KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

University of Antwerp

ANTWERP DOCTORAL SCHOOL
Faculty of Science

Components Matter: Considering Compositionality for Visual Representations

Kaili Wang

Supervisors:
Prof. dr. ir. T. Tuytelaars
(KU Leuven)
Prof. dr. ir. J. A. Oramas Mogrovejo
(University of Antwerp)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering at KU Leuven, and Doctor of Science (PhD): Computer Science at University of Antwerp

November 2021

Components Matter: Considering Compositionality for Visual Representations

Kaili WANG

Examination committee:
Prof. dr. ir. O. Van der Biest, chair
Prof. dr. ir. T. Tuytelaars, supervisor
Prof. dr. ir. J. A. Oramas Mogrovejo, supervisor
(University of Antwerp)
Prof. dr. ir. M-F. Moens
Prof. dr. ir. T. Goedemé
Prof. dr. ir. D. Martens
(University of Antwerp)
Prof. dr. B. Zhou
(The Chinese University of Hong Kong)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering at KU Leuven, and Doctor of Science (PhD): Computer Science at University of Antwerp

November 2021

Componenten zijn belangrijk: Compositionaliteit overwegen voor visuele representaties

Kaili WANG

Examination committee:
Prof. dr. ir. O. Van der Biest, chair
Prof. dr. ir. T. Tuytelaars, supervisor
Prof. dr. ir. J. A. Oramas Mogrovejo, supervisor
(University of Antwerp)
Prof. dr. ir. M-F. Moens
Prof. dr. ir. T. Goedemé
Prof. dr. ir. D. Martens
(University of Antwerp)
Prof. dr. B. Zhou
(The Chinese University of Hong Kong)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering at KU Leuven, and Doctor of Science (PhD): Computer Science at University of Antwerp

November 2021

Acknowledgements

All I want to say is *Thank You*.

I want to thank Prof. Tinne Tuytelaars for giving me the opportunity to pursue the Ph.D. degree with her. She is a wonderful supervisor with broad and deep scientific knowledge, and always being kind and considerate. I really appreciate the freedom of research she gives to me, so that I can focus on the topics that interest me most. I still remember what Tinne told me in the beginning of my Ph.D.: learn to defend your own ideas. These words encourage me a lot, especially when I was a beginner. I will not forget our weekly meeting during the past four years and us staying up late before the deadlines with Jose (and the monthly trip to Aalst and Gent for the project meeting, which is an interesting experience to me). I learn a lot from these weekly meetings: research, paper writing/rebuttal, as well as presentation skill.

I want to thank Prof. José Antonio Oramas Mogrovejo. Jose practically taught me how to research from the very beginning when I did my master thesis in VISICS. He is a very kind and helpful person, he can always explain a difficult question in a simple manner. I appreciate that he could be my co-supervisor when he became a professor at University of Antwerp. I really enjoy the conversations (both academic and daily-life, both face-to-face and remote) with him.

I also want to thank my jury members: Prof. Marie-Francine Moens, Prof. Toon Goedemé, Prof. David Martens, and Prof. Bolei Zhou for their inspiring words and useful comments throughout my Ph.D.

I want to thank all the colleagues in our lab. Thank Paul for helping me handle a lot of odds and ends, Patricia for my contract and reimbursement staff, and Bert for helping me handle every computer-related problem. Thanks to Ali Diba, Vicent, Rahaf, Davy, Bert, Xuanli, Yu-hui, Maxim, Klass, Tom, Ali

Varamesh, Soroush, Yevhen, Simon, Thomas, Wouter, Dusan, Matthias, Zehao, Bo, Han, Gorjan, Abhishek, Cedrik, Eli, Tim and Vinod. I really enjoyed the time we shared during the coffee break, lunch, and birthday cake. Hope the pandemic can end soon, so that some traditional VISICS activities can revive. Thanks to Salma, Saja, and Hamed. Although we haven't met face to face in Antwerp due to the pandemic, our online chatting is fun and interesting.

I also want to thank VLAIO, FWO and AI Flanders for the support and founding to realize my Ph.D. works.

I want to thank all my friends I met in Leuven. Particularly, I want to say thank you to 段竞宇，马里千，陈颖意，魏博渊，邓雪莹，RP, 张扬，石茜，李翊涵，韩朋轩，李文遂。 I also want to thank my friends who are in Chengdu: 曾丽颖, 刘瑞琪。

最后，我想谢谢爸爸和妈妈，谢谢他们尊重我的每一个人生选择，让我过我想过的生活。还有家里的亲人：奶奶，姑姑，婆婆，外公，小姨，舅舅，舅妈和表弟。谢谢他们关心我，想念我。 (Finally, I want to thank my dad and mom, thank them for respecting my choices. Of course, I also want to thank my family for their support and love.)

Four-year hard work makes this book; I will be very much honored if readers find anything useful for their scientific research.

Abstract

In this thesis, we investigate the compositionality properties of single images and image sets for visual representations, with the goal of exploring the benefits of considering these properties. In realistic images this compositionality property can be observed in the multiple different features that compose them. Similar to single images, image sets also have the compositionality property, where a set is composed by multiple images consisting of multiple different features.

In the first two parts of the thesis, we focus on the single image scenario. We firstly propose a method to identify the component representations that are important to the prediction of a pre-trained model, given an input image. Leveraging the representation visualization method, we also generate the visual explanations, highlighting the important regions of the input images. From a complementary direction, in the next chapter, we actively encode two types of features, present on the inputs, separately. Assuming that images are composed by style and shape features, we disentangle the two features and then combine the disentangled style and shape representations to synthesize a novel image, where the appearance (style) of the novel image is the same with the original one while the shape is different, thus, achieving unpaired shape translation (changing shape, preserving appearance).

Then we move our research interest to the scenario where sets of images are considered. We firstly explore a set composed by images containing the same object to accurately localize the object. Based on the fact that the object-specific representations should be very similar across different images from the same class, we design a regularization to adjust the Class Activation Mapping based localization map. Secondly, we utilize a set of high-resolution face images as exemplars to help a model hallucinate high-resolution images. We believe more exemplars bring more useful visual information. To optimally extract the information from a set of exemplars, we design a module to find and combine the most useful component representation from the sets. Finally, we tackle the general multiple instance learning problem, where the model learns and

predicts based on unordered sets of elements. We propose to iteratively learn set-level representations via LSTMs. While not often used for this, we show LSTMs are capable of modeling unordered sets, based on their memory ability. The performance is competitive and even surpasses methods tailored to solve multiple instance learning problems. We also show that LSTMs can indirectly capture instance-level information using only set-level annotations.

Beknopte samenvatting

In deze dissertatie onderzoeken we de compositionaliteitseigenschappen van afzonderlijke beelden en beeldreeksen voor visuele representaties, met als doel de voordelen van het beschouwen van deze eigenschappen te onderzoeken. In realistische beelden kan deze compositionele eigenschap worden waargenomen in de veelvoud verschillende kenmerken waaruit ze zijn opgebouwd. Vergelijkbaar met enkelvoudige beelden, hebben beeldreeksen ook de compositionele eigenschap, waarbij een set is samengesteld uit meerdere beelden die bestaan uit meerdere verschillende kenmerken.

In de eerste twee delen van het proefschrift richten we ons op het scenario met een enkel beeld. Eerst stellen we een methode voor om de componentrepresentaties te identificeren die belangrijk zijn voor de voorspelling van een vooraf getraind model, gegeven een inputbeeld. Door gebruik te maken van de methode om representaties te visualiseren, genereren we ook visuele verklaringen die de belangrijke regio's van de inputbeelden benadrukken. Vanuit een complementaire richting, in het volgende hoofdstuk, coderen we actief twee soorten kenmerken, aanwezig op de inputs, afzonderlijk. Ervan uitgaande dat afbeeldingen zijn samengesteld uit stijl- en vormkenmerken, ontwarren we de twee kenmerken en combineren dan de ontwarren stijl- en vormvoorstellingen om een nieuw beeld te synthetiseren, waarbij het uiterlijk (stijl) van het nieuwe beeld hetzelfde is als het originele beeld, terwijl de vorm verschillend is, waardoor we ongepaarde vormvertaling bereiken (vorm veranderen, uiterlijk behouden).

Vervolgens verplaatsen we onze onderzoeksinteresse naar het scenario waarin sets van afbeeldingen worden beschouwd. Eerst onderzoeken we een set bestaande uit beelden die hetzelfde object bevatten om het object nauwkeurig te lokaliseren. Gebaseerd op het feit dat de object-specifieke representaties zeer gelijkaardig moeten zijn over verschillende beelden van dezelfde klasse, ontwerpen we een regularisatie om de Class Activation Mapping gebaseerde lokaliseringskaart aan te passen. Ten tweede gebruiken we een set van hoge-resolutie gezichtsbeelden als voorbeelden om een model te helpen hoge-resolutie beelden te hallucineren.

Wij geloven dat meer voorbeelden meer bruikbare visuele informatie opleveren. Om optimaal de informatie uit een set van voorbeelden te halen, ontwerpen we een module om de meest bruikbare componentrepresentatie uit de sets te vinden en te combineren. Tenslotte pakken we het algemene multiple instance leerprobleem aan, waarbij het model leert en voorspelt op basis van ongeordende sets van elementen. We stellen voor om iteratief representaties op set-niveau te leren via LSTMs. Hoewel LSTM's hier niet vaak voor gebruikt worden, laten we zien dat ze in staat zijn om ongeordende sets te modelleren, gebaseerd op hun geheugencapaciteit. De prestaties zijn concurrerend en overtreffen zelfs methoden die zijn toegesneden op het oplossen van multiple instance leerproblemen. We laten ook zien dat LSTM's indirect informatie op instellingsniveau kunnen vastleggen door alleen annotaties op instellingsniveau te gebruiken.

List of Abbreviations

AI Artificial Intelligence. 1

ANN Artificial Neural Network. 1

CAM Class Activation Map. 7

CNN Convolutional Neural Network. 3

CRR Common Region Regularization. 75

DCGAN Deep Convolutional Generative Adversary Network. 21

DNN Deep Neural Network. 1

FRR Full Region Regularization. 75

GAN Generative Adversary Network. 11

GAP Global Average Pooling. 16

HR High Resolution. 8

I2I Image-to-image translation. 6

IDU Instance Description Unit. 117

IoU Intersection over Union. 83

LR Low Resolution. 8

LSGAN Least Squares Generative Adversarial Networks. 22

LSTM Long short-term memory. 10

MIL Multiple Instance Learning. 10

NLP Natural Language Processing. 1

PSNR Peak Signal-to-Noise Ratio. 100

PWAve Pixel Weighted Average. 92

PxAP Pixel Averaged Precision. 83

SR Super Resolution. 92

SRE Set Representation Encoder. 117

SSL Self-supervised Learning. 141

TCAV Testing with Concept Activation Vectors. 45

UST Unpaired Shape Transformer. 12

ViT Vision Transformer. 19

WGAN Wasserstein Generative Adversary Network. 21

WSOL Weakly Supervised Object Localization. 7

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer Vision, *Artificial Intelligence*, and *Deep Learning*: these words often appear in newspapers and magazines nowadays. Many people may not have a clear idea regarding the relationship between the three above concepts. Artificial Intelligence (AI) is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals, which involves consciousness and emotionality [173]. AI is not a fresh idea, it can actually be traced back to the 1950s. The definition of "Intelligence" in AI actually is defined by the scientists who create it, from the ability to play chess to driving a car. It is a collective concept that expands over several fields, such as computer vision, natural language processing (NLP), etc. Computer vision or machine vision is the technique that handles how computers can understand the content of digital images or videos. In other words, this technique tries to equip computers with capabilities similar to those of the sense of sight. In terms of Deep learning, briefly speaking, it is a class of algorithms designed and implemented to solve modern AI problems. Inspired by biological neural networks that constitute animal brains, scientists designed such the artificial neural network (ANN) to tackle "intelligence tasks". ANN consists of computation nodes distributed in multiple hierarchical layers. Thanks to recent advances of Graphical Processing Units (GPU) to accelerate the computation, hardware can handle ANN with much more hidden layers. The deep ANN can also be called deep neural network (DNN). Deep learning can be extensively used in the computer vision community as well as NLP, Robotics, etc., which led to recent breakthroughs in the AI community. In short, computer vision is a sub-field of AI, and deep learning is the key family of algorithms making modern computer vision models work.

Current computer vision models have demonstrated a strong ability on many

tasks. In image classification, for example, state-of-the-art models can even surpass human performance. Beyond just assigning a class label to images, current models can even create a novel synthetic image from noise while humans can hardly perceive its fakeness. In addition, face swapping and animalization, pedestrian re-identification and autonomous driving, these popular applications are more or less related to computer vision techniques.

Then, compared to humans, how does a machine understand an image? To humans, for example, an image depicting a cat is perceived as a cat because our prior knowledge tells us that the combination of the color, texture, and shape, or the *features* of the image matches the concept of "cat" in our brain. To machines, an image is just a three-dimension matrix, nothing else. In order to make the machine understand an image, it has to be able to extract *features* or *representations* from images and store them. The core of the computer vision task is representation learning, but what is a "good representation"? It depends on the task. Imagine an image of a wild cat catching a butterfly on green grass. If the machine's task is to classify cat images from dogs, the good representation should represent the cat. If the machine is used to distinguish wild cats from domestic cats, then a good representation may need to represent additional contextual cues, e.g., the butterfly and/or the grass, since now the cat itself may not provide sufficient discriminative information.

Prior to the deep learning era, scientists and engineers manually designed different features, e.g., HOG [35], SIFT [116], SURF [13], etc., for a specific type of data and task. Nowadays, DNNs are able to extract the representation of any type of data guided by a reasonable amount of the data itself and the suitable cost function(s) or *loss(es)*. Millions of learnable parameters in DNNs can automatically learn the mapping from input to output while learning a representation, automatically, in the middle. It has a significant advantage: experts do not have to manually design different features for different images or tasks, all these jobs can be taken over by DNNs and data itself. In the meantime, questions and criticism have been raised due to the black-box characteristic of DNNs. Therefore, understanding what DNNs have learned exactly or why DNNs make such decisions are also interesting and necessary tasks in this community.

In Section 1.1, we present the motivation of this thesis. Then, based on the motivation, we present the tasks we will address in Section 1.2. In Section 1.3, we list our research questions. Finally, the overview of the thesis and our main contributions are presented in Section 1.4.

Figure 1.1: a. A cat, taken from imageNet dataset [37]. b. A cat and a dog, taken from the Internet [16]. c. The Starry Night, taken from the Internet [174].

1.1 Motivation

Let us talk about images first. Most of the realistic images are quite complex. An image of a cat, for instance (Fig. 1.1 a), actually contains several different features: the cat's body shape, fur color/pattern, eyes color, and the background grass etc. If there are multiple objects in one image, things become even more complex (Fig. 1.1 b). In addition, these features are not necessarily concrete, they can be also abstract. For example, paintings consist of the style and content features (Fig. 1.1 c). These features are the *components* of an image. In short, images can manifest themselves into a combination of different components. In this thesis, we refer to it as the *compositionality* of images. Intuitively, to understand images well, the components should be noticed.

In most cases, however, a DNN encodes one image into one representation for classification or regression task no matter how complex the image is, ignoring its compositionality characteristics (Fig. 1.2). The representation can be obtained at the output of any layer. Therefore, the representation encoded by the DNN actually contains several hidden components. Hence, it can be considered as a *compositional representation*. In this thesis, we are interested in the underlying components that define images and their representations. Moreover, we explore the possibilities/potential benefits brought by considering them.

Nowadays, DNNs, especially convolutional neural networks (CNN), have achieved close and even superior performance than humans on several tasks, e.g., image classification[72, 94], detection[142, 192] and generation[55]. However, the criticism from outside of academia has never stopped on these techniques due to their black-box characteristics. Apparently, pixels of inputs do not contribute equally to predictions made by models for a given task. Implicitly learning different components in the input images leads to confusion for humans about the

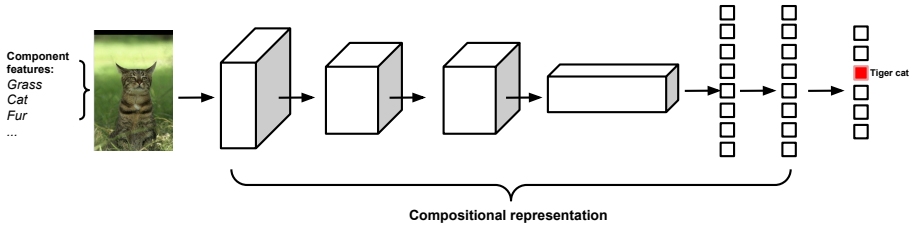

Figure 1.2: Normally one image is encoded into single representations layer by layer, even if it consists of several component features.

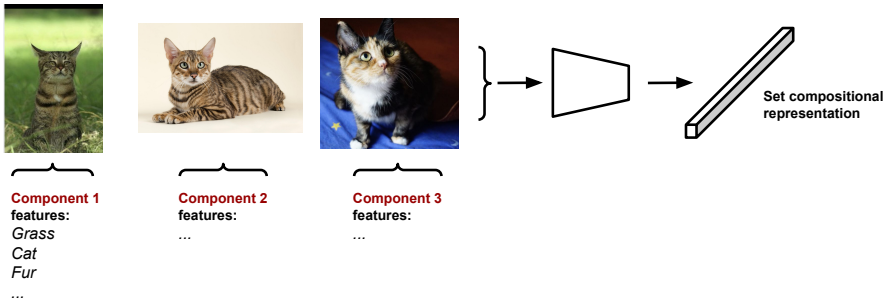

Figure 1.3: For the image set case, the first type of component is each image. In the meantime the component images have their own feature components.

model's prediction. Therefore, if there is a method that can reveal this important information, i.e., discovering the representations of hidden components used by the model to make predictions, it can help explain and interpret the DNN, making them more transparent to some extent.

Besides discovering the hidden components from the compositional representation, a complementary direction is separately learning different component features in the training phase. Models can have a better understanding of images by disentangling different component features and several advanced tasks beyond classification/regression can be achieved, such as video prediction [76], domain adaptation [115] and image translation [207]. In this thesis, we mainly focus on the image generation task. For instance, a novel T-shirt image can be generated by combining the style and shape representations disentangled from two different T-shirts images. This technique equips machines with the ability to create, and of course, the fashion industry can also benefit from it.

Apart from learning single images by dealing with the different component features, it is also interesting to consider the image sets case, where it is

similar but can be more complex than the single-image scenario. The sets can be composed of several images, and each image has its own feature-level components, which looks like a two-level hierarchy (Fig. 1.3). Inherently, image sets also have *compositionality* characteristics and the definition of *compositional representation* can be extended from a single image to an image set.

Usually, a set of images from the same class can provide more useful information than a single one, which is one of the motivations for considering image sets. For example, a single image may have occlusion on the foreground object, blocking some useful information. With the help of other images from the same class, it is possible to obtain complete information. Apart from that, while deep learning is a data-driven technique, it is difficult to collect and label a huge amount of data with 100% correct annotations. Sometimes a set of images with only a set-level label is easier to acquire. Therefore, a method capable of inspecting the component representation by learning the set-level representation of a group of images is also highly desirable.

1.2 Tasks of Interest

Based on the motivation, in this thesis, we tackle various tasks. There is always one spirit behind these tasks: considering the compositionality of the data no matter whether it is for single images or image sets. For single images scenario, we tackle DNN explanation/interpretation and unpaired image shape translation. The next three tasks are related to image sets: weakly-supervised object localization, exemplar-guided face image hallucination, and general multiple instance learning.

1.2.1 Deep Model Explanation and Interpretation

Methods based on DNNs have achieved impressive results for several computer vision tasks, such as image classification, object detection and image generation. Usually, despite of the complexity of an image, DNNs (e.g. VGG, ResNet) encode it as single representation after each layer, which means the representation of different components in the input is implicitly learned by the model. In this thesis, we try to discover these hidden representations that are crucial to predictions made by the model and provide their visualizations for people (Fig. 1.4).

There are mainly two approaches for model interpretation in the literature. Zeiler et al. [191] and Yosinsk et al. [186] manually inspected the visualizations

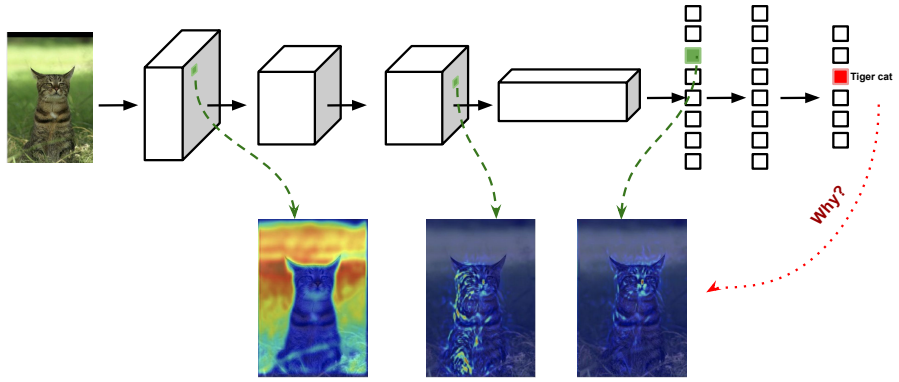

Figure 1.4: The DNN explanation task. Given a pre-trained DNN and an image with the predicted class, the explanation method finds the most important neurons (green square in the figure) for the model to make such prediction. By using the visualization technique, these important features can be visualized as heatmaps on the input image.

of every single filter, or a random subset thereof from every layer of a DNN. Differently, Bau et al. [12] and Fong et al. [48] considered using a dataset with pixel-wise annotations of possibly relevant concepts to provide references and compared the unknown internal activation with it. Apparently, the two methods have some weaknesses, such as introduced subjective bias [56], the need of concept annotation [12] and only being computable for convolutional layers (not applicable for fully connected layers). In this thesis, our method aims to overcome these weaknesses, providing pleasant visual-feedback from any layers by using the same annotations used to train the initial model only (i.e., image class label).

1.2.2 Unpaired Shape Translation

In the previous section, we aimed at explaining and interpreting a classification DNN. The key idea is to discover the important component representations implicitly learned by the model. In this task, we try to explicitly learn the different component representations of the input, i.e., style and shape representation of the fashion data, and combine different style and shape representations to synthesis a new image, achieving the Image-to-image shape translation.

Image-to-image translation (I2I) refers to a process that generates a novel image

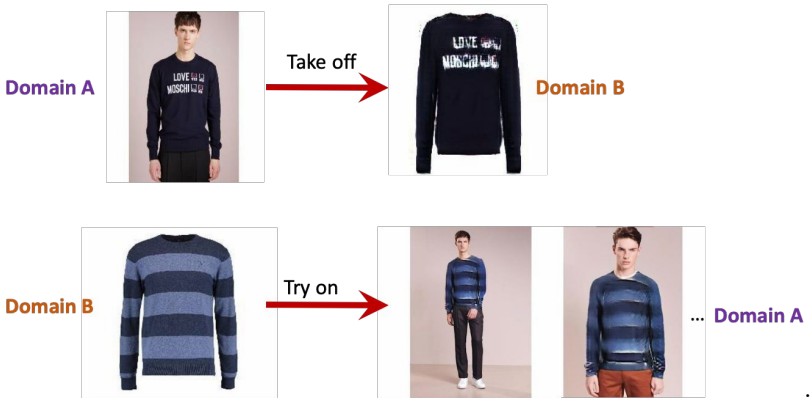

Figure 1.5: The task of unpaired shape translation: Translating a clothing item from a "catalog" image domain to the domain of individuals wearing the indicated item (try-on task, bottom), and vice versa (take-off task, top).

based on the original input image but is different in some aspects. Typically, I2I is also known as style transfer, where the geometry of the input image remains the same but the appearance/style changes between the novel and original input image, such as transferring photos to paintings [85] and colorizing the grey-scale images [18].

In contrast to the traditional setting (i.e., keeping shape, changing style) [121, 185, 203], in this thesis we focus on the opposite setting. The image semantics (e.g., the clothing pattern) should be preserved while the image geometry changes (e.g., the clothing shape or face angle). Fig. 1.5 shows some examples for this setting. This new setting is significantly more challenging as the image geometry changes. We refer to this task as shape transfer.

1.2.3 Weakly Supervised Object Localization

Weakly Supervised Object Localization (WSOL) is a type of localization task where the object coordinates are not provided as an annotation. The only supervision signal needed is the class label, the same as with the traditional classification task. One of the most used methods for this task is based on Class Activation Mapping (CAM) [205], which produces class activation maps as the localization heatmaps. This method enriches the CNN model with pseudo object localization ability when the model is only trained for classification.

Basically, there are two main problems of using CAM-based methods for this

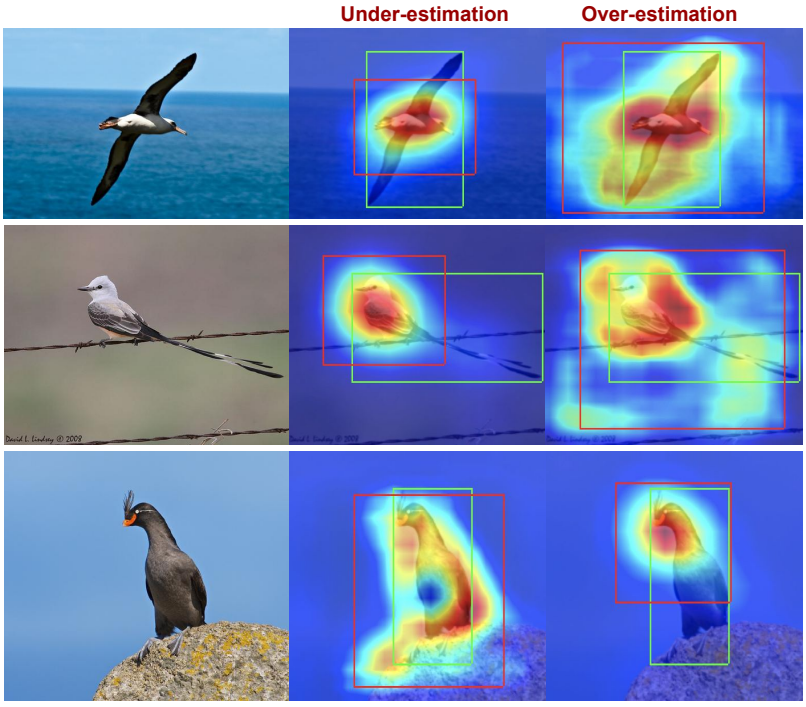

Figure 1.6: The task of weakly supervised object localization: Localization maps generated from CAM method have two problems, either under-estimating the region of the object (left) or over-estimating it (right). Green and red box refer to the ground truth and predicted box, respectively.

task: under-estimating the coverage of the object [34, 201, 33, 155, 189, 200, 199] (i.e., the localization heatmap focuses on the most discriminative region of the input), or over-estimating the coverage of the object [181] (i.e., the localization heatmap activates a lot on background regions). Please see Fig. 1.6. Current state-of-the-art methods focus on solving either one of them, while in this thesis we solve the two problems in a unified model.

1.2.4 Face Image Hallucination

The next task we address in this thesis is face image hallucination, which consists of two aspects: super-resolution and editing. Image super-resolution tries to restore a high-resolution (HR) image from a low-resolution (LR) image without changing the details. For example, facial expression, iris color, etc., should

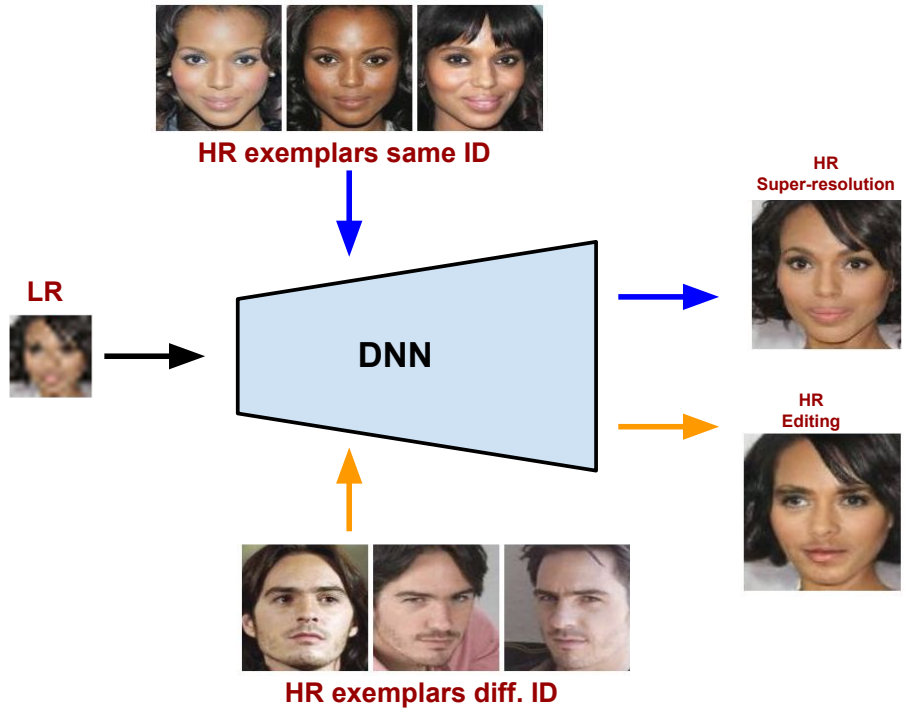

Figure 1.7: The face image hallucination task. Given a LR image, the model converts it to the HR version via generating the necessary visual details.

be consistent across both LR and HR images. Apart from that, the lack of visual details on the LR image gives some space to generate an HR image with different but reasonable visual features, which is called image editing. Fig. 1.7 displays some examples of this task.

Restoring HR images from input LR images needs extra information. In order to provide the visual details for the model, there are mainly two approaches. [31, 17, 187, 38] proposed using one HR exemplar to guide the model. Differently, [188, 118, 103, 86, 196] used the generic face priors, such as facial key points, segmentation.

In this thesis we plan to investigate along with the first approach but use multiple HR images. We believe multiple HR images can provide more useful information. Thus one of the important objectives is how to effectively handle the component of the set and the own feature components of each image.

1.2.5 Multiple Instance Learning via LSTM

This section moves our interest to general *Multiple Instance Learning* (MIL) problems, where the DNN makes decisions (e.g., classification, regression, etc.) based on the representation over a set of unordered images while image-level labels are not accessible. For example, to judge whether sets contain a specific component, or to count the number of occurrences of a specific component.

Long short-term memory (LSTM) networks [75] have been proposed to learn information in sequential (ordered) data over a long time. They have been used extensively and very successfully in the computer vision field for action recognition, future frame prediction, etc.

As they mostly handle ordered data, LSTMs do not seem appropriate for analyzing unordered sets at first - but is that so? In this thesis, we systematically analyze the capabilities of LSTMs for handling unordered sets.

1.3 Research Questions

From the previous sections we can see that compositionality characteristics of images do affect various computer vision tasks. By designing proper methods, the benefit of taking compositionality into account can be expected promisingly in different tasks. We believe it is worth studying and is not fully explored yet. Therefore, the main research question of this thesis is:

> *How does considering the compositionality of images help in computer vision tasks?*

We try to answer it from the single-image and image sets scenarios. For the sake of clarity of presentation, we split the main question into several sub-questions for each of the scenarios. The first two sub-questions are based on learning from single images.

> 1. Can different hidden components implicitly encoded in the compositional representations be discovered? Can they be used to explain and interpret a pre-trained DNN?

> 2. Is it possible to separately encode the component features of a single object rather than learning a single compositional representation and what are the benefits of doing so?

The next three questions are based on the image set scenario.

3. Given a set composed of images from the same class, how can we accurately localize each image's common object?

4. How to select useful visual details from a set composed by HR exemplars in a super-resolution model?

5. Is it possible to learn set representations, composed by unordered elements, by using a model which usually processes sequential data? What are the advantages?

1.4 Overview and Thesis Contributions

This thesis explores the potential benefits by considering the compositionality of single images and image sets. The work presented in this thesis has been spread across several papers. Each paper is assigned to one chapter and they address the research question in the previous section as a whole. In addition, each paper achieved state-of-the-art performance at the time of submission. In this section, we will give an overview of this thesis as well as more details of the contribution in each chapter and corresponding paper.

Chapter 2 presents the background knowledge used in this thesis, which covers mostly class activation mapping and its variant, attention mechanisms in computer vision, and some popular Generative Adversary Networks (GAN).

From Chapter 3 to Chapter 7, we try to answer the research questions by presenting our research works. In Chapter 3 we investigate DNN explanation and interpretation. Usually, in the training phase a deep model takes the whole image as input no matter how complex the input is and encodes it into several single representations layer by layer. In other words, different component representations of the input are implicitly encoded. In this chapter, we propose a method based on feature selection to identify these hidden component representations that are important for the trained model prediction. Given the assumption that only a small subset of the internal filters of a DNN encode features that are important for the task, we firstly identify these filters by formulating the feature selection as a $\mu - lasso$ optimization problem. Then we propose a method to generate the visual feedback for these relevant filters, which has the higher-level of details and improved quality over the deconvolution+guided back-propagation methods. By presenting Chapter 3, we

can answer Question 1. The content of this chapter is based on the following paper:

- J. Oramas M[1], K. Wang[1], T. Tuytelaars, Visual Explanation by Interpretation: Improving Visual Feedback Capabilities of Deep Neural Networks. International Conference on Learning Representation (ICLR) 2019.

Chapter 4 focuses on the unpaired shape translation task, where we disentangle the component representations (i.e., style and shape) in the training phase. We propose the Unpaired Shape Transforming (UST) method, which does not need any paired data or refinement post-processing. The model contains two asymmetrical streams, one is for transforming images in category domain with standard shape to context domain where it is more contextualized with large shape variation and vice versa. The first stream is apparently a "one-to-many mapping" problem while the latter is "many-to-one", where the output is deterministic. To tackle the "one-to-many" mapping problem we propose to use the context information to constrain the output. Specifically, we propose a Fit-in module to limit the variety of the output, making the output deterministic. In addition, we show the potential of the style representation learned by our model on the cross-(shape)-domain item retrieval task. This chapter can answer the second question and the content is based on the following two papers:

- K. Wang, L. Ma, J. Oramas M, L. Van Gool, T. Tuytelaars, Unpaired Image Shape Translation Across Fashion Data. International Conference on Image Processing (ICIP) 2020.
- K. Wang[2], L. Ma[2], J. Oramas M, L. Van Gool, T. Tuytelaars, Unpaired Shape Transforming for Image Translation and Cross-domain Retrieval. submitted to Computer Vision and Image Understanding).

In Chapter 5 we focus on the weakly-supervised object localization task. We propose a new CAM-based method MinMaxCAM. As we mentioned earlier, CAM-based methods suffer either over-estimation or under-estimation of the coverage of the object, according to different backbones. To address these two problems within one model, we design two regularizations, Common Region Regularization (*CRR*) and Full Region Regularization (*FRR*). They can serve as objective functions for the model to optimize the last linear layer. *CRR*

[1] J. Oramas M and K. Wang contribute equally. Jose worked more on the literature survey and main idea. Kaili worked more on the implementation and evaluation.

[2] K. Wang and L. Ma contribute equally. Kaili worked more on the implementation and writing. Liqian worked more on the main idea and provided help on implementation.

is based on the fact that for multiple images containing the same object, the object-specific representations from different images should be close to each other. Therefore, by minimizing *CRR* the activations of the localization map on the background region will be suppressed. On the other hand, *FRR* minimizes the distance between the full image representation and the object-specific representation, which can expand the activations centered on the small, most discriminative region. Moreover, the proposed MinMaxCAM is light-weight, it only relies on a standard classification model; no extra network is needed. It saves computation resources and is relatively simple to train. Regarding the performance, MinMaxCAM achieves the state-of-the-art performance on ImageNet, CUB-200-2011 and OpenImages-segmentation datasets at the time of submission of the work. This chapter addresses Question 3. The content of this chapter is mainly based on the following paper:

- K. Wang, J. Oramas M, T. Tuytelaars, MinMaxCAM: Improving object coverage for CAM-based Weakly Supervised Object Localization. British Machine Vision Conference (BMVC) 2021.

In Chapter 6, we tackle the face image super-resolution/hallucination problem. We utilize a set of high-resolution exemplars to provide useful high-frequency information with the model, guiding the model to super-resolve/hallucinate very low resolution (16×16, 8×8) images. Compared with using one exemplar, considering multiple exemplars gives more flexibility for the model to select the useful representations that best fit the LR input image. To effectively exploit useful component representation from the exemplars, we propose the Pixel Weighted Average (PWAve) module. This module consists of several 1×1 convolutional layers, producing spatial attention maps for all the exemplars. By aggregating the component representations based on the attention maps, the model learns how to select useful regions across different exemplars and produces superior results compared to simply averaging the representations. In addition, the conducted user study indicates our results are hard to distinguish from real images. The qualitative analysis suggests that our model outperforms the literature baselines. This chapter aims at answering Question 4. The content of this chapter is based on the following paper:

- K. Wang, J. Oramas M, T. Tuytelaars, Multiple Exemplars-based Hallucination for Face Super-resolution and Editing. Asian Conference on Computer Vision (ACCV) 2020.

In Chapter 7, we investigate the general Multiple Instance Learning (MIL) problem. MIL aims at learning how to encode a set composed of unordered

instances. In this thesis, the instances are images. Besides, MIL problems usually only have set-level supervision and can be solved in a weakly supervised manner. The model needs to learn the useful component representation from sets based on the requirement of the task. In this chapter, we advocate LSTMs handle this job. As we know, LSTM is widely utilized to learn a set of sequential data, such as action recognition (e.g. opening door vs. closing door). Apparently, the capability of capturing sequential information can be attributed to the memory ability of LSTM. The sequential dependency is one of the possible information carried by the set. We believe this memory ability is capable of capturing other types of information beyond order as well. In this chapter, we systematically analyze the performance of LSTMs when addressing MIL problems and conduct several experiments based on MNIST and a realistic dataset. The results indicate that indeed LSTM is competitive with or even surpasses state-of-the-art methods which are designed for handling specific MIL problems. Moreover, we find the component distribution of sets can also be obtained by modeling the set representation for specific tasks, which can be regarded as weakly supervised learning. Question 5 is well-answered by presenting this chapter. The content of Chapter 7 is based on the following paper:

- K. Wang, J. Oramas M, T. Tuytelaars, MinMaxCAM: In Defense of LSTMs for addressing MultipleInstance Learning Problems. Asian Conference on Computer Vision (ACCV) 2020.

For sake of the content and consistency of this thesis, we do not include the following works, which are also finished during my four-year Ph.D study, in the later chapters.

- K. Wang, Y. Huang, J. Oramas M, L. Van Gool, T. Tuytelaars, An Analysis of Human-Centered Geolocation. Winter Conference on Applications of Computer Vision (WACV) 2018.

- K. Wang, J. Oramas M, T. Tuytelaars, Towards Human-Understandable Visual Explanations: High Frequency Imperceptible Cues Can Better Be Removed. Submitted to International Conference on Learning Representation (ICLR) 2022.

Chapter 2

Background

In this thesis, all the proposed methods are built on different DNN architectures including CNNs and LSTMs, etc. Given these and other fundamental concepts (e.g. Deep learning, general ANN, CNN etc.) have been frequently discussed in depth and presented in many other theses, we assume the readers are familiar with these concepts and we will not cover them in this chapter. Here, we present the more advanced background knowledge that our proposed methods are built on. We start from introducing Class Activation Mapping [205], a method that is extensively used in model explanation and object localization. Then we present the attention mechanism used in computer vision tasks. Finally, we introduce generative adversarial networks (GANs).

2.1 Class Activation Mapping and Grad-CAM

2.1.1 Class Activation Mapping

Class Activation Mapping [205] is a method that can reveal what CNN mainly focuses in the images. The idea of Class Activation Mapping is based on the fact that the activation of neurons in the network is proportional to the importance of the feature. Here we refer the output of the layers as *activation map*. It is the same concept with *feature map* which is also widely used in other papers. However, there are multiple layers in DNNs and the corresponding activation maps usually have multiple channels (e.g. 512, 1024). This begs the question - which channels from which activation maps should be used? Class Activation Mapping proposes to use the activation map of the last convolutional block.

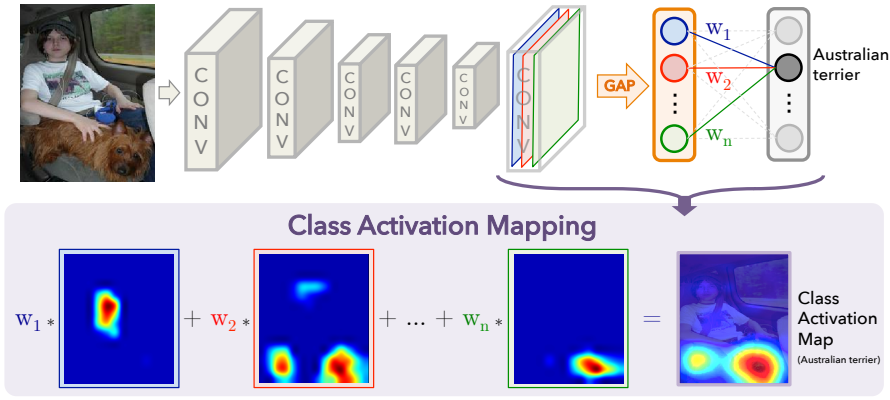

Figure 2.1: Illustration of how Class Activation Mapping works. Image is taken from [205].

