibility of designing robust and integrable event detection systems for real-world applications. Although the research was only validated on a limited set of event detection scenarios, it effectively demonstrated how to address data and system challenges, while also including spatial context. We hope this work serves as a source of inspiration for further exploration into the many untapped opportunities for event detection, which would contribute to efficiency, decision-making, and safety across numerous domains.