Firstly, the fully-connected layers in a DNN (e.g. VGG) are replaced by a global average pooling (GAP) layer followed by one prediction linear layer[1]. Then, after introducing these changes, the model is either finetuned or trained. After re-training, the feature map of the last convolutional block is chosen to generate the class activation map (CAM). The weight of the prediction linear layer w.r.t. the predicted class is applied to combine different channels. Since the spatial resolution is much smaller than the input (e.g. 14×14 for VGG, 7×7 for ResNet when the input size is set 224×224), the combined feature map is then up-scaled via interpolation method (e.g. Bi-linear interpolation) to the size of input. Mathematically, given the backbone (B) of a DNN (before GAP), input image I, $w \in \mathbb{R}^{C \times K}$ is the weight matrix of the prediction linear layer, where C is the number of classes:

$$f = B(I) \tag{2.1}$$

$$CAM = \sum_{k=0}^{K-1} w_k^{\hat{c}} f^k \tag{2.2}$$

where f is the feature map (i.e. activation) of the last convolutional block, K is the number of channels of f, $\hat{c}$ is the predicted class of I. Please see Fig. 2.1

CAM is extensively applied for DNN explanation [205], weakly supervised object localization [205, 155, 200, 199, 34, 189, 201] etc, but also it has some weaknesses. The explanation cannot be very accurate, when highlighting small details on

[1]The implementation for different architectures can be slightly different.

the image space, because of the up-scaling operation. On the localization task, CAM mostly localizes the most discriminiative part of an image, rather than the whole object. We will discuss them in Chapter 5.

2.1.2 Grad-CAM

To avoid modifying the architecture and re-train a DNN, Selvaraju et al. [148] proposed a gradient-based approach to generate CAM, where the fully-connected layers can be retained. After pushing an image through the DNN and obtaining the raw scores y for all categories (i.e. before $softmax$), the raw scores are set to 0 for all the classes except the predicted one that is set to 1, noted as y^c. Gradients are obtained by computing the derivative of y^c w.r.t. the feature map f^k of a convolutional layer, noted as $\frac{\partial y^c}{\partial f^k}$. In other words, y^c is backpropagated from the last linear layer, through all the fully-connected layers, up to the last convolutional block. GAP operation is then applied on $\frac{\partial y^c}{\partial f^k}$ to average the two spatial dimensions, obtaining the combination weight α_k^c for the feature map of the last convolutional block.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial f_{i,j}^k} \tag{2.3}$$

Afterwards, Grad-CAM is calculated like CAM, different channels are weighted combined, but followed by a ReLU operation.

$$GradCAM = ReLU(\sum_{k=0}^{K-1} \alpha_k^c f^k) \tag{2.4}$$

In addition, Selvaraju et al. [148] proved that the expression of w_k^c used in CAM is identical to α_k^c in Grad-CAM. By calculating the gradients flow, Grad-CAM is more flexible, i.e. the task is not limited to image classification, where it can be image captioning and Visual Question Answering. Grad-CAM will be used and discussed in Chapter 3.

2.2 Attention Mechanisms in Computer Vision

In the Oxford Dictionary, *Attention* is defined as "the act of listening to, looking at or thinking about something/somebody carefully; interest that people show in somebody/something". In computer vision, the meaning of *attention* is similar,

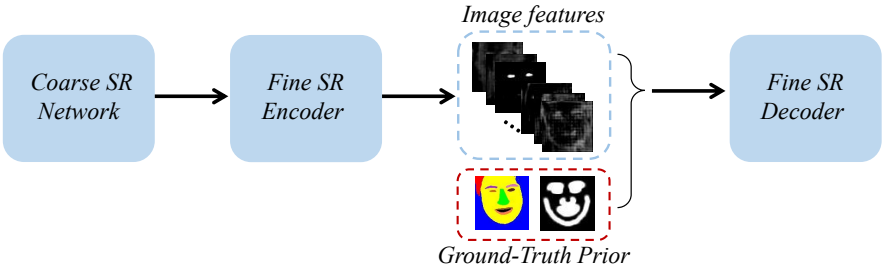

Figure 2.2: An example of using facial prior information as attention to help model generate HR images. The image is taken from [31].

which means making DNNs focus on some specific part of the representation. Usually, attention is represented as a *mask* or *heatmap*. The idea is simple, but it is quite useful and extensively applied in representation learning [67, 112, 81], image generation tasks [120, 209, 38] etc. The attention can be either obtained from prior knowledge, such as facial key points, body segmentation, or be computed by the representation itself (self-attention).

2.2.1 Prior Knowledge as Attention

For the image generation task, prior knowledge related to the data domain can help the DNN improve the generation quality. For example, [31, 17, 187, 38] use facial component heatmaps to guide the DNN for the face super-resolution task (Fig. 2.2). More specifically, the attention heatmap is concatenated with a feature representation (i.e. feature map) so that the model can learn to focus more on the region specified in the attention map. For other generation tasks, the attention heatmap can also serve as a guide for the target output. [120, 209] employ human pose information as the attention heatmap to force the model generate output images having the same pose with the attention. Similarly, facial attribute can also be used as one kind of attention map so that the model can transfer it to the target image [194]. In Chapter 4 we leverage prior knowledge-based attention in our generative model.

2.2.2 Self-attention

Another type of attention is called *self-attention*, which means the attention heatmap/weight is computed by the representation itself. Given the middle representation tensor $f \in \mathbb{R}^{N \times K \times H \times H}$ (assume height equals width), the idea

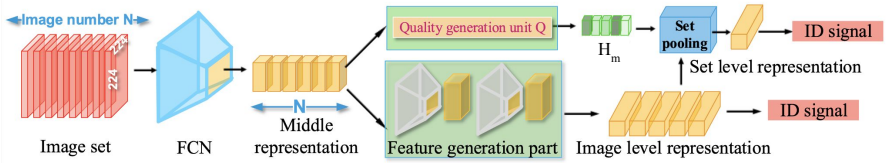

Figure 2.3: An example of using attention-generation network (which is called "Quality generation unit" in the image) F. It is inserted as a branch in the middle of the main network. The image is taken from [112].

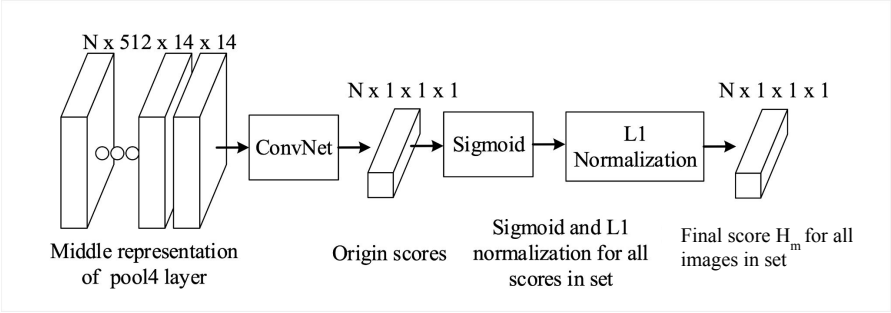

Figure 2.4: An example of the attention-generation network F, in this case, F produces a element-wise attention H_m. The image is taken from [112].

is to employ a branch network F besides the main network for f to produce the attention heatmap H_m, i.e. $H_m = F(f)$, so that H_m can indicate the important information of f. Depending on the task, the extra network F varies and H_m can be defined as spatial-wise attention [26, 202] ($H_m \in \mathbb{R}^{N\times 1\times H\times H}$), channel-wise attention [26, 77] ($H_m \in \mathbb{R}^{N\times K}$) or element-wise attention [112, 81] ($H_m \in \mathbb{R}^{N\times 1}$). Fig. 2.3 and Fig. 2.4 show two examples of how to insert F in the main network. A new attention-based representation can be obtained by either concatenating or multiplying f with H_m. Apparently, similar to some pooling operations (e.g. Global average pooling), H_m can be also used to aggregate f. In Chapter 6, we will introduce our proposed self-attention generation module that is integrated in the face hallucination model while in Chapter 7, we will discuss some self-attention-based methods in terms of MIL tasks.

The above attention mainly serves as a guidance for representation and models. Recently, Vision Transformer (ViT) [43] proposed to directly use the attention as the representation for the model to make prediction. Since we do not use ViT or any type of transformers in this thesis, we will not go further in their description. We suggest readers to related papers [165, 43] for more details.

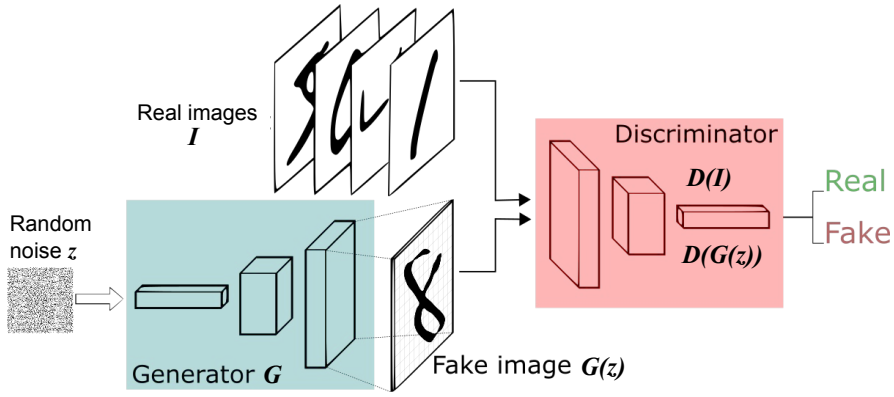

Figure 2.5: Illustration of the GAN models. It contains two components: the Generator (G) and the discriminator (D). The images is taken from [150].

2.3 Generative Adversary Networks

Generative Adversary Networks (GAN) [57] are extensively used in the image generation/translation task, the core of the model is finding a distribution mapping from the input data (e.g. Gaussian noise) to the target data (e.g. realistic images) by means of adversarial training. GAN consists of two networks: a generative network (generator, denoted as G) and a discriminative network (discriminator, denoted as D). G outputs fake images given the random noise variable input z, denoted as $G(z)$. It is trained to make $G(z)$ as real as possible. D is designed to estimate the probability of samples taken from real images.

2.3.1 Vanilla GAN

In [57], Goodfellow et al. use cross-entropy to measure the loss. The real and fake samples are assigned label 1 and 0, respectively. On the one hand, D should be accurate on distinguishing the real and fake samples, in other words, $\mathbb{E}_{I\sim p_{data}}[logD(I)]$ and $\mathbb{E}_{z\sim p_{model}}[log(1-D(G(z)))]$ should be maximized. On the other hand, $G(z)$, the generated samples (i.e. fake images), should get closer to the real samples during the training, thus, $\mathbb{E}_{z\sim p_{model}}[log(1-D(G(z)))]$ should be minimized. Therefore, D and G together play the *minmax* game, see Fig. 2.5. The following loss function should be optimized:

$$\min_G \max_D V(G,D) = \mathbb{E}_{I\sim p_{data}}[logD(I)] + \mathbb{E}_{z\sim p_{model}}[log(1-D(G(z)))] \quad (2.5)$$

What is the optimal value for $D(I)$? Intuitively, the perfect G can generate fake samples $G(z)$ which cannot be distinguished by the perfect D, noted as D^*, i.e. $D^*(I) = D^*(G(z)) = 1/2$. Mathematically, when G is fixed, the optimal D^* is $\frac{p_{data}(I)}{p_{model}(I)+p_{data}(I)}$. Apparently, the optimal case is that the generated images look exactly the same with real images, i.e. $p_{model}(I) = p_{data}(I)$. Hence, $D^* = 1/2$.

2.3.2 DCGAN

Radford et al. [135] integrate CNN architectures into GANs and proposed the Deep Convolutional Generative Adversary Network (DCGAN). DCGAN increases the quality of the generated images as well as makes the training process more stable. The most significant difference from GAN is that the convolutional layers are introduced. More specifically, DCGAN replaces any pooling layers with strided convolutional layers in the discriminator, and with fractional convolutional layers in the generator. In other words, the fully connected layers in the deeper architecture are removed. Normally, only the first fully connected layer is kept, which projects the input low-dimension noise to higher dimension. Besides, Batch Normalization is applied in both generator and discriminator. In terms of the activation function, *ReLU* is applied for every layer in the generator except the last one, where $Tanh$ is utilized. *LeakyReLU* is applied in the discriminator. Fig. 2.6 shows the architecture of the generator used in DCGAN. The original DCGAN generates fake images with 64×64, while the state-of-the-art DCGAN-based method [177] achieved 1024×1024 image generation.

2.3.3 WGAN and WGAN-GP

Wasserstein GAN (WGAN) [7] improves GAN in the context of the loss function. The authors proposed to use *Wasserstein* distance to measure the data distributions between the real and fake data. Cross-entropy is not suitable to measure the distance of two distribution where there is no overlapping or the amount of overlapping can be disregard between the real and fake data. Particularly, *Sigmoid* activation is removed in the last layer of the discriminator, which means the output no longer represents the probability of an image being fake or real. The discriminator is trained to make the output larger for real

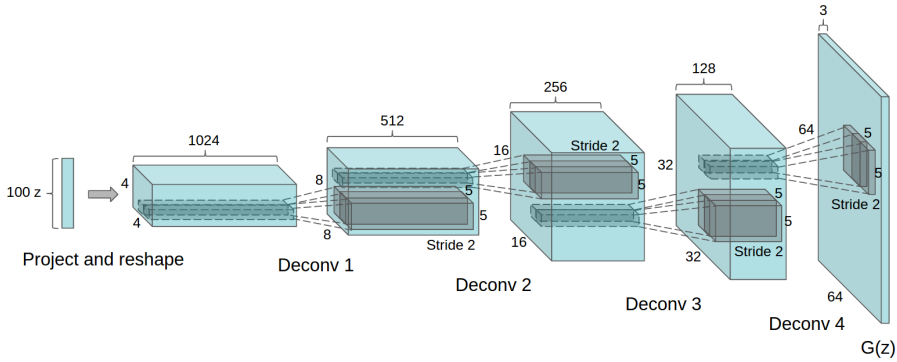

Figure 2.6: Generator architecture of DCGAN. The image is taken from [135].

data than for fake data. Therefore, *Critic* is a better name for this component. Besides, the logarithm operation is not applied in the GAN loss function anymore. In addition, the trainable weight of the discriminator is forced to be clipped into a pre-defined range after being updated (weight clipping) in order to meet the Lipschitz continuous condition. Regarding the optimizer, [7] proposed not to use momentum-based optimizer, such as Momentum and Adam.

$$\max_{C} V_{WGAN}(C) = \mathbb{E}_{I \sim p_{data}}[C(I)] - \mathbb{E}_{z \sim p_{model}}[C(G(z))] \tag{2.6}$$

$$\min_{G} V_{WGAN}(G) = -\mathbb{E}_{z \sim p_{model}}[C(G(z))] \tag{2.7}$$

Simply clipping the weight can cause gradient vanishing or explosion. Rather than apply weight clipping on discriminator, [64] proposed the *Gradient Penalty* to limit the gradient of discriminator, where WGAN-GP penalizes the model if the gradient norm moves away from its target norm value 1.

$$\min_{C} V_{WGAN-GP}(C) = \mathbb{E}_{\tilde{I} \sim p_{model}}[C(\tilde{I})] - \mathbb{E}_{I \sim p_{data}}[C(I)] + \lambda \mathbb{E}_{\hat{I} \sim p_{\hat{I}}}[(||\nabla C(\hat{I})||_2 - 1)^2] \tag{2.8}$$

$\hat{I}$ samples from real data I and fake data $\tilde{I}$. WGAN-GP is utilized in Chapter 6.

2.3.4 LSGAN

Mao et al. [124] proposed the Least Squares Generative Adversarial Networks (LSGAN) to improve the generative quality and training stability of the vanilla

GAN. Instead of calculating cross-entropy loss for the output of discriminator, LSGAN computes the least square loss. One problem of the cross-entropy loss is that the generator will not optimize the fake images once they are predicted as real by the discriminator even if they are still far away from the decision boundary. Least square loss, however, can not only do what cross-entropy loss does, but also force the generator to generate fake images toward the decision boundary. In this thesis, we adopt LSGAN in Chapter 4.

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{I\sim p_{data}}[(D(I)-1)^2] + \frac{1}{2}\mathbb{E}_{z\sim p_{model}}[(D(G(z)))^2] \quad (2.9)$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_{model}}[(D(G(z))-1)^2] \quad (2.10)$$

Chapter 3

DNN Explanation and Interpretation

In this chapter we investigate the task of explanation and interpretation for DNNs. Usually, DNNs encode one image into a single representation, ignoring its compositionality characteristic, and produces the prediction. Explanation methods and representation visualization technique can reveal what exact visual features of the input are taken by the model to make such a prediction. These important visual features are (part of) the component features of the input. Specifically, we propose a novel scheme for both interpretation as well as explanation in which, given a pretrained model, we automatically identify a small set of filters relevant to the classes considered by the model, without relying on additional annotations. The representations encoded by these filters are the most important component representations of the input images. We *interpret* the model through average visualizations of this reduced set of filters. Then, at test time, we *explain* the network prediction by accompanying the predicted class label with supporting visualizations derived from the identified component representations, where we find that indeed they focus on different regions of the input image. In addition, we propose a method to address the artifacts introduced by strided operations in deconvNet-based visualizations. Moreover, we introduce *an8Flower*, a dataset specifically designed for objective quantitative evaluation of methods for visual explanation. Experiments on the MNIST, ILSVRC12, Fashion144k and an8Flower datasets show that our method produces detailed explanations with good coverage of relevant component representation of the classes of interest.

The presented content is based on the following paper:

- J. Oramas M*, K. Wang*, T. Tuytelaars, Visual Explanation by Interpretation: Improving Visual Feedback Capabilities of Deep Neural Networks. International Conference on Learning Representation (ICLR) 2019.

3.1 Introduction

Methods based on deep neural networks (DNNs) have achieved impressive results for several computer vision tasks, such as image classification, object detection and image generation. Combined with the general tendency in the Computer Vision community of developing methods with a focus on high quantitative performance, this has motivated the wide adoption of DNN-based methods, despite the initial skepticism due to their black-box characteristics. Given an input image, a DNN (e.g. VGG, ResNet etc.) usually encodes it into a single representation layer by layer, despite how complex the input image is. Intuitively, pixels do not contribute equally for different tasks. Discovering the most influential (i.e. relevant to the prediction) ones of the input image can help humans better understand the working mechanism of the DNN. In this work, we aim for more visually-descriptive predictions and propose means to improve the quality of the visual feedback capabilities of DNN-based methods. Our goal is to bridge the gap between methods aiming at model *interpretation*, i.e., understanding what a given trained model has actually learned, and methods aiming at model *explanation*, i.e., justifying the decisions made by a model.

Model interpretation of DNNs is commonly achieved in two ways: either by a) manually inspecting visualizations of every single filter (or a random subset thereof) from every layer of the network ([186, 191]) or, more recently, by b) exhaustively comparing the internal activations produced by a given model w.r.t. a dataset with pixel-wise annotations of possibly relevant concepts ([12, 48]). These two paths have provided useful insights into the internal representations learned by DNNs. However, they both have their own weaknesses. For the first case, the manual inspection of filter responses introduces a subjective bias, as was evidenced by Gonzalez and Garcia et al. [56]. In addition, the inspection of every filter from every layer becomes a cognitive-expensive practice for deeper models, which makes it a noisy process. For the second case, as stated by Bau et al. [12], the interpretation capabilities over the network are limited by the concepts for which annotation is available. Moreover, the cost of adding annotations for new concepts is quite high due to its pixel-wise nature. A third weakness, shared by both cases, is inherited by the way in which they generate spatial filter-wise responses, i.e., either through deconvolution-based heatmaps ([157, 191]) or by up-scaling the activation maps at a given

layer/filter to the image space ([12, 205]). On the one hand, deconvolution methods are able to produce heatmaps with high level of detail from any filter in the network. However, as can be seen in Fig. 3.1 (bottom), they suffer from artifacts introduced by strided operations in the back-propagation process. Up-scaled activation maps, on the other hand, can significantly lose details when displaying the response of filters with large receptive field from deeper layers. Moreover, they have the weakness of only being computable for convolutional layers.

In order to alleviate these issues, we start from the hypothesis proven by Bau et al. [12] and Yosinski et al. [186], that only a small subset of the internal filters of a network encode representations that are important for the task that the network addresses. These encoded representations are the part of the component representations that are relevant to the prediction. Based on that assumption, we propose a method which, given a trained DNN model, automatically identifies a set of relevant internal filters whose encoded representations serve as indicators for the class of interest to be predicted (Fig. 3.1 top). These filters can originate from any type of internal layer of the network, i.e., *convolutional, fully connected*, etc. Selecting them is formulated as a μ-*lasso* optimization problem in which a sparse set of filter-wise responses are linearly combined in order to predict the class of interest. At test time, we move from interpretation to explanation. Given an image, a set of identified relevant filters (groups of neurons), and a class prediction, we accompany the predicted class label with heatmap visualizations of the top-responding relevant filters (i.e. relevant compositional representation) for the predicted class, see Fig. 3.1 (middle). The visualized relevant component representations are distributed on different regions of the input image, while the input is encoded into one representation as a whole when the DNN is trained.

In addition, by improving the resampling operations within deconvnet-based methods, we are able to address the artifacts introduced in the back-propagation process, see Fig. 3.1 (bottom). Overall, the proposed method removes the requirement of additional expensive pixel-wise annotation, by relying on the same annotations used to train the initial model. Moreover, by using our own variant of a deconvolution-based method, our method is able to consider the spatial response from any filter at any layer while still providing visually pleasant feedback. This allows our method to reach some level of explanation by interpretation.

Finally, recent approaches to evaluate explanation methods measure the validity of an explanation either via user studies ([191, 148]) or by measuring its effect on a proxy task, e.g. object detection/segmentation ([206, 195]). While user studies inherently add subjectivity, benchmarking through a proxy task steers the optimization of the explanation method towards such task. Here we propose an objective evaluation via *an8Flower*, a synthetic dataset where

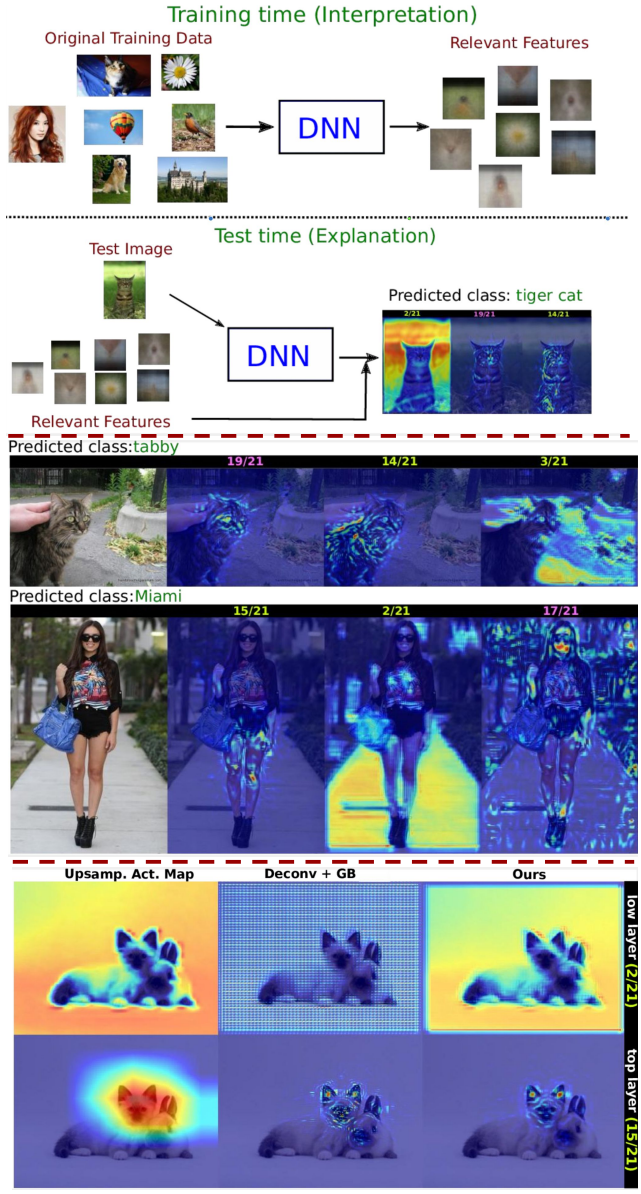

Figure 3.1: Top: Proposed training/testing pipeline. Middle: Visual explanations generated by our method. For each image, we visualize the component representation encoded by the identified filters. Note the relevant component representations can come from both object itself as well as from the context. Predicted class labels are enriched with heatmaps indicating the pixel locations, associated to the compositional representation, that contributed to the prediction. On top of each heatmap we indicate the number of the layer where the features come from. The layer type is color-coded (green for convolutional and pink for fully connected). Bottom: Visualization comparison. Note how our heatmaps attenuate the grid-like artifacts introduced by deconvnet-based methods at lower layers. At the same time, our method is able to produce a more detailed visual feedback than up-scaled activation maps.

the discriminative feature between the classes of interest is controlled. This allows us to produce ground-truth masks for the regions to be highlighted by the explanation. Furthermore, it allows us to quantitatively measure the performance of methods for model explanation.

The main contributions of this work are four-fold.

- We propose an automatic method based on feature selection to identify the hidden-encoded component representations that are important for the prediction of a given class. This alleviates the requirement of exhaustive manual inspection or additional expensive pixel-wise annotations required by existing methods.
- The proposed method is able to provide visual feedback with higher-level of detail over up-scaled raw activation maps and improved quality over recent deconvolution+guided back-propagation methods.
- The proposed method is general enough to be applied to any type of network, independently of the type of layers that compose it.
- We release a dataset and protocol specifically designed for the evaluation of methods for model explanation. To the best of our knowledge this is the first dataset aimed at such task, at the time that the corresponding paper was submitted.

3.2 Related Work

Interpretation. Zeiler et al. [191] and Zhou et al. [206] proposed to visualize properties of the function modelled by a network by systematically covering (part of) the input image and measuring the difference of activations. The assumption is that occlusion of important parts of the input will lead to a significant drop in performance. This procedure is applied at test time to identify the regions of the image that are important for classification. However, the resolution of the explanation will depend on the region size. Another group of works focuses on linking internal activations with semantic concepts. Escorcia et al. [45] proposed a feature selection method in which the neuron activations of a DNN trained with object categories are combined to predict object attributes. Similarly, Bau et al. [12] Fong et al. [48] and Zhang et al. [197] proposed to exhaustively match the activations of every filter from the convolutional layers against a dataset with pixel-wise annotated concepts. While both methods provide important insights on the semantic concepts encoded by the network, they are both limited by the concepts for which annotation is available. Similar

to [45], we discover relevant internal filters through a feature selection method. Different from it, we link internal activations directly to the same annotations used to train the initial model. This removes the expensive requirement of additional annotations. A third line of works aims at discovering frequent visual patterns ([40, 141]) occurring in image collections. These patterns have a high semantic coverage which makes them effective as means for summarization. We adopt the idea of using visualizations of (internal) mid-level elements as means to reveal the relevant features encoded, internally, by a DNN. More precisely, we use the average visualizations used by these works in order to interpret, visually, what the network has actually learned.

Explanation. For the sake of brevity, we ignore methods which generate explanations via bounding boxes ([89, 132]) or text ([73]), and focus on methods capable of generating visualizations with pixel-level precision. Zeiler et al. [191] proposed a deconvolutional network (Deconvnet) which uses activations from a given top layer and reverses the forward pass to reveal which visual patterns from the input image are responsible for the observed activations. Simonyan et al. [152] used information from the lower layers and the input image to estimate which image regions are responsible for the activations seen at the top layers. Similarly, Bach et al. [9] decomposed the classification decision into pixel-wise contributions while preserving the propagated quantities between adjacent layers. Later, Springenberg et al. [157] extended these works by introducing "guided back-propagation", a technique that removes the effect of units with negative contributions in forward and backward pass. This resulted in sharper heatmap visualizations. Zhou et al. [205] propose Global Average Pooling, i.e., a weighted sum over the spatial locations of the activations of the filters of the last convolutional layer, which results in a class activation map. Finally, a heatmap is generated by upsampling the class activation map to the size of the input image. Selvaraju et al. [148] extended this by providing a more efficient way for computing the weights for the activation maps. Recently, Chattopadhyay et al. [24] extended this with neuron specific weights with the goal of improving object localization on the generated visualizations. Here, we take DeconvNet with guided-backpropagation as starting point given its maturity and ability to produce visual feedback with pixel-level precision. However, we change the internal operations in the backward pass with the goal of reducing visual artifacts introduced by strided operations while maintaining the network structure.

Benchmarking. Zhou et al. [205] and Zhang et al. [195] proposed a saliency-based evaluation where explanations are assessed based on how well they highlight complete instances of the classes of interest. Thus, treating model explanation as a weakly-supervised object detection/segmentation problem. This saliency-based protocol assumes that explanations are exclusive to intrinsic

object features, e.g. color, shape, parts, etc. and completely ignores extrinsic features, e.g. scene, environment, related to the depicted context. Zeiler et al. [191] and Selvaraju et al. [148] proposed a protocol based on crowd-sourced user studies. These type of evaluations are not only characterized by their high-cost, but also suffer from subjective bias ([56]). Moreover, Das et al. [36] suggest that deep models and humans do not necessarily attend to the same input evidence even when they predict the same output. Here we propose a protocol where the regions to be highlighted by the explanation are predefined. The goal is to *objectify* the evaluation and relax the subjectivity introduced by human-based evaluations. Moreover, our protocol makes no strong assumption regarding the type of features highlighted by the the explanations.

3.3 Proposed Method

The proposed method consists of two parts. At training time, a set of relevant layer-filter pairs are identified for every class of interest c. This results in a relevance weight w_c, associated to class c, for every filter-wise response x computed internally by the network. At test time, an image I is pushed through the network producing the class prediction $\hat{c}{=}F(I)$. Then, taking into account the internal responses x, and relevance weights $w_{\hat{c}}$ for the predicted class $\hat{c}$, we generate visualizations of the identified component representation, indicating the image regions that contributed to this prediction.

3.3.1 Identifying Relevant Features

One of the strengths of deep models is their ability to learn abstract concepts from simpler ones. That is, when an example is pushed into the model, a conclusion concerning a specific task can be reached as a function of the results (activations) of intermediate operations at different levels (layers) of the model. These intermediate results may hint at the "semantic" concepts that the model is taking into account when making a decision. From this observation, and the fact that activations are typically sparse, we make the assumption that some of the internal filters of a network encode features that are important for the task that the network addresses. To this end, we follow a procedure similar to Escorcia et al. [45], aiming to predict each class c by the linear combination $w_c{\in}\mathbb{R}^m$ of its internal activations x, with m the total number of neurons/activations.

As an initial step, we extract the image-wise response x_i. To this end, we compute the L_2 norm of each channel (filter response) within each layer and

produce a 1-dimensional descriptor by concatenating the responses from the different channels. This layer-specific descriptor is L_1-normalized in order to compensate for the difference in length among different layers. Finally, we concatenate all the layer-specific descriptors to obtain x_i. In this process, we do not consider the last layer whose output is directly related to the classes of interest, e.g. the last two layers from VGG-F [23].

Following this procedure, we construct the matrix $X \in \mathbb{R}^{m \times n}$ by passing each of the n training images through the network F and storing the internal responses x. As such, the i^{th} image of the dataset is represented by a vector $x_i \in \mathbb{R}^m$ defined by the filter-wise responses at different layers. Furthermore, the possible classes that the i^{th} image belongs to are organized in a binary vector $l_i \in \{0,1\}^C$ where C is the total number of classes. Putting the annotations from all the images together produces the binary label matrix $L=[l_1, l_2, ..., l_n]$, with $L \in \mathbb{R}^{C \times n}$. With these terms, we resort to solving the equation:

$$W^* = argmin_W \; ||X^T W - L^T||_F^2 \quad subject\ to: \; ||w_c||_1 \leq \mu \; , \; \forall_j = 1, ..., C \tag{3.1}$$

with μ a parameter that allows controlling the sparsity. This is the matrix form of the μ-lasso problem. This problem can be efficiently solved using the Spectral Gradient Projection method [123, 163]. The μ-lasso formulation is optimal for cases like the ones obtained by ResNet where the number of internal activations is large compared to the number of examples. After solving the μ-lasso problem, we have a matrix $W=[w_1, w_2, ..., w_C]$, with $W \in \mathbb{R}^{m \times C}$. We impose sparsity on W by enforcing the constraints on the L_1 norm of w_c, i.e., $||w_c||_1 \leq \mu \; , \; \forall_c = 1, ..., C$. As a result, each non-zero element in W represents a pair of network layer p and filter index q (within the layer) of relevance.

3.3.2 Generating Visual Feedback

During training time (Section 3.3.1), we identified a set of relevant features (indicated by W) for the classes of interest. At test time, we generate the feedback visualizations by taking into account the response of these features on the content of the tested images. Towards this goal, we push an image I through the network producing the class prediction $\hat{c}=F(I)$. During that pass, we compute the internal filter-wise response vector x_i following the procedure presented above. Then we compute the response $r_i^{\hat{c}}=(w_{\hat{c}} \circ x_i)$, where $\circ$ represents the element-wise product between two vectors. Note that the $w_{\hat{c}}$ vector is highly sparse, therefore adding an insignificant cost at test time. The representations, i.e., layer-filter pairs (p^*, q^*), with strongest contribution in the prediction $\hat{c}$ are selected as those with maximum response in $r_i^{\hat{c}}$. Finally, we feed this information to the Deconvnet-based method with guided backpropagation from Grun et

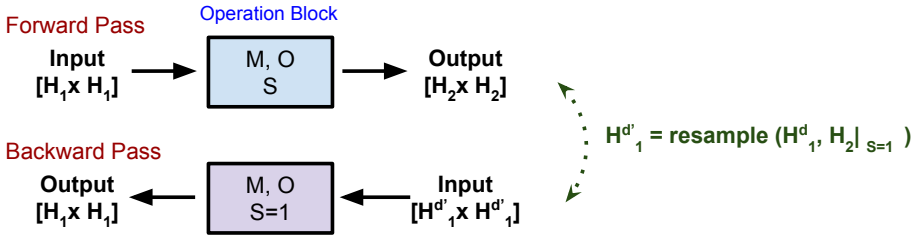

Figure 3.2: To attenuate artifacts , during the backward pass, we set the stride to 1 ($S = 1$) and compensate by resampling the input so that $A'_d = B|_{S=1}$.

al. [63] to visualize the important representations as defined by the layer-filter pairs (p^*, q^*). Following the visualization method from [63], given a filter p from layer q and an input image, we first push forward the input image through the network, storing the activations from each filter at each layer, until reaching the layer p. Then, we backpropagate the activations from filter q at layer p with inverse operations until reaching back to the input image space. As result we get as part of the output a set of heatmaps, associated to the relevant features, defined by (p^*, q^*), indicating the influence of the pixels that contributed to the prediction. See Fig.3.1 (top) for an example of the visual feedback provided by our method. Please refer to [63, 157, 191] for further details regarding Deconvnet-based and Guided backpropagation methods.

3.3.3 Improving Visual Feedback Quality

Deep neural networks addressing computer vision tasks commonly push the input visual data through a sequence of operations. A common trend of this sequential processing is that the input data is internally resampled until reaching the desired prediction space. As mentioned in Sec. 3.2, methods aiming at interpretation/explanation start from an internal point in the network and go backwards until reaching the input space - producing a heatmap. However, due to the resampling process, heatmaps generated by the backwards process tend to display grid-like artifacts. More precisely, we find that this grid effect is caused by the internal resampling introduced by network operations with stride larger than one (s>1). To alleviate this effect, in the backwards pass, we set the stride s=1 and compensate for this change by modifying the input accordingly. As a result, the backwards process can be executed while maintaining the network structure.

More formally, given a network operation block defined by a convolution mask

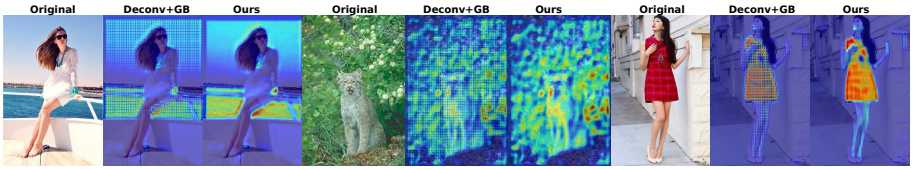

Figure 3.3: Heatmap visualization at lower layers of VGG-F. Note how our method attenuates the grid-like artifacts introduced by existing DeconvNet+GB methods ([157]).

with size $[M \times M]$, stride $[S, S]$, and padding $[O, O, O, O]$, the relationship between the size of its input $[H_1 \times H_1]$ and its output $[H_2 \times H_2]$ (see Fig. 3.2) is characterized by the following equation:

$$H_1 + 2 \cdot O = M + (H_2 - 1) \cdot s \tag{3.2}$$

from where,

$$H_2 = [\ (H_1 + 2 \cdot O - M)/s\ ] + 1 \tag{3.3}$$

Our method starts from the input ($[H_1^d \times H_1^d]$), which encodes the contributions from the input image, carried by the higher layer in the Deconvnet backward pass. In order to enforce a "cleaner" resampling when $S{>}1$, during the backward pass, the size of the input ($[H_1^d \times H_1^d]$) of the operation block should be the same as that of the feature map ($[H_2 \times H_2]$) produced by the forward pass if the stride s were equal to one, i.e., $H_1^{d'} = H_2|_{s=1}$. According to Eq. 3.3 with $S{=}1$, H_1^d should therefore be resampled to $H_1^{d'} = H_2|_{s=1} = H_1 + 2 \cdot O - M + 1$. We do this resampling via the nearest-neighbor interpolation algorithm given its proven fast computation time which makes it optimal for real-time processing. By introducing this step, the network will perform the backwards pass at every layer with stride $S{=}1$ and the grid effect will disappear. See Fig. 3.3 for some examples of the improvements introduced by our method.

3.4 Evaluation

We conduct four sets of experiments. First, in Sec. 3.4.1, we verify the importance of the identified relevant features in the task addressed by the network. Then, in Sec. 3.4.2, we evaluate the improvements on visual quality provided by our method. In Sec. 3.4.3, we quantify the capability of our visual explanations to highlight the regions that are descriptive for the classes of interest. Finally, in Sec. 3.4.4, we assess the sensitivity of the proposed method w.r.t. the predicted classes.

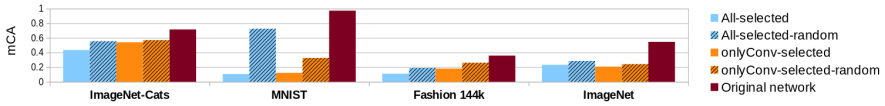

Figure 3.4: Changes in mean classification accuracy (mCA) as the identified relevant filters are ablated.

We evaluate the proposed method on an image recognition task. We conduct experiments on three standard image recognition datasets, i.e., MNIST [97], Fashion144k [151] and imageNet (ILSVRC'12) [145]. Additionally, we conduct experiments on a subset of cat images from imageNet (imageNet-cats). MNIST covers 10 classes of hand-written digits. It is composed by 70k images in total (60k for training/validation, 10k for testing). The imageNet dataset is composed of 1k classes. Following the standard practice, we measure performance on its validation set. Each class contains 50 validation images. For the Fashion144k dataset [151], we consider the subset of 12k images from Wang et al. [169] used for the geolocation of 12 city classes. The imageNet-cats subset consists of 13 cat classes, containing both domestic and wild cats. It is composed of 17,550 images. Each class contains 1,3k images for training and 50 images for testing.

3.4.1 Importance of Identified Relevant Features

In this experiment, we verify the importance of the "relevant" representations identified by our method at training time (Sec. 3.3.1). To this end, given a set of identified representation, we evaluate the influence they have in the network by measuring changes in classification performance caused by their removal. We remove representation in the network by setting their corresponding layer-filter to zero. The expected behavior is that a set of features with high relevance will produce a stronger drop in performance when ablated. Fig. 3.4 shows the changes in classification performance for the tested datasets. We report the performance of four sets of features:

- *All*, selected with our method by considering the whole internal network architecture.
- *OnlyConv*, selected by considering only the convolutional layers of the network.
- A *Random* selection of features (filters) selected from the layers indicated in the sets *All* and *OnlyConv*, and for reference.

- The performance obtained by the original network.

Note that the *OnlyConv* method makes the assumption that relevant features are only present in the convolutional layers. This is a similar assumption as the one made by state-of-the-art methods [12, 178, 205]. When performing feature selection (Sec.3.3.1), we set the sparsity parameter μ=10 for all the tested datasets. This produces subsets of 92|101, 46|28, 104|111, 248|180 relevant features for the *All* | *OnlyConv* methods, on the respective datasets from Fig. 3.4. Differences in the number of the selected features can be attributed to possibly redundant or missing predictive information between the initial pools of filter responses x used to select the *All* and *OnlyConv* features.

A quick inspection of Fig. 3.4 shows that indeed classification performance drops when we remove the identified features, *All* and *OnlyConv*. Moreover, it is noticeable that a random removal of features has a lower effect on classification accuracy. This demonstrates the relevance of the identified features for the classes of interest. In addition, it is visible that the method that considers the complete internal structure, i.e., *All*, suffers a stronger drop in performance compared to the *OnlyConv* which only considers features produced by the convolutional layers. This suggests that there is indeed important information encoded in the fully connected layers, and while convolutional layers are a good source for features, focusing on them only does not reveal the full story.

Regarding the effect of the sparsity value μ in the μ-lasso formulation (Sec. 3.3.1), we note that increasing μ increases the number of selected features. Note that $\mu \neq$ the number of features. This leads to more specialized features that can better cope with rare instances of the classes of interest. We decided to start from a relatively low value, e.g. μ=10, in order to focus on a small set of relevant features that can generalize to the classes of interest while, at the same time, keeping the computational cost low.

Qualitative Analysis. In order to get a qualitative insight into the type of information that these representations encode we compute an average visualization by considering the top 100 image patches where such representations have a high response. Towards this goal, given the set of identified relevant representations, for every class, we select images with higher responses. Then, we take the input image at the location with maximum response for a given filter and crop it by considering the receptive field of the corresponding layer-filter of interest. Selected examples of average images, with rich semantic representation, are presented in Fig. 3.5 for the tested datasets.

We can notice that for imageNet-Cats, the identified representations cover descriptive characteristics of the considered cat classes. For example, the dark head of a Siamese cat, the nose/mouth of a cougar, or the fluffy-white

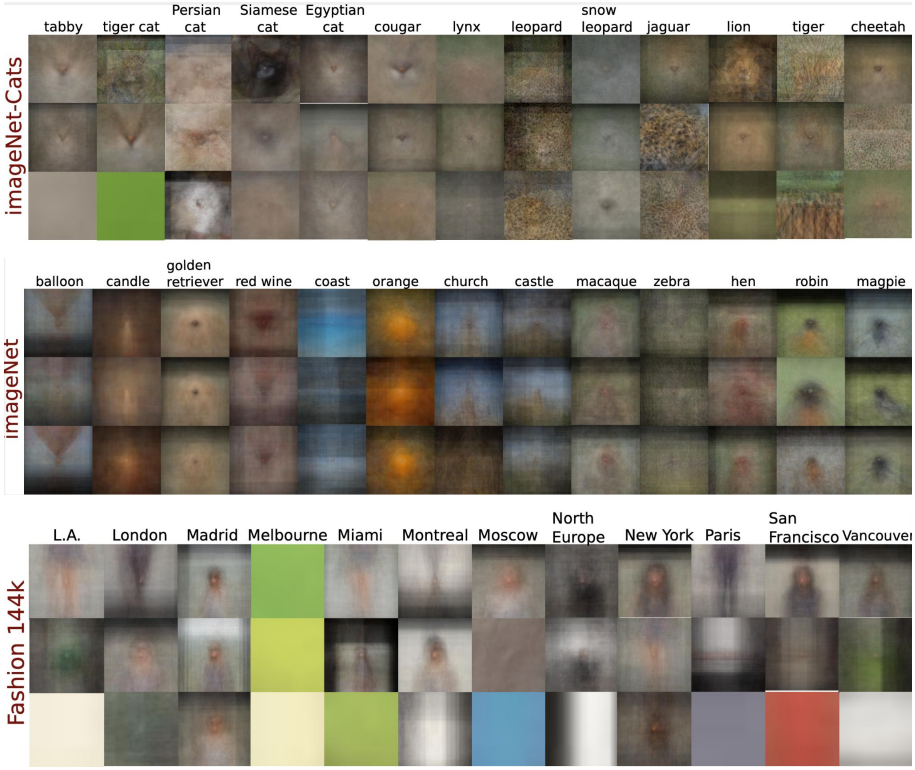

Figure 3.5: Average Images from the identified relevant filters for the ImageNet-Cats subset (top), some selected classes from the full ImageNet (middle) and the Fashion144K (bottom) datasets, respectively.

body shape of Persian cat. Likewise, it effectively identifies the descriptive fur patterns from the jaguar, leopard and tiger classes and colors which are related to the background. We see a similar effect on a selection of other objects from the rest of the imageNet dataset. For instance, for scene-type classes, i.e., coast, castle and church, the identified representations focus on the outline of such scenes. Similarly, we notice different viewpoints for animal-type classes, e.g. golden-retriever, hen, robin, magpie. On the Fashion144k dataset (Fig. 3.5 (bottom)) we can notice that some classes respond to features related to green, blue, red, and beige colors. Some focus on legs, covered and uncovered, while others focus on the upped body part. It is interesting that from the upper body parts, some focus on persons with dark long hair, short hair, and light hair. Similarly, there is a class with high response to horizontal black-white gradients where individuals tend to dress in dark clothes. These visualizations

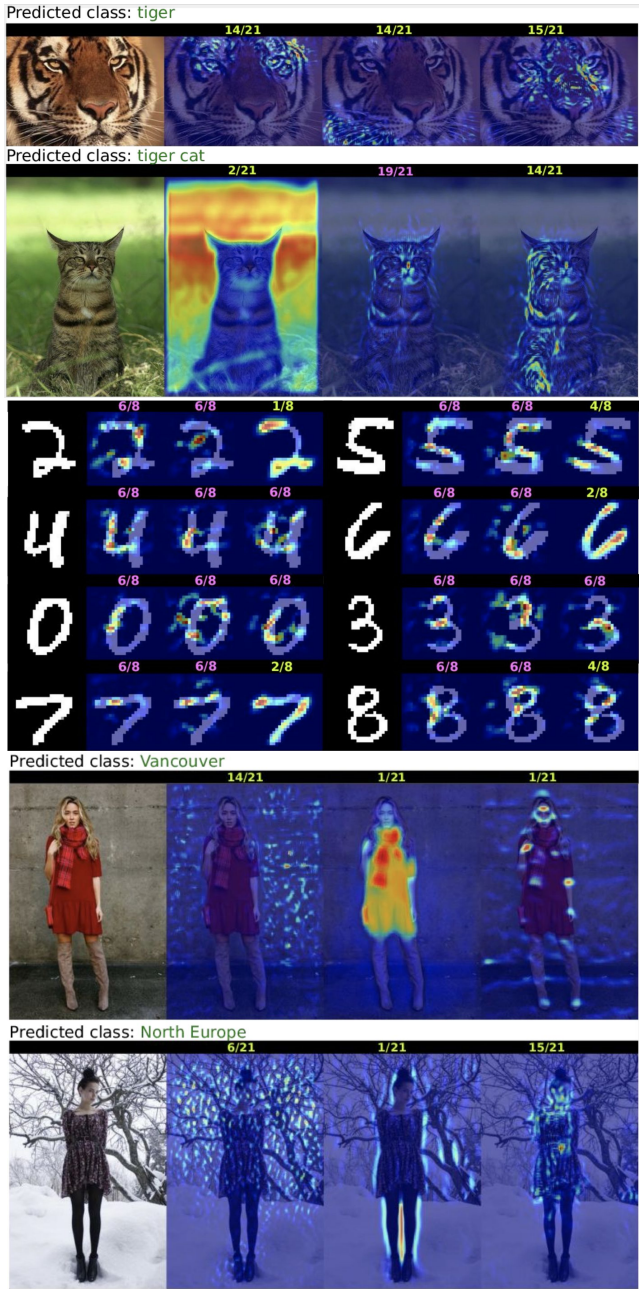

Figure 3.6: Our visual explanations. We accompany the predicted class label with our heatmaps indicating the pixel locations, associated to the discovered compositional representation, that contributed to the prediction. These representations may come from the object itself as well as from its context. See how for MNIST, some features support the existence of gaps, as to avoid confusion with another class. On top of each heatmap we indicate the number of the layer where the features come from. The layer type is color-coded, i.e., convolutional (green) and fully connected (pink).

answer the question explored in [169] and why the computer outperforms the surveyed participants. It shows that the model effectively exploits human-related representations (legs clothing, hair length/color, clothing color) as well as background-related representations, mainly covered by color/gradients and texture patterns. In the visual explanations provided by our method we can see that the model effectively uses this type of representations to reach its decision.

Finally, in Fig. 3.6 we show some examples of the visual explanations produced by our method. We aggregate the predicted class label with our heatmap visualizations indicating the pixel locations, associated to the relevant representations, that contributed to the prediction. For the case of the ILSVRC'12 and Fashion144k examples, we notice that the relevant representations come from the object itself as well as from its context. For the case of the MNIST examples, in addition to the representations firing on the object, there are representations that support the existence of a gap (background), as to emphasize that the object is not filled there and avoid confusion with another class. For example, see for class 2 how it speaks against 0. Likewise, 6 goes against 4.

3.4.2 Visual Feedback Quality

In this section, we assess the visual quality of the visual explanations generated by our method. In Fig. 3.7, we compare our visualizations with upsampled activation maps from internal layers ([12, 205]) and the output of DeconvNet with guided-backpropagation ([157]). We show these visualizations for different layers/filters throughout the network.

A quick inspection reveals that our method to attenuate the grid-like artifacts introduced by Deconvnet methods (see Sec 3.3.3) indeed produces noticeable improvements for lower layers. See Fig. 3.3 for additional examples presenting this difference at lower layers. Likewise, for the case of higher layers (Fig. 3.7), the proposed method provides more precise visualizations when compared to upsampled activation maps. In fact, the rough output produced by the activation maps at higher layers has a saliency-like behavior that gives the impression that the network is focusing on a larger region of the image. This could be a possible attribution to why in earlier works [206], manual inspection of network activations suggested that the network was focusing on "semantic" parts. Please see Gonzalez and Garcia et al. [56] for an in-depth discussion of this observation. Finally, for the case of FC layers, using upsampled activation maps is not applicable. In addition, to quantitatively measure the quality of our heatmaps we perform a box-occlusion study [191]. Given a specific heatmap, we occlude the original image with patches sampled from the distribution defined

Method	**single-6c**	**double-12c**
Upsam. Act.	16.8±2.63	16.1±1.30
Deconv+GB, [157]	21.3±0.77	21.9±0.72
Grad-CAM, [36]	17.5±0.25	14.8±0.16
Guided Grad-CAM, [36]	19.9±0.61	19.4±0.34
Grad-CAM++, [24]	15.6±0.57	14.6±0.12
Guided Grad-CAM++, [24]	19.6±0.65	19.7±0.27
Ours	**22.5±0.82**	**23.2±0.60**

Table 3.1: Area under the IoU curve (in percentages) on an8Flower over *5-folds*.

by the heatmap. We measure changes in performance as we gradually increase the number of patches up to covering the 30% most relevant part of the image. Here our method reaches a mean difference in prediction confidence of 2% w.r.t. to Springenberg et al. [157]. This suggests that our method is able to maintain focus on relevant class features while producing detailed heatmaps with better visual quality.

3.4.3 Measuring Visual Explanation Accuracy

We generate two synthetic datasets, *an8Flower-single-6c* and *an8Flower-double-12c*, with 6 and 12 classes respectively. In the former, a fixed single part of the object is allowed to change color. This color defines the classes of interest. In the latter, a combination of color and the part on which it is located defines the discriminative feature. After defining these features, we generate masks that overlap with the discriminative regions (Fig. 3.8 (top)). Then, we threshold the heatmaps at given values and measure the pixel-level intersection over union (IoU) of a model explanation (produced by the method to be evaluated) w.r.t. these masks. We test a similar model as for the MNIST dataset (Sec. 3.4.1) trained on each variant of the *an8Flower* dataset. In Table 3.4.3 we report 5-fold cross-validation performance of the proposed feature selection method using three different means (Upsamp. Act. Maps, Deconv+GB [157] and ours heatmap variant) and other state-of-the-art methods to generate visual explanations.

We can notice in Fig. 3.8 (bottom) that our method effectively identifies the pre-defined discriminative regions regardless of whether they are related to color and/or shape. Likewise, Fig. 3.8 (top) shows that our explanations accurately highlight these features and that they have a better balance between level of detail and coverage than those produced by existing methods. The quantitative results (Table 3.4.3) show that our method has a higher mean IoU of the discriminative features when compared to existing methods. However, it should

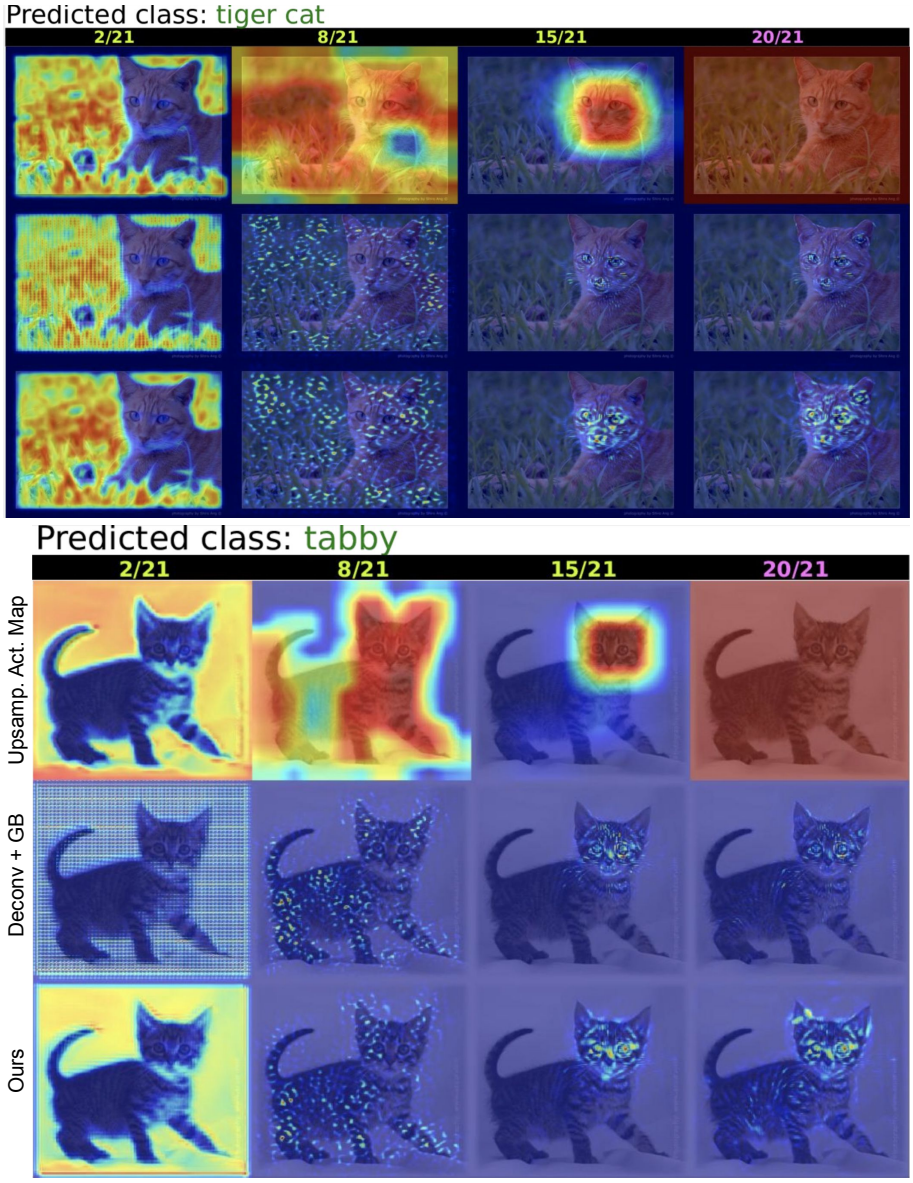

Figure 3.7: Pixel effect visualization for different methods. Note how for lower layers (8/21), our method attenuates the grid-like artifacts introduced by Deconvnet methods. For higher layers (15/21), our method provides a more precise visualization when compared to upsampled activation maps. For the case of FC layers (20/21), using upsampled activation maps is not applicable.

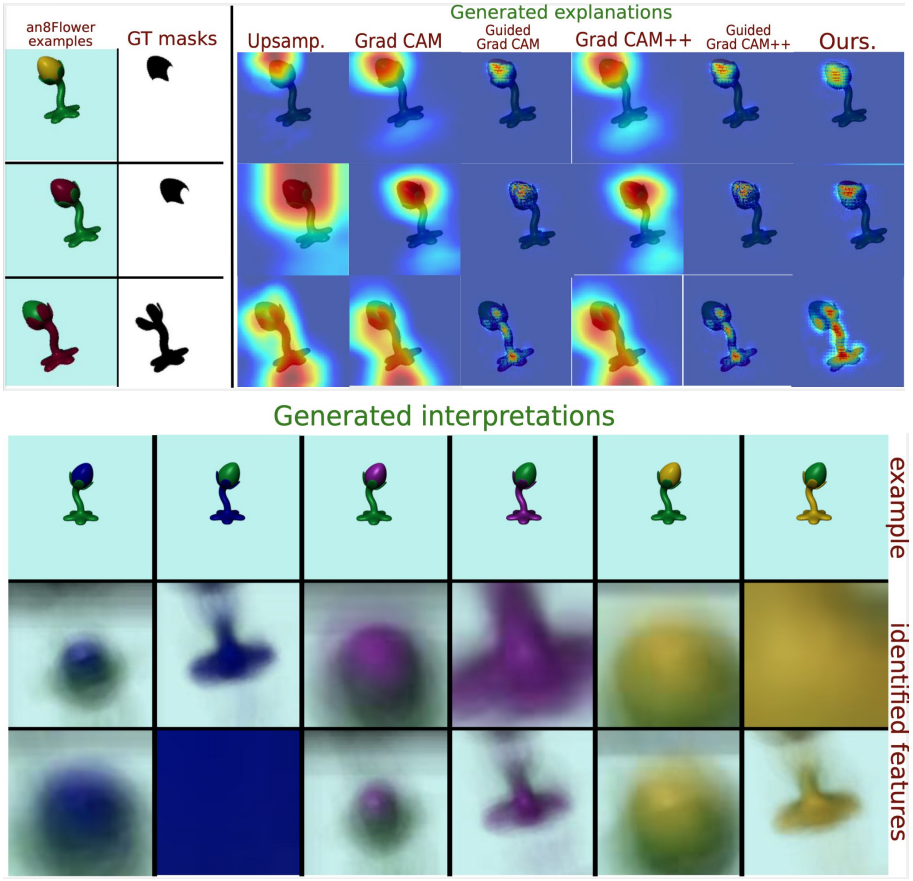

Figure 3.8: Top left: Examples and GT-masks from the proposed *an8FLower* dataset. Top right: Comparison of generated visual explanations. Bottom: Examples of the generated visual interpretations.

be noted that, different from the compared methods, our method involves an additional process, i.e., feature selection via μ-lasso, at training time. Moreover, for this process an additional parameter, i.e μ, should be defined (Sec. 3.3.1).

Methods for model explanation/interpretation aim at providing users with insights on what a model has learned and why it makes specific predictions. Putting this together with the observations made in our experiments, there are two points that should be noted. On the one hand, we believe that our objective evaluation should be complemented with simpler user studies. This should ensure that the produced explanations are meaningful to the individuals they aim to serve. On the other hand, our proposed evaluation protocol enables objective quantitative comparison of different methods for visual explanation. As such it is free of the weaknesses of exhaustive user studies and of the complexities that can arise when replicating them.

3.4.4 Checking the Sanity of the Generated Visual Explanations

Beyond the capability of generating accurate visual explanations, recent works ([92, 88, 130]) have stressed the importance of verifying that the generated explanations are indeed relevant to the model and the classes being explained. Towards this goal, we run a similar experiment to that conducted in Nie et al. [130] where the visual explanation produced for a *predicted class* of a given model after observing a given image is compared against those when a *different class* is considered when generating the explanation. A good explanation should be sensible to the class, and thus generate different visualizations. In Fig. 3.9 we show qualitative results obtained by running this experiment on our models trained on the ILSVRC'12 ([145]) and imageNet-cats datasets.

As can be noted in Fig. 3.9, the explanation generated for the predicted class, i.e., *'cat'/'tabby'*, focuses on different regions than those generated for randomly selected classes. This is more remarkable for the case of the imageNet-cats model, which can be considered a fine-grained classification task. In this setting, when changing towards a random class, i.e., *'jaguar, panther, Panthera onca, Felis onca'*, the generated explanations only highlight the features that are common between the random class and the *'tabby'* class depicted in the image. In their work, [130] and [88] found that explanations from DeconvNet and Guided-Backpropagation methods are not performing well in this respect, yielding visualizations that are not determined by the predicted class, but by the filters of the first layer and the edge-like structures in the input images. Although our method relies on DeconvNet and Guided-Backpropagation, our explanations go beyond regions with prominent gradients - see Fig. 3.1, 3.6 &

Figure 3.9: Sensitivity of the proposed method w.r.t. the predicted class. Note how the generated explanation focuses on different regions of the input image (left) when explaining different classes. We see a similar trend on explanations generated from the models trained on the imageNet (center) and imageNet-cats (right) datasets.

3.8. In fact, in classes where color is a discriminative feature, uniform regions are highlighted. This different result can be understood since, in our method, DeconvNet with Guided-Backpropagation is merely used as a means to highlight the image regions that justify the identified relevant features, not the predicted classes themselves. If a better, more robust or principled visualization method is proposed in the future by the community, we could use that as well.

3.4.5 Potential Users

In this section, we briefly discuss the potential users of our proposed method. On the one hand, scientists, with related technical knowledge, can benefit from or improve our proposed method since they are familiar with DNN architectures and their inner-working mechanisms. Some advanced feature visualization methods can also be developed and applied to visualize the identified layer-filter pairs (p, q) and further improve our current results. The generated heatmaps, on the other hand, can provide average people (lay users) with an intuitive idea about the regions of the input that are important for the prediction. Consequently, to some extent, This helps people understand DNNs (so-called black boxes) better.

3.5 Current Research Trends

In this section we present some recent research work that works along DNN explanation and interpretation that appeared after we published our work. Rafegas et al. [136] focuses on understanding the internal representations of CNN. Different from our methods, where we consider the top 100 image patches where the identified relevant filters have a high response, [136] suggests

to weighted average of them. In addition, the authors propose a framework to associate selectivity indices to individual neurons, such as color selectivity index and class selectivity index, where the neurons have strong responses for images with a certain color or class. Rather than focus on the individual sample features, Ghorbani et al. [54] propose a concept-based explanation method, where the identified concepts are human-understandable and can apply for the entire dataset. In order to identify the dataset-wise concepts, the proposed method processes image segments of a group of images from the same class and clusters the similar segments based on the feature space distance. Testing with Concept Activation Vectors (TCAV) [91], a concept-based importance score, is used to rank the identified concepts based on the importance to the model prediction. In the context of the DNN visualization method, Rebuffi et al. [139] proposes a novel saliency method called NormGrad that is based on the spatial contribution of gradients of convolutional weights. NormGrad uses Frobenius norm to transform these spatial contributions into saliency maps, while GradCAM and Gradient backprogation utilize filtering and take the maximum absolute value of the vector, respectively. NormGrad can be integrated in our proposed method since it is a visualization method that can visualize the relevant filters once they are identified.

3.6 Conclusion

In this chapter, we succeed in discovering the component representations which are important to the task addressed by the DNN, even if the model encodes the entire image in the training phase. Particularly, we propose a method to enrich the prediction made by DNNs by indicating the visual features that contributed to such prediction. Our method identifies the features that are relevant for the task addressed by the DNN. It also allows *interpretation* of these representations by the generation of average feature-wise visualizations. In addition, we propose a method to attenuate the artifacts introduced by strided operations in visualizations made by Deconvnet-based methods. This empowers our method with richer visual feedback with pixel-level precision without requiring additional annotations for supervision. Finally, we have proposed a novel dataset designed for the objective evaluation of methods for explanation of DNNs.

Chapter 4

Unpaired Shape Translation

In the previous chapter we present a method that automatically identifies the relevant filters in a pre-trained DNN and visualizes these important hidden-encoded component representations of an image. In this chapter, we try to deal with the component features in a complementary manner, i.e. disentangle two component features of input images in the training phase and synthesize a novel image with different shape by combining the disentangled representations.

Images can be regarded as being composed of style and shape features. Standard image-to-image style transfer methods aim at transferring the style features (i.e. appearance) of an image as a whole while preserving the shape features (i.e. structure) of its content. In this work, we propose a method to tackle the unpaired geometric image-to-image translation task, a complementary problem where the goal is to translate the shape of an object as depicted in different domains while preserving its appearance. Specifically, we focus on one canonical domain with standard shape (e.g. a clothing item) and one contextualized domain (e.g. a clothed person). Our method can be learned in an unpaired manner, i.e., without the need of paired training images, and can perform all shape transfer steps within a single model without additional post-processing. In addition, our model can produce a compact low-dimension (e.g. 8-dimensional) style feature representation which is useful to the item retrieval task. Extensive evaluations on the VITON, CMU-Multi-PIE and our own FashionStyle datasets demonstrate the capabilities of the proposed method.

The presented content is based on the following paper:

- K. Wang, L. Ma, J. Oramas M, L. Van Gool, T. Tuytelaars, Unpaired Image Shape Translation Across Fashion Data. International Conference

on Image Processing (ICIP) 2020.

- K. Wang*, L. Ma*, J. Oramas M, L. Van Gool, T. Tuytelaars, Unpaired Shape Transforming for Image Translation and Cross-domain Retrieval. submitted to Computer Vision and Image Understanding.

4.1 Introduction

Image-to-image translation (I2I) refers to the process of generating a novel image, which is similar to the original input image yet different in some aspects. Typically, the input and output images belong to different *domains*, with images in the same domain sharing a common characteristic, e.g. going from photographs to paintings [85], from greyscale to color images [18], or from virtual (synthetic) to real images [204]. Apart from direct applications [98], I2I has proven valuable as a tool for data augmentation [50] or to learn a representation for cross-domain image retrieval [66].

We assume that images are composed by style and shape features. Traditionally, each image domain is characterized by a different style (i.e. appearance), and I2I is therefore sometimes referred to as style transfer [85]. While the translation process may drastically change the appearance or style of the input image, the image structures are to be preserved, i.e. both input and output should represent the same objects and scene. Moreover, in most works, also the image geometry, i.e. the shape of the objects and the global image composition, is preserved. We refer to this as the image *shape*.

Most methods for I2I build on top of Generative Adversarial Networks (GANs) [57, 135, 124, 7] and are data-driven. They learn a translation model from example images of the two domains. While most methods require paired examples [82, 185, 203], some recent methods do not [135, 207, 5]. To constrain the complexity of the problem, the training data is often restricted to a specific setting, e.g. close-ups of faces [78, 203], people [120, 121], traffic scenes [108], etc.

In contrast to the traditional setting [121, 185, 203], we focus on the challenge where input and output do *not* belong to domains that share the same geometrical information. Instead, we work with one object-centric domain with standard shape and one that is more contextualized with large shape variation (using a reference image to provide the right context). For instance, we go from a single piece of clothing to a person wearing that same item; or from a frontal face crop to a wider shot with arbitrary viewpoint of that same person (see Fig. 4.1 & 4.12). In other words, we want to transfer the shape feature while

Figure 4.1: Translating a clothing image from a canonical domain of clothing items to a contextualized domain of clothed persons (top), and vice versa (bottom). For both try-on (top) and take-off (bottom) tasks the appearance of the item remains constant while its shape is determined by the pose of the person wearing it.

keep the style feature of an image. This setting is significantly more challenging, as the image geometry changes. At the same time, the image semantics (e.g. the clothing pattern or face identity) should be preserved. Analogous to the term style transfer, we refer to this as *shape transfer*. While a couple of recent works have looked into this setting [121, 185, 203], to the best of our knowledge we are the first to propose a solution that does *not* require paired data, across different domains, for model training. This is important, as collecting paired data is cumbersome or even impossible. Either way, it limits the amount of data that can be used for training, while access to large amounts of data is crucial for the quality of the results. Methods working with unpaired training data have been proposed for style transfer [80, 207], relying on low-level local

transformations. However, these are not suited for the more challenging shape transfer setting, as clearly illustrated in Fig. 4.2.

Translating shapes in a unpaired way is an unsolved task that is of interest for several reasons. First, it can be considered an alternative formulation of the novel-view synthesis problem, in the 2D image space, using only a single image as input. Second, shape translation can recover missing/occluded characteristics of an object instance which can help other tasks, such as recognition or tracking.

The main contributions of this chapter are four-fold:

- We address the unpaired shape translation problem. To the best of our knowledge, we are the first addressing it in a cross-domain setting from an unpaired perspective when the corresponding paper was submitted.
- We address this problem via the proposed Unpaired Shape Transforming (UST) method, which does not need any paired data or refinement post-processing. In a single pass, along one direction, an object with standard shape is transformed to a contextualized domain with arbitrary shape, and vice versa along the other direction.
- The proposed method achieves a one-to-many mapping by utilizing context and structure information guidance.
- We show the potential of the features learned by our model on the cross-domain item retrieval task.

4.2 Related Work

Isola et al. [82] first formulate the image-to-image translation problem with a conditional GAN model which learns a mapping from the source image distribution to the output image distribution using a U-Net neural network in an adversarial way. Zhu et al. [207] propose cycle-consistency to solve the I2I problem with unpaired data. This enables a lot of applications since it is usually expensive or even impossible to collect paired data for many tasks. Liu et al. [110] assume that there exists a shared latent space for the two related domains and propose a weights-sharing based framework to enforce this constraint. These methods learn a one-to-one mapping function, i.e. the input image is mapped to a deterministic output image. [80, 3, 119, 101] propose unpaired multimodal methods which either sample multiple styles from a Gaussian space or capture the styles from exemplar images to generate diverse outputs.

Figure 4.2: Comparisons with CycleGAN [207] and MUNIT [80] for try-on (left) and take-off (right) on FashionStyle dataset.

All the above methods focus on appearance transfer where the content depicted in the input and output images has an aligned geometric structure. [203, 78, 120, 121, 10, 137] aim at the case when the geometry itself is to be transferred. However, these methods focus on within-domain tasks (e.g. person-to-person and face-to-face), which depicts reduced variability when compared to its cross-domain counterpart (e.g. person-to-clothing). Focused on images of clothing items, Yoo et al. [185] propose one of the first methods addressing cross-domain pixel-level translation. Their method semantically transfers a natural image depicting a person (source domain) to a clothing-item image corresponding to the clothing worn by that person on the upper body (target domain), and vice versa. Recently, Han et al. [69] and Wang et al. [167] propose two-stage warping-based methods aimed at virtual try-on of clothing items. These methods focus on learning a thin-plate spline (TPS) operation to transfer the pixel information directly. They rely on paired data to learn to transfer the shape in a first stage and then refine it in a second stage. In contrast, we propose a more general method that utilizes the context and shape guidance to perform translation across different domains without any paired data. In addition, different from previous works which divide the translation process into multiple stages, our method is able to handle the full appearance-preserving translation, in both directions, within a single model.

Outside of the I2I literature, Jaderberg et al. [83] proposes a spatial transformer network (STN) which also aims at object-level transformations. Different from our method, which learns the plausible transformations from data and allows for user-suggested transformations through the use of "desired" target images, STNs start from a predefined set of possible transformations. In addition, STNs apply the same transformation to every pixel. Differently, our method implicitly allows deformable objects since different pixel-level transformations are possible as depicted in the training data. Finally, STNs makes no distinction between

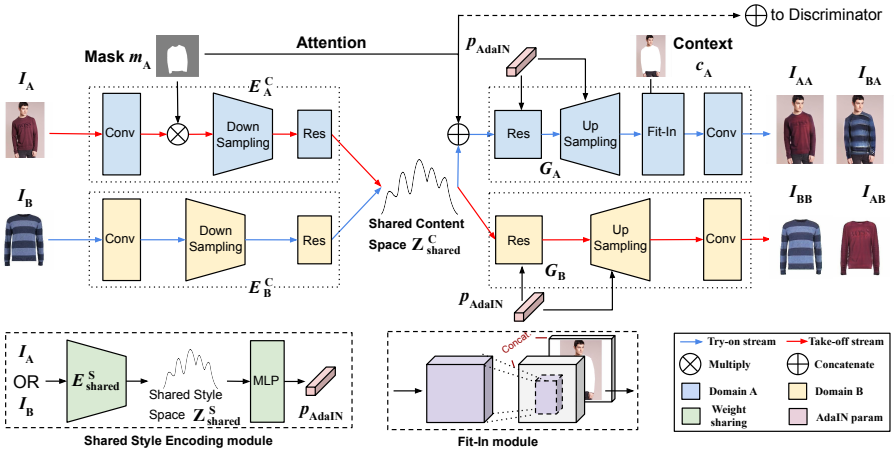

Figure 4.3: Our proposed Unpaired Shape Transforming (UST) method. The try-on and take-off streams are trained jointly with shared style/content space constraints. To learn the one-to-many mapping in the try-on stream, the context information is utilized in the Fit-in module to constrain the output to be deterministic. Besides, an attention mechanism is applied to encourage the network to focus on the object. To learn the many-to-one mapping in the take-off stream, adversarial learning is adopted directly.

content and style information. Along the same line, recently Lin et al. [106] and Lee et al. [100] proposed methods focused on the image composition task. [106] learns how to add object segments with the correct shape in the semantic space. This is more constrained than our instance-level transfer in the RGB space. Moreover, it requires expensive supervision in the form of pixel-wise labels and instance-level contours. [100] operates in the RGB space. However, it can only handle homography transformations (in rigid objects) related to changes in viewpoint and scale. This is un-applicable on articulated/deformable objects and unsuitable to handle self-occlusions.

4.3 Proposed Method

In this section, we describe our model using the clothing try-on / take-off as an example. It should be noted though that our method can also be applied to other types of data, such as the face try-on / take-off illustrated in Sec. 4.4. Our goal is to transfer the shape information while keeping the appearance information, all trained without access to paired data. For this, we propose the

asymmetric two-stream model shown in Fig. 4.3. The asymmetry reflects the fact that one of the two domains (domain B) is object-focused (e.g. catalog images of clothing items) while the other one (domain A) shows the objects in context (e.g. pictures of clothed persons). In the one-to-many try-on stream (blue arrows), we transfer from the object-focused to the contextualized setting. This requires synthesizing a new image, where the shape of the object is first altered after which it is merged seamlessly with the provided context (in our setting, a segmented image of a person wearing a different piece of clothing). During this process, the color, texture and anything else specific to the object instance is to be preserved. In the many-to-one take-off stream (red arrows), our goal is to synthesize the clothing catalog image in a standard frontal view starting from a natural person image with varying pose.

Here, we use I_A and I_B to refer to images from domain A and domain B respectively. I_{AB} refers to images transferred from domain A to domain B, and vice versa for I_{BA}.

4.3.1 Assumptions

In previous works [185, 69], the try-on and take-off tasks are solved in a supervised way, respectively. Here, we solve both tasks in one model using unpaired data based on shared-latent space and context-structure constraints.
Cycle-consistency. For the unpaired setting, translating an image can be formulated as a distribution matching problem, i.e. learning a mapping that makes the translated result $I_{\mathrm{AB}} = F_{\mathrm{A}\to\mathrm{B}}(I_{\mathrm{A}})$ look like images sampled from domain B's distribution. Such a mapping function F can be learned with adversarial learning. However, the learning procedure is highly under-constrained and ill-posed, since there could be multiple mappings from domain A to B. To reduce the space of possible mapping functions, we utilize the cycle-consistency [207], i.e. $I_{\mathrm{A}} = F_{\mathrm{B}\to\mathrm{A}}(F_{\mathrm{A}\to\mathrm{B}}(I_{\mathrm{A}}))$ and vice versa, this constrains and stabilizes the adversarial training.

Shared-latent space constraint. Similar to [80, 119, 101], we decompose the latent space into a content space and a style space. Different from previous works, we have two assumptions: 1) content space constraint, i.e. the content space *can be shared* by the two domains; 2) style space constraint, i.e. images from the two domains *do share* the same style space. We use $\mathrm{Z}_{\mathrm{A}}^{\mathrm{C}}$ and $\mathrm{Z}_{\mathrm{B}}^{\mathrm{C}}$ to denote the content space of domains A and B, respectively. We assume $\mathrm{Z}_{\mathrm{A}}^{\mathrm{C}}$ and $\mathrm{Z}_{\mathrm{B}}^{\mathrm{C}}$ are both embedded in a larger latent space $\mathrm{Z}_{\mathrm{shared}}^{C}$. Symbols $\mathrm{Z}_{\mathrm{A}}^{\mathrm{S}}$ and $\mathrm{Z}_{\mathrm{B}}^{\mathrm{S}}$ denote the style space of domain A and B, respectively. Note that we assume $\mathrm{Z}_{\mathrm{A}}^{\mathrm{S}}$ and $\mathrm{Z}_{\mathrm{B}}^{S}$ are the same space, which is a stronger constraint.

To achieve the content space constraint, we use two encoders $E_{\mathrm{A}}^{\mathrm{C}}$ and $E_{\mathrm{B}}^{\mathrm{C}}$ to encode images from domain A and B, respectively. Then, we use a latent content code reconstruction loss to enforce the latent content reconstruction, similar to Huang et al. [80] and Lee et al. [101]. To achieve the style space constraint, we utilize both the weight-sharing technique [110] and the latent style code reconstruction loss (see Fig. 4.3).

Context constraint. Although the above shared-latent space constraints enable the unpaired I2I and work well for style transfer tasks [110, 119, 101], it is not enough for geometry transfer when the output is multi-modal (i.e. multiple possible outputs). To address this, Yoo et al. [185] proposed triplet adversarial learning with paired data. However, for the unpaired setting, the adversarial learning on its own is too weak (see Fig. 4.2). Here, we propose to use context information guidance to constrain the output to be deterministic, i.e. decompose the one-to-many mapping into one-to-one mappings. In particular, for the try-on stream, we propose a Fit-in module which combines the feature maps with the context information. As to the take-off stream, we assume the output is unimodal and directly use the adversarial learning to learn the deterministic many-to-one mapping.

Attention. Since domain A is contextualized, we need to constrain the network to focus on the object instead of the background. Therefore, we introduce the attention mechanism in both generator and discriminator, i.e. concatenating the shape mask m_A with the inputs before G_A and the discriminator D_A (see Fig. 4.3).

4.3.2 Network architecture

The model can be divided into several sub-networks. For the content encoder $E_{\mathrm{A}}^{\mathrm{C}}$ and $E_{\mathrm{B}}^{\mathrm{C}}$, we use a convolution block and several down-sampling layers followed by several residual blocks. The decoders G_{A} and G_{B} are symmetric with the encoding part except for the Fit-in module which is key to learn the one-to-many mapping and the shape mask attention that helps preserving the appearance. The Fit-in module is a simple convolution block which receives the generated feature map and the context information of the desired target shape. The shared style encoding module contains a style encoder $E_{\mathrm{shared}}^{\mathrm{S}}$ and a multilayer perceptron (MLP). It encodes the style information of both domains.

Try-on stream. The catalog image I_{B} first passes through the domain B content encoder $E_{\mathrm{B}}^{\mathrm{C}}$ producing the content code $z_{\mathrm{B}}^{\mathrm{C}}$ in the shared content space $\mathrm{Z}_{shared}^{\mathrm{C}}$. In parallel, I_{B} is also encoded into a style code $z_{\mathrm{B}}^{\mathrm{S}}$ in the shared style space $\mathrm{Z}_{\mathrm{shared}}^{\mathrm{S}}$ by the shared style encoder E_{shared}^{S}. To combine

the content and style information in the decoder, we use adaptive instance normalization (AdaIN) [79] layers for all residual and up-sampling blocks. The AdaIN parameters p_{AdaIN} are dynamically computed by a multilayer perceptron from the style code z_{B}^{S} to ensure the generated person image I_{BA} has the same style as I_{B}.

$$\text{AdaIN}(z, \gamma, \beta) = \gamma \frac{(z - \mu(z))}{\delta(z)} + \beta, \tag{4.1}$$

where z is the activation of the previous convolution layer. μ and δ are the mean and standard deviation computed per channel. Parameters γ and β are the output of the MLP of the shared style encoding module.

During decoding, the content code z_{B}^{C} concatenated with the shape mask m_{A} are fed to the decoder G_{A}. There the content and style are fused by AdaIN and then fed to the Fit-in module. We apply AdaIN in both the residual blocks and up-sampling layers. This helps stabilize and speed up the convergence during training, and also helps preserve appearance better. This is due to the fact that AdaIN can be treated as a skip-connection between the encoder and decoder to alleviate the exploding and diminishing gradient problems. The Fit-in module is designed to enforce the context information constraint. We first obtain the bounding box of the mask from the context image. Then, we resize and align the up-sampled feature maps to this bounding box. Finally, this ouput is concatenated with the context image. The main goal of this design is to integrate the context information which helps the deterministic shape transform. The final try-on image I_{BA} is generated after the last convolution block.

In addition, inspired by Isola et al. [82], we introduce an attention mechanism to both generator and discriminator. We concatenate the mask m_{A} with the content code z_{B}^{C} before the generator G_A and concatenate the mask m_{A} with the generated image I_{BA} before the discriminator D_A, respectively. This simple but effective attention operation encourages the network to focus on the generated clothing instead of the context part. This improves the results, especially when the objects to be translated have a highly variable scale/location within the images.

Take-off stream. For the take-off stream, the clothed person image I_{A} first passes through a convolution block and then gets multiplied with the clothing mask m_{A} in order to exclude the background and skin information. Similar to the try-on stream, the masked feature maps are then encoded into a content code z_{A}^{C} in the shared content space $\text{Z}_{\text{shared}}^{\text{C}}$.

For the decoding part, the only difference with the try-on stream is that there is no "Fit-in" module or mask attention. The final take-off catalog image I_{AB} is

generated by decoding $z_{\mathrm{A}}^{\mathrm{C}}$, through the decoder G_{B} with AdaIN residual blocks, up-sampling blocks and convolution blocks.

4.3.3 Learning

In this section, we only describe A→B translation for simplicity and clarity. The B→A is learned in a similar fashion. We denote the content latent code as $\mathrm{z}_{\mathrm{A}}^{\mathrm{C}}{=}E_{\mathrm{A}}^{\mathrm{C}}(I_{\mathrm{A}})$, style latent code as $\mathrm{z}_{\mathrm{A}}^{\mathrm{S}}{=}E_{\mathrm{shared}}^{\mathrm{S}}(I_{\mathrm{A}})$, within domain reconstruction output as $I_{\mathrm{AA}}{=}G_{\mathrm{A}}(\mathrm{z}_{\mathrm{A}}^{\mathrm{C}},\mathrm{z}_{\mathrm{A}}^{\mathrm{S}})$, cross domain translation output as $I_{\mathrm{AB}}{=}G_{\mathrm{B}}(\mathrm{z}_{\mathrm{A}}^{\mathrm{C}},\mathrm{z}_{\mathrm{A}}^{\mathrm{S}})$. Our loss function contains terms for the bidirectional reconstruction loss, cycle-consistency loss and adversarial loss [80, 101]. Besides, we also use a composed perceptual loss to preserve the appearance information across domains, and a symmetry loss capturing some extra domain knowledge [78, 203].

Bidirectional reconstruction loss ($L_{LaR}^{I_A}$, $L_{SeR}^{I_A}$). This loss consists of the feature level latent reconstruction loss $\mathcal{L}_{\mathrm{LaR}}$ and the pixel level image self-reconstruction loss $\mathcal{L}_{\mathrm{SeR}}$. The former contains both content and style code reconstructions. The whole bidirectional reconstruction loss encourages the network to learn encoder - decoder pairs that are inverses of one another and also stabilizes the training.

$$\begin{aligned}\mathcal{L}_{\mathrm{LaR}}^{I_{\mathrm{A}}} =& \mathbb{E}_{I_{\mathrm{AB}},z_{\mathrm{A}}^{\mathrm{C}}}\left[\|E_{\mathrm{B}}^{\mathrm{C}}(I_{\mathrm{AB}})-\mathrm{z}_{\mathrm{A}}^{\mathrm{C}}\|_1\right]\\ &+\mathbb{E}_{I_{\mathrm{AB}},z_{\mathrm{A}}^{\mathrm{S}}}\left[\|E_{\mathrm{shared}}^{\mathrm{S}}(I_{\mathrm{AB}})-\mathrm{z}_{\mathrm{A}}^{\mathrm{S}}\|_1\right] \qquad (4.2)\\ \mathcal{L}_{\mathrm{SeR}}^{I_{\mathrm{A}}} =& \mathbb{E}_{I_{\mathrm{A}}}\left[\|I_{\mathrm{AA}}-I_{\mathrm{A}}\|_1\right], \qquad (4.3)\end{aligned}$$

Adversarial loss ($L_{GAN}^{I_A}$). To make the translated image look domain realistic, we use an adversarial loss to match the domain distribution. For the A→B translation, the domain B discriminator D_B tries to distinguish the generated fake images with the real domain B images, while the generator G_B will try to generate domain B realistic images.

$$\mathcal{L}_{GAN}^{I_{\mathrm{A}}} = \mathbb{E}_{I_{\mathrm{B}}}\left[\log D_{\mathrm{B}}(I_{\mathrm{B}})\right] + \mathbb{E}_{I_{\mathrm{AB}}}\left[\log\left(1-D_{\mathrm{B}}(I_{\mathrm{AB}})\right)\right] \qquad (4.4)$$

Cycle-consistency loss ($L_{CC}^{I_A}$). To enable unpaired translation, the cycle-consistency loss [207] is applied to stabilize the adversarial training.

$$\mathcal{L}_{\mathrm{CC}}^{I_{\mathrm{A}}} = \mathbb{E}_{I_{\mathrm{AB}},I_{\mathrm{A}}}\left[\|G_{\mathrm{A}}(E_{\mathrm{B}}^{\mathrm{C}}(I_{\mathrm{AB}}),E_{\mathrm{shared}}^{\mathrm{S}}(I_{\mathrm{AB}}))-I_{\mathrm{A}}\|_1\right] \qquad (4.5)$$

Perceptual loss ($L_P^{I_A}$). To preserve the appearance information, we apply a composed perceptual loss.

$$\begin{aligned}\mathcal{L}_{\mathrm{P}}^{I_{\mathrm{A}}} =&(\mathbb{E}_{I_{\mathrm{AA}},I_{\mathrm{A}}}\big[\|\Phi(I_{\mathrm{AA}})-\Phi(I_{\mathrm{A}})\|_2^2\big])\\ &+(\mathbb{E}_{I_{\mathrm{AB}},I_{\mathrm{B}}}\big[\|\Phi(I_{\mathrm{AB}})-\Phi(I_{\mathrm{A'}})\|_2^2\big])\\ &+\lambda\ \mathbb{E}_{I_{\mathrm{AB}},I_{\mathrm{B}}}\big[\|Gram(I_{\mathrm{AB}})-Gram(I_{\mathrm{A'}})\|_1\big],\end{aligned} \tag{4.6}$$

where $I_{A'}$ is the Region of Interest (RoI) of I_A. For clothing items, it is the segmented clothing region. For the face experiments, it is the facial region (without context information). Φ is a network trained on external data, whose representation can capture image similarity. Similar to Gatys et al. [51] and Johnson et al. [87], we use the first convolution layer of all five blocks in VGG16 [153] to extract the feature maps to calculate the Gram matrix which contains non-localized style information. λ is the corresponding loss weight.

Symmetry loss ($L_{Sym}^{I_A}$). To utilize the inherent prior knowledge of clothing and human faces, we apply a symmetry loss [78, 203] to the take-off stream.

$$\mathcal{L}_{\mathrm{Sym}}^{I_{\mathrm{A}}} = \mathbb{E}_{I_{\mathrm{AB}}}\big[\frac{1}{W/2\times H}\sum_{w=1}^{W/2}\sum_{h=1}^{H}\|I_{\mathrm{AB}}^{w,h}-I_{\mathrm{AB}}^{W-(w-1),h}\|_1\big], \tag{4.7}$$

According to custom, W and w refer to the width of the image and the horizontal coordinates of each pixel here. Similarly, H and h denote the height of the image and the vertical coordinates of each pixel, and $I_{\mathrm{AB}}^{w,h}$ refers to a pixel in the transferred image I_{AB}.

Total loss. Our model, including encoders, decoders and discriminators, is optimized jointly. The full objective is as follows,

$$\begin{aligned}&\min_{\substack{E_{\mathrm{A}}^{\mathrm{C}},E_{\mathrm{B}}^{\mathrm{C}},E_{\mathrm{shared}}^{\mathrm{S}},\\ G_{\mathrm{A}},G_{\mathrm{B}}}}\ \max_{D_{\mathrm{A}},D_{\mathrm{B}}}\ \mathcal{L}(E_{\mathrm{A}}^{\mathrm{C}},E_{\mathrm{B}}^{\mathrm{C}},E_{\mathrm{shared}}^{\mathrm{S}},G_{\mathrm{A}},G_{\mathrm{B}},D_{\mathrm{A}},D_{\mathrm{B}})\\ &=\mathcal{L}_{\mathrm{GAN}}^{I_{\mathrm{A}}}+\mathcal{L}_{\mathrm{GAN}}^{I_{\mathrm{B}}}+\lambda_{\mathrm{CC}}(\mathcal{L}_{\mathrm{CC}}^{I_{\mathrm{A}}}+\mathcal{L}_{\mathrm{CC}}^{I_{\mathrm{B}}})+\lambda_{\mathrm{SeR}}(\mathcal{L}_{\mathrm{SeR}}^{I_{\mathrm{A}}}+\mathcal{L}_{\mathrm{SeR}}^{I_{\mathrm{B}}})\\ &+\lambda_{\mathrm{LaR}}(\mathcal{L}_{\mathrm{LaR}}^{I_{\mathrm{A}}}+\mathcal{L}_{\mathrm{LaR}}^{I_{\mathrm{B}}})+\lambda_{\mathrm{P}}(\mathcal{L}_{\mathrm{P}}^{I_{\mathrm{A}}}+\mathcal{L}_{\mathrm{P}}^{I_{\mathrm{B}}})+\lambda_{\mathrm{Sym}}\mathcal{L}_{\mathrm{Sym}}^{I_{\mathrm{A}}}.\end{aligned} \tag{4.8}$$

where λ_{CC}, λ_{SeR}, λ_{LaR}, λ_{P} and λ_{Sym} are loss weights for different loss terms.

4.4 Evaluation

We evaluate our method on both clothing try-on / take-off and face try-on / take-off tasks. We perform an ablation study on our own FashionStyle dataset.

Then, the full model results on VITON and MultiPIE datasets are reported. Finally, we assess the potential of the learned style/appearance representation for clothing item retrieval across domains.

Datasets. We use three datasets: VITON [69], Fashion-Style and CMU MultiPIE [62]. VITON and FashionStyle are fashion related datasets, see Figs. 4.1, 4.4, 4.9 for some example images. VITON has around 16,000 images for each domain. However, we find that there are plenty of image duplicates with different file names. After cleaning the dataset, there are 7,240 images in each domain left. The FashionStyle dataset, provided by an industrial partner, has 5,230 training images and 1,320 test images of clothed people (domain A), and 2,837 training images and 434 test images of clothing items (domain B). For domain A, FashionStyle has multiple views of the same person wearing the same clothing item. We present results on this dataset for one category, namely pullover/sweater. CMU MultiPIE is a face dataset under pose, illumination and expression changes. Here we focus on a subset of images with neutral illumination and expression, and divide the subset in two domains: 7,254 profile images (domain A) and 920 frontal views (domain B).

Metrics. We use paired images from different domains depicting the same clothing item to quantitatively evaluate the performance our method. For the case of the try-on task we measure the similarity between the ROI of original image (from domain A) and the ROI of a generated version (where its corresponding clothing item has been translated to fit in a masked out version of the image). Thus, we call it *Try-on ROI*. To create this masked image we first run a clothing-item segmentation algorithm [104] that we use to remove the clothing-item originally worn by the person. For the case of the take-off task, given an image from domain A, we measure the similarity of its corresponding clothing item (from domain B) with the generated item. On both cases similarity between images is computed using the SSIM [172] and LPIPS [198] metrics. We report the mean similarity across the whole test set.

For the retrieval task performance is reported in terms of Recall rate given that every query image has only one corresponding image in the database.

Implementation details. The perceptual feature extractors Φ in Eq. 4.6 are LPIPS [198] and Light-CNN [175] networks for clothing translation and face translation, respectively. In all our experiments, we use the Adam [93] optimizer with β_1=0.5 and β_2=0.999. The initial learning rate is set to 2×10^{-6}. Models are trained with a minibatch of size 1 for FashionStyle and VITON, and 2 for the face experiment. We use the segmentation method [104] to get the clothing mask and its bounding box. For faces, we detect the face landmarks using the algorithm proposed by Cao et al. [19] and then connect each point to get the face mask. The shared content code is a tensor whose dimension is determined

Method	Try-on ROI (SSIM/LPIPS-VGG)	Take off (SSIM/LPIPS-VGG)
W/O P. Loss	**66.78** / 27.37	58.96 / 36.62
W/O shared S.E.	66.72 / 27.38	60.33 / 34.94
W/O mask attention	64.63 / 28.13	60.94 / 34.49
W/O Fit-in module	N/A / N/A	60.11 / 37.64
Full model	66.42 / **27.02**	**61.19** / **34.37**
Supervised model	***69.51*** / ***24.14***	***61.54*** / ***32.56***

Table 4.1: Mean SSIM and LPIPS-VGG similarity of each setting from our ablation study. Higher SSIM values and lower LPIPS indicate higher similarity. Both metrics are in the range $[0, 100]$.

by the data. The shared style code is a vector, we use 8/32/128 dimensions in our experiments.

4.4.1 Ablation Study: Clothing try-on / take-off

We conduct a study in order to analyze the importance of four main components of our model. More precisely, the *perceptual loss*, shared style encoder (*Shared S. E.*), *mask attention* and *Fit-in module*, on the FashionStyle dataset. Note that *mask attention* is applied to generator G_A and discriminator D_A. Towards this goal we test different variants of our architecture (Sec. 4.3) where one of these four components has been removed. In addition, we run an experiment using a *supervised model* (paired data). The model architecture is a residual block based on U-net similar to PG^2 [120], but extended to get closer to our model. It is extended by applying our mask multiplication operation after the first convolution block for the supervised take-off experiment. Likewise, we add our Fit-in module for the supervised try-on experiment. We present quantitative results on the translation performance of the try-on / take-off tasks in Table 4.1 for the FashionStyle dataset with related qualitative results presented in Fig. 4.4 and Fig. 4.5. In addition, Fig. 4.6 displays some qualitative comparison w.r.t. the Fit-in module. It clears shows that even the supervised model cannot generate the sharp and clean images without using the Fit-in module.

Discussion. A quick inspection of Table 4.1 reveals that, based on the LPIPS metric, the full model generates images with the highest similarity to the ground-truth on the try-on task among the unpaired variants. Our full model generates sharper and more consistent results than other models, but does not obtain the highest SSIM. This is also observed in person generation and super-resolution papers [120, 87]. The try-on ROI scores of W/O Fit-in module is not applicable

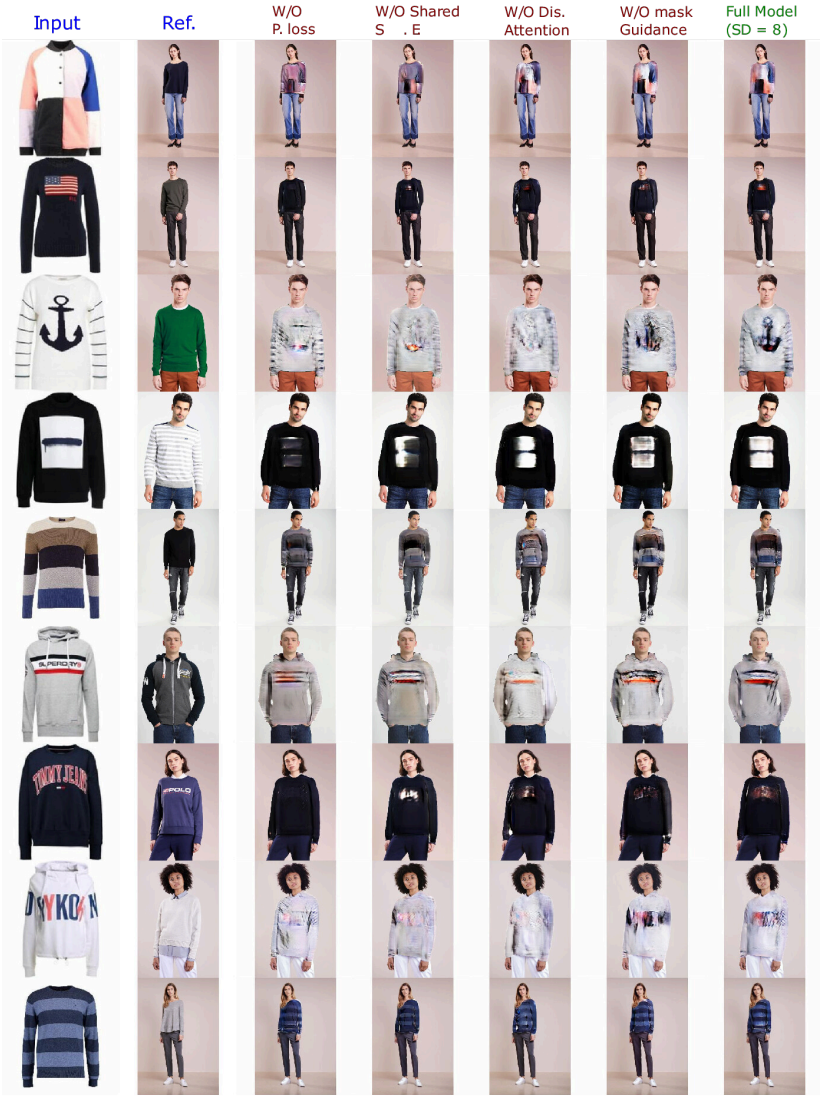

Figure 4.4: Ablation study on the FashionStyle dataset for the try-on results. The first two columns show the input image and the reference ground truth image. The other columns show the generated results of different model settings. Please zoom in for more details.

Figure 4.5: Ablation study on the FashionStyle dataset for the take-off results. The first two columns show the input image and the reference ground truth image. The other columns show the generated results of different model settings. Please zoom in for more details.

Figure 4.6: Ablation study on FashionStyle Dataset w.r.t. the Fit-in Module. It clearly shows without the Fit-in module, even the supervised model can not generate the sharp, clean images.

since without the context information, the network cannot determine the target generated shape, i.e., ROI cannot be determined. This task seems to be affected most when the *mask attention* is dropped. This confirms the relevance of this feature when translating shape from images in this direction (try-on).

For the case of the take-off task, results are completely dominated by the full model among the unpaired variants. However, different from the try-on task, the take-off task is mostly affected by the removal of the perceptual loss(i.e., LPIPS) and Fit-in modules. The Fit-in module is set in the try-on stream, but since the two streams are trained jointly, the take-off stream is indirectly affected by the performance of the try-on stream. Therefore, the take-off result of W/O Fit-in module is the worst. Although this trend is different from the try-on task, it is not surprising given that for the take-off task, the expected shape of the translated image is more constant when compared with that of the try-on task which is directly affected by the person's pose. Moreover, the output of the take-off task is mostly dominated by uniformly-coloured regions, which is a setting in which perceptual similarity metrics, such as LPIPS, excel.

A close inspection of Fig. 4.7 and Fig. 4.8 confirm the trends previously observed. Note how the full model produces the most visually-pleasing result; striking a good balance between shape and level of details on the translated items. The other unpaired variants (except *W/O Fit-in module*) tend to generate blurry results and lose details, e.g. patterns and logos, while maintaining the basic shape and context well. More critical, *W/O Fit-in module* no longer preserves both shape and loses details. Especially, without context information guidance, it is difficult for the model to learn the one-to-many mapping, which results in inconsistent outputs. We have noted that failures are mostly caused by incorrectly estimated masks and heavy occlusion.

It is remarkable that quantitatively speaking (Table 4.1), the performance of our method is comparable to that of the *supervised model*. Moreover, while the *supervised model* is very good at translating logos, our method still has an edge when translating patterns (e.g. squares from the 1^{st} row and stripes from the 9^{th} row of Fig. 4.4), without requiring paired data.

4.4.2 Clothing try-on / take-off on VITON

We complement the results presented previously with a qualitative experiment (see Fig. 4.9) on the VITON dataset using the full model. We see that our method is able to effectively translate the shape of the clothing items across the domains. It is notable that on the try-on task, it is able to preserve the texture information of the items even in the presence of occlusions caused by

Figure 4.7: The quality results of try-on task. The whole model is trained by using unpaired data.

Figure 4.8: The quality results of take-off task. Please note the "GT" in the take-off is just a reference. The whole model is trained by using unpaired data.

Figure 4.9: Try-on and take-off results on the VITON dataset. For try-on (top) each column shows a person (from the top row) virtually trying on different clothing items. For take-off (bottom) each example consists of three images: input image, generated take-off image and the ground-truth (GT) image. Zoom in for more details.

arms. This is handled by the proposed Fit-in module (Sec. 4.3) which learns how to combine foreground and contextual information.

Method	Take off (SSIM/LPIPS-VGG)
CycleGAN	45.63 / 47.47
MUNIT	45.97 / 46.53
MUNIT, shared S.E.	50.44 / 49.15
Ours	***61.19*** / ***34.37***

Table 4.2: Comparisons with CycleGAN, MUNIT & MUNIT with shared Style Encoder Mean SSIM and LPIPS-VGG similarity of CycleGAN, MUNIT and MUNIT with shared Style Encoder. Higher SSIM values and lower LPIPS indicate higher similarity. Both metrics are in the range $[0, 100]$.

4.4.3 Comparisons with existing methods

We compare our model w.r.t. CycleGAN [207], MUNIT [80] and VITON [69]. Fig. 4.2 shows qualitative results from our model, CycleGAN and MUNIT (with/without shared S.E.). It is clear that these unpaired methods cannot handle the one-to-many shape transfer task. CycleGAN can only work for one-to-one mapping task. MUNIT has the ability to do one-to-many mapping for style translation but it is unable to transfer shapes. We present quantitative results in Table 4.2. We do not provide the Try-on ROI scores for the same reason explained in Sec. 4.4.1. The comparison with the supervised method VITON is shown in Fig. 4.10. It is motivating that even without any supervised paired data, our method achieves competitive results.

4.4.4 Clothing retrieval

We present the in-shop clothing retrieval results using the extracted style features. We apply the shared style encoder as feature extractor to extract the style codes and then use Euclidean distance to measure the similarity for retrieval.

Protocol. The shared style encoder is trained and tested on the FashionStyle training and test sets, respectively. During retrieval, there are 1,302 query images and 434 database images. The query images are all from domain A, i.e., clothed people. and database images are from domain B, i.e., clothing items. For fair comparison, we apply the clothing masks to the query input of both our method and other methods. As shown in Table 4.3, we provide four baselines: Color histogram, Autoencoder+GAN (AE+GAN), ResNet-50/152[72] and FashionNet[113]. Following Ji et al. [84], we only use the triplet branch of FashionNet. In addition, for the comparison purpose, we use 8 dimensions and 128 dimensions feature by adding one more fully connected layer after the

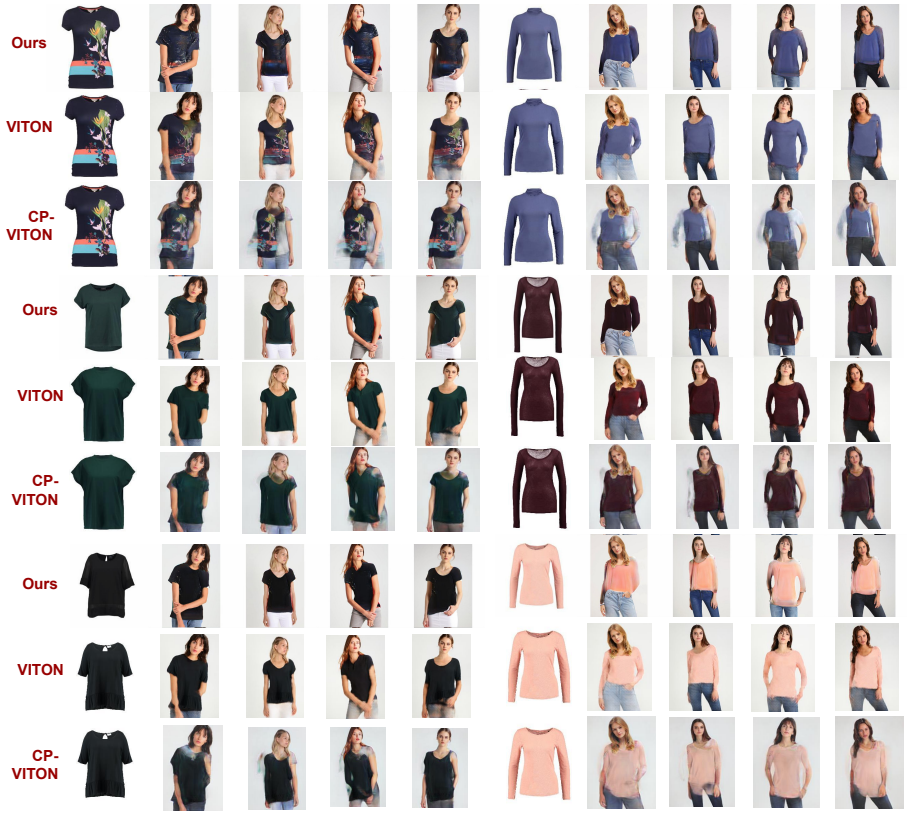

Figure 4.10: Comparison with VITON (supervised) [69] on the try-on task.

original one. For AE+GAN, the latent code of the AE is 128-dimension. We train the model using both domain A and domain B images. ResNet-50 and ResNet-152 are trained from imageNet.

Discussion. Our method outperforms all the baselines except LPIPS-Alex and FashionNet. It is noted that LPIPS-Alex extracts the feature maps of different layers as clothing features, resulting in a very high dimensional feature vector (∼640K dimensions). This costs a lot, both in compute time as well as in storage costs, which both scale linearly with the dimensionality. FashionNet is trained in a supervised way and uses a triplet loss. It is not surprising that its results are better than ours. Our extracted style code on the other hand has a very low dimension (e.g. 8), which can significantly reduce (over 80K times) the computation. Furthermore, combining our method with LPIPS-Alex

Method	Dim	top-1	top-5	top-20	top-50
Color histogram	128	1.5	3.7	9.3	17.4
AE+GAN	128	9.4	21.9	39.6	57.3
ResNet-50 [72]	2048	11.9	25.0	40.9	56.4
ResNet-152 [72]	2048	14.4	29.1	47.6	62.8
LPIPS-Alex [198]	640K	25.2	42.0	59.5	72.0
FashionNet (*D=4096*) [113]	4096	30.0	56.2	79.3	90.9
FashionNet (*D=128*) [113]	128	26.3	52.3	78.0	89.6
FashionNet (*D=8*) [113]	8	19.4	47.0	74.5	88.2
Ours (SD = 8)	8	17.1	37.6	58.1	74.3
Ours (SD = 32)	32	18.7	39.6	62.5	76.1
Ours (SD = 128)	128	19.4	41.1	64.1	77.6
LPIPS-Alex + Ours(SD = 8, κ = 20)	-	24.4	41.4	58.1	74.3
LPIPS-Alex + Ours(SD = 128, κ = 5)	-	24.4	41.1	64.1	77.6

Table 4.3: Retrieval recall rate in the FashionStyle dataset.

in a simple coarse-to-fine way, i.e. first using our method to quickly obtain the coarse top-κ results and then using LPIPS-Alex to re-rank these results, can achieve the best performance among the unpaired methods while reducing the aforementioned costs significantly. The κ value can be selected as the point where the performance of our method and LPIPS gets close. e.g. κ=20/κ=5 for Ours (SD=8/SD=128), or adapted based on user requirements. A similar gain in performance can be achieved for the case of the VITON dataset (Table 4.4).

In addition, we provide a clothing retrieval ablation study on FashionStyle, as shown in Fig. 4.11. It is interesting to observe that the performance of the retrieval process is affected by different factors than that of the image translation process (Sec. 4.4.1). We hypothesize that the translation task directly exploits shape related components in order to achieve detailed image generation. On the contrary, the retrieval task considers representative features regardless of whether they grant accurate shape transfer.

We also provide the *computation complexity* analysis for the retrieval. We use Euclidean distance to measure the difference between the features extracted from two images. For each query, computation complexity is $\mathcal{O}(d \cdot n)$ which scales linearly with the feature dimension d and the number of database images n. Thus, the computation complexity of our method is 80k times (SD=8) or 5k times (SD=128) smaller than LPIPS-Alex according to the dimension in Table 4.3. As to LPIPS-Alex+Ours, the computation complexity is $\mathcal{O}(d_{Ours} \cdot n + d_{LPIPS} \cdot \kappa), \kappa \ll n$ which maintains the performance and significantly reduces the computation compared to $\mathcal{O}(d_{LPIPS} \cdot n)$, $d_{Ours} \ll d_{LPIPS}$ for our naive implementation. While more efficient retrieval algorithms exist, the dependence on the feature dimensionality remains.

Figure 4.11: Top-5 retrieval results on the FashionStyle dataset sorted in decreasing order from left to right. Correct items are marked in green.

Method	Dim	top-1	top-5	top-20	top-50
Ours	128	20.2	39.9	64.9	79.3
LPIPS-Alex [198]	640K	42.3	61.3	77.2	88.7
LPIPS-Alex + Ours(k = 50)	-	41.6	57.7	72.8	79.3

Table 4.4: Retrieval recall rate in the VITON dataset

Method	Try-on ROI (SSIM/LPIPS-VGG)	Take off (SSIM/LPIPS-VGG)
Face experiment	69.48 / 15.65	43.82 / 39.97

Table 4.5: Mean SSIM and LPIPS-VGG distance of face experiment.

4.4.5 Face shape transfer

We further conduct experiments related to face translation. In the first experiment, given the input face and the target context (body), we generate a new image where the input face is fitted on the target context (try-on task). In the second experiment, we perform a face take-off task where given a face image with a side viewpoint, we generate an image where the face from the input is rotated towards the front and zoomed-in. We conduct these experiments on the CMU MultiPIE dataset. Qualitative results are presented in Fig. 4.12. We present translation similarity measurements in Table 4.5.

Discussion. As can be noted in Fig. 4.12, images from the different domains, i.e., frontal and side view faces, exhibit many differences regarding to scale and the presence of other parts of the body. Yet, the proposed method is able to achieve both translation tasks with a decent level of success. Fig. 4.12 shows that, for both tasks, apart from facial orientation features such as facial hair, lip color, accessories, and skin color are to some level properly translated. It is remarkable that this has been achieved without using facial landmarks like eyes, nose, mouth, ears, as in existing work [78, 203]. Failures are mainly caused by incorrectly estimated masks, large pose variation, and inconsistent skin colors. Table 4.5 shows that the proposed method has a comparable performance on both faces and clothing related datasets.

4.5 Current Research Trends

There are a few researches working along the clothing try-on task. Based on the current work, we find that warping the clothing items can always achieve

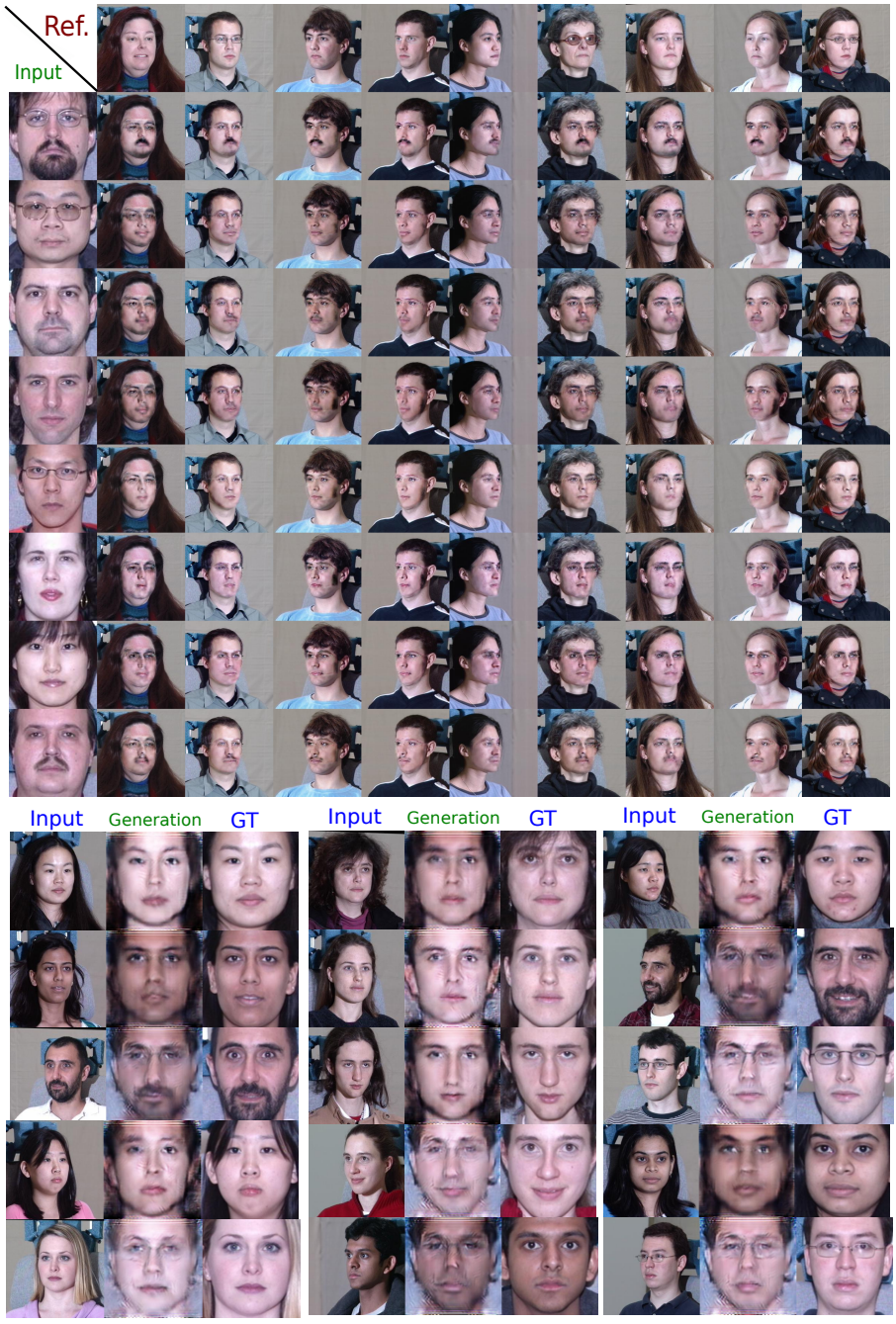

Figure 4.12: Face translation. Top, given the input face and the target body, we generate a new image where the input face is fitted to the target body (try-on), and vice versa (take-off) at the bottom.

a better visual result as long as the geometry transferring matches the shape of the target person. Instead of using a fixed clothing mask, which can lead to a failure case when the shape of the input and target clothing items are different, e.g. long sleeves v.s. short sleeves, Dong et al. [42] and Neuberger et al. [129] integrate a human body parsing network, where the network can dynamically generate the suitable clothing mask on the generated image based on the input image. In addition, Wu et al. [176] Neuberger et al. [129] and Mir et al. [127] tackle this problem with a 3D technique, such as DensePose [144]. The dense human pose estimation maps each pixel of an image to a dense pose point, where the estimated dense poses can be used for clothing region warping and pose alignment. In terms of the learning process, most of them are supervised. Interestingly, Ge et al. [52] proposes a self-supervised framework including the warping procedure based on cycle consistency and achieve a pleasant visual results. [129] proposes the O-VITON to achieve virtual try-on task without using unpaired data. The model consists of three main parts, Shape Generation to generate the segmentation mask for the target person image, Appearance Generation to transfer the clothing items to the target person image and Appearance Refinement to refine the details of the clothing items. The authors use the same input images to train the model while in the testing phase, the segmentation and appearance feature can be computed from multiple different images.

4.6 Conclusion

In this chapter we present a method to translate the shape of an object across the canonical and contextualized domains, where in the encoding phase we manage to encode the style and shape representations separately. Extensive empirical evidence suggests that our method has comparable performance on both faces and clothing data. Moreover, our ablation study shows that the proposed mask attention and Fit-in module assist the translation of shape, thus, improving the generation process. Finally, we show that the features learned by the model have the potential to be employed for retrieval tasks, in spite of their low dimensionality.

Chapter 5

Weakly Supervised Object Localization

We have explored the compositionality of individual images in the first two chapters. From this chapter on, we move our interest to image sets, where we believe models can obtain more useful information by considering multiple images at the same time. In this chapter we present the weakly supervised object localization (WSOL) task, where we take advantage of multiple images from the same class. Since these images contain a common object, the object-specific representations from different images should be similar, which gives an opportunity to mine the object locations. Our method is based on CAM. In Chapter 3 we have observed that CAM can highlight some region of the predicted class in the image without the need of the object coordinates, besides providing explanation for the DNN model.

One of the most common problems of CAM-based WSOL is that of inaccurate object coverage. In the context of state-of-the-art methods based on CAM, this is caused either by localization maps which focus, exclusively, on the most discriminative region of the objects of interest, or by activations occurring in background regions. To address these two problems, we propose two representation regularization mechanisms: *Full Region Regularization* which tries to maximize the coverage of the localization map inside the object region, and *Common Region Regularization* which minimizes the activations occurring in background regions. We evaluate the two regularizations on the imageNet, CUB-200-2011 and OpenImages-segmentation datasets, and show that the proposed regularizations tackle both problems, outperforming the state-of-the-art by a significant margin.

The presented content is based on the following paper:

- K. Wang, J. Oramas M, T. Tuytelaars, MinMaxCAM: Improving object coverage for CAM-based Weakly Supervised Object Localization. British Machine Vision Conference (BMVC) 2021.

5.1 Introduction

Learning how to localize objects in images without relying on data paired with expensive location-specific annotations is a highly desirable capability. Therefore, it is no surprise that this task, usually referred to as Weakly-supervised Object Localization (WSOL), has gained attention in recent years.

One of the most used methods for this task is based on CAM [205], see [155, 200, 199, 34, 189, 201]. In these works, it has been noticed that the localization map generated by Class Activation Mapping focuses on the most discriminative region of the image. The reason is simple: the backbone is trained for classification since there is no access to the coordinates of the object, so it learns the discriminative features for each class. As a result, the object coverage is under-estimated.

Existing efforts to address this problem follow one of three common strategies. They either iteratively occlude/replace relevant regions of input in order to force the model to learn features that enable localization [155, 189], or rely on additional networks to assist with the localization task [200, 199]. Alternatively, a more simple strategy is to update the representation learned by the convolutional layers of the model [34, 201].

For CAM-based methods, the localization map is generated by a weighted linear combination of the feature map of the last convolutional layer and then rescaled to the image's size. If the size of the feature map is too small, say 7×7 while the input size is 224×224, the resized localization map will have poor precision. In order to increase the size while still loading the original pre-trained weights of the backbone, a common strategy is to change the stride in some convolutional block [34, 201, 33]. However, we observe the localization map generated this way can activate a lot on the background, i.e. the object coverage is over-estimated (see Fig. 5.2). To solve it, activations on the background should be suppressed. To this end, Yang et al. [181] compute all possible CAM of one image and use some pre-defined combination functions to combine them. Clearly, this method costs more computation and the combination functions are not optimized.

Then the research question for us is to find a way to actively control the activation distribution on the localization map, maximizing or minimizing the

activations as needed. We propose *MinMaxCAM*, a method that iteratively learns the classification task which provides the coarse feature maps (Stage I) and how to re-weight them so that it is capable of shifting the mass of their internal activations (Stage II). This not only enables accurate object localization but is relatively stable to train and does not need additional networks. In particular, we design two regularizations, *Common Region Regularization (CRR)* and *Full Region Regularization (FRR)*, that can serve as objective functions for the model to optimize the *linear layer* after global average pooling (GAP). *CRR* is based on the fact that multiple images from the same class share a very similar representation for the common object. During the training, the coarse localization map obtained from Stage I via Class Activation Mapping is used to extract the object-specific representation. By minimizing *CRR* we force the model only optimizes the linear layer, whose weight can be utilized to combine the feature map, generating more object-only towards localization maps. On the other hand, inner-class differences can reduce the common region to a very part of the object only. The same may happen due to failed localization of the most discriminative region. To tackle these situations, *FRR* is proposed. It stimulates covering a larger part of the object by only optimizing the linear layer like *CRR*. Stage I and II are optimized every mini-batch, optimizing Stage I guarantees that the coarse localization map is object-centered.

MinMaxCAM has a number of advantages:

- It is light-weight: it only relies on a standard classification model; no extra network is needed. It saves computation resources and is relatively simple to train.
- The proposed method produces more precise or tighter bounding boxes, addressing the problem of over- and under-estimating the object within one model.
- Despite its simplicity, the proposed method is capable of setting a new state-of-the-art performance on the imageNet, CUB-200-2011 and OpenImages-segmentation datasets, outperforming existing methods by a significant margin.

5.2 Related Work

Most existing works related to WSOL [155, 199, 200, 34, 189, 201, 181] are based on CAM [205]. They address the WSOL task, indirectly, by solving the problem that the generated localization map only focuses on the most

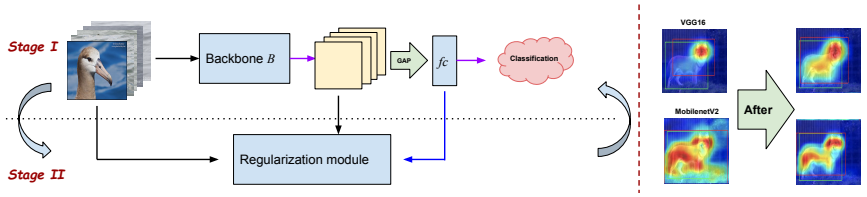

Figure 5.1: The overview of our MinMaxCAM. Flows with gradients are color-coded. Stage I optimizes both backbone and the linear layer via the classification task (purple arrow), Stage II only optimizes the linear layer (blue arrow) via our proposed two regularizations. After regularizing the model, the localization map can overcome either the under-estimation (top right) or the over-estimation (bottom right) problem. localization optimization module

discriminative regions of the image. These methods can be divided into two types: non-parametric (w.r.t. CAM) and parametric methods. In this section, we focus on representative works from these two types. Please refer to Zhang et al. [193] for a more comprehensive survey on the WSOL task.

[155, 34, 189, 201] and our work belong to the first type. These methods do not need extra networks during the training and inference phases. This makes the methods lightweight, easy to implement and saves computation resources. Singh et al. [155] force the neural networks to focus on other relevant regions of the objects of interest by randomly occluding some patches of the input image when the network is trained for a classification task. Recently, Babar et al. [8] propose to occlude one image with two complementary occlusion patterns, creating input image pairs, to tackle the WSOL task. Yun et al. [189] extend this idea by using patches from other images as occluding regions in a given image. Choe et al. [34] propose a simple but effective method: randomly drop out the most highly activated region or apply an attention mask on the feature maps when the classification network is trained. Zhang et al. [201] use information shared by two images from the same class to improve the localization map. They apply two constraints to improve the quality of the localization map. The first constraint is to learn the consistent features of two images of the same class by randomly sampling the features located in the most activated region and minimizing their distance. The second constraint is to compensate the limitation that features can only keep consistency within batches, where it learns a global class center for each class. To increase the coverage of the localization map, Mai et al. [122] propose to erase the most discriminative regions in the feature map when training a classification task. Yang et al. [181] found that CAM can also activate on the background. To suppress the activations of an image, they propose to compute all the possible CAMs firstly, and combine

them via a combination function. This combination function is not learned during the training but pre-defined and related to the prediction probability of each possible class. Differently, our method only computes the localization map once for each image.

Zhang et al. [199, 200] add extra components based on the CAM model. [199] proposes a two-head architecture where the activation map generated by one stream is used to suppress the most discriminative region of the activation map generated by another one. By doing this, the model learns to use information from other relevant regions instead of the most discriminative one. [200] proposes to generate self-produced guidance (SPG) masks that separate the target object from the background. The masks are learned by the high-confidence regions within attention maps progressively and they also provide the pixel-level supervisory signal for the classification networks. Guo et al. [65] add a regressor which takes pseudo-locations generated by the model to learn the coordinates of the object. In addition, two novel losses are proposed to keep the localization map cover the whole object during the training process. Instead of using CAM-based methods, Lu et al. [117] use a encoder-decoder architecture to learn the location of objects, by leveraging the geometry constrains of objects. A novel loss function that considers the object's geometrical shape is proposed.

Different from these methods, we focus on the linear combination part of the CAM-based method rather than the feature extraction part, or the structure of the input images. The linear layer is optimized based on the proposed regularizations, which provides the optimal combination factors to generate the localization map. Similar to the non-parametric methods, our method does not have more trainable parameters. We only introduce two more hyper-parameters which are the weights for the two regularizations.

5.3 Methodology

5.3.1 Problem statement

CAM [205] is widely used to localize an object of interest in an image, in a weakly supervised manner. Given a backbone B applied to input image I, followed by GAP and a linear layer fc, in which $w \in \mathbb{R}^{C \times K}$ is the weight matrix, where C is the number of classes. The CAM of an image is computed as:

$$CAM = \sum_{k=0}^{K-1} w_k^c B(I) \tag{5.1}$$

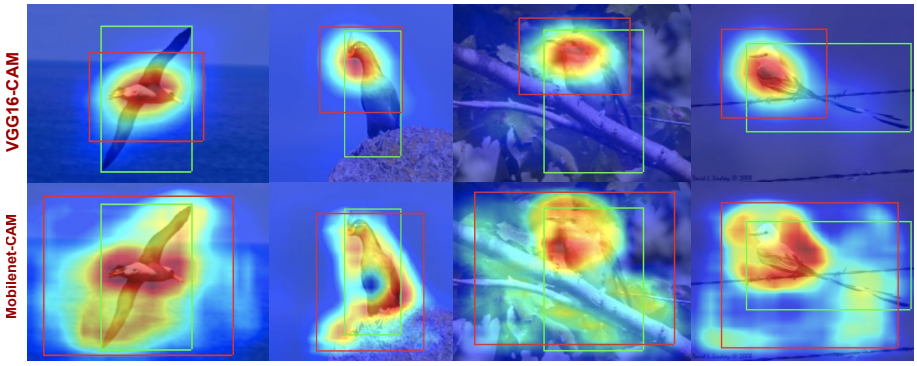

Figure 5.2: CAM generated with the VGG16 (top) and MobileNet (bottom) backbones with their estimated bounding boxes (red) and ground-truth (green).

where c is the class of image I. In short, the localization map is a linear combination of the feature maps of the last convolutional block of the backbone B. The weights of this combinations are taken directly from the weights of the linear layer w.r.t. the class which I belongs to. Choe et al. [33] and Zhou et al.[205] noted that localization maps generated by VGG often focus on the most discriminative region of the image, rather than on the whole object, i.e. under-estimating the location. We and Yang et al. [181] also observe that for different backbones B, the localization map can sometimes even cover the whole image, i.e. over-estimating the location. Fig. 5.2 shows some examples of the two cases. Current methods solve either one of the two problems. Here we propose two regularizations that can address these problems within one model.

5.3.2 Common region regularization

The idea of the *Common region regularization (CRR)* is that different images depicting foreground objects of the same class should share a very similar feature of the common object. To obtain the object-specific representation from the whole image I, we first acquire the localization map H_m via Eq. 5.1. We freeze the trainable parameters of B (noted as B^*) and make H_m can only update fc. Then we extract the object-specific feature f via $B^*(I \odot H_m)$ and GAP. Notably, B is still frozen since it serves as a feature extractor. Therefore, $f{=}g(B^*(I \odot H_m))$, where g and $\odot$ refer to GAP and the element-wise multiplication. To save computation resources, we use the same backbone B in Sec. 5.3.1. We will present the learning process in Sec. 5.3.4. Given m different

images from the same class, we have

$$CRR = \frac{1}{m(m-1)} \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} ||f_i - f_j||_2^2 \tag{5.2}$$

CRR calculates the pair-wise distance of the feature of $I \odot H$.

There are two conditions in place for CRR to work: 1) a set of images from the same class should have different background and 2) f should have different activation values for different backgrounds. The first assumption is dependent on the dataset. We will discuss the second assumption later.

5.3.3 Full region regularization

For the case where the localization map H_m under-estimates the object region, we propose *Full region regularization (FRR)* to enlarge the localization map. In addition, FRR can also compensate a possible side-effect of using CRR, that is B only focusing on a very small part of the object if the objects from the same class have some inner difference in different images.

$$FRR = \frac{1}{m} \sum_{i=0}^{m-1} ||f_i - f_i^o||_2^2 \tag{5.3}$$

f^o is the feature of the original image I, i.e. f^o=$g(B^*(I))$. Notably, f^o cannot update the model. FRR calculates the distance between the object-specific feature f and the image feature f^o. Minimizing FRR make f_i closer to f^o, hence H_m will change towards the identity matrix $\mathbf{1}$ since $B^*(I)$ can be also interpreted as $B^*(I \odot \mathbf{1})$ In other words, Minimizing FRR has the effect of **maximizing** the activations on the localization map. Similar to CRR, there is one assumption in place for FRR to work: B should not be invariant to changes in intensity. We will discuss it in Sec.5.4.4.

5.3.4 Training process

The training process has two stages. For stage I, it takes $N \times m$ images as input. The model is trained for the classification task to provide the coarse localization map, i.e. update the backbone B and linear layer fc via the cross-entropy loss.

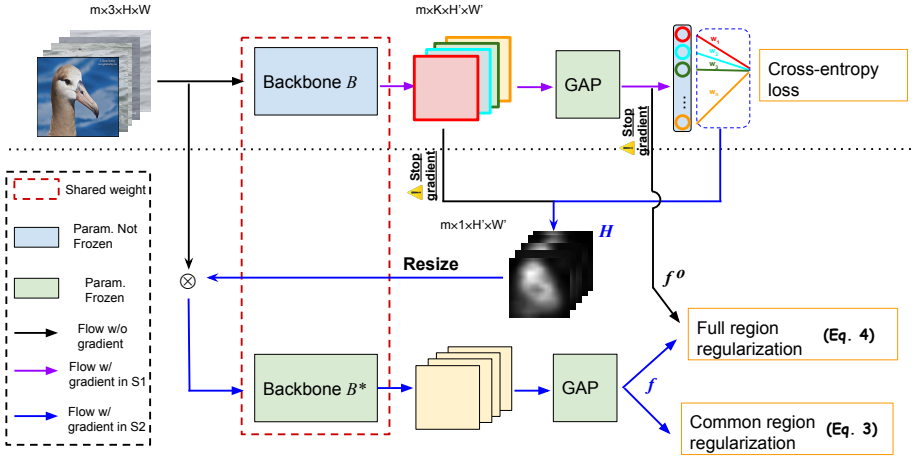

Figure 5.3: Proposed method architecture. In Stage I (above the dashed line) we train B and fc for a classification task. In Stage II (below) we multiply the localization map H_m with the images to extract object-specific features and compute CRR and FRR, where B is frozen and is noted as B^*. The two regularizations update the weights of fc (blue arrow).

It is the same as Class Activation Mapping.

$$L_{S1} = -\sum_{i=1}^{N \times m} c_i log(\hat{c}_i) \tag{5.4}$$

For stage II, the backbone is frozen as a feature extractor, noted as B^*. It receives the images multiplied by the localization map ($I \odot H$) as input to extract the object-specific representation. H_m is obtained via Eq. 5.1 and can only update fc. This step introduces an intensity change on the original image. We discuss its effects on B in Sec. 5.4.4. The features (f and f^o) extracted by B^* are used for the two regularizations. By minimizing CRR and FRR (L_{S2}), the loss updates fc, making H_m gradually more accurate by adjusting the combination weight (w_k^c).

$$L_{S2} = \lambda_1 CRR + \lambda_2 FRR \tag{5.5}$$

L_{S1} and L_{S2} update the model **every mini-batch**.. Fig. 5.3 shows the proposed method. To train our model there is no extra hyper-parameters besides the weights for the two regularizations. During testing, the CAM is generated as localization map, (Eq. 5.1), therefore, there is no need for a set of images per class.

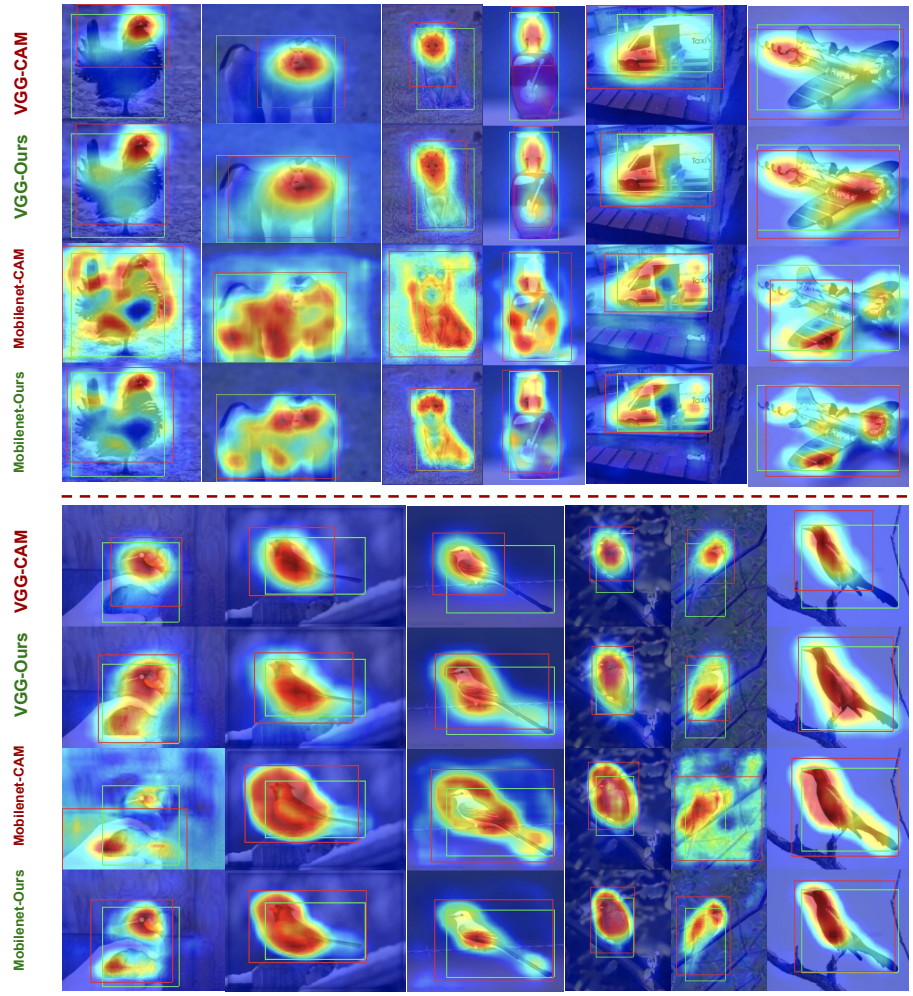

Figure 5.4: Qualitative comparison of the localization map H_m on the imageNet and CUB dataset. For reference we show the ground truth bounding box (green) and the one estimated by H_m (red) based on the optimal threshold τ.

Why freeze the backbone (B) in Stage II? The goal of stage II is to adjust the weights w_k^c which are used to generate the localization map by minimizing CRR and FRR. B serves as a feature extractor in this stage. If B were also updated, after minimizing FRR, it would become invariant to intensity changes. In the later training process, FRR could then not measure the difference between f^o and f.

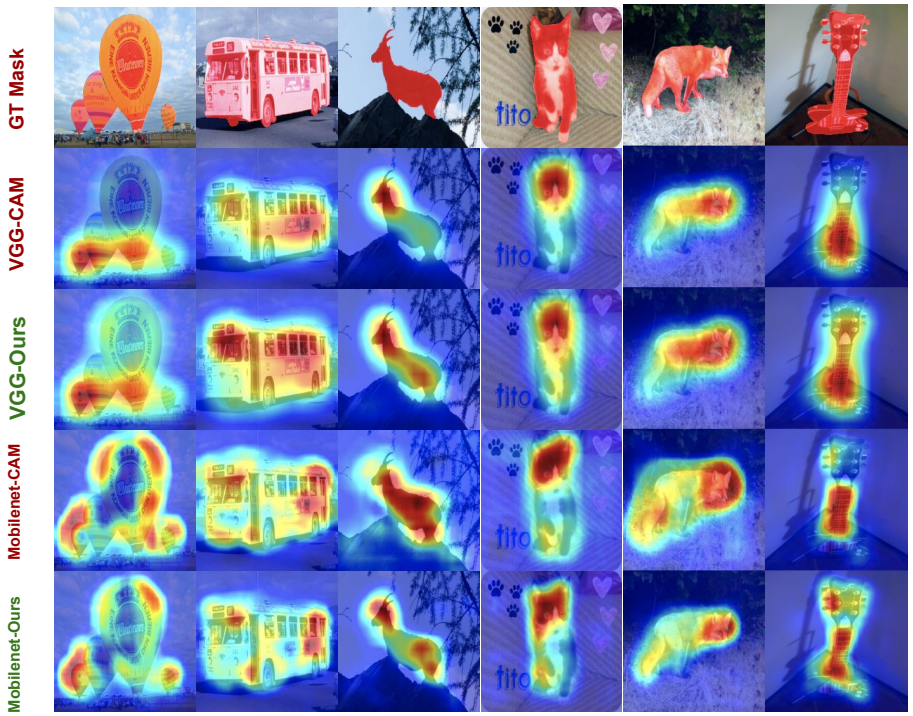

Figure 5.5: Qualitative comparison of the localization map H_m on OpenImages dataset. For the OpenImages dataset, the first row shows the input image with the target segmentation mask.

5.4 Experiments

5.4.1 Dataset and performance metric

We consider three widely used datasets imageNet [37], CUB-200-2011 (CUB) [166] and OpenImages instance segmentation subset [14, 33] to evaluate our method. contains 1,000 classes with over 1 million images. Following Choe et al. [33], we use imageNetV2 [140] as the validation set to tune our model. This validation set contains 10 images per class with the object bounding boxes annotated by Choe et al. [33]. CUB has 200 fine-grained classes of birds with 5,994 images for training and 5,794 images for testing. Similarly, we follow Choe et al. [33] to use a validation set collected by them to tune the model. The validation set contains 1,000 images in total, around 5 images per class. OpenImages instance segmentation subset (OpenImages) [32] covers 100 classes.

It contains 29,819, 2,500 and 5,000 images for training, validation and testing, respectively. Every image has the object segmentation as annotation.

Performance metric [205, 34, 189, 199, 155, 201] use a pre-defined threshold (0.2) for the generated CAM to produce a localization region. Choe et al. [33] argues that using a fixed pre-defined threshold can be disadvantageous for certain methods since the ideal threshold may depend on the data and architecture that are used. In short, for different datasets, architectures and methods, the ideal threshold is different. We follow this idea and use the metric proposed by Choe et al. [33]. For imageNet and CUB datasets we use two threshold-free metrics to evaluate the localization map, i.e. MaxBoxAcc and MaxBoxAccV2. MaxBoxAcc is equivalent to *GT-known localization accuracy* where one localization map is counted correct when the intersection over union (IoU) of the estimation and ground truth bounding box is larger than 0.5. Differently, to avoid using a fixed pre-defined threshold for binarizing the localization map, here we set various τ thresholds to find the best performance. MaxBoxAccV2 is the average of three MaxBoxAcc when the IoU is 0.3, 0.5 and 0.7. For OpenImages dataset, since we have access to the segmentation mask, we use the *pixel averaged precision (PxAP)* proposed by Choe et al. [33]. Similarly, *PxAP* is also threshold-free. Please refer to [33] for more details.

5.4.2 Implementation Details

We consider three different backbones: VGG16 [153], ResNet-50 [72] and the lightweight MobilenetV2 [147]. Following [33, 201], for ResNet-50 and MobilenetV2 we increase the size of the last feature map by changing the stride of convolution layers from 2 to 1. By doing this, the model can still load the pretrained weights. We set the cardinality m=5, N=12 (i.e. batch size = 60) for all the experiments except those with the ResNet-50 backbone, where N=10 due to GPU memory limitations. For τ we set 100 intervals between 0 and 1.

5.4.3 Comparison with State-of-the-art Methods

In this part, we compare the proposed method w.r.t. several state-of-the-art methods on imageNet, CUB and OpenImages datasets. Table 5.1 shows quantitative results. The quantitative results from these methods except I2C are taken from [33, 32] where the authors used the validation set to select the final models. We implement the I2C method and use their suggested hyper-parameters to train the models. The results clearly show that our method outperforms the competing methods on all three datasets except when ResNet-50 is selected as backbone for the CUB dataset. In this case, our method is only

lower by 0.1 pp, in *MaxBoxAcc* w.r.t. HaS. Interestingly, for imageNet with a ResNet-50 backbone, only our method outperforms CAM. We believe it is due to the proposed CRR which minimizes the activations in the background. It is expected that I2C works better when MobilenetV2 or ResNet-50 are used as backbone, since the constraints proposed by I2C prevent the model from activating highly in background regions, which is a weakness that Mobilenet and ResNet suffer from. Please note the performance can be further improved when the optimal hyperparameters are found.

Fig. 5.5 shows qualitative comparisons w.r.t. CAM on the imageNet, CUB and OpenImages datasets. It clearly shows that our method can enlarge H_m when it originally focuses on a small region and reduces it when it is highly-activated in the background. In the fifth-column example from the imageNet dataset generated by the VGG backbone, the effect of different optimal thresholds τ can be noticed. The same effect can be seen in the Mobilenet-based localization map of the last example in the imageNet dataset. In addition, in some cases although the estimated bounding box of CAM has a large IoU with the ground truth, the object region has a stronger activation for our method (e.g. the last example of the imageNet dataset).

5.4.4 Analysis of our method

In this section we analyze several aspects of our method. The experiments are conducted on the CUB dataset.

Background activation We have introduced in Sec. 5.3.2 that B should activate differently for different background regions for CRR to work. If the assumption holds, then the object-specific representation $f{\in}\mathbb{R}^K$ ($f{=}g(B(I{\odot}H_m))$) from the same class should be distributed together and more centered in the K-dim representation space, compared with $f^o \in \mathbb{R}^K$ ($f^o{=}g(B(I))$). To verify it, we first use t-SNE [164] to visualize the representation f and f^o from the same class, see Fig. 5.6. Please note, we use the generated localization map H_m to obtain f. To quantitatively measure the statistical dispersion of these representations, for each class, we first $L2$-normalize the representations and calculate their distance to the class center which is calculated by averaging all the representations of that class, then compute the average distance within one class. More centered distribution means a shorter distance between each representation and the class center. Then we compute the average distance for all classes. For f and f^o, the average distance are 1.03 ± 0.34 and 1.47 ± 0.38 ,respectively, which suggests that f is distributed more centered compared with f^o. The result proves our assumption,

Method	Backbone	imageNet		CUB		OpenImages
		MaxBoxAcc (%)	MaxBoxAccV2 (%)	MaxBoxAcc (%)	MaxBoxAccV2 (%)	PxAP (%)
CAM	VGG16	61.1	60.0	71.1	63.7	58.1
HaS	VGG16	0.7	0.6	5.2	0	-1.2
ACoL	VGG16	-0.8	-2.6	1.2	-6.3	-3.4
SPG	VGG16	0.5	-0.1	-7.4	-7.4	-2.2
ADL	VGG16	-0.3	-0.2	4.6	2.6	0.2
CutMix	VGG16	1.0	-0.6	0.8	-1.4	0.1
I2C	VGG16	-	-	-2.7	-3	-1
WTL	VGG16	2.3	-	6.4	-	–
Ours	VGG16	***3.5***	***2.4***	***12.8***	***6.5***	***1.9***
CAM	ResNet-50	64.2	63.7	73.2	63.0	58.0
HaS	ResNet-50	-1	-0.3	4.9	1.7	0.2
ACoL	ResNet-50	-2.5	-1.4	-0.5	3.5	-0.2
SPG	ResNet-50	-0.7	-0.4	-1.8	-2.6	-0.3
ADL	ResNet-50	0	0	0.3	-4.6	-3.7
CutMix	ResNet-50	-0.3	-0.4	-5.4	-0.2	-0.7
I2C	ResNet-50	-	-	0.3	1.0	**2.9**
WTL	ResNet-50	0.6	-	**5.3**	-	–
Ours	ResNet-50	***3.9***	***2.5***	*5.0*	***4.8***	***2.9***
CAM	MobilenetV2	60.8	59.5	65.3	58.1	54.9
I2C	MobilenetV2	-	-	1.9	1.5	3.3
Ours	MobilenetV2	***4.5***	***3.8***	***10.5***	***6.9***	***4.4***

Table 5.1: Quantitative comparison w.r.t. state-of-the-art. The numbers indicate the difference w.r.t. the baseline method CAM. The scores of CAM [205], HaS [155], ACoL [199], SPG [200], ADL [34], CutMix [189] are taken from [33, 32] while WTL [8] is taken from itself. Performance of I2C [201] was computed by ourselves. Due to limited computation resources we limit ourselves to report performance only on the CUB and OpenImages datasets

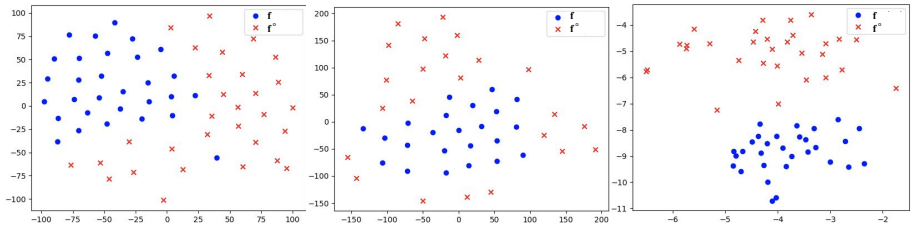

Figure 5.6: t-SNE visualizations of f and f^o for class 64, 67 and 70 of CUB dataset. The blue dots and red crosses refer to f and f^o, respectively. Please zoom in for more details.

that indeed B is activated differently for different backgrounds of the input image, and this is one of the reasons that *CRR* works.

Intensity change For *FRR* to work, B should *not* be robust to changes in intensity of its input, otherwise, there would be no difference between f and f^o. The experiment based on VGG backbone has implied that this condition

Method	*MaxBoxAcc* (%)	*MaxBoxAccV2*(%)
VGG-Ours	83.9	70.2
Apply H_m on f'	-19.1	-12.5
MobilenetV2-Ours	75.8	65.0
Apply H_m on f'	-4.1	-3.0

Table 5.2: Masking inputs vs.masking features

holds. Here we conduct an extra experiment to confirm the necessity of the condition for *FRR* from another direction. We add an intensity change as one of the data augmentations in stage I. In stage II we do not change anything. By doing this, we force B to become *more* robust to intensity changes on its input (Please note, we cannot make B completely robust). We use VGG16 as backbone. The performance drops 2.5pp and 1.9pp for MaxBoxAcc and MaxBoxAccV2, respectively. This shows that indeed the performance drops when B becomes more robust to intensity changes.

Masking inputs vs. masking features In order to get the localized-object representation f, we suggest applying element-wise product between H_m and the input image I firstly and then send it to the same backbone B. It can be argued that H_m can be applied on the extracted feature map (noted as $f' \in \mathbb{R}^{N \times K \times H \times H}$) from stage I directly, which can avoid the re-computation of the feature. However, this may produce some problems.

On the one hand, f' not only has spatial information but also has K channels (around 1024 or 2048 for our backbones), in Chapter 3 we find that different channels can represent different concepts of the image. On the other hand, the localization map $H \in \mathbb{R}^{H \times H}$ only contains spatial information, no channel-wise information. Information can be lost if H_m is directly applied on f' since the activations on the same spatial location but different channels will be encouraged/suppressed in the same scale. To verify our analysis, we conduct an experiment where we apply H_m directly on f'. Table 5.4.4 suggests that our analysis is correct. Especially, for VGG16 the performance is even worse than CAM.

Classification v.s. localization In Chapter 3 we find the classification task sometimes rely on information from the background, while the localization task only focuses on the foreground object. Therefore, a good localization model is not necessarily a good classification model. The evaluation metric *top-1/5 Loc.* takes into account both localization and classification accuracy of a given localization model,therefore, it is not able to accurately measure the localization performance [33]. *Top-1/5 Loc.* can be low because of the classification accuracy even if the localization accuracy is good.

Method	*Top 1 Loc. / Top 5 Loc.*
VGG-ACoL	45.9 / -
VGG-ADL	52.3 / -
VGG-CCAM	50.1 / 63.8
VGG-Ours	***66.0*** / ***83.9***
MobilenetV1-HaS	44.7 / -
MobilenetV1-ADL	47.7 / -
MobilenetV1-Ours	***54.8*** / ***69.4***

Table 5.3: Top-1 and Top-5 localization rate.

In order to compute *Top-1/5 Loc.* fairly, we propose a simple path: train a separate classification model to provide the predicted class for the object localization model. In practice, we use ResNet-50 to train the classifier, whose classification accuracy on CUB dataset is 77.3%. Table 5.3 shows the *Top-1/5 Loc.* results. In order to compare with the competitive methods, here we use MobilenetV1 as backbone. The numbers of the competitive methods are taken from Choe et al. [34] and Yang et al.[181].

5.4.5 Ablation study

Effects of *CRR* and *FRR* for different backbones (B) We analyze the effect of *CRR* and *FRR* on different backbones. In practice, intuitively, if the localization map H_m always localizes the most discriminative region of the object, e.g. VGG16, *FRR* should play a more important role in the training process. On the contrary, *CRR* should be more essential if H_m is relatively highly activated in background regions, e.g. Mobilenet. To verify the influence of *CRR* and *FRR*, we gradually increase/decrease the weight of one regularization with the other one fixed.

Fig. 5.7 shows the performance curve. For VGG16, performance decreases gradually as the weight for *CRR* increases. The opposite occurs with *FRR*. The performance increases and reach the peak when the weight of *FRR* is 10, afterwards the performance decreases slightly (weight=20). On the contrary, increasing the weight of *CRR* boosts the performance for MobilenetV2. The performance decreases slightly when the weight of *FRR* is 0 while it drops dramatically when a larger weight is applied. These trends are expected. VGG16 focuses on small discriminative regions rather than the whole object, hence *FRR* which maximizes the activations on the localization map improves the object localization ability while *CRR* which suppresses the activations makes the model worse. An opposite trend can be observed from MobilenetV2

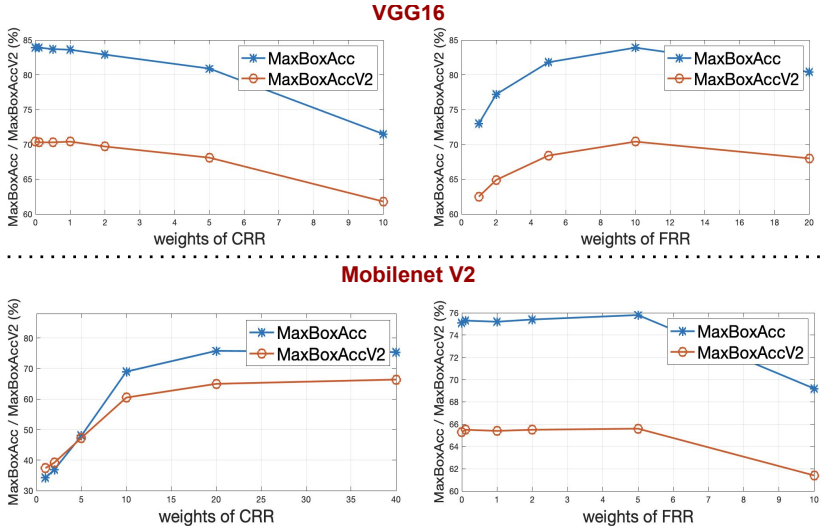

Figure 5.7: Ablation study w.r.t. *CRR* and *FRR* for the VGG16 (top) and MobilenetV2 (bottom) backbones, respectively. For each plot, We fix one regularization and ablate the other.

Cardinality m	*MaxBoxAcc(%)*	*MaxBoxAccV2(%)*
m=5	75.8	65.0
m=4	74.7	64.6
m=3	74.4	64.4
m=2	73.9	63.9
CAM	65.3	58.1

Table 5.4: Effect of the cardinality m.

which activates frequently on the background. The ablation study confirms the effectiveness of our proposed *FRR* and *CRR*.

Effect of the cardinality (m) Here we discuss how the cardinality m influences the performance of *CRR*. *CRR* can benefit from using a relatively large m because more representations can be grouped together simultaneously. We conduct series of experiments, where we decrease m from 5 to 2 gradually. We use Mobilenet as backbone and keep the batch size the same (60 images). The results in Table 5.4.5 indicate that for a larger cardinality m, *FRR* indeed works better.

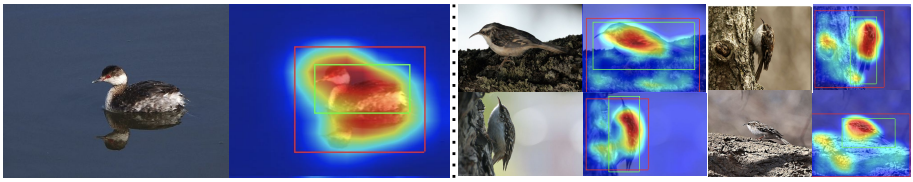

Figure 5.8: Failure cases of the proposed method.

5.4.6 Failure cases

We analyze some failure cases in this section. The first type is caused by nature, which is unavoidable, like reflections of water. The cause of the second type is related to our first assumption for *CRR*, that different images from a class have very similar background. For example, brown creeper always appear with trunks in the image. Fig. 5.8 shows some examples.

5.5 Current Research Trends

We credit the success of CAM-based WSOL to the potential ability of attention mechanism of CNNs. The filters activate highly when processing the object related features since they are the important cue for the model, thus the activation maps can reflect the location of the object to some extent. The limitation of the CAM-Based methods is, however, also due to the usage of activation maps: the downscaled activation map cannot very precisely cover the object in the input scale. Recently, transformer [165], a type of DNN almost completely relying on self-attention mechanism, has shown its ability on various tasks that are traditionally handled by CNN, such as classification [43], object detection [21] and segmentation [168]. The visualization of the attentions in the vision transformer (ViT) [43] shows the important patches on the input image based on the model prediction, where the concept is quite similar to what CAM derives from. Currently, there is no such work which systematically analyzes the localization ability of transformers, but in [43, 25] the authors have shown some visualizations of the attentions, which can indeed cover the object regions. Therefore, we believe the vision-based transformer also has a good ability on object localization based on the internal attentions.

5.6 Conclusion

In this chapter, we propose two representation regularizations, *Common Region Regularization* and *Full Region Regularization*, to overcome the weaknesses of weakly supervised object localization methods based on CAM. *CRR* benefits from considering the sets composed by multiple images containing the same object, which helps obtain more precise object location information. Our method relies only on a standard classification model; no extra network is needed. Through extensive experiments and analysis, we discuss relevant aspects of our method and show that it is capable of surpassing the state-of-the-art by a significant margin.

Chapter 6

Face Image Hallucination

In the previous chapter, we leverage multiple images from the same class to mine the accurate location for the common objects, based on the similarity of the object-specific representations. In this chapter we present another task that can be assisted with using multiple images: Face image hallucination. Given a really low resolution input image of a face (say 16×16 or 8×8 pixels), the goal of face hallucination is to reconstruct a high-resolution version thereof. This, by itself, is an ill-posed problem, as the high-frequency information is missing in the low-resolution input and needs to be hallucinated, based on prior knowledge about the image content. Rather than relying on a generic face prior, in this chapter we explore the use of a set of HR exemplars, i.e. other high-resolution images of the same person. These guide the neural network as we condition the output on them. To obtain the information from multiple exemplars effectively, we introduce a pixel-wise weight generation module to find and combine the important component representations from each HR exemplar in the exemplar set based on the input LR image, since the HR exemplars are randomly selected. Besides standard face super-resolution, our method allows to perform subtle face editing simply by replacing the exemplars with another set with different facial features in the inference phase. A user study is conducted and shows the super-resolved images can hardly be distinguished from real images on the CelebA dataset. A qualitative comparison indicates our model outperforms methods proposed in the literature on the CelebA and WebFace datasets.

The presented content is based on the following paper:

- K. Wang, J. Oramas M, T. Tuytelaars, Multiple Exemplars-based Hallucination for Face Super-resolution and Editing. Asian Conference

on Computer Vision (ACCV) 2020.

6.1 Introduction

Super resolution (SR) imaging consists of enhancing, or increasing, the resolution of an image, i.e., going from a coarse low-resolution (LR) input to one with HR depicting more details. Especially challenging is the setting where there is a large scale factor between the resolution of the input and that desired for the output. In that case, there is insufficient information in the LR input. This leads to the need to "hallucinate" what the detailed content of the HR output would look like. As can be seen in Fig. 6.1 this makes the SR process an ill-defined problem with multiple valid HR solutions for a given LR input.

Using side-information is a promising avenue towards guiding the SR process and addressing the above issue. In particular, we consider using multiple HR reference images, or *exemplars* as we call them, as a form of side information. Typically, these are HR images of the same person depicted in the LR input but randomly chosen. They provide visual cues that are expected to occur in the HR output – think e.g. of the color or shape of the eyes of the person. By guiding the SR process these exemplars help to constrain the output and disambiguate the ill-defined characteristic of the problem. The output becomes conditional on the provided exemplars and a different set of exemplars will lead to a different output.

In this work, we advocate the use of *multiple* exemplars, rather than just one. This gives more flexibility, as it allows the network to pick the visual features that best fit the LR input image, in terms of facial expression, illumination or pose. Our method effectively exploits available information from the exemplars via the proposed Pixel Weighted Average (PWAve) module. This module learns how to select useful regions across the exemplars and produces superior results compared to simply averaging the representations [111].

Rather than just superresolving the LR input, we can also exploit the ambiguity of the problem to our advantage and use different sets of exemplars as a means to inject different visual features in the produced HR image (see again the examples in Fig. 6.1). This turns the model into a flexible tool for subtle image editing. Since in this manuscript we focus on images depicting faces, we refer to the general SR process as "face/facial hallucination". Similarly, we refer as "face editing" to the task where new features are injected. Investigating these two tasks constitutes the core of this chapter.

Taking a broader perspective, the method proposed here to condition a network

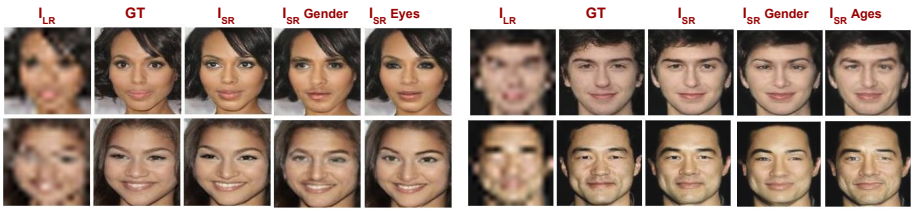

Figure 6.1: Super-resolving images with different facial features by using different exemplars. From left to right: input LR image, ground truth HR image, SR image using exemplars of the same person and SR images using two sets of exemplars with different facial features. Please note, the ethnicity of the LR image does not change.

on a set of exemplars, can be seen as a novel, lightweight and flexible scheme for model personalization, that could have applications well beyond face hallucination or face editing. Indeed, instead of finetuning a model on user-specific data, which requires a large set of labeled data and extra training, or adapting a model using domain adaptation techniques, which requires access to the old training data as well as a large set of unlabeled target data and extra training, the scheme we propose allows to adapt a generic model to a specific user without the need for any retraining. All that is needed is a small set of exemplars and a standard forward pass over the network. This opens new perspectives in terms of on-the-edge applications, where personalization is often a desirable property yet computational resources are limited.

Our contributions are three-fold:

- We propose to use multiple exemplars to guide the model to super-resolve very low resolution (16×16, 8×8) images.
- Our model does not require any domain-specific features, e.g. facial landmarks, in the training phase. This makes our model general to adapt to other datasets and tasks. Our user study indicates our results are hard to distinguish from real images. Our qualitative analysis suggests that our model outperforms the literature baselines.
- Besides achieving face super-resolution, our model can also address the face editing task. Unlike traditional conditional generative adversarial networks which can only generate images by modifying pre-defined discrete visual features, our model can dynamically generate HR face images with arbitrary facial features, benefiting from the usage of exemplars.

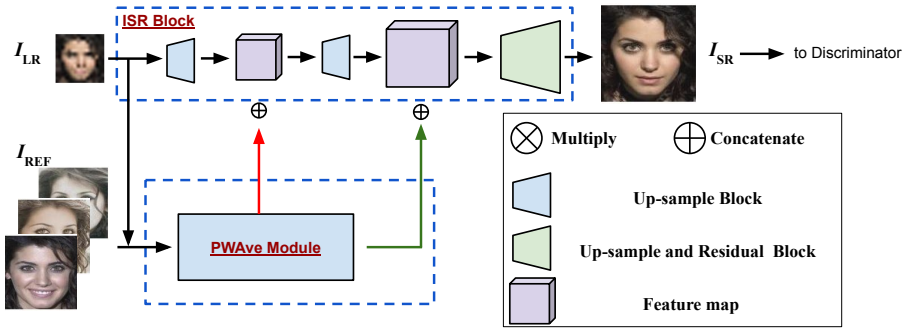

Figure 6.2: The proposed model in our chapter. For details of the PWAve module, please refer to Sec. 6.3 and Fig. 6.3.

6.2 Related Work

Face super-resolution

Using deep neural networks to super-resolve LR images has been applied quite widely. The methods can be gathered into two groups: those using facial priors like landmarks or heatmaps, and those not using any facial priors. [31, 17, 187, 38] belong to the first group. Yet to use these facial priors, extra annotations and networks are required. In the other group, [188, 118, 103, 41, 86, 196] use a guidance-image or semantic labels (e.g. facial attributes or face identity) to help the model super-resolve the LR images. [188, 118] utilize pre-defined facial attribute labels. Besides super-resolving the LR image, as a conditional adversary model, it allows users to change the feature labels to control the generated SR image. However, to achieve this, they require additional data with annotations for the visual attributes to be added/edited. Furthermore, editing of visual features is also limited to the set of visual attributes pre-defined at design time, and fixed to the possible values that these features may take. Moreover, when new features are desired to be integrated in the editing model, additional data and re-training are required. In addition, Yu et al. [188] manually rotates the input LR image and uses a spatial transformer to cope with this change. Compared with Yu et al. [188] and Lu et al. [118], our method can be regarded as conditioned on multiple exemplars, which gives us more flexibility on the subtle editing of facial features. Xin et al. [86] does not use any guiding image, but uses the facial feature labels to train a channel-wise attention model which guides the model to recombine the basic features in the super resolution process. Compared with [86], our method uses pixel-level weights generated from our PWAve module, which provides the user with a more clear visualization on the

usage of exemplars. In addition, the way of using facial features in [86] does not allow users to control the generation. Zhang et al. [196] considers the facial identity label, where they use a specific person identity loss. Li et al. [103] and Dogan et al. [41] use one guiding exemplar while we propose to use multiple guiding images. They warp the guiding exemplar (HR) via a flow field and use a specifically designed network to align with the input LR image. [103] also uses facial landmarks in the training of the warping network. Li et al. [102] also considers using multiple exemplars. However, [102] selects one exemplar from multiple exemplars and only processes the selected exemplar, while we handle multiple exemplars simultaneously by using our PWAve module.

Conditional Image Generation

[131, 128, 29, 11, 179] use labels as condition to guide neural networks to generate images. Therefore, the generated images are limited to the pre-defined classes. [111, 46, 107, 80] achieve image translation by using condition image(s). In the training process, they aim to disentangle the visual features from both input and condition images and re-combine the features from different images later. Liu et al. [111] simply averages the features from condition images and uses adaptive instance normalization to insert them. To achieve the modification of facial features, our method does not need to re-train the original SR model. Our SR model is originally trained to restore the LR image with the help of several exemplars that have similar visual features to the LR image. Besides, our PWAve module generates the pixel-wise weights among the multiple exemplars. In addition, the input image in our method is a LR image while conditional image translation efforts [111, 46, 107, 80] receive the HR image as part of its input.

6.3 Methodology

In this section, we describe the architecture of our model and the objective functions required for its optimization. For each of the LR images to be super-resolved we assume the availability of a set of high-resolution exemplars. These exemplars contain a person with the same identity as the person depicted in the LR image. Specifically, we assume that the exemplars possess detailed facial features, e.g. eye shape, iris color, gender etc. that are expected to appear in the SR image. These exemplars are used to provide the network with these facial features that are invisible in the LR image.

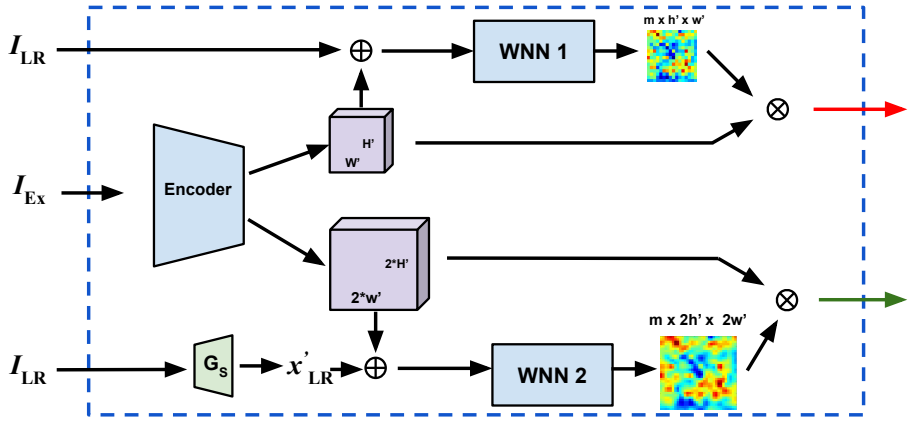

Figure 6.3: The proposed PWAve module in this chapter. To concatenate with two times higher resolution feature map, we train a small generator to get x'_{LR}. For the whole model, please see Fig. 6.2.

6.3.1 Architecture

The generator model consists of two parts: the image super resolution block (ISR) and the PWAve module (see Fig. 6.2). The ISR block contains M_1 upsample blocks and M_2 residual blocks . The ISR block takes the LR image (I_{LR}) as input and integrates at different parts of its feed-forward pass combined feature maps from the exemplars as generated by the PWAve module.

Besides face super-resolution, the combination of feature maps from the input LR image and the exemplars allows the model to dynamically generate HR images with similar facial features to those present in the exemplars.

PWAve module

For each LR input we randomly assign a set of HR exemplars, thus not every exemplar contains the needed information. The goal of this module is to learn how to perform a good combination of the set of intermediate feature maps computed from the exemplars. The idea is to provide the freedom to the model to learn a different, and perhaps more suitable, combination method rather than the commonly used average method [111] or image-level weighted-average. Our intuition is that the exemplars could depict different features, e.g. facial expression, angle, makeup, etc., that are in the LR image and are expected to appear in the SR image. Therefore, it is meaningful to consider the exemplars

when the model generates the HR image so that the model can take into account proper regions of the exemplars.

The PWAve module consists of an encoder E_{Ex} and two pixel weight generation networks (WNN). Assume the batch size is 1, E_{Ex} encodes the exemplars $I_{Ex} \in \mathbb{R}^{m\times 3\times H\times H}$, with height and width H, into feature maps $f_{Ex} \in \mathbb{R}^{m\times K\times H'\times H'}$ of different scales, m is the number of HR exemplars (cardinality of the exemplar set), respectively. We use two scales: the same scale with and two times higher than I_{LR}. WNN takes f_{Ex} and I_{LR} of corresponding size and generates the pixel-wise weight matrix W ($W \in \mathbb{R}^{m\times 1\times H'\times H'}$) across the exemplars. Afterwards, W is L_1-normalized along the first dimension. In order to concatenate with the two times higher resolution feature maps and provide a more precise guidance in WNN-2, we use a small generator G_S to generate $I^{'}_{LR}$ rather than just up-sample I_{LR}. Finally, the combined exemplar feature maps f^{c}_{Ex} are calculated by applying dot-product between f_{Ex} and W, followed by the summation along the first channel, i.e. $f^{c}_{Ex} \in \mathbb{R}^{1\times K\times H'\times H'}$. Different from Liu et al. [112], we generate pixel-wise weights while taking into account the LR image itself. Taking into account I_{LR} (and $I^{'}_{LR}$) in the combination process is a more direct way to identify the important region on the exemplars compared with simply relying on loss penalty.

Discriminator (Critic)

The architecture of the discriminator (critic) is similar to StyleGAN[90], where it takes either super-resolved images or original images as input and tries to match the distribution between the super-resolved image and real ground-truth HR images.

6.3.2 Objective Functions

We formulate this task as a supervised learning problem given the simplicity to obtain a LR/HR image pair from a single high-resolution image.

Content Loss (L_c)

We apply a $L1$ loss on the ground truth HR image I_{HR} and the super-resolved image I_{SR}.

$$\mathcal{L}_{\mathrm{c}} = \|I_{SR} - I_{HR}\|_1 \tag{6.1}$$

We also calculate the content loss for G_S,

$$\mathcal{L}_{\mathrm{c}}^{\mathrm{s}} = \|I_{SR}^{s} - I_{HR}^{d}\|_1 \tag{6.2}$$

where I_{HR}^{d} and I_{SR}^{s} are the real image that is downsampled to match the resolution of output of G_S and the output of G_S, respectively.

Perceptual Loss (L_p)

This loss aims at preserving the face appearance and identity information of the depicted persons. We use the perceptual similarity model Φ_p [198], which is trained on an external dataset, and the last layer before the classification layer of the face emotion model Φ_{ID} [2] to calculate the perceptual loss.

$$\mathcal{L}_{\mathrm{p}} = \|\Phi_p(I_{SR}) - \Phi_p(I_{HR})\|_2^2 + \|\Phi_{ID}(I_{SR}) - \Phi_{ID}(I_{HR})\|_2^2 \tag{6.3}$$

where I_{HR} and I_{SR} are the real HR image and corresponding super-resolved HR image, respectively. Similarly, we apply the perceptual loss on $I_{LR}^{'}$ and I_{SR}^{d}.

$$\mathcal{L}_{\mathrm{p}}^{s} = \|\Phi_p(I_{SR}^{s}) - \Phi_p(I_{HR}^{d})\|_2^2 + \|\Phi_{ID}(I_{SR}^{s}) - \Phi_{ID}(I_{HR}^{d})\|_2^2 \tag{6.4}$$

Adversary Loss (L_{adv})

In order to generate a more realistic and sharper image, we use the adversary loss to match data's distribution. Furthermore, here we use Wasserstein GAN with gradient penalty (WGAN-GP) [64] in order to have a more stable training process. The critic loss, used to update the discriminator, is defined as follows:

$$\mathcal{L}_{\mathrm{critic}} = D(I_{SR}) + D(I_{HR}) - \lambda_{gp}\mathcal{L}_{gp} \tag{6.5}$$

where $D(.)$ is the discriminator (critic) network and λ_{gp} is the coefficient parameter for gradient penalty. The adversary loss for the generator part is $-D(I_{SR})$.

Total loss (L_{total})

With the previous terms in place, the total loss is defined as:

$$\mathcal{L}_{\mathrm{total}} = \left[\mathcal{L}_c + \lambda_1\mathcal{L}_p + \lambda_2\mathcal{L}_{adv}\right] + \left[\mathcal{L}_c^s + \lambda_3\mathcal{L}_p^s\right] \tag{6.6}$$

where λ_1, λ_2 and λ_3 are the trade-off coefficients of the model. The losses in the first bracket will update the whole model except for G_S, where G_S is updated by the losses in the second bracket.

6.4 Evaluation

In this section, we start with the introduction of the dataset used in our experiments. This is followed by qualitative and quantitative results obtained by our method. We also conduct a user study on the quality of our super-resolved images. Then, we perform a qualitative comparison of our method w.r.t. methods from the literature. In addition, we conduct an ablation study to show the effectiveness of the number of exemplars and the proposed PWAve module. Finally, we show our method is capable of dynamically introducing features on the generated images via the exemplars.

Dataset

We use two datasets depicting human faces: CelebA[114] and WebFace[184]. CelebA has 202,599 images with 10,177 identities, each identity has 20 images on average. We follow Yu et al. [188] and crop the images to the size of 128×128 and drop the identities which have less than 5 images. We downsample the original 128×128 image to the size of 16×16 and 8×8. We also follow the training/testing splits in the original CelebA dataset. There is no identity overlapping between these splits. Images from the WebFace dataset have a higher 256×256 resolution. There are 10,575 identities and each of them has several images, ranging from 2 to 804. For each identity, we follow Li et al. [103] and select the top 10 best quality images. We drop the identities which have less than 10 images. The images of the first 9121 identities are used for training and the rest of them for testing. Unlike [103], for both datasets, we select the LR/HR images and the corresponding exemplars at random. That is, there is no constraint on the facial angle and expression.

Implementation details

We implement our model in PyTorch[134]. Height and width of I_{LR} are both H_{LR}. For the CelebA dataset with the scaling factor of 8/16, the setting is M_1=3/4, M_2=1/1, H_{LR}=16/8, N=8. For the WebFace dataset with the scaling factor of 8/16, the setting is M_1=3/4, M_2=1/1, H_{LR}=32/16, N=4. We use Adam [93] optimizer with β_1=0 and β_2=0.99. The initial learning rate is set to

Method	SSIM (CelebA/WebFace)	PSNR(dB) (CelebA/WebFace)
Bicubic (×8)	0.61/0.68	20.72/22.04
Ours(×8, w/o Discriminator)	0.74/0.78	23.45/24.92
Ours(×8, w/ Discriminator)	0.72/0.76	23.18/24.38
Bicubic (×16)	0.48/0.57	17.56/19.04
Ours (×16 w/o Discriminator)	0.62/0.67	19.82/20.55
Ours (×16 w/ Discriminator)	0.59/0.61	19.13/19.42

Table 6.1: SSIM and PSNR scores on CelebA and WebFace dataset.

0.003 for the whole model except for the two WNNs, whose initial learning rate is 0.0001.

6.4.1 Qualitative and quantitative results

Fig. 6.4 and Fig. 6.5 shows qualitative results from the celebA and WebFace dataset. It is clear that our model can recover most of the details, such as wrinkles, iris color and stubble. In addition, our model is robust to changes in face angle, i.e. not only the front view but side view images can be super-revolved as well. Even when a scaling factor of 16 is used, it is interesting to see that the generated images (the last column of Fig. 6.4) can also keep the original identity and details for most of the cases. For the fifth example in Fig. 6.4, the 8×8 LR image has a black region around the eyes due to the downsampling of 16 times, which suggests the model to generate the sunglasses.

For the WebFace dataset with the scaling factor of 16, the task is harder since the original size of the image is larger (256×256), they are not aligned and have much more non-facial regions. However, the super-resolved images are still reasonable good based on the 16×16 input and the corresponding exemplars. Table 6.1 shows SSIM and Peak Signal-to-Noise Ratio (PSNR) scores on the testing set.

To assess how the PWAve module helps the network in the generation process, we visualize the weight matrix W, generated by the PWAve module, using the jet color scale and overlaying it on top of each of the exemplars. Fig. 6.6 shows some examples. The generated heatmaps clearly show that the PWAve module does learn how to select different parts from the exemplars rather than doing a random selection or a uniform average across images. For example, the second row in the left part of this figure, the eyes and mouth parts of the first exemplar image have very low weight. This is reasonable since the face of the target image has a more frontal angle and does not have glasses. If both the exemplar and target images have similar face angle, expression, etc., the PWAve

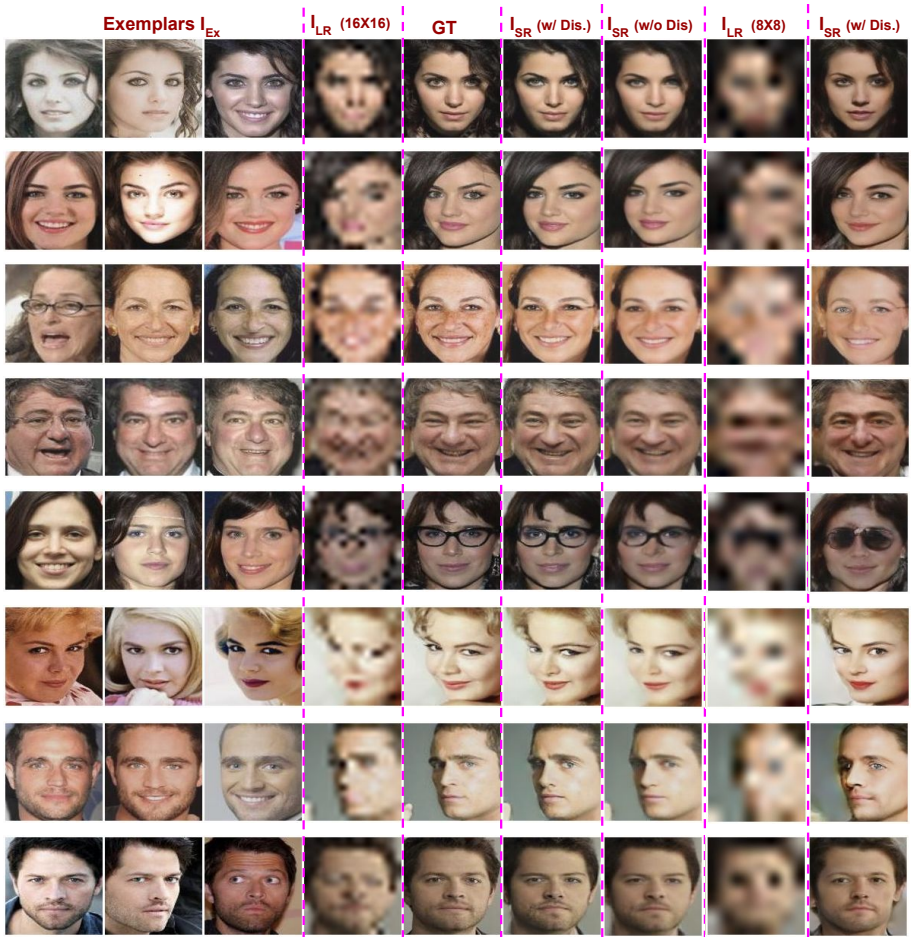

Figure 6.4: Qualitative results from the CelebA dataset. We show two scaling factors: $\times 8$ and $\times 16$. The resolution of HR images is 128×128. All the images are from testing set.

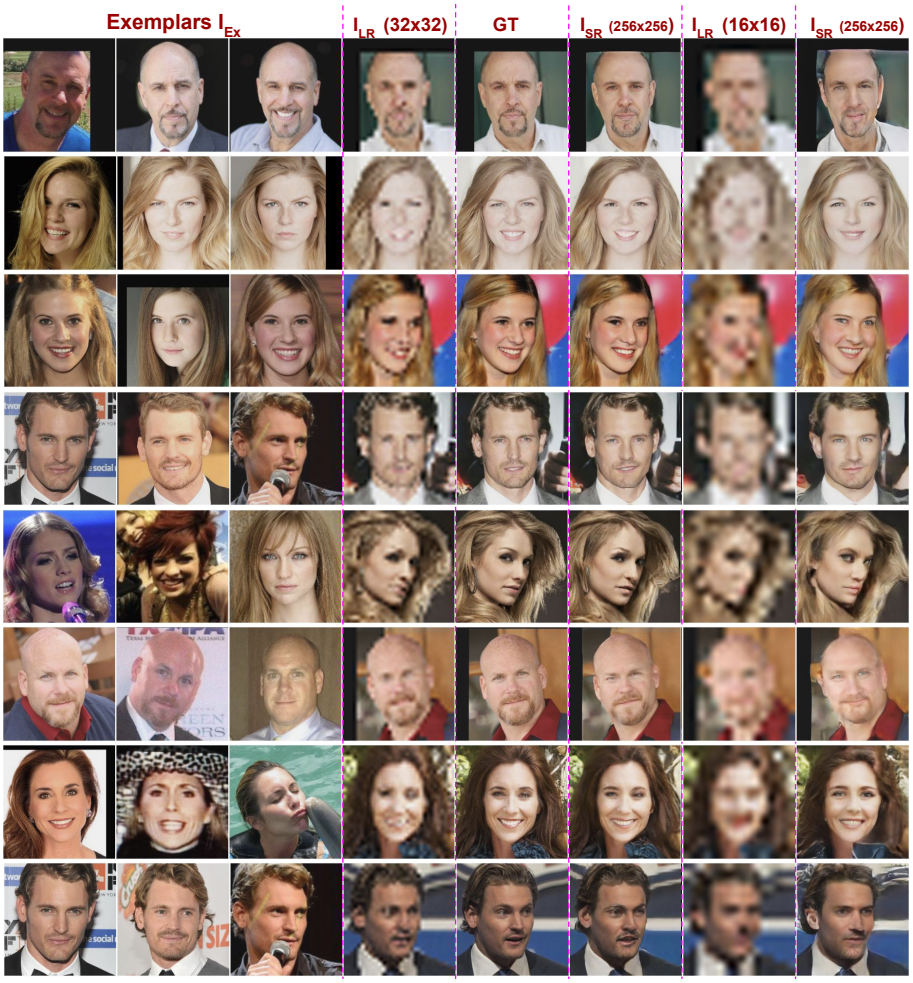

Figure 6.5: Qualitative results on the WebFace dataset. We show two scaling factors: $\times 8$ and $\times 16$. The resolution of HR images is 256×256 All the images are from testing set.

Error rate (%)	37.70 ±12.53
TPR (%)	59.79±15.35
TNR (%)	66.80±17.80
FPR (%)	33.34±17.85
FNP (%)	37.29±19.30

Table 6.2: Results from our user study on the celebA dataset.

module will generate weights that resemble an average operation. This can be seen in the third row on the right part. Table 6.1 shows the scores without discriminators (i.e. no adversary loss), we also remove the perceptual loss on the CelebA dataset (×8), resulting 0.69/22.35 for SSIM/PSNR. This shows that using only the L_1 content loss is not sufficient to produce realistic HR images while preserving identity.

6.4.2 Survey

We conduct an user study in order to quantitatively assess the quality of our super-resolved images as perceived by humans. Our survey was joined by 51 participants from around the world. From these participants, 35 have experience in computer science or informatics. We refer to this group as "CS". The rest of them do not have such experience, we refer to them as "Non-CS".

The survey was composed by two parts. The first part aims to check whether the participants could distinguish between the super-resolved images from real GT (HR) images. For each question, we put one HR image (128×128) next to the upsampled version of the corresponding LR image (16×16). Then we ask users to judge whether the shown HR image is the real, i.e. original, one. The HR image is randomly sampled from the ground truth and the images generated by our method. All the generated images shown on the survey are randomly sampled from our testing set. In total, there are 200 questions, half of them with a ground truth HR image. For each survey we randomly sample 25 questions. Before starting the survey, we also "train" the users by showing 12 examples with label (Real or Fake) in order to make them familiar with the task at hand.

The second part is more subjective, it consists of asking the participants to judge which type of the super-resolved images they consider more realistic: the sharp one (with Discriminator) or the smooth one (without Discriminator).

Table 6.2 shows the results of the first part of the study. The error rate is close to the random guess (50%). The FPR and FNR are quite balanced. In terms of the two user groups, CS group gets (34.57±11.22)% error rate while the

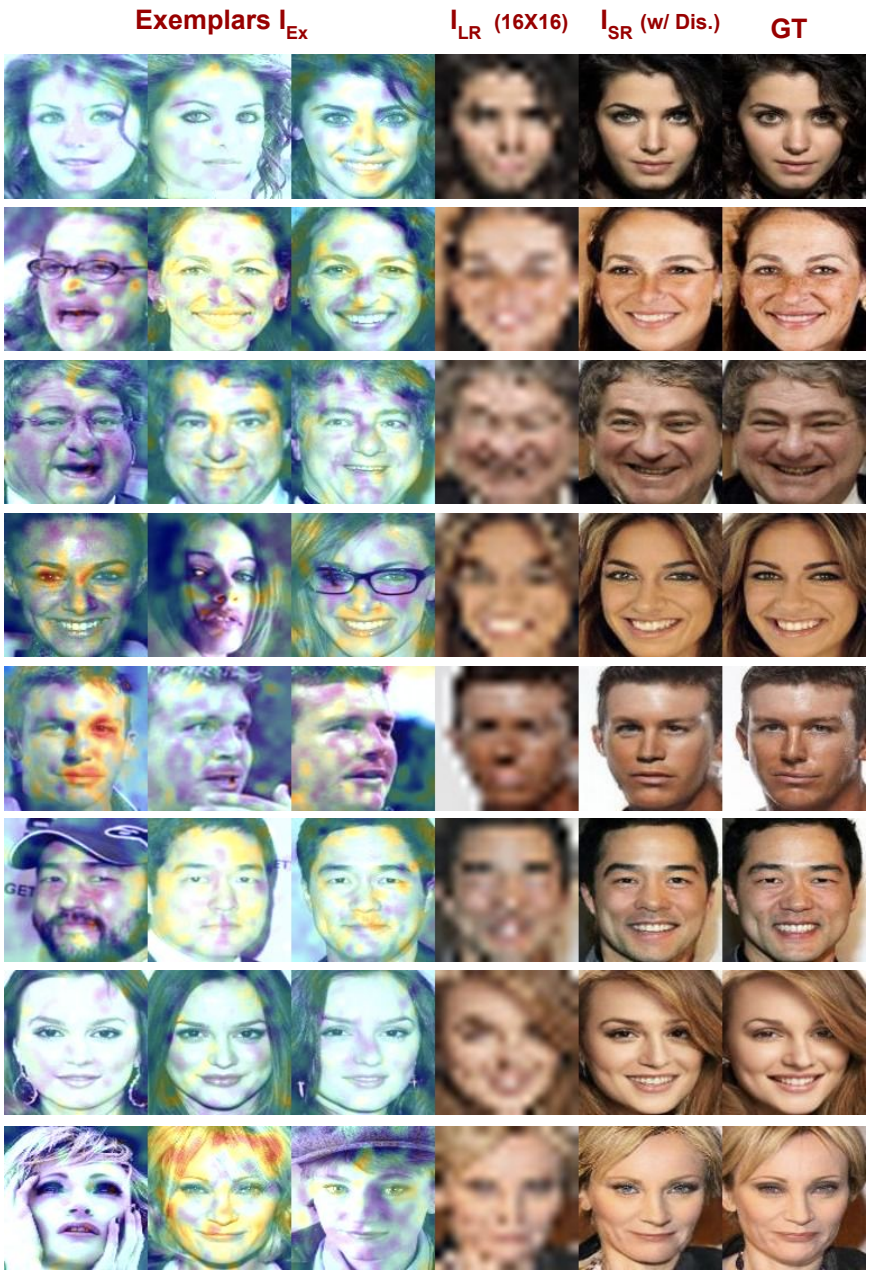

Figure 6.6: The visualization of W (Here $W \in \mathbb{R}^{3\times1\times16\times16}$) generated by the PWAve module. Please note W is normalized along the second channel. The heatmap is in jet color space, the warmer the color, the higher the weight.

Non-CS group achieves (43.62±13.51)%, respectively. As for the second part, although the SSIM and PSNR scores are lower (Table 6.1), around 60% of the participants think the sharp images look more realistic than the smooth ones. Therefore, we will focus on the sharper qualitative results in the next sections.

6.4.3 Qualitative comparison with respect to the state-of-the-art

Why there is no quantitative comparison?

We do not provide a quantitative comparison w.r.t. existing methods for the following reasons:

- The SSIM and PSNR scores do not provide a definitive answer: blurry images may have a higher SSIM and PSNR score [87][70] since the optimal solution to minimize reconstruction error in image space is averaging all possible solutions [41][44][99]. This observation was further ratified in our previous experiment.
- Existing methods are implemented in different frameworks and, currently, there is no unified benchmark for this task. Re-implementation may raise doubt on the quality of the code, and the used data splits and pre-processing. Therefore, we only provide the conducted user-study (Sec. 6.4.2) and a qualitative comparison.

In this experiment, the presented qualitative results (images) are taken directly from the corresponding papers. These selected images will constitute the point of comparison. Fig. 6.7 shows a qualitative comparison on celebA dataset w.r.t. Yu et al. [188] and on the WebFace dataset w.r.t. Li et al. [103] and Dogan et al. [41]. Generally, our model can generate more detailed results and preserve the person's original identity. Yu et al. [188] uses a manually rotated LR image as input while a few spatial transform networks are utilized to compensate this manual rotation. The results from [188] change the identity and the face shape for some cases. Moreover, some details are missing, e.g. the moustache is lost during the super resolution process in the first image in the top part of Fig. 6.7. The images generated by [103] have blurry hair as well as a blurry background. [41] overcomes this problem but also loses some details, e.g. the eyes' shape (i.e. the third row in the bottom left of Fig. 6.7) and the iris color (i.e. the second row in the bottom right of Fig. 6.7).

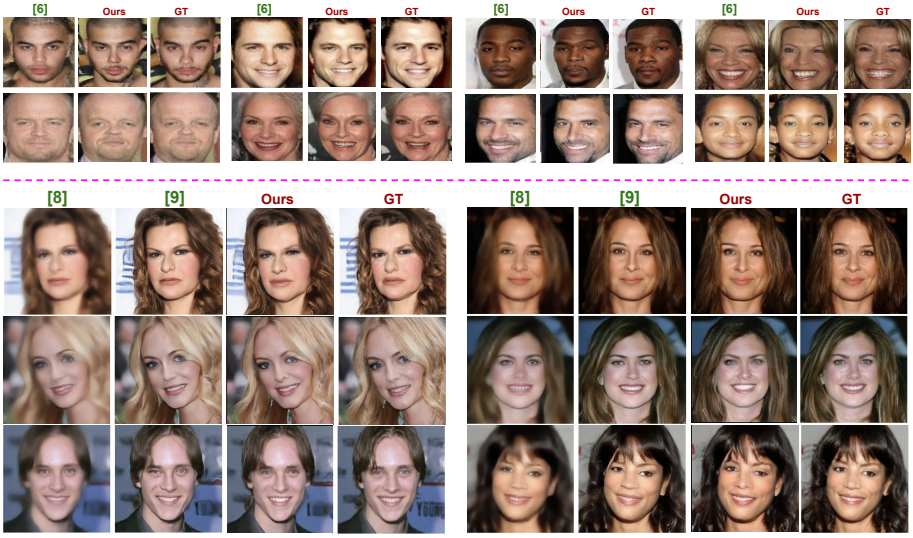

Figure 6.7: Top: Qualitative comparison w.r.t. state-of-the-art methods on the celebA dataset. The resolution of I_{LR} is 16×16 and the HR image is 128×128. The first column shows the results from [188]. Bottom: Qualitatively comparison w.r.t. state-of-the-art baselines on WebFace dataset. The I_{LR} has the resolution of 32 × 32 and HR image is 256 × 256. The first and second column are results from [103] and [41].

6.4.4 Ablation study

Our ablation study aims at assessing the effect of i) the number of considered exemplars, and ii) the PWAve module. Towards this goal, we conduct two experiments: gradually increase the cardinality m of the exemplar set, from 0 (no exemplar) to 5, and replace the PWAve module by simply averaging the feature maps of exemplars. Fig 6.8 shows the SSIM and PSNR scores under each setting. The trend is clear - more exemplars bring more benefits. The model with scaling factor of 16 can gain more, this is because the I_{LR} contains less information and the model can gain extra information from the exemplars.

Regarding the PWAve module, the scores are also better than the averaging method. To give a more straightforward comparison on this module, we show some examples in Fig. 6.9. These examples show that the combination of the feature maps will influence the details in the generated image, such as mouth, eyebrows, etc. With the help of the weights generated by the PWAve module, the model can take into account useful regions from the exemplars and produce

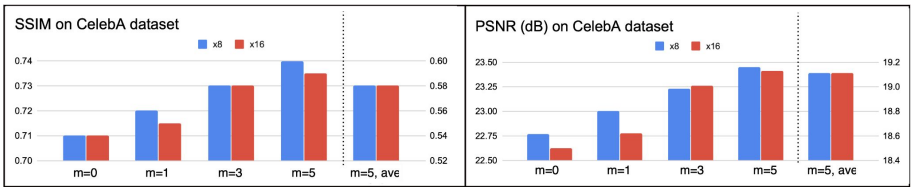

Figure 6.8: Ablation study on the cardinality of the exemplar set and feature map fusion method. The blue bar and the left y-axis are for the experiment with the scaling factor of 8 while the red bar and the right y-axis are for 16. Higher values are better.

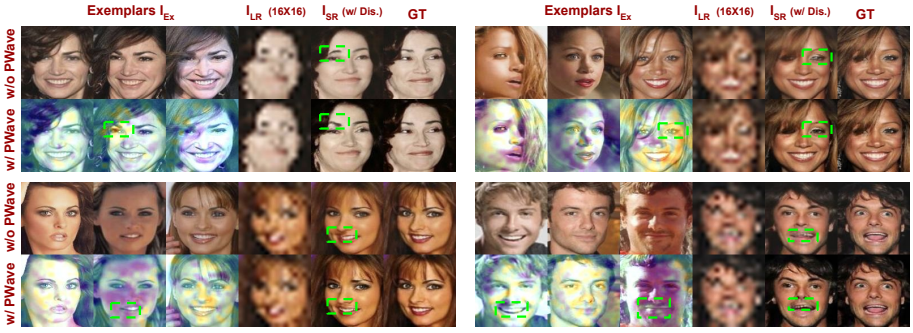

Figure 6.9: Examples of ablation study on the PWAve module. For each set, the first row is without PWAve module (standard average). We show the heatmap of W generated by PWAve on each the second row. Please give your attention to the region within the dashed green box highlighting keys aspects of the combination process.

a HR image closer to the ground truth.

6.4.5 Facial features editing via exemplars

Based on the assumption that for most cases, the exemplars possess facial features such as eye shape, iris color, gender etc. that are in the LR image and are expected to appear in the SR image. we have shown that the exemplars do provide useful information to help the model super-resolve the LR image. This observation raises the question, what if we use exemplars with different facial features? Will the super-resolved images still contain the original features from the LR image or adopt the new features from the exemplars? In the

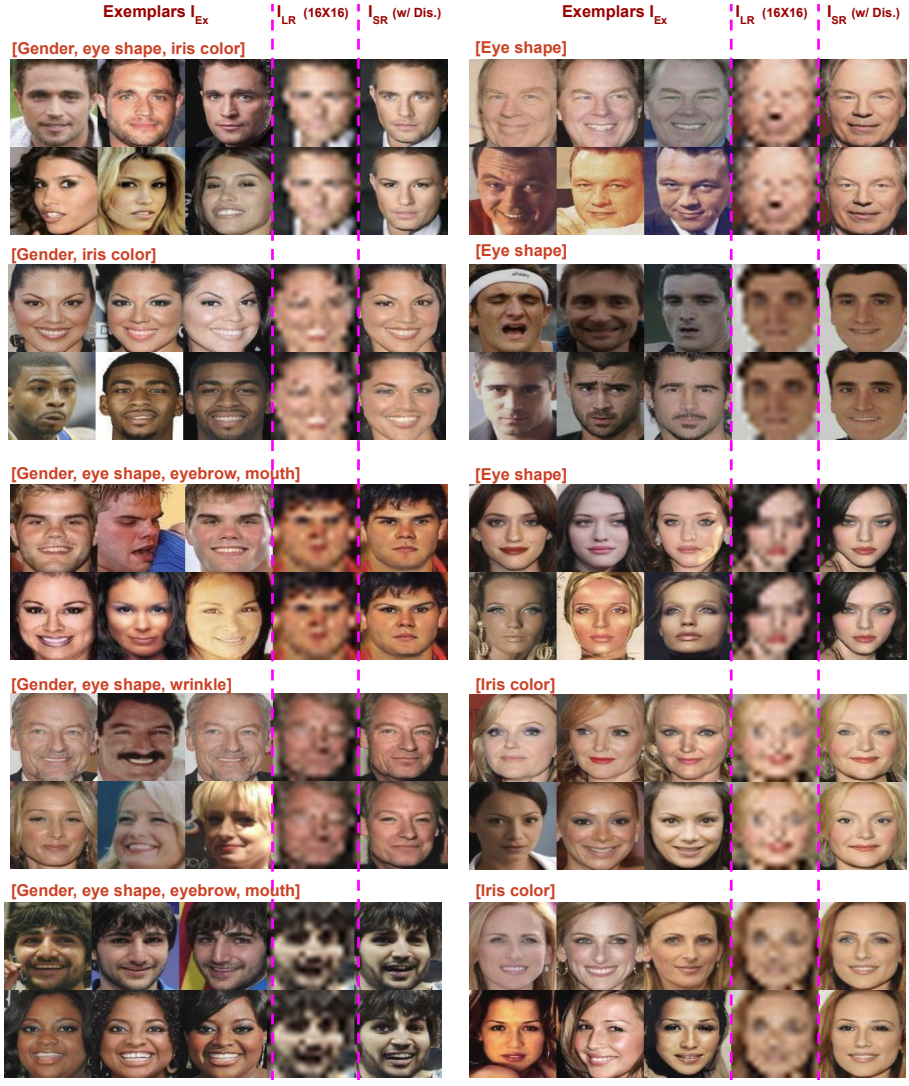

Figure 6.10: Examples of editing/modifying facial features via exemplars. In this figure, we show the hallucinated images guided by exemplars I_{Ex} with the same and different identity. The edited facial features are displayed on the top of each set.

Figure 6.11: Examples of editing/modifying facial features via exemplars. In this figure, Exemplars I_{Ex} has the same identity but different facial features.

CelebA dataset, we still find some cases where the facial features within the same identity are different. We run two experiments considering exemplars with different and the same identity. Please note, in this experiment we just replace the exemplars in the testing phase, no re-training is required.

Fig. 6.10 and Fig. 6.11 clearly answer the question above. For both experiments, the super-resolved images adopt the features from exemplars, such as gender, makeup, eyes, mouth etc. More specifically, if the gender of exemplars is different (Fig. 6.10), the model will change the gender as well as other facial features. We believe changing gender is more difficult since it is a high-level characteristic which is related to other low-level attributes such as eyes, eyebrows, mouth etc. If the gender of the exemplars is the same (Fig. 6.11), then the identity of the original HR image will be maintained but changes in corresponding facial features occur. Compared with [188], benefiting from using the exemplars, our method can generate face HR image with arbitrary facial features rather than only the pre-defined ones. The experiment shows that our model is capable of dynamically introducing features on the generated images via the exemplars. This suggests that our model can be also applied to subtle image editing tasks without the need of re-retraining or additional modifications on the original model.

6.5 Ethical Discussion

In this section, we briefly discuss potential ethical issues related with our method. The datasets we use (i.e., CelebA and WebFace) are public and collected online. The images depict public individuals such as celebrities, singers, movie stars, etc. These datasets have been widely used in the computer vision community for various tasks. Therefore, we have no control over the underlying gender, age, race, etc., distributions of these datasets. Compared with the method from [125], to the best of our knowledge, our model will not inject any gender or race bias after the face hallucination process. This is because the usage of the exemplars. They can guide the model so that these biases have a low effect on the hallucination process. Additionally, unlike the popular deepfake technique, where the model can swap any face to another person in any scene, our method focuses more on restoring the HR version of an existing LR image. Our method can be applied for entertainment by restoring the HR face from an old, LR digit image. We think the users have to ensure they have the copyright of the images used and will not do anything harmful to the person on the image.

6.6 Current Research Trends

One of the limitations of this work is that we do not generate very high resolution (e.g. 1024× 1024) output. Based on styleGAN [90], recently Menon et al. [125] propose a method to hallucinate the HR image from 16×16 to 1024×1024. The method can generate multiple reasonable HR images based on the LR input because there is no strict constraint. The different facial attributes (e.g. eyes, hair etc) come from sampled noise just like styleGAN. In addition, rather than directly using the HR exemplars, Chan et al. [22] propose to utilize a generative latent bank to guide the model. In their work, the authors leverage a pre-trained generator as a latent bank to provide priors for texture and detail generation. This method is able to capture the distribution of the images and has potentially unlimited size and diversity for the output.

6.7 Conclusion

In this chapter, we propose to use multiple HR exemplars as conditions to guide the model to hallucinate LR images. This is complemented by the proposed PWAve module which is capable of selecting and combining the most useful component features across the exemplar set. We empirically show the effectiveness of using more than one exemplar and that our method outperforms baselines from the literature. In addition, benefiting from the exemplars, our model can dynamically generate HR face images with arbitrary facial features.

Chapter 7

Multiple Instance Learning

In the last chapter, we move our interest to the general Multiple Instance Learning (MIL) problem, where a set can be composed by images (instances) from different classes and only the set-level label is accessible. Different from the previous two chapters, here we mainly investigate how to learn a good set representation. It is also interesting to study whether any information of the unannotated instances can be obtained via learning the set representation.

In this chapter, we propose an iterative learning strategy to learn the set representation. Particularly, we use LSTM networks to realize this strategy, although LSTMs have a proven track record in analyzing sequential data. While not often used for unordered data, we show LSTMs excell under this setting too. In addition, we show that LSTMs are capable of indirectly capturing instance-level information using only set-level annotations. Thus, they can be used to learn instance-level models in a weakly supervised manner. Our empirical evaluation on both simplified (MNIST) and realistic (Lookbook and Histopathology) datasets shows that LSTMs are competitive with or even surpass state-of-the-art methods specially designed for handling specific MIL problems. Moreover, we show that their performance on instance-level prediction is close to that of fully-supervised methods.

The presented content is based on the following paper:

- K. Wang, J. Oramas M, T. Tuytelaars, In Defense of LSTMs for addressing Multiple Instance Learning Problems. Asian Conference on Computer Vision (ACCV) 2020.

7.1 Introduction

Traditional single-instance classification methods focus on learning a mapping between a feature vector (extracted from a single instance) w.r.t. a specific class label. In a complementary fashion, MIL [146] algorithms are tasked with learning how to associate a set of elements, usually referred to as a "set", with a specific label. In comparison, MIL methods usually require weaker supervision in the form of set-level labels.

The MIL problem has a long history, and various solutions have been proposed over time. Here, we advocate the use of standard LSTM networks in this context, as a strong baseline, yielding competitive results under a wide range of MIL settings.

Long short-term memory (LSTM) networks [75] have been proposed as an extension over standard recurrent neural networks, to store information over long time intervals in sequential data. They have been used extensively and very successfully for modeling sentences (sequences of words) in text documents [158], e.g. for machine translation [159] or sentiment analysis [95]. Later, they have been employed in several other fields including computer vision [109, 149, 1, 208, 154] and speech processing [58, 158]. LSTMs provide great flexibility for handling data sequences. There is no need to know the length of the sequence beforehand, and they can operate on sequences of variable size. In addition, they are capable of accumulating information by using a memory mechanism with add/forget[53] functionality.

As they need the input to be provided as a sequence, LSTMs do not seem an appropriate choice for analyzing unordered sets at first - but is that so ? Obviously, the capability to remember the temporal (order) information can be attributed to the *memory* ability of LSTMs. This memory ability is capable of capturing other types of information beyond order as well. Take the LSTMs used for action recognition as an example. For some finegrained actions (e.g. opening or closing a door), the order of the events is key and this is picked up by the LSTM. However, for other actions the context information provides the most important cue (e.g. playing tennis or cooking). This does not depend on the temporal order, but can still be learned using LSTM.

Starting from an unordered set, we can always transform it into a sequence by imposing a random order on the instances, making it suitable for LSTMs. The order is not relevant, but that does not matter: the LSTM can still process the data and extract useful information from it. In fact, this is also how humans often deal with unordered sets: e.g, if one is asked to count the number of rotten apples in a basket, most of us would just pick the apples one by one, in random order, inspect them and keep track of the count. The order does not matter,

but nevertheless, treating them in a sequential order comes very naturally.

The observations above clearly hint to a promising capability of LSTMs for addressing MIL problems. Yet, LSTMs are not often used in this way (see our related work section for a few notable exceptions). Therefore, we present a systematic analysis on the performance of LSTMs when addressing MIL problems.

More specifically, we conduct a series of experiments considering different factors that may affect the performance of LSTMs. First, we consider the standard MIL problem [4, 20, 49], with sets of instances without sequential order. Second, we study the effect of the order in which the instances of each set are fed to the LSTM network. Likewise, in a third test, we investigate the influence of the cardinality (size) of the set on performance. Fourth, we assess the effect that the complexity of the data has on the previous observations. Toward this goal we conduct experiments considering sets derived from the MNIST dataset [97], clothing-item images from the Lookbook dataset [185] and Histopathology images [81]. Fifth, we inspect how the internal state of the LSTM changes when observing each of the instances of the set. Moreover, we propose an LSTM-based framework that can predict the instance-level labels by only using set-level labels, in a weakly supervised manner.

Our contributions are three-fold:

- We advocate the application of LSTM networks on general MIL problems, as a means for encoding more general underlying structures (i.e. not limited to ordered data) within sets of instances.
- We conduct an extensive systematic evaluation showing that LSTMs are capable of capturing information within sets that go beyond ordered sequences. Moreover, we show that their performance is, surprisingly, comparable or even superior to that of methods especially designed to handle MIL problems.
- We propose a framework for weakly supervised learning based on LSTM, capable of modeling distributions at the instance-level, using only set-level annotations.

7.2 Related Work

Efforts based on LSTMs/RNNs aiming at modelling unordered sets are quite rare. Graves et al. [59] and Graves et al. [60] propose a memory-based recurrent architecture and apply it on sorting [59] and traversal of graphs [60], where

the input data can be regarded as unordered sets. Han et al. [68] considers a fashion outfit to be a sequence (from top to bottom) and each item in the outfit as an instance. Then, a LSTM model is trained to sequentially predict the next item conditioned on previous ones to learn their compatibility relationships. Later, Yazici et al. [183] used a CNN-RNN model to perform multiple label predictions, where LSTMs were used to decode the labels in an unordered manner. Different from this work which focused on the decoding part, we investigate the encoding of the unordered sets. Zaheer et al. [190] proposed to learn permutation-invariant set representations by summing features across instances and applying non-linear transformations. This can be regarded as a specific case of [81] where the weight of the instances are uniform.

Pabbaraju et al. [133] uses LSTM to capture the function over the sets, which is different from ours, where we use LSTM to model the set representation and learn the instance label from the set label. These works either use LSTM to handle unordered data on some specific settings [68, 190] or use them just as side experiments [59, 60]. Here, we propose the use of LSTMs to address more *general* MIL problems.

On the task of modeling general set representations, we position our approach w.r.t. efforts based on neural networks, specifically those with deep architectures since our work is based on LSTMs. Please refer to [4, 20, 49] for detailed surveys covering non-deep methods. Ramon et al. [138] proposed a multiple instance neural network to estimate instance probabilities . This idea is further extended in Wang et al. [171] which uses a neural network to learn a set representation and directly carry out set classification without estimating instance-level probabilities or labels. In parallel, Ilse et al. [81] proposed an attention mechanism to learn a pooling operation over instances. The weights learned for the attention mechanism on instances can serve as indicators of the contribution of each instance to the final decision – thus, producing explainable predictions. Liu et al. [112] proposed a similar idea, using the computed set representations, to measure distances between image sets. Yan et al. [180] proposed to update the contributions of the instances by observing all the instances of the set a predefined number of iterations. Along a different direction, Tibo et al. [161] proposed a hierarchical set representation in which each set is internally divided into subsets until reaching the instance level. Very recently, Ming et al. [162] proposed to consider the instances in the sets to be non-i.i.d. and used graph neural networks to learn a set embedding.

Similar to [81, 171] we embed the instance features from each set into a common space and the set representation is used to make direct set predictions. Similar to [162, 180] we aim at learning the underlying structure within the sets. Different from [162], our method does not rely on hand-tuned parameters, e.g. manual graph construction. Moreover, the improvement in performance displayed by

our method is not sensitive to the possible lack of structure within each set. Compared to [180], our method only requires *a single pass* through all the instances. Moreover, our method is able to go beyond binary classification tasks and handle more complex classification and regression tasks. Finally, most of the works mentioned above operate under the standard Multiple Instance (MI) assumption. In contrast, the proposed approach is able to learn the underlying structure of sets of instances, thus, being robust to several MI assumptions/problems Foulds et al. [49].

7.3 Methodology

We begin our analysis by defining the different parts that compose the proposed pipeline. First, we formally define MIL problems and draw pointers towards different MI assumptions that they commonly consider. Then, we introduce the LSTM-based pipeline that will be considered to model sets of instances throughout our analysis.

7.3.1 Underlying Structures within Sets of Instances

As was said earlier, underlying sequential structures between the instances within a set is a cue that LSTMs are capable of encoding quite effectively. In fact, this capability have made them effective at handling problems defined by these sequences, e.g. actions, speech, text. However, this sequential order is just one of many possible underlying structures that could be present between the instances within the sets processed by a LSTM. Encoding these underlying structures and making predictions about sets of instances is the main objective of MIL [4, 20, 49]. We will conduct our analysis from the perspective of MIL, where LSTMs will play an active role in modeling the underlying set structure.

Given the set $\mathbb{I}_j=\{I_1, I_2, ..., I_m\}$ of instances x_i with latent instance-level labels $\mathbb{C}_j=\{c_1, c_2, ..., c_m\}$, traditional MIL problems aim at the prediction of set-level labels y_j^{set} for each set $\mathbb{I}_j$. The MIL literature covers several underlying set structures, referred to as *assumptions*, that have been commonly considered in order to define set-level labels. We refer the reader to [4, 20, 49] for different surveys that have grouped these assumptions based on different criteria.

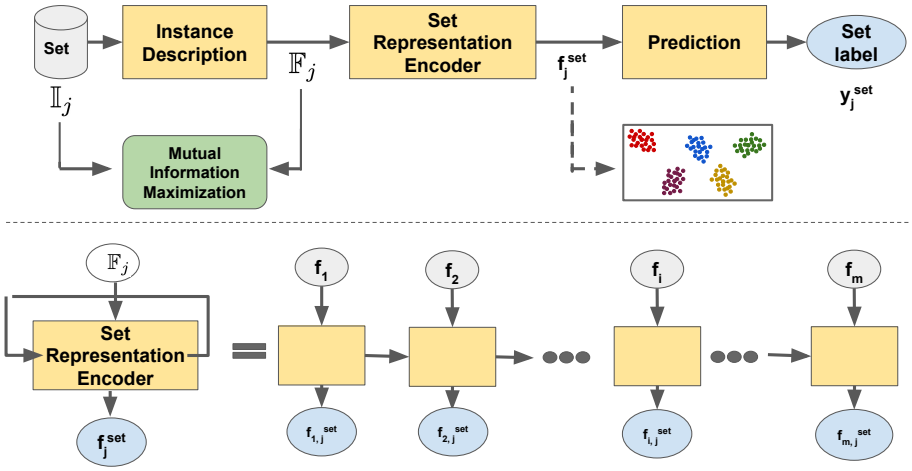

Figure 7.1: Top: Proposed approach pipeline. Bottom: Iterative set representation encoder. The set representation $f_{i,j}^{set}$ is updated each time the representation f_i of an element is observed.

7.3.2 Proposed Pipeline

The proposed pipeline consists of three main components. Given a set $\mathbb{I}$ of m instances I_i, each of the instances I_i is encoded into a feature representation f_i through the *Instance Description Unit (IDU)*. Then, each element is fed to the Iterative *Set Representation Encoder (SRE)*, producing the aggregated set representation m. Finally, a prediction $\hat{y}$ is obtained by evaluating the set representation via the *Prediction Unit*.

Instance Description Unit

This component receives the set of instances in raw form, i.e. each of the instances $I_i \in \mathbb{R}^d$ that compose it, in its original format. It is tasked with encoding the input set data into a format that can be processed by the rest of the pipeline. As such, it provides the proposed method with robustness to different data formats/modalities. More formally, given a dataset $\{\mathbb{I}_j, y_j\}$ of sets $\mathbb{I}_j$ paired with their corresponding set-level labels y_j^{set}, each of the sets $\mathbb{I}_j$ is encoded into a feature $\mathbb{F}_j = \{f_1, f_2, ..., f_m\}$. This is achieved by pushing each of the instances I_i that compose it, through a feature encoder $\tau(.)$ producing the instance-level representation $f_i = \tau(I_i), f_i \in \mathbb{R}^K$.

Selection of this component depends on the modality of the data to be processed, e.g. VGG [153] or ResNet [72] features for still images, Word2Vec [126] or BERT [39] for text data, or rank-pooled features [47] or dynamic images [15] for video data.

Maximizing Mutual Information from Instances

Mutual information can be used to measure the (possibly non-linear) dependency between two variables, noted as $I(A;B)$. Maximizing mutual information between input instance and its representation helps the model learn a better representation [74][182]. It is useful as a regularizer especially when learning a model from scratch. In our method, we follow Devon Hjelm et al. [74] where the total objective function is:

$$L = \alpha \cdot max(MI_{global}) + \beta \cdot max(MI_{local}) + \gamma \cdot PriorMatching \qquad (7.1)$$

MI_{global} and MI_{local} are the global and local Mutual information where the latter one is calculated between intermediate feature map and final representation. $PriorMatching$ is used to match the output of the IDU to a prior: combined with the maximization of mutual information, Eq. 7.1 can constrain representations according to desired statistical properties. Please refer to [74] for more details regarding the derivation of Eq. 7.1.

Set Representation Encoder

The main goal of this component is to derive a set-level representation $f_j^{set} \in \mathbb{R}^K$ that is able to encode all the instances I_i, and any possible underlying structure between them. As mentioned earlier, LSTM is utilized to model the underlying [unordered] structure in the set.

We aim at learning a set representation that is independent of both the cardinality m of the set and the nature of the underlying structure. Starting from the element-level representations $\mathbb{F}_j$ computed in the previous step, this is achieved by iteratively looking at the representations f_i, from each of the instances I_i, one at a time. In each iteration i an updated set-level representation $f_{i,j}^{set}$ is computed. In parallel, following the LSTM formulation, a feedback loop provides information regarding the state of the set representation that will be considered at the next iteration $i+1$. Finally, after observing all the m instances I_i in the set, the final set representation $f_{m,j}^{set}|_{i=m}$ is taken as the output f_j^{set} of this component.

The notion behind this iterative set pooling idea is that instances observed at specific iterations can be used to compute a more-informed set-level representation at later iterations. Thus, allowing to encode underlying relationships or structures among the instances of the set. While this iterative assumption may hint at a sequence structure requirement within each set, our empirical evaluation strongly suggests this not to be the case. (see Sec. 7.4.2)

In practice, we use Bi-directional LSTMs which observe the instances in a set from the left-to-right and right-to-left directions. This will further ensure that the context in which the instances of the set occur is properly modelled.

Prediction Unit

Having a set-level representation f_j^{set} for set $\mathbb{I}_j$, this component is tasked with making a set-level prediction $\hat{y_j}{=}g(f_j^{set})$ that will serve as final output for the pipeline. The selection of the prediction function $g(.)$ is related to the task of interest. This unit provides our method with flexibility to address both classification and regression tasks.

7.4 Analysis

7.4.1 What kind of information can be captured by LSTMs?

This experiment focuses on performing multiple instance predictions based on visual data. Following the protocol from Ilse et al. [81] we use images from the MNIST dataset [97] to construct image sets to define four scenarios, each following a different assumption: *Single digit occurrence*, *Multiple digit occurrence*, *Single digit counting* and *Outlier detection*.

For this series of experiments, we use a LeNet [96] as IDU and an LSTM with an input and cell state with 500 dimensions as SRE, respectively. Both the IDU and SRE components are trained jointly from scratch. We compare the obtained performance w.r.t. the attention-based model from [81] and the dynamic pooling method from [180]. Mean error rate in the binary classification task is adopted as performance metric in these experiments. Please note, we *do not* traverse all the possible permutations in the following experiments, on the contrary, only small proportion of them are seen by the model.

The main objective of this experiment is to answer the following questions: i) whether other underlying set structures, outside of sequential order, can be encoded properly by LSTMs?, and ii) *how competitive are LTSMs when*

compared with methods from the MIL literature specifically designed for modeling the underlying set structures?

Single Digit Occurrence

In this scenario we follow the standard MI assumption and label a set as positive if at least one digit '9' occurs in the set. The digit '9' is selected since it can be easily mistaken with digit '4' and '7' [81], thus, introducing some instance-level ambiguity. We define sets with mean cardinality m=10, and verify the effect that m has on performance by testing two standard deviation values, σ=2 and σ=8. We repeat this experiment five times generating different sets and weight initializations. We report mean performance in Table 7.1 (col. II and III).

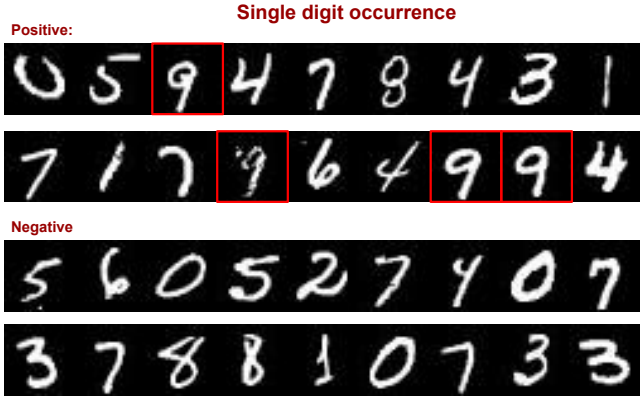

Figure 7.2: Examples of the data for the Single digit occurrence experiment.

Discussion: The results indicate that, in this task, our performance is comparable with the state-of-the-art for lower values of σ and superior as σ increases. This is to some extent expected, since at lower σ the cardinality (i.e. the number of instances) of each set is almost fixed. This setting is favorable for the attention-based method since it operates in a feed-forward fashion. Yet, note the high standard deviation in performance produced by this baseline. On the contrary, at higher σ values there is a higher variation of cardinality across sets. Under this setting, feed-forward approaches start to produce higher errors. Here our method produces superior performance, $\sim$1.4 percentage points (pp) w.r.t. to the state-of-the-art.

Method	single digit(σ=2)	single digit(σ=8)	multiple digits	digit counting	outlier detection
Atten. Based	**2.8 ± 4.8**	4.5 ± 0.4	28.5 ± 0.7	33.4 ± 19.3	37.0*
Gated Atten. Based	4.0 ± 0.9	4.6 ± 0.5	27.4 ± 0.9	11.9 ± 3.6	37.4*
Dyn. Pool	5.6 ±1.1	6.1 ± 1.2	28.5 ± 6.6	25.4 ± 1.8	40.9*
Ours w/o Mut. Info.	3.5 ± 1.1	**3.1± 0.5**	6.4± 1.4	9.0± 2.7	50.0
Ours	4.0± 0.4	4.1±1.4	**3.5±1.3**	**7.4±1.2**	**2.07**

Table 7.1: Mean error rate (in percentage points) of experiments considering digits from the MNIST dataset. (*) refers to baselines which include the Mutual Information loss.

Multiple Digit Occurrence

This is an extension of the previous scenario in which instead of focusing on the occurrence of a single digit class, the model should recognize the occurrence of instances of two digit classes (presence-based MI assumption [49]). More specifically, a set is labeled positive if both digits '3' and '6' occur in it, without considering the order of occurrence. For this scenario 1,000 sets are sampled for training. Results are reported in Table 7.1 (col. IV).

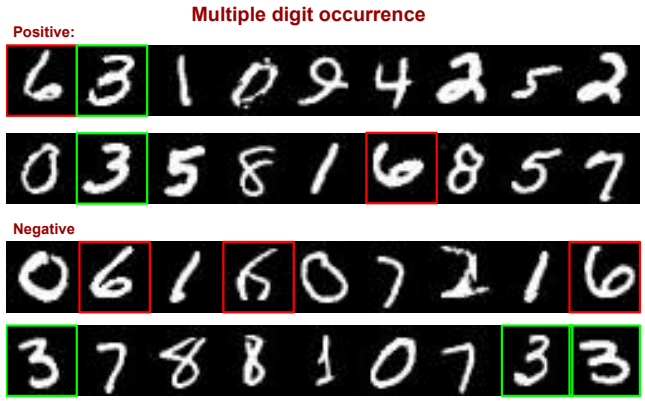

Figure 7.3: Examples of the data for the Multiple digit occurrence experiment.

Discussion: It is remarkable that when making this simple extension of considering the occurrence of multiple digits, i.e. '3' and '6', the state-of-the-art methods suffer a significant drop in performance. This drop put the state-of-the-art methods ∼27 pp below, on average, w.r.t. the performance of our method. Please note that in this experiment the order (or location) of the two digits does *not* matter. This supports previous observations that LSTMs can indeed handle multiple instances of interest, independent of the ordering in which they occur within the sets. In this scenario, where observing multiple

instances is of interest, the model needs to "remember" the information that it has seen in order to asses whether instances of the classes of interest have been encountered. The feed-forward models lack information persistence mechanisms; which translates to a poor ability to remember and to handle multiple instances of interest. Surprisingly, in spite of its iterative nature, the Dynamic pooling method is not able to preserve the information it has observed across iterations, resulting in similar performance as the other baselines.

Digit Counting

Previous scenarios addressed the classification task of predicting positive/negative set-level labels. In contrast, in this scenario, we focus on the regression task of counting the number of instances of a specific digit class of interest within the set (presence-based MI assumption). In order to make our approach suitable to address a regression problem, instead of using a classifier as prediction unit we use a regressor whose continuous output is rounded in order to provide a discrete count value as output. In this experiment the digit '9' is selected as the class to be counted. The mean cardinality of each set is fixed to m=15. Performance is reported in Table 7.1 (col. V).

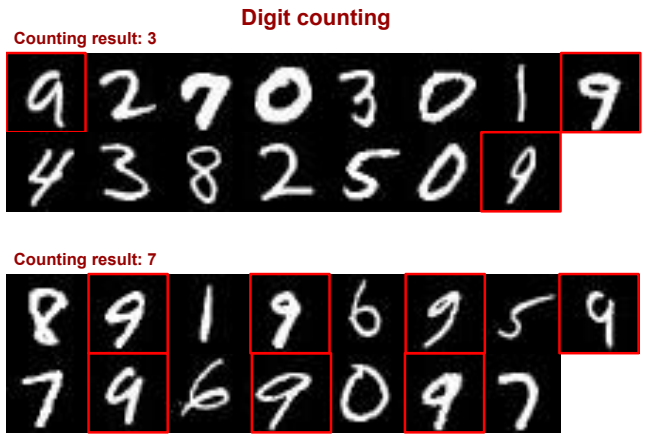

Figure 7.4: Examples of the data for the Digit counting experiment.

Discussion: From Table 7.1 (col. V) the same trend can be observed: our method has superior performance and higher stability than the attention-based model and other baselines. When conducting this counting task, our method obtains a performance that is superior by 24 pp w.r.t. the attention-based model and by 16 pp w.r.t. the dynamic pooling. These results further confirm

the capability of LSTMs to handle this type of unordered regression problems [160].

Digit outlier detection

This task is concerned with identifying whether a set contains a digit which is different from the majority (outlier). Different from *Single digit occurrence*, this task is more difficult since the model has to understand: i) the two digit classes that might be present in the set, and ii) the proportion condition that makes the set an outlier. This is different from *Single digit occurrence* where it only needed to identify the "witness" digit '9'. Besides, there is no restriction on the outlier and majority digits, they can be any digit class from MNIST dataset. This constitutes a collective MI assumption since all the instances determine the underlying structure of the set. Therefore, given the complexity of this task, in this experiment we apply the mutual information loss on every baseline method in order to assist their training. We use 10,000 sets to train the model and 2,000 sets to test. The set cardinality is 6 with 1 standard deviation. Table 7.1 (col. VI) shows quantitative results of this experiment.

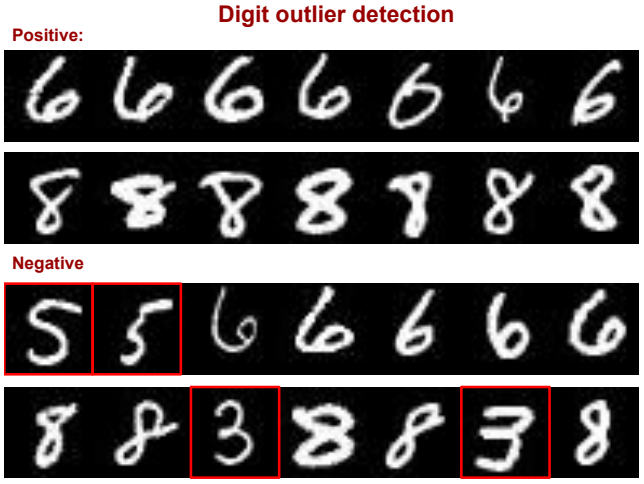

Figure 7.5: Examples of the data for the Digit outlier detection experiment.

Discussion: It is remarkable that, even after applying the mutual information loss on the other baselines, they still have a low performance on this task. We notice that the Attention and Gated Attention methods work slightly better than Dynamic Pooling. More importantly, our method, based on LSTMs, outperforms the baselines by a large margin (∼36 pp). This suggests that

LSTMs are quite capable at modeling this type of set structure, even to the point of outperforming MIL methods tailored to model set-based structures.

7.4.2 Does the result depend on the order chosen?

The short answer is no. The reason is that in the training phase we push the sets with different orders (as a form of data augmentation) to the model while the set labels are the same. By following this procedure the loss function will not penalize differences in the order in which the instances are observed. To further verify this, we repeated the test phase of our experiments 100 times with the contents of each set (cardinality=m, in total $m!$ combinations) shuffled thus producing sets with 100 different orders. Then, similar to Sec. 7.4.1, we measure the error rate and report the mean performance. The obtained error rate is (4.2±0.6)%, (3.5±0.4)%, (7.8±0.7)% for the Single-digit, Multiple-digit and Digit counting experiments, respectively (m=10, 12, 15 respectively). They are very close to the numbers reported in Table 7.1. The results verify that the LSTM is able to learn, to a good extent, that the underlying MIL assumptions were permutation invariant - changing the order of instances of a set has a relatively low effect on the prediction in most of the cases.

7.4.3 Does the result depend on the cardinality of the set?

No, modeling set representations via LSTMs seems robust enough to sets with different cardinalitiy (sizes). We verify this by conducting an extended experiment based on the *multiple instance occurrence* scenario. Firstly, we consider sets with higher cardinality but keep *only one relevant instance pair ('3','6') present* using one of our trained models (mean set cardinality m=12). We obtained error rates of 7%, 14.5%, 42%, and 44% for set cardinality 20, 50, 100 and 200, respectively. This result is not surprising since during training the set cardinality was much lower. To have a fair experiment, we use sets with higher cardinality to finetune our model, using 1/5 amount of the original number of training sets (i.e. now we use 200 sets). Similarly, the larger sets still contain only one pair of relevant digits. This results in error rates of (2.38±0.41)%, (3.13±0.89)%, (4.25±1.3)% for mean set cardinality of 50, 100 and 200, respectively. This shows that LSTMs are still capable of modeling unordered sets even when sets with significantly higher cardinality are considered, although, unsurprisingly, training and testing conditions should match.

Figure 7.6: Examples of instances for the original (left), occluded (middle) and database images (right) in our cross-domain clothing retrieval experiment.

7.4.4 Effect of the Complexity of the Data

In this section, we shift our analysis to real-world data. Summarizing, the results show that our method still works comparable and even better than the baselines.

Cross-domain clothing retrieval

For this experiment, we divide images from the Lookbook dataset into two domains: catalog clothing images and their corresponding human model images where a person is wearing the clothing product, see Fig. 7.6. Each clothing product has one catalog image and several human model images. We only consider the products with five or more human model images, resulting in 6616 unique products (latent classes c_i) with around 63k images in total. Every product image has 5-55 human model images. The training set contains 4000 classes while the validation and test sets have 616 and 2000 classes, respectively. We run two experiments on this dataset as described in the following sections. Given the higher complexity of images in this dataset, we use a pre-trained VGG16 [153] as IDU. Since this unit is pretrained, the mutual information loss is not applied for this unit in this experiment. Moreover, we set the dimensionality of the input and cell state of our LSTM to K=2048.

For this experiment, human model images are used as queries while catalog images serve as database, thus, defining a many-to-one retrieval setting. The cardinality of each set is the same as the number of human model images of each product (class). We conduct two variants of this experiment. On the first variant we use the complete image, as it is originally provided. The second is an occluded variant where every human model image in a set is divided into a 4×4 grid of 16 blocks. 12 of these blocks are occluded by setting all the pixels therein to black. By doing so, every single image in a set can only show part of the information while their combination (i.e. the whole set) represents the complete clothing item. Catalog images in the database are not occluded in this

Method	rec.@1	rec.@10	rec.@20	rec.@50
Atten.	13.75	39.25	49.70	63.60
Dyn. Pool	16.75	47.65	59.45	73.60
Single AVE	20.55	57.05	68.25	81.90
Single MIN	22.60	58.15	**69.20**	82.50
Single Fea. AVE	20.15	56.25	67.85	81.50
Ours w/o mut. info.	**22.95**	**58.65**	68.70	**83.00**

Table 7.2: Retrieval on the original Lookbook dataset.

Method	rec.@1	rec.@10	rec.@20	rec.@50
Atten.	3.55	20.6	32.95	53.65
Dyn. Pool	1.95	11.95	29.35	32.55
Single AVE	3.65	23.85	35.06	56.10
Single MIN	5.25	26.05	37.35	55.00
Single Fea. AVE	5.10	25.60	36.95	54.65
Ours w/o mut. info.	**9.25**	**34.75**	**45.00**	**61.80**

Table 7.3: Retrieval on the occluded Lookbook dataset.

experiment. This experiment can be regarded as an extreme case of standard MI assumption, where all the instances in each set is positive.

As baselines, in addition to the attention-based model we follow DeepFashion [113], and train a model to perform retrieval by computing the distances by considering single image representations instead of set-based representations. Following the multiple queries approach from Arandjelovic et al. [6], we report performance of three variants of this method: *Single-AVE*, where the distance of each set is computed as the average of the distances from every image in the set w.r.t. an item in the database; *Single-MIN*, where the distance of the set is defined as the minimum distance of an image in the set w.r.t. an item in the database; and *Single Fea. AVE* , where the distance of the set is calculated as the distance of a prototype element w.r.t. an item in the database. As prototype element we use the average feature representation $\overline{f_i}$ from the representation f_i of every element in the set. We refer to these baselines as *Single-image models*.

Discussion: Table 7.2 shows that in the original setting our method tends to obtain superior recall values in the majority of the cases, with the exception of the case when the closest 20 items (recall@20) are considered. When looking at the occluded variant of the experiment, a quick glance at Table 7.3 shows that, compared to the original setting, absolute performance values on this setting are much lower. This is to be expected since this is a more challenging scenario where the model needs to learn the information cumulatively by aggregating information from parts of different images. In this occluded setting, our method clearly outperforms all the baselines. This could be attributed to the information

persistence component from the LSTMs. This component allows our method to select what to remember and what to ignore from each of the instances that it observes when updating the set representation used to compute distances. The difference w.r.t. to the *Single-AVE* and *Single-MIN* baselines is quite remarkable given that they require a significant larger number of element-wise distance computations w.r.t. items in the database. This may lead to scalability issues when the dataset size increases, as the computation cost will grow exponentially.

Moreover, in both occluded and non-occluded datasets, we notice that the *Single-image model* baselines have a superior performance w.r.t. the attention-based model and dynamic pooling model. We hypothesize that is because the single-image models can better exploit important features, e.g. discriminative visual patches, since they compute distances directly in an instance-wise fashion. In contrast, it is likely that some of these nuances might get averaged out by the feature aggregation step that is present in the attention-based model.

Colon Cancer Prediction

This task consists of predicting the occurrence of Colon cancer from histopathology images. The used Colon cancer dataset contains 100 500×500 H&E images with a total of 22k annotated nuclei. There are four types of nuclei: *epithelial, inflammatory, fibroblast*, and *miscellaneous*. This experiment focuses on identifying whether colon cancer histopathology images contain a specific type of nuclei. We follow the protocol from Ilse et al. [81] and treat every H&E image as a set composed by instances (patches) of 27×27 pixels centered on detected nuclei. The set cardinality varies from 6 to 796 depending on the number of nuclei present in the image. Fig. 7.7 shows some examples. Following a standard MI assumption, a set is considered positive if it contains *epithelial* nuclei since Colon cancer originates from epithelial cells [81][143] This produces a dataset with 51 and 48 positive and negative set examples, respectively. We extend this dataset via data augmentation as in [81].

We adapt an architecture which is similar to Sirinukunwattana et al. [156] to define the IDU and a 512 dimension input and cell state to define the LSTM (SRE). The whole model is trained from scratch. Following the protocol, only set-level binary labels are used for training. We conduct experiments considering the same baselines as in previous experiments. We apply five-fold cross validation and report the mean performance and standard deviation. For reference, we also provide the results presented in [81] for the baselines Atten.* and Gated Atten.*. Table 7.4.4 shows quantitative results in terms of Accuracy and F1-Score.

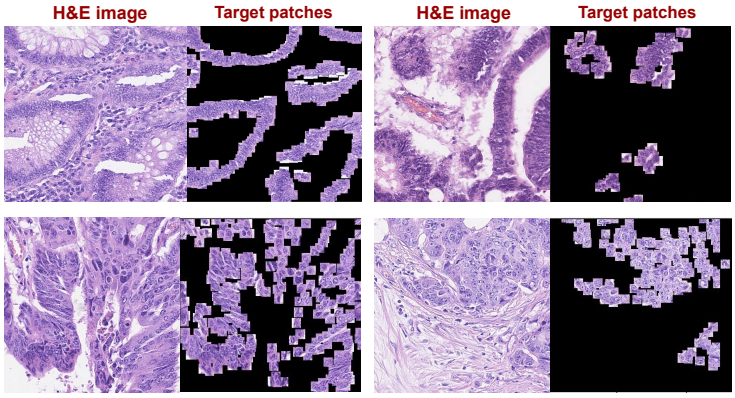

Figure 7.7: Examples of the Colon cancer H&E images and the target nuclei patches.

Method	Accuracy	F1-Score
Atten.*	90.40±1.10	90.10±1.10
Gated Atten.*	89.80±2.00	89.30±2.20
Atten.	88.79±6.16	88.85±6.35
Gated Atten.	86.89±3.93	86.87±6.67
Dyn. Pool	87.89±2.37	88.18±2.11
Ours w/o mut. info.	90.89±2.06	90.66±2.80
Ours	**92.74±2.41**	**93.08±1.36**

Table 7.4: Colon cancer experiment results.

Discussion: This experiment, where a set can have up to 796 instances, serves a good test-bed to assess the performance of the proposed method on sets with high cardinality. From the results in Table 7.4.4, we can notice that our method still outperforms all the considered baselines.

7.5 Internal State Inspection and its application

Previous efforts [75, 105, 149] based on ordered sets have shown that the internal state of the representation within the LSTMs can be used to predict future instances. Here, we have shown that LSTMs can also encode other types of set-based information internally. This begs the question - *what else can the internal representation in LSTMs reveal in the unordered setting?*. Here we conduct an analysis aiming to answer this question.

7.5.1 Internal States Inspection

We begin by making an inspection of the internal states of the LSTM as it observes instances and use t-SNE [164] to visualize these internal states. Fig. 7.8 clearly shows that when the model meets (one of) the condition(s), the current internal state changes evidently, which means LSTMs can distinguish the condition instances with other instances within the sets. Based on this observation, we wonder whether it is possible to predict the instance level label based on the set level label.

7.5.2 Weakly Supervised Learning of Instance-Level Distributions

We have presented using LSTM to make predictions from a set-level representation f_j^{set} through the use of a prediction function $g(\cdot)$. There is a connection between the MIL task and the distribution of the instance representation. Based on this observation we put forward the following hypotheses:

- *Hypothesis 1: A model trained for a MIL task can learn the underlying distribution over the instances.*

- *Hypothesis 2: A prediction function $g(\cdot)$ trained on the set representation f^{set} can be used to make instance-level predictions if the distribution from f^{set}, influenced by the underlying MI assumption, is close to that of $\mathbb{F}$.*

We propose the following approach to recover the underlying instance-level representation and make instance-level predictions. We break down the instance set $\mathbb{I}=\{I_1, I_2, ..., I_m\}$ into m singleton sets $\mathbb{I}_1=\{I_1\}$, $\mathbb{I}_2=\{I_2\}$, ..., $\mathbb{I}_m=\{I_m\}$. The singleton set $\mathbb{I}_j=\{I_j\}$ is sent to the model, passing the IDU and the LSTM. Afterwards, the output f_j^{set} of the LSTM from every singleton is collected into a feature matrix $\mathbb{F}^{set}, \mathbb{F}^{set} \in \mathbb{R}^{m \times n}$, where n is the total number of singletons. Then, K-means clustering algorithm is applied on $\mathbb{F}^{set}$ with the number of clusters determined by the corresponding MIL task. We use a similar metric to clustering purity, where we calculate the purity of each cluster first and average them instead of calculating the purity of all samples. By doing this we avoid problems caused by imbalanced data. The clustering performance reflects the ability of modeling the distribution of instances for the model.

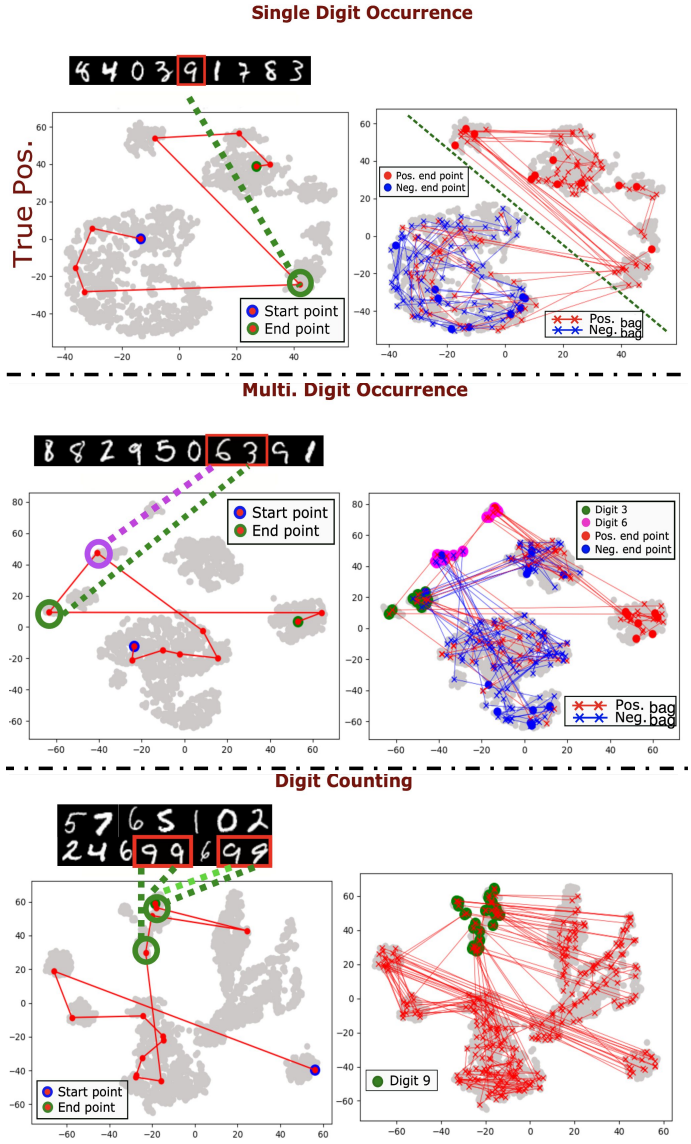

Figure 7.8: t-SNE visualization of the interal states for three MNIST-based experiments. The left figures show an example of predictions on true positive set except for the Digit Counting experiment which shows a set containing 4 instances of interest. The right figures show the prediction of 20 examples overlaid on the t-SNE space for the digit-based experiments. The red and blue lines refer to positive and negative sets, respectively. Best viewed in color.

Task	Gated Atten. (test/train)	Ours (test/train)
single digit (2 classes)	98.69/98.92	97.59/97.42
multiple digits (3 classes)	85.92/87.47	**97.94/97.06**
digit counting (2 classes)	99.22/99.31	99.15/99.23
outlier detection (10 classes)	59.33/57.02	**97.96/97.52**

Table 7.5: Instance clustering accuracy from MNIST-set task models.

Method	TP (test/train)	TN (test/train)	mean Acc (test/train)
Atten.	32.42 / 21.25	**98.45 / 99.22**	65.43 / 63.60
Ours	**73.47 / 70.73**	92.39 / 92.28	**82.93 / 81.51**
Supervised	*78.92 / 92.09*	*91.14 / 98.22*	*85.03 / 95.16*

Table 7.6: Instance label accuracy for Colon cancer dataset.

7.5.3 Weakly Supervised Instance-level Learning

In Sec. 7.5.2 we presented two hypotheses related to the weakly supervised instance-level learning. We will address them in this section.

Modelling Instance-level Representations In Sec. 7.4.1 and 7.4.4, we trained both IDU and LSTM from scratch by considering the set-level labels only. This can be regarded as weakly supervised learning if the goal is to make instance-level predictions. For attention-based methods, we collect the output of IDU and multiply with weight 1, since it is a singleton set and there is no LSTM. Following this procedure, both methods use the features after their respective units handling the MIL task. We evaluate instances from both testing/training set for the baseline and our model, respectively. We choose the Gated-Attention model as a baseline since it works best among the attention-based methods in Sec. 7.4.1. Table 7.5 reports the clustering performance metric described in Sec. 7.5.2.

Discussion: Table 7.5 indicates that for simple tasks, such as *single digit occurrence* and *digit counting*, both attention-based and our methods can distinguish the background digits and witness digits. To handle the MIL task, the model just needs to differentiate between the witness digit ("9") from other digits. Therefore, there should only be two clusters/classes. Three clusters/classes are assigned to *multiple digits* because the model needs to distinguish the two witness digits from the others.

For the case of *outlier detection*, in order to detect the outlier(s) from a set, the model needs to distinguish every digit. For this reason, once capable of handling

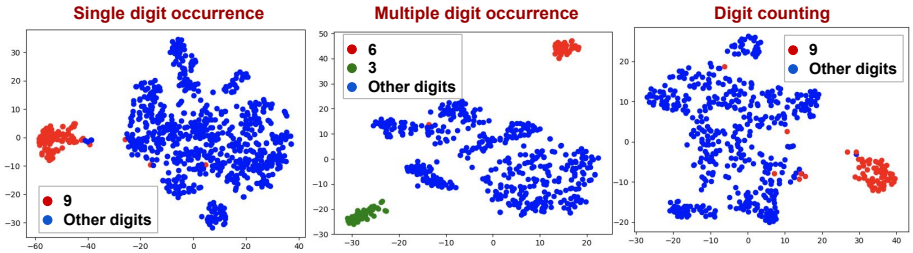

Figure 7.9: t-SNE visualization of features extracted from **our** MIL model in three MNIST set tasks.

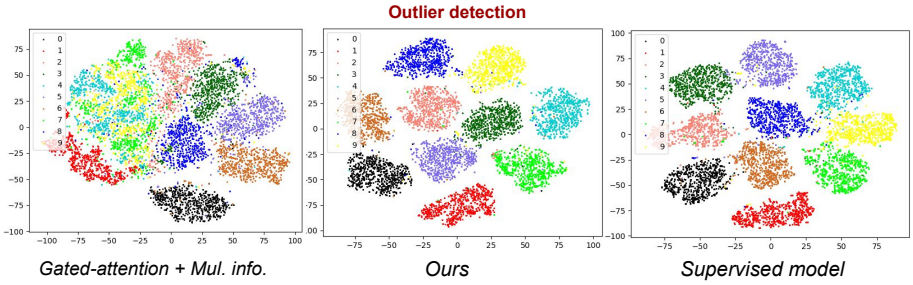

Figure 7.10: t-SNE visualization of features extracted from **our** MIL model in three MNIST set tasks.

this MIL task, the models should also have the ability to cluster/classify the 10 digits. It is clear that our model trained for this task has learned very good discriminative features for all 10-class digits, while the attention-based method fails, even when the mutual information loss is still applied on top of it. The clustering accuracy is close to the known performance of ∼98% accuracy of the supervised LeNet model [97]. This is strong evidence showing that our method is able to learn an instance-level representation in a weakly supervised manner. In addition, Fig. 7.9 and Fig. 7.10 show the t-SNE visualizations for features extracted by our method in the testing set of the four tasks. The figure clearly shows how discriminative the singleton features are.

These results prove that our *Hypothesis 1* is correct.

Instance-level prediction: The colon cancer dataset contains 7,722 epithelial nuclei and 14,721 other nuclei. We select one of the models we trained earlier and treat the patches as singleton sets (i.e. sets only contain one patch). The singleton sets are sent to the model to make instance-level predictions: epithelial or not. In the meantime we also use the same training-test split to train a

fully supervised model. We report the instance-level accuracy in Table 7.6. In addition, Fig. 7.11 shows the patches that are classified as epithelial nuclei.

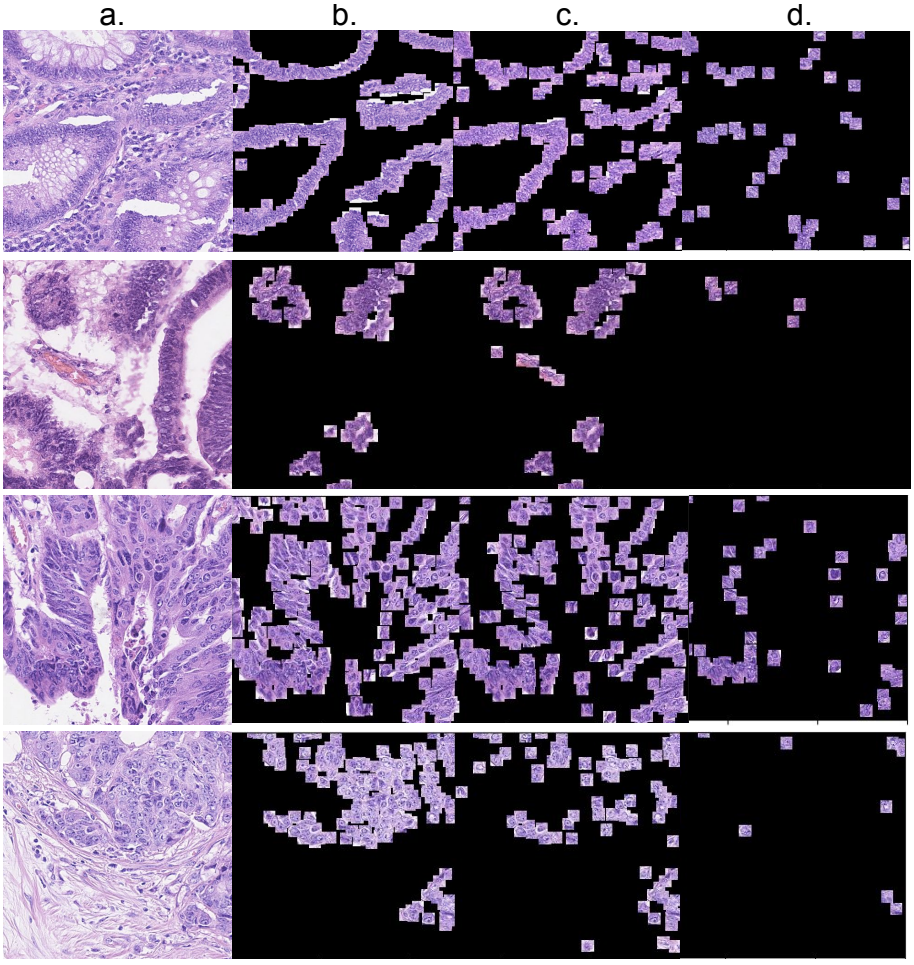

Figure 7.11: a) The original H&E image. b) The epithelial nuclei patches (Ground-Truth). c) The epithelial nuclei patches detected by our MIL model. d) The epithelial nuclei patches detected by attention-based MIL model

Discussion. This task meets the requirement of *Hypothesis 2*: the set representation f^{set} contains the information whether the epithelial nuclei exist in a set, which is close to what would be expected from instance-level feature $\mathbb{F}$. Our model achieves the best performance for set-level prediction. It also has a good performance on the instance-level prediction. The mean accuracy

is close to the supervised model and significantly better (∼18 pp) than that of the Attention-based model. It clearly shows that our MIL model can be used to predict the instance labels. In addition, Fig. 7.11 shows that our model has a better ability to identify the nuclei of interest, which can be useful for pathologists.

7.6 Current Research Trends

The paper based on this chapter was published in summer 2020, there is few follow-up work. However, we notice the large-scale usage of transformers in the computer vision field, where they can be used to take over LSTMs in some sequential-dependent tasks. Intuitively, a possible research direction is to use a transformer to handle the MIL tasks. Notably, transformers process all the elements simultaneously and compute the self-attention of the input set rather than in a recurrence manner. We will discuss it in Chapter 8.

7.7 Discussion

Go back to the previous two chapters, where we also handle image sets. Could the method proposed in this chapter be also applied to them? First of all, the essence of this method is learning set-level representation for classification, regression tasks. What we want is a good set-level representation that describes the set well. Secondly, here we propose to use LSTMs to handle this task; therefore, the encoded set-level representation is projected from the CNN feature space to the LSTM feature space, where the information directly related to the image itself can be lost. Therefore, it cannot replace the PWAve module proposed in Chapter 6, where the model is not interested in the set-level representation of the HR exemplars. A possible option is to integrate this method in the discriminator, where it takes a set of HR images as a real set and the same set with one element replaced by a generated image as a fake set. Compared with the traditional CNN-based discriminator, the LSTM-based discriminator will become deeper with more parameters, which could be a potential problem for training. On the other hand, it can handle the case where there is a various number of conditions. In Chapter 5, we try to accurately locate the object by minimizing the pair-wise distance of two object-specific representations of two images from the same class, which is simple and effective. Similar to Chapter 6, we do not need the set-level representation to compute *CRR*.

7.8 Conclusion

In this chapter, we investigate the potential of LSTMs at solving MIL problems. Through an extensive analysis we have shown that LSTMs can indeed capture additional information of the underlying structure within the sets. Our results suggest that the performance at modeling more general set structures is comparable and even better than that from methods specifically designed for MIL problems. The proposed method can also model the instance-level distribution in a weakly supervised manner.

Chapter 8

Conclusion and Discussion

In this thesis we explore the potential benefits that can be obtained by taking into account the compositionality of single images and image sets. For the single-image scenario, we explore them from two complementary cases: a) Given an input image, find the component representations that are relevant to the prediction made by a pre-trained model, while the model encodes images into single representations layer by layer in the training phase. b) Separately encode different component features (i.e. shape and style) of an image in the training phase. The former case can be extended to DNN explanation and interpretation, and the latter case can be applied to the image synthesis task. Regarding the image sets scenario, we investigate the weakly supervised object localization and image super-resolution tasks, where the sets composed by multiple component images help the model in different ways. We also tackle the general MIL problem, where the component images can be from different classes and only set-level annotation is available. In this chapter, we start by summarizing the contributions of this thesis. Then we revisit the research questions raised in Chapter 1. Last but not least, we introduce the current research trends and propose some future research directions for our work.

8.1 Summary of Contributions

We first investigate DNN explanation and interpretation in Chapter 3. Specifically, we aim to automatically identify a small subset of filters in DNNs relevant to the predictions made by a pre-trained DNN without using expensive pixel-level annotation. Based on the assumption that only a small subset of

the filters of a DNN is important for the task that the DNN addresses, we collect the activations of every filter in a DNN from all the training images and formulate the filter selection as a $\mu-Lasso$ optimization problem. By solving the problem we can obtain the filters important to each predicted class. The ablation study with randomly removing the same amount of filter-wise activation in the DNN verifies the effectiveness of our method. To interpret the feature/informantion encoded by the filter we average the specific region of images that activate highly on that given filter, where its receptive field determines the region size. In the inference phase, given an image with the pre-trained DNN, our method can generate the explanation by calculating the relevant filters and visualizing the representations encoded by these relevant filters thanks to the DNN visualization method. In addition, our proposed method is generic: it can be applied to any type of feed-forward DNN. We also provide a dataset and an evaluation protocol specifically designed for the model explanation task and it was the first dataset at such task when the corresponding paper was submitted.

Then we switch to a generative task in Chapter 4, where we disentangle two component features (i.e. style and shape) of input images in the training phase rather than identify them from a pre-trained model in Chapter 3. In this chapter, we address the unpaired shape translation problem and propose the unpaired shape transforming method that does not need any paired data to simultaneously achieve clothing try-on and taking-off tasks. It is significantly harder than the style transfer task since changes in geometry and shape introduce side effects, e.g., viewpoint changes, self-occlusion, etc. Besides, the translation is expected not to affect the semantic properties of the images, e.g., the clothing appearance. The take-off task is relatively easier because the output is always the catalog image in the front view, which is deterministic. On the contrary, the output of the try-on task is various, there are multiple reasonable outputs such as different models with different poses, etc., which confuses the generative model. In order to solve the one-to-many mapping problem we design the Fit-in module which uses the context information to constrain the output, making the output deterministic. In addition, we find the style representation is highly compact (i.e., only 8 dimensions) but powerful for the cross-(shape) domain item retrieval task, which is even comparable with LPIPS [198] whose representation dimension is approximate 640K. We also apply our methods to MultiPIE [62] face dataset, where the method can achieve face frontalization (similar to the clothing take-off) and face swapping (similar to the clothing try-on).

From Chapter 5 we move our interest to explore sets composed by multiple images. We first address the weakly supervised object localization problem in Chapter 5. The weakly supervised object localization is based on CAM method [205] where the final localization heatmap is computed by leveraging the

activation maps of the last convolutional block and the weight of the predicted class in the linear layer. The problem with current localization methods is that the localization map either under-estimates the object region, i.e., focuses on the small, most discriminative region of the object, or over-estimates it, i.e., localizes a lot of background region. We propose two regularizations, *FRR* and *CRR*, to optimize the linear layer. Particularly, benefiting from using multiple images containing the same objects, *CRR* is able to suppress the activations of the localization map on the background region. On the contrary, *FRR* can expand the activations from the small, most discriminative region. By adjusting the weight of the two regularizations, the backbone model can obtain more accurate object localization no matter which problem the backbone suffers. In addition, the proposed method is light-weight: it only relies on a standard CAM model, and there is no need for an extra network. Our proposed method achieves the new state-of-the-art on several datasets, including ImageNet, CUB-200-2011 and OpenImages-segmentation.

In Chapter 6, we handle the face image super-resolution task where a set composed of multiple HR face images having the same identity with the LR image serves as exemplar to provide the missing high-frequency information to the LR input. The HR images are randomly selected, hence not all the information is useful. In order to selectively extract information rather than simply take the average from the HR image set, we propose the PWAve module to select the useful component features from each HR exemplar in a set, based on the LR input. Benefiting from using HR exemplars, our method can also address the face editing task. Different from traditional conditional GAN which relies on the pre-defined discrete visual features, our method can dynamically generate the arbitrary facial features based on the provided HR exemplars. In addition, we observe a similar trend in Chapter 5 that the super-resolution performance increases when there are more HR exemplars in a set.

Finally, we tackle general MIL problem in Chapter 7. We aim at learning compositional set representation supervised by the set-level signal. In this chapter, we propose to iteratively learn each component in a set. Specifically, we advocate the application of LSTM networks on the general MIL problems to encode the general underlying structure of image sets, where the set components can be in any order. Benefiting from the memory ability of LSTM, it can learn beyond sequential information. The results of our extensive experiments show the effectiveness of LSTMs on capturing the information within sets that go beyond ordered sequences. The performance of using LSTMs is comparable and even superior to the competitive methods specially designed to handle MIL problems. Furthermore, we notice that the LSTM is able to model the component distribution through learning the set-level representation in a weakly supervised learning manner.

8.2 Revisiting Research Questions

In Chapter 1 we present the main research question of this thesis "How does component representation help with computer vision tasks?". To make it more concrete we decompose the main research question into five sub-questions, which are answered from Chapter 3 to Chapter 7. At the end of this thesis, we revisit the five questions and summarize the answers we propose.

1. Can different hidden components implicitly encoded in the compositional representations be discovered? Can they be used to explain and interpret a pre-trained DNN?

Both answers are Yes. By applying the method proposed in Chapter 3 we can find these component representations that are important for a DNN making predictions. In Chapter 3 we present a method that can systematically identify the relevant filters given an input to a pre-trained DNN. These filters are the most important ones of the tens-of-thousands of internal filters in a DNN, in other words, they are the explanation to the prediction made by the DNN. By visualizing the representations encoded by them on the input image we can interpret these component representations, where we find different regions of the input are highlighted. This suggests two things: the discovered representations are the hidden component representation of the input; and different regions of the input indeed contribute differently although the whole image is encoded into a single representation for training.

2. Is it possible to separately encode the component features of a single object rather than learning a single compositional representation and what are the benefits of doing so?

Yes, it is possible. In Chapter 4 we design a model to disentangle representations of two components (i.e. style and shape) of images from one domain and then combine them to transfer the image to another domain. Specifically, we focus on fashion-related data, where the dataset is composed by contextualized images (i.e. people wearing garment, domain A) and object-centered images (i.e. garment category image, domain B). We assume that the shape representations from the two domains *can be* shared while originally they do not; and the style representations *are* shared. The two content encoders respectively encode the shape representation for images from two domains and the style encoder encodes the style representation for all images. For the take-off task, the domain-B decoder then takes the shape and style representation from the contextualized image input to synthesize the novel category image. Regarding the try-on task, besides the shape and style representations from the category image input, the domain-A decoder also takes the context information that constrains the output variety and generate the novel try-on image.

3. Given a set composed of images from the same class, how can we accurately localize each image's common object?

This question is explored in Chapter 5. The solution is based on the fact that the object-specific representations should be very similar even if the objects are from different images. In Chapter 5 we design the *CRR* regularization, where we minimize the distance between the object-specific representations from two image components in a set in the second stage of training process. The object-specific representation is obtained by extracting the representation of the image multiplied by its corresponding object localization map which is computed as CAM. The object localization map only optimizes the fully connected layer, in other words, minimizing *CRR* will optimize the combination weight used in computing CAM. The activations on the background region will be gradually suppressed with *CRR* decreasing and in the meantime the localization map will accurately cover the object region. In addition, our ablation study shows that the localization performance is proportional to the number of components in a set.

4. How to select useful visual details from a set composed by HR exemplars in a super-resolution model?

The content in Chapter 6 can answer this question. In Chapter 6 we tackle the face image super-resolution task with help from a set of HR images having the same identity with the LR input as exemplars. The basic idea is utilizing the HR exemplars to provide the model with the missing high frequency information. We design PWAve module to find and select the most useful component features from each HR exemplar in a set. PWAve module takes representations of HR exemplar as well as LR image as input. The LR input works as a condition, where PWAve module will learn the proper regions of the exemplars based on this condition. The proper regions learned by the model will be assigned higher weights when combining the representations of the HR exemplars. PWAve module provides the freedom to the model to learn a different and more suitable combination method rather than the commonly used average. We show the effectiveness of using PWAve module and the visualization of the learned attention heatmap confirms that it focuses on the reasonable regions on the HR exemplars.

5. Is it possible to learn set representations, composed by unordered elements, by using a model which usually processes sequential data? What are the advantages?

Yes, it is. In Chapter 7 we learn the set representation over a set of unordered images by using LSTM networks given the memory ability of LSTMs. The performance of using LSTMs is competitive and even surpass state-of-the-art

methods which are specially designed for handling MIL problems. In addition, by learning the set representation w.r.t. a task, we find that the model also learns the underlying distribution over the components from the set. For example, the learned representation of the interest components are separated from the rest of them when the model is trained to judge whether a specific component is in a set. Furthermore, analyzing and visualizing the hidden representations of the LSTM network reveal the learning process, where it reflects the acceptance and rejection of each coming component w.r.t. a task and provides people with some insight of the learning process of sets.

8.3 Future Work

At the end of each chapter, we have discussed the limitation and current research trends w.r.t. the presented research work. In this section, we will mainly focus on possible future research directions. There are mainly two directions: Transformer-based network and Self-supervised Learning (SSL).

Transformer

Researchers have shown that transformers have massive potential for computer vision tasks. There can be two stages for our further research. The first stage is to verify if our proposed methods still hold if the transformer replaces the backbone, e.g. CNN or LSTM. More specifically, as we discussed in Chapter 5, the visualization of the patch-based self-attention can also localize the classified object. We can systematically evaluate the localization ability and compare it with CAM. Based on the preliminary results, we can test the effectiveness of *CRR* and *FRR* in this new setting. In Chapter 7, we propose to use the LSTM to process unordered image sets found on the memory ability of LSTMs. It is interesting and worthy to investigate whether transformers can learn the underlying structure of image sets, beyond the order information. Compared with LSTM, transformers are designed to handle the long dependency problem where LSTMs may suffer since the self-attention mechanism learns the global dependency, which implies it can have better ability for sets with large cardinality. In addition, the way ViT [43] handles images can be regarded as handling an ordered MIL task, where it processes the embeddings of non-overlapped image patches with corresponding position embeddings. This also inspires us to explore the possibility of using transformers for MIL tasks. The next stage can explore the general transformer. Currently, the basic architecture of transformers is directly brought from NLP field, which is not specifically designed for visual data. This leaves research space for computer vision researchers to investigate. For instance, [170] proposes a hierarchical transformer inspired by CNN, where the

transformer can also capture the local information. By aggregating both local and global information, the model can solve some problems vanilla transformer suffers, like missing detection of the small objects for the detection task. In addition, the performance improvement gained from the current transformers (e.g., ViT [43]) is based on a huge amount of pre-trained data (e.g., JFT dataset, 300 million images, an internal dataset of Google), which is extremely expensive in terms of computation power and time. Therefore, we think our future research direction can be integrating some core ideas of transformers, such as obtaining global representation, into CNNs which are more suitable and computation-friendly for visual data.

Self-supervised Learning

Self-supervised Learning (SSL) is another popular research trend in the community since accessing annotation can be very expensive for a large dataset. We think exploring the object localization task under SSL setting is interesting. The challenge for the self-supervised object localization is in line with the existing SSL problem, which is to train an encoder (e.g. CNN) that can learn representation from the object. The localization map can be obtained from the activation map of the CNN. Therefore, similar to the previous future direction, the first step is to evaluate the localization ability of the SSL model, such as MoCo [71, 30], SimCLR[27, 28] and BYOL [61] and investigate the compatibility of our proposed regularization with the SSL learning framework. The next step is to improve the current SSL models since the optimization goal is in line with the localization task. In addition, we think it is worthy of applying the explanation and interpretation methods to the SSL-trained model, inspecting the similarity and difference of the learned representation between the supervised-trained model.

Bibliography

[1] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. Social lstm: Human trajectory prediction in crowded spaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[2] Albanie, S., Nagrani, A., Vedaldi, A., and Zisserman, A. Emotion recognition in speech using cross-modal transfer in the wild. In *ACM international conference on Multimedia (MM)* (2018).

[3] Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., and Courville, A. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *International Conference on Machine Learning (ICML)* (2018).

[4] Amores, J. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence* (2013).

[5] Anonymous. Unsupervised one-to-many image translation. In *Submitted to International Conference on Learning Representations* (2019). under review.

[6] Arandjelović, R., and Zisserman, A. Multiple queries for large scale specific object retrieval. In *British Machine Vision Conference (BMVC)* (2012).

[7] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)* (2017).

[8] Babar, S., and Das, S. Where to look?: Mining complementary image regions for weakly supervised object localization. In *Winter Conference on Applications of Computer Vision (WACV)* (2021), pp. 1010–1019.

[9] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* (2015).

[10] Balakrishnan, G., Zhao, A., Dalca, A. V., Durand, F., and Guttag, J. Synthesizing images of humans in unseen poses. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[11] Bao, J., Chen, D., Wen, F., Li, H., and Hua, G. CVAE-GAN: fine-grained image generation through asymmetric training. *International Conference on Computer Vision (ICCV)* (2017).

[12] Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition (CVPR)* (2017).

[13] Bay, H., Tuytelaars, T., and Van Gool, L. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)* (2006).

[14] Benenson, R., Popov, S., and Ferrari, V. Large-scale interactive object segmentation with human annotators. *Computer Vision and Pattern Recognition (CVPR)* (2019).

[15] Bilen, H., Fernando, B., Gavves, E., and Vedaldi, A. Action recognition with dynamic image networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2018).

[16] Bluecross. Introducing dogs and cats, 2019. [Online; accessed 31-Auguest-2021].

[17] Bulat, A., and Tzimiropoulos, G. Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. *Computer Vision and Pattern Recognition (CVPR)* (2018).

[18] Cao, Y., Zhou, Z., Zhang, W., and Yu, Y. Unsupervised diverse colorization via generative adversarial networks. In *ECML/PKDD* (2017).

[19] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *Computer Vision and Pattern Recognition (CVPR)* (2017).

[20] Carbonneau, M.-A., Cheplygina, V., Granger, E., and Gagnon, G. Multiple instance learning: A survey of problem characteristics and applications. *arXiv: 1612.03365* (2018).

[21] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. *European Conference on Computer Vision (ECCV)* (2020).

[22] Chan, K. C., Wang, X., Xu, X., Gu, J., and Loy, C. C. Glean: Generative latent bank for large-factor image super-resolution. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (2021).

[23] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. Return of the devil in the details: delving deep into convolutional nets. In *British Machine Vision Conference (BMVC)* (2014).

[24] Chattopadhyay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *Winter Conference on Applications of Computer Vision (WACV)* (2018).

[25] Chefer, H., Gur, S., and Wolf, L. Transformer interpretability beyond attention visualization. In *Computer Vision and Pattern Recognition (CVPR)* (2021).

[26] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., and Chua, T.-S. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Computer Vision and Pattern Recognition (CVPR)* (2017).

[27] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv: 2002.05709* (2020).

[28] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. Big self-supervised models are strong semi-supervised learners. *arXiv: 2006.10029* (2020).

[29] Chen, X., Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Neural Information Processing Systems (NeurIPS)* (2016).

[30] Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv: 2003.04297* (2020).

[31] Chen, Y., Tai, Y., Liu, X., Shen, C., and Yang, J. Fsrnet: End-to-end learning face super-resolution with facial priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018).

[32] Choe, J., Oh, S. J., Chun, S., Akata, Z., and Shim, H. Evaluation for weakly supervised object localization: Protocol, metrics, and datasets. *arXiv: 2007.04178* (2020).

[33] Choe, J., Oh, S. J., Lee, S., Chun, S., Akata, Z., and Shim, H. Evaluating weakly supervised object localization methods right. In *Computer Vision and Pattern Recognition (CVPR)* (2020).

[34] Choe, J., and Shim, H. Attention-based dropout layer for weakly supervised object localization. In *Computer Vision and Pattern Recognition (CVPR)* (2019).

[35] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR)* (2005).

[36] Das, A., Agrawal, H., Zitnick, C. L., Parikh, D., and Batra, D. Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? In *EMNLP* (2016).

[37] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition (CVPR)* (2009).

[38] Deokyun Kim, Minseon Kim, G. K. D.-S. K. Progressive face super-resolution via attention to facial landmark. In *British Machine Vision Conference (BMVC)* (2019).

[39] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv: 1810.04805* (2018).

[40] Doersch, C., Singh, S., Gupta, A., Sivic, J., and Efros, A. A. What makes paris look like paris? *Communications of the ACM* (2015).

[41] Dogan, B., Gu, S., and Timofte, R. Exemplar guided face image super-resolution without facial landmarks. In *Computer Vision and Pattern Recognition (CVPR) Workshops* (2019).

[42] Dong, H., Liang, X., Shen, X., Wang, B., Lai, H., Zhu, J., Hu, Z., and Yin, J. Towards multi-pose guided virtual try-on network. In *International Conference on Computer Vision (ICCV)* (2019).

[43] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)* (2020).

[44] Dosovitskiy, A., and Brox, T. Generating images with perceptual similarity metrics based on deep networks. *arXiv: 1602.02644* (2016).

[45] Escorcia, V., Niebles, J. C., and Ghanem, B. On the relationship between visual attributes and convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)* (2015).

[46] Esser, P., Sutter, E., and Ommer, B. A variational u-net for conditional appearance and shape generation. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[47] Fernando, B., Gavves, E., Oramas M., J., Ghodrati, A., and Tuytelaars, T. Rank pooling for action recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2016).

[48] Fong, R., and Vedaldi, A. Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[49] Foulds, J., and Frank, E. A review of multi-instance learning assumptions. *The Knowledge Engineering Review 25*, 1 (2010).

[50] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. Synthetic data augmentation using gan for improved liver lesion classification. In *ISBI* (2018).

[51] Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)* (2016).

[52] Ge, C., Song, Y., Ge, Y., Yang, H., Liu, W., and Luo, P. Disentangled cycle consistency for highly-realistic virtual try-on. *Computer Vision and Pattern Recognition (CVPR)* (2021).

[53] Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. *Neural Computation* (1999).

[54] Ghorbani, A., Wexler, J., Zou, J. Y., and Kim, B. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems* (2019).

[55] Ghosh, A., Kulharia, V., Namboodiri, V. P., Torr, P. H. S., and Dokania, P. K. Multi-agent diverse generative adversarial networks. *arXiv: 1704.02906* (2017).

[56] Gonzalez-Garcia, A., Modolo, D., and Ferrari, V. Do semantic parts emerge in convolutional neural networks? *International Journal of Computer Vision (IJCV)* (2017).

[57] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *Neural Information Processing Systems (NeurIPS)* (2014).

[58] Graves, A., and Schmidhuber, J. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS* (2005).

[59] Graves, A., Wayne, G., and Danihelka, I. Neural turing machines. *arXiv: 1410.5401* (2014).

[60] Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., Badia, A. P., Hermann, K. M., Zwols, Y., Ostrovski, G., Cain, A., King, H., Summerfield, C., Blunsom, P., Kavukcuoglu, K., and Hassabis, D. Hybrid computing using a neural network with dynamic external memory. *Nature* (2016).

[61] Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. Á., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning. *Neural Information Processing Systems (NIPS)* (2020).

[62] Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. Multi-pie. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition* (2008).

[63] Grün, F., Rupprecht, C., Navab, N., and Tombari, F. A taxonomy and library for visualizing learned features in convolutional neural networks. In *International Conference on Machine Learning (ICML) Workshops* (2016).

[64] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Neural Information Processing Systems (NeurIPS)* (2017).

[65] Guo, G., Han, J., Wan, F., and Zhang, D. Strengthen learning tolerance for weakly supervised object localization. In *Computer Vision and Pattern Recognition (CVPR)* (2021).

[66] Guo, L., Liu, J., Wang, Y., Luo, Z., Wen, W., and Lu, H. Sketch-based image retrieval using generative adversarial networks. In *ACM international conference on Multimedia (MM)* (2017).

[67] Han, K., Guo, J., Zhang, C., and Zhu, M. Attribute-aware attention model for fine-grained representation learning. *ACM international conference on Multimedia (MM)* (2018).

[68] Han, X., Wu, Z., Jiang, Y.-G., and Davis, L. S. Learning fashion compatibility with bidirectional lstms. In *ACM international conference on Multimedia (MM)* (2017).

[69] Han, X., Wu, Z., Wu, Z., Yu, R., and Davis, L. S. Viton: An image-based virtual try-on network. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[70] Hanhart, P., Korshunov, P., and Ebrahimi, T. Benchmarking of quality metrics on ultra-high definition video sequences. In *2013 18th International Conference on Digital Signal Processing (DSP)* (2013).

[71] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv: 1911.05722* (2019).

[72] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)* (2016).

[73] Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T. Generating visual explanations. In *European Conference on Computer Vision (ECCV)* (2016).

[74] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations* (2019).

[75] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural Computation* (1997).

[76] Hsieh, J.-T., Liu, B., Huang, D.-A., Fei-Fei, L., and Niebles, J. C. Learning to decompose and disentangle representations for video prediction. In *Neural Information Processing Systems (NeurIPS)* (2018).

[77] Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[78] Huang, R., Zhang, S., Li, T., He, R., et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *International Conference on Computer Vision (ICCV)* (2017).

[79] Huang, X., and Belongie, S. J. Arbitrary style transfer in real-time with adaptive instance normalization. In *International Conference on Computer Vision (ICCV)* (2017).

[80] Huang, X., Liu, M., Belongie, S., and Kautz, J. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)* (2018).

[81] Ilse, M., Tomczak, J. M., and Welling, M. Attention-based deep multiple instance learning. *arXiv:1802.04712* (2018).

[82] Isola, P., Zhu, J., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)* (2017).

[83] Jaderberg, M., Simonyan, K., Zisserman, A., and kavukcuoglu, k. Spatial transformer networks. In *Neural Information Processing Systems (NeurIPS)* (2015).

[84] Ji, X., Wang, W., Zhang, M., and Yang, Y. Cross-domain image retrieval with attention modeling. In *ACM international conference on Multimedia (MM)* (2017).

[85] Jing, Y., Yang, Y., Feng, Z., Ye, J., and Song, M. Neural style transfer: A review. *arXiv: 1705.04058* (2017).

[86] Jingwei, X., Nannan, W., Xinbo, G., and Li, J. Residual attribute attention network for face image super-resolution. In *AAAI Conference on Artificial Intelligence* (2019).

[87] Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)* (2016).

[88] Julius Adebayo, Justin Gilmer, I. G. M. H. B. K. Sanity checks for saliency maps. In *Neural Information Processing Systems (NeurIPS)* (2018).

[89] Karpathy, A., and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2017).

[90] Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. *Computer Vision and Pattern Recognition (CVPR)* (2019).

[91] Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning (ICML)* (2018).

[92] Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. The (Un)reliability of saliency methods. In *Neural Information Processing Systems (NeurIPS) workshop on Explaining and Visualizing Deep Learning* (2017).

[93] Kingma, D., and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2015).

[94] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)* (2012).

[95] Le, Q. V., and Mikolov, T. Distributed representations of sentences and documents. In *International Conference on Machine Learning (ICML)* (2014).

[96] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* (1998).

[97] LeCun, Y., and Cortes, C. MNIST handwritten digit database.

[98] Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A. P., Tejani, A., Totz, J., Wang, Z., and Shi, W. Photo-realistic single image super-resolution using a generative adversarial. In *Computer Vision and Pattern Recognition (CVPR)* (2015).

[99] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. Photo-realistic single image super-resolution using a generative adversarial network. In *Computer Vision and Pattern Recognition (CVPR)* (2017).

[100] Lee, D., Liu, S., Gu, J., Liu, M.-Y., Yang, M.-H., and Kautz, J. Context-aware synthesis and placement of object instances. In *Neural Information Processing Systems (NeurIPS)* (2018).

[101] Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision (ECCV)* (2018).

[102] Li, X., Li, W., Ren, D., Zhang, H., Wang, M., and Zuo, W. Enhanced blind face restoration with multi-exemplar images and adaptive spatial feature fusion. In *Computer Vision and Pattern Recognition (CVPR)* (2020).

[103] Li, X., Liu, M., Ye, Y., Zuo, W., Lin, L., and Yang, R. Learning warped guidance for blind face restoration. In *European Conference on Computer Vision (ECCV)* (2018).

[104] Liang, X., Gong, K., Shen, X., and Lin, L. Look into person: Joint body parsing & pose estimation network and a new benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2018).

[105] Liang, X., Lee, L., Dai, W., and Xing, E. P. Dual motion gan for future-flow embedded video prediction. In *International Conference on Computer Vision (ICCV)* (2017).

[106] Lin, C.-H., Yumer, E., Wang, O., Shechtman, E., and Lucey, S. St-gan: Spatial transformer generative adversarial networks for image compositing. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[107] Lin, J., Xia, Y., Qin, T., Chen, Z., and Liu, T.-Y. Conditional image-to-image translation. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[108] Liu, G., Wang, J., Zhang, C., Liao, S., and Liu, Y. Realistic view synthesis of a structured traffic environment via adversarial training. In *CAC* (2017).

[109] Liu, J., Shahroudy, A., Xu, D., and Wang, G. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision (ECCV)* (2016).

[110] Liu, M.-Y., Breuel, T., and Kautz, J. Unsupervised image-to-image translation networks. In *Neural Information Processing Systems (NeurIPS)* (2017).

[111] Liu, M.-Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., and Kautz, J. Few-shot unsupervised image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)* (2019).

[112] Liu, Y., Yan, J., and Ouyang, W. Quality aware network for set to set recognition.

[113] Liu, Z., Luo, P., Qiu, S., Wang, X., and Tang, X. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Computer Vision and Pattern Recognition (CVPR)* (2016).

[114] Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)* (2015).

[115] Long, M., CAO, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. In *Neural Information Processing Systems (NeurIPS)* (2018).

[116] Lowe, D. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)* (1999).

[117] Lu, W., Jia, X., Xie, W., Shen, L., Zhou, Y., and Duan, J. Geometry constrained weakly supervised object localization. In *European Conference on Computer Vision (ECCV)* (2020).

[118] Lu, Y., Tai, Y.-W., and Tang, C.-K. Attribute-guided face generation using conditional cyclegan. In *The European Conference on Computer Vision (ECCV)* (2018).

[119] Ma, L., Jia, X., Georgoulis, S., Tuytelaars, T., and Van Gool, L. Exemplar guided unsupervised image-to-image translation with semantic consistency. *International Conference on Learning Representations (ICLR)* (2019).

[120] Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., and Van Gool, L. Pose guided person image generation. In *Neural Information Processing Systems (NeurIPS)* (2017).

[121] Ma, L., Sun, Q., Georgoulis, S., Van Gool, L., Schiele, B., and Fritz, M. Disentangled person image generation. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[122] Mai, J., Yang, M., and Luo, W. Erasing integrated learning: A simple yet effective approach for weakly supervised object localization. In *Computer Vision and Pattern Recognition (CVPR)* (2020).

[123] Mairal, J., Bach, F., and Ponce, J. Sparse modeling for image and vision processing. *Found. Trends. Comput. Graph. Vis.* (2014).

[124] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. Least squares generative adversarial networks. In *International Conference on Computer Vision (ICCV)* (2017).

[125] Menon, S., Damian, A., Hu, S., Ravi, N., and Rudin, C. PULSE: self-supervised photo upsampling via latent space exploration of generative models. *Computer Vision and Pattern Recognition (CVPR)* (2020).

[126] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems (NeurIPS)* (2013).

[127] Mir, A., Alldieck, T., and Pons-Moll, G. Learning to transfer texture from clothing images to 3d humans. In *Computer Vision and Pattern Recognition (CVPR)* (2020).

[128] Mirza, M., and Osindero, S. Conditional generative adversarial nets. *arXiv: 1411.1784* (2014).

[129] Neuberger, A., Borenstein, E., Hilleli, B., Oks, E., and Alpert, S. Image based virtual try-on network from unpaired data. In *Computer Vision and Pattern Recognition (CVPR)* (2020).

[130] Nie, W., Zhang, Y., and Patel, A. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *Proceedings of the 35th International Conference on Machine Learning, ICML* (2018).

[131] Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In *International Conference on Machine Learning (ICML)* (2017).

[132] Oramas M., J., and Tuytelaars, T. Modeling visual compatibility through hierarchical mid-level elements. *arXiv: 1604.00036* (2016).

[133] Pabbaraju, C., and Jain, P. Learning functions over sets via permutation adversarial networks. *arXiv: 1907.05638* (2019).

[134] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch.

[135] Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)* (2016).

[136] Rafegas, I., Vanrell, M., and Alexandre, L. A. Understanding trained cnns by indexing neuron selectivity. *Pattern Recognition Letters* (2019).

[137] Raj, A., Sangkloy, P., Chang, H., Lu, J., Ceylan, D., and Hays, J. Swapnet: Garment transfer in single view images. In *European Conference on Computer Vision (ECCV)* (2018).

[138] Ramon, J., and De Raedt, L. Multiple instance neural networks. In *International Conference on Machine Learning (ICML) Workshop on Attribute-value and Relational Learning* (2000).

[139] Rebuffi, S., Fong, R., Ji, X., Bilen, H., and Vedaldi, A. Normgrad: Finding the pixels that matter for training. *arXiv: 1910.08823* (2019).

[140] Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? *International Conference on Machine Learning (ICML)* (2019).

[141] Rematas, K., Fernando, B., Dellaert, F., and Tuytelaars, T. Dataset fingerprints: Exploring image collections through data mining. In *Computer Vision and Pattern Recognition (CVPR)* (2015).

[142] Ren, S., He, K., Girshick, R. B., and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks.

[143] Ricci-Vitiani, L., Lombardi, D. G., Pilozzi, E., Biffoni, M., Todaro, M., Peschle, C., and De Maria, R. Identification and expansion of human colon-cancer-initiating cells. *Nature* (2007).

[144] Riza Alp Güler, Natalia Neverova, I. K. Densepose: Dense human pose estimation in the wild. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[145] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* (2015).

[146] Sammut, C., and Webb, G. I. Multi-instance learning. *Encyclopedia of Machine Learning* (2011).

[147] Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *Computer Vision and Pattern Recognition (CVPR)* (2018).

[148] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision (ICCV)* (2017).

[149] SHI, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and WOO, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Neural Information Processing Systems (NeurIPS)* (2015).

[150] Silva, T. A short introduction to generative adversarial networks, 2017. [Online; accessed 3-November-2021].

[151] Simo-Serra, E., Fidler, S., Moreno-Noguer, F., and Urtasun, R. Neuroaesthetics in fashion: modeling the perception of fashionability. In *Computer Vision and Pattern Recognition (CVPR)* (2015).

[152] Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations (ICLR) Workshops* (2014).

[153] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)* (2015).

[154] Singh, B., Marks, T. K., Jones, M., Tuzel, O., and Shao, M. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Computer Vision and Pattern Recognition (CVPR)* (2016).

[155] Singh, K. K., and Lee, Y. J. Hide-and-Seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *International Conference on Computer Vision (ICCV)* (2017).

[156] Sirinukunwattana, K., Raza, S. E. A., Tsang, Y., Snead, D. R. J., Cree, I. A., and Rajpoot, N. M. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Transactions on Medical Imaging* (2016).

[157] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. Striving for simplicity: the all convolutional net. In *International Conference on Learning Representations (ICLR) Workshops* (2015).

[158] Sundermeyer, M., Schlüter, R., and Ney, H. Lstm neural networks for language modeling. In *INTERSPEECH* (2012).

[159] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems (NeurIPS)*. 2014.

[160] Suzgun, M., Gehrmann, S., Belinkov, Y., and Shieber, S. M. LSTM networks can perform dynamic counting. *ACL 2019 Workshop on Deep Learning and Formal Languages* (2019).

[161] Tibo, A., Jaeger, M., and Frasconi, P. Learning and interpreting multi-multi-instance learning networks. *arXiv:1810.11514* (2018).

[162] Tu, M., Huang, J., He, X., and Zhou, B. Multiple instance learning with graph neural networks. In *International Conference on Machine Learning (ICML) Workshop on Learning and Reasoning with Graph-Structured Representations* (2019).

[163] van den Berg, E., and Friedlander, M. P. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* (2008).

[164] van der Maaten, L., and Hinton, G. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research* (2008).

[165] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems* (2017), vol. 30.

[166] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, 2011.

[167] Wang, B., Zheng, H., Liang, X., Chen, Y., Lin, L., and Yang, M. Toward characteristic-preserving image-based virtual try-on network. In *European Conference on Computer Vision (ECCV)* (2018).

[168] Wang, H., Zhu, Y., Adam, H., Yuille, A. L., and Chen, L. Max-deeplab: End-to-end panoptic segmentation with mask transformers. *Computer Vision and Pattern Recognition (CVPR)* (2021).

[169] Wang, K., Huang, Y.-H., Oramas M, J., Van Gool, L., and Tuytelaars, T. An analysis of human-centered geolocation. In *Winter Conference on Applications of Computer Vision (WACV)* (2018).

[170] Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv: 2102.12122* (2021).

[171] Wang, X., Yan, Y., Tang, P., Bai, X., and Liu, W. Revisiting multiple instance neural networks. *Pattern Recognition* (2018).

[172] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.* (2004).

[173] Wikipedia. Artificial intelligence — Wikipedia, the free encyclopedia, 2021. [Online; accessed 09-June-2021].

[174] Wikipedia. The starry night, 2021. [Online; accessed 31-Auguest-2021].

[175] Wu, X., He, R., Sun, Z., and Tan, T. A light cnn for deep face representation with noisy labels. *IEEE Trans. on Information Forensics and Security* (2018).

[176] Wu, Z., Lin, G., Tao, Q., and Cai, J. M2e-try on net: Fashion from model to everyone. *ACM international conference on Multimedia (MM)* (2019).

[177] Xiangli, Y., Deng, Y., Dai, B., Loy, C. C., and Lin, D. Real or not real, that is the question. *Internation Conference on Learning Representation (ICLR)* (2020).

[178] Xie, N., Sarker, M. K., Doran, D., Hitzler, P., and Raymer, M. Relating input concepts to convolutional neural network decisions. In *Neural Information Processing Systems (NeurIPS) Workshops* (2017).

[179] Yan, X., Yang, J., Sohn, K., and Lee, H. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision (ECCV)* (2016).

[180] Yan, Y., Wang, X., Guo, X., Fang, J., Liu, W., and Huang, J. Deep multi-instance learning with dynamic pooling. In *Asian Conference on Machine Learning* (2018).

[181] Yang, S., Kim, Y., Kim, Y., and Kim, C. Combinational class activation maps for weakly supervised object localization. In *Winter Conference on Applications of Computer Vision (WACV)* (2020).

[182] Yang, X., Deng, C., Zheng, F., Yan, J., and Liu, W. Deep spectral clustering using dual autoencoder network. *Computer Vision and Pattern Recognition (CVPR)* (2019).

[183] Yazici, V. O., Gonzalez-Garcia, A., Ramisa, A., Twardowski, B., and van de Weijer, J. Orderless recurrent models for multi-label classification. *arXiv: 1911.09996* (2019).

[184] Yi, D., Lei, Z., Liao, S., and Li, S. Z. Learning face representation from scratch. *arXiv: 1411.7923* (2014).

[185] Yoo, D., Kim, N., Park, S., Paek, A. S., and Kweon, I. Pixel-level domain transfer. In *European Conference on Computer Vision (ECCV)* (2016).

[186] Yosinski, J., Clune, J., Nguyen, A. M., Fuchs, T. J., and Lipson, H. Understanding neural networks through deep visualization. In *International Conference on Machine Learning (ICML) Workshops* (2015).

[187] Yu, X., Fernando, B., Ghanem, B., Porikli, F., and Hartley, R. Face super-resolution guided by facial component heatmaps. In *European Conference on Computer Vision (ECCV)* (2018).

[188] Yu, X., Fernando, B., Hartley, R., and Porikli, F. Super-resolving very low-resolution face images with supplementary attributes. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[189] Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)* (2019).

[190] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *Neural Information Processing Systems (NeurIPS)*. 2017.

[191] Zeiler, M. D., and Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)* (2014).

[192] Zeng, X., Ouyang, W., Yan, J., Li, H., Xiao, T., Wang, K., Liu, Y., Zhou, Y., Yang, B., Wang, Z., Zhou, H., and Wang, X. Crafting gbd-net for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[193] Zhang, D., Han, J., Cheng, G., and Yang, M. Weakly supervised object localization and detection: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence abs/2104.07918* (2021).

[194] Zhang, G., Kan, M., Shan, S., and Chen, X. Generative adversarial network with spatial attention for face attribute editing. In *European Conference on Computer Vision (ECCV)* (2018).

[195] Zhang, J., Lin, Z., Brandt, Jonathan, S. X., and Sclaroff, S. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision (ECCV)* (2016).

[196] Zhang, K., Zhang, Z., Cheng, C.-W., Hsu, W. H., Qiao, Y., Liu, W., and Zhang, T. Super-identity convolutional neural network for face hallucination. In *European Conference on Computer Vision (ECCV)* (2018).

[197] Zhang, Q., Nian Wu, Y., and Zhu, S.-C. Interpretable convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[198] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[199] Zhang, X., Wei, Y., Feng, J., Yang, Y., and Huang, T. Adversarial complementary learning for weakly supervised object localization. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[200] Zhang, X., Wei, Y., Kang, G., Yang, Y., and Huang, T. Self-produced guidance for weakly-supervised object localization. In *European Conference on Computer Vision (ECCV)* (2018).

[201] Zhang, X., Wei, Y., and Yang, Y. Inter-image communication for weakly supervised localization. In *European Conference on Computer Vision (ECCV)* (2020).

[202] Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C. C., Lin, D., and Jia, J. Psanet: Point-wise spatial attention network for scene parsing. In *European Conference on Computer Vision (ECCV)* (2018).

[203] Zhao, J., Cheng, Y., Xu, Y., Xiong, L., Li, J., Zhao, F., Jayashree, K., Pranata, S., Shen, S., Xing, J., Yan, S., and Feng, J. Towards pose invariant face recognition in the wild. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[204] Zheng, C., Cham, T., and Cai, J. T ˆ2 2 net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *European Conference on Computer Vision (ECCV)* (2018).

[205] Zhou, B., Khosla, A., A., L., Oliva, A., and Torralba, A. Learning Deep Features for Discriminative Localization. In *Computer Vision and Pattern Recognition (CVPR)* (2016).

[206] Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations (ICLR)* (2015).

[207] Zhu, J., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)* (2017).

[208] Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *AAAI Conference on Artificial Intelligence* (2016).

[209] Zhu, Z., Huang, T., Shi, B., Yu, M., Wang, B., and Bai, X. Progressive pose attention transfer for person image generation. *Computer Vision and Pattern Recognition (CVPR)* (2019).

Curriculum

Kaili Wang was born on March 6^{th}, 1992, in Chengdu, China. He obtained his Bachelor's degree with a major in Measurement and Control Technology and Instrumentation from the University of Electronic Science and Technology of China, Chengdu, China, in 2015. Then he moved to Leuven, Belgium, to continue his graduate study. He obtained his Master of Science degree with a major in Electrical Engineering with Cum laude from KU Leuven, Leuven, Belgium, in 2017. For his Master's thesis, he worked on image geolocation by leveraging human-centered images in PSI-VISICS, KU Leuven, supervised by Dr. Jose Oramas and Yu-hui Huang and promoted by Prof. Tinne Tuytelaars Since September 2017, he pursues a Ph.D. at the Center for Processing Speech and Images (ESAT-PSI) under the supervision of Prof. Tinne Tuytelaars and Prof. Jose Oramas. Since in 2019 Prof. Oramas moved to the University of Antwerp, from that date he is pursuing a joint degree in collaboration with the University of Antwerp, imec-IDLab, as the partner institution. His researches focus on neural network explanation and interpretation, image generation, and object localization. During his Ph.D., he has contributed several research papers in international conferences like ICLR, ICIP, ACCV, and BMVC. In 2018, he was recipient of the Bell Labs Student Award Benelux 2018 for his work on explanation and interpretation for deep neural networks, later published at International Conference on Learning Representations (ICLR) 2019. In 2020, his work on image-to-image shape translation was selected as runner-up for best paper award (top 0.3%) at the IEEE International Conference on Image Processing (ICIP) 2020.

Publications

Conference Articles

- **K. Wang**, Y. Huang, J. Oramas M, L. Van Gool, T. Tuytelaars, An Analysis of Human-Centered Geolocation. Winter Conference on Applications of Computer Vision (WACV) 2018. (Poster)
- J. Oramas M*, **K. Wang***, T. Tuytelaars, Visual Explanation by Interpretation: Improving Visual Feedback Capabilities of Deep Neural Networks. International Conference on Learning Representation (ICLR) 2019. (Poster)
- **K. Wang**, L. Ma, J. Oramas M, L. Van Gool, T. Tuytelaars, Unpaired Image Shape Translation Across Fashion Data. International Conference on Image Processing (ICIP) 2020. (Poster, Finalist for Best paper award)
- **K. Wang**, J. Oramas M, T. Tuytelaars, Multiple Exemplars-based Hallucination for Face Super-resolution and Editing. Asian Conference on Computer Vision (ACCV) 2020. (Poster)
- **K. Wang**, J. Oramas M, T. Tuytelaars, MinMaxCAM: In Defense of LSTMs for addressing MultipleInstance Learning Problems. Asian Conference on Computer Vision (ACCV) 2020. (Oral)
- **K. Wang**, J. Oramas M, T. Tuytelaars, MinMaxCAM: Improving object coverage for CAM-based Weakly Supervised Object Localization. British Machine Vision Conference (BMVC) 2021 (Poster).

Preprints and work under submission:

- **K. Wang***, L. Ma*, J. Oramas M, L. Van Gool, T. Tuytelaars, Unpaired Shape Transformer for Image Translation and Cross-domain Retrieval. submitted to Computer Vision and Image Understanding).

- **K. Wang**, J. Oramas M, T. Tuytelaars, Towards Human-Understandable Visual Explanations: High Frequency Imperceptible Cues Can Better Be Removed. Submitted to International Conference on Learning Representation (ICLR) 2022.

Supervised Master's thesis:

- **Z. Wang**, Information Compensation and Interpretation for Deep Conditional Generative Networks. 2018 $\sim$ 2019
- **B. Kellens**, **J. Smeets**, From comfort classification to position prediction supported by co-teaching for self-driving cars. 2019 $\sim$ 2020

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
PSI
Arenberg Castle Park 10 PO Box 2440
B-3001 Leuven