A Protocol for Evaluating Model Interpretation Methods from Visual Explanations

Hamed Behzadi-Khormouji José Oramas
University of Antwerp, imec-IDLab

Abstract

With the continuous development of Convolutional Neural Networks (CNNs), there is an increasing requirement towards the understanding of the representations they internally encode. The task of studying such encoded representations is referred to as model interpretation. Efforts along this direction, despite being proved efficient, stand with two weaknesses. First, there is low semanticity on the feedback they provide which leads toward subjective visualizations. Second, there is no unified protocol for the quantitative evaluation of interpretation methods which makes the comparison between current and future methods complex.

To address these issues, we propose a unified evaluation protocol for the quantitative evaluation of interpretation methods. This is achieved by enhancing existing interpretation methods to be capable of generating visual explanations and then linking these explanations with a semantic label. To achieve this, we introduce the Weighted Average Intersection-over-Union (WAIoU) metric to estimate the coverage rate between explanation heatmaps and semantic annotations. This is complemented with an analysis of several binarization techniques for heatmaps, necessary when measuring coverage. Experiments considering several interpretation methods covering different CNN architectures pre-trained on multiple datasets show the effectiveness of the proposed protocol.

1. Introduction

Convolutional Neural Networks (CNNs) have recently shown impressive performance in a vast variety of tasks [1, 14, 9]. Up to now, significant efforts have been mounted towards the justification of the predictions made by CNNs, i.e. *model explanation* [25, 8]. See [8, 26, 3, 23, 19, 2] for surveys on the subject. Comparatively, the task of identifying critical elements of the representations encoded inside a given model (*model interpretation*) has remained relatively unexplored.

Efforts related to model interpretation include the characterization of sparse features encoded in the model that are relevant for the prediction task [18] and assessing the alignment between the activation of internal units and annotation masks [27]. [7] clusters the activation maps from segmented inputs. [24] utilizes activation maps to discover topics in the dataset. [12, 16, 4] learn compressed latent-representations of activation maps that encode visual patterns.

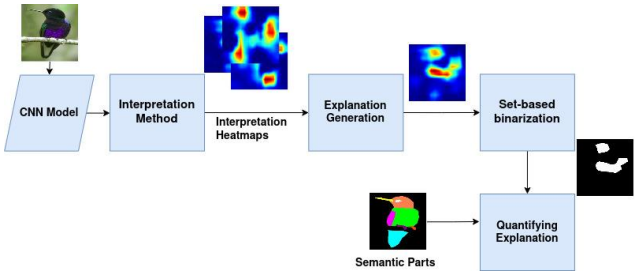

Figure 1. Proposed evaluation protocol. The interpretation method is enhanced to be capable of generating explanation heatmaps via the feedback provided by the interpretation method. Then, a set_based *binarization* operation is applied to the explanation heatmap. Finally, *Quantifying Explanation* measures the coverage rate and assigns a semantic label to the explanation heatmap.

While these methods have proven effective at providing insights on the representations learned by a model, they suffer from three weaknesses. First, most of the existing methods evaluate the provided interpretation feedback in a qualitative manner. For instance, by illustrating the aggregation of visualizations from the identified relevant features [18], or by visualizing the visual patterns in a few input examples [7, 24, 12]. In cases [18, 24] where quantitative analysis is conducted, the analysis focuses on the effect that the provided interpretations have on the prediction performance of the base model. As a result, there is a disparity in the criteria that the interpretation methods use to evaluate their feedback. Hence, there is a lack of a unified evaluation protocol for the quantitative comparison of interpretation methods. Second, with the exception of [27, 6], most of the existing methods [18, 7, 24, 4, 12] are not capable of linking semantic labels to their provided interpretation feedback. This results in visualizations that tend to become ambiguous or subjective. Finally, while capable of provid-

ing some visual feedback on the relevant features encoded by a given model [18, 7, 24, 5, 12], the majority of these methods do not exploit this information for the explanation of the predictions made from a given input.

Our Proposal. To address the above weaknesses, this work puts forward the following four contributions.

First, we propose a procedure to enhance the interpretation methods with the capability of generating an explanation heatmap based on the relevant elements identified in (or learned from) the latent representation of a given input example. This enables the evaluation of both post-hoc and interpretable-by-design (aka. inherently-interpretable) methods. Second, we propose a method for linking the interpretation feedback, produced from the identified relevant elements, with a semantic label. This is achieved by measuring the coverage rate between the explanation heatmaps and the annotation masks corresponding to a given input example. As part of this procedure, we introduce the Weighted Average Intersection-over-Union (WAIoU) metric, which helps measure the overlap between an explanation heatmap and a given annotation mask. Third, a common practice when measuring the overlap between a pixel-level annotation mask and a heatmap is to apply a binarization operation on the latter. We show that thresholds for this binarization operation are more effective when estimated by considering multiple heatmaps at a time. This differs from the standard practice which selects these parameters in a heatmap per heatmap basis.

The three aspects from above are components of the proposed evaluation protocol for the quantitative comparison of interpretation methods (Figure 1).

Fourth, as part of our evaluation, we further extended the CUB-200 dataset by adding part-level segmentation masks for 70 classes. We hope this new set of annotations (*CUB-70*[1]) promotes quantitative comparison w.r.t. future work.

2. Related Work

In this section, we review existing model interpretation methods regarding four aspects:

Capability of generating explanations. [18] utilizes Deconvnet with guided-backpropagation [21] to generate an explanation heatmap from relevant units with a high response on a given input image. ACE [7] visualizes segments of a limited set of examples pertaining to the clustered activation maps. [16, 24] highlight patches in training examples whose activation maps have high closeness to the learned "interpretable" representations. [12] extracts visual patterns by binarizing the similarity matrix between learned latent space vectors and encoders embedding maps, located between the last convolutional layer and the classifier part. [4] generates explanation heatmaps only for those patches of images in the training set such that their activations have the lowest Euclidean distance to the learned prototypes. [5] applies whitening and orthogonality transformations on convolutional filters to increase the distance between latent spaces learned by the filters. Then, it highlights top activated input images as detected concepts for each transformed filter. As can be seen, with the exception of [18], these interpretation methods are not capable of providing explanation feedback for a given input. To address this issue, we enhance some of these methods (Sec. 4) to produce visualizations based on highly activated relevant units as means of explanation for a given input.

Evaluation Protocol. [18, 7, 4] evaluate the provided interpretation in a qualitative manner. For example, [18] uses average visualizations generated from the identified critical units as means to visually illustrate what the network has actually learned. [24, 12] proposed an evaluation protocol based on user studies. This type of evaluation not only is time-consuming, but also can be hard to replicate.

In the case of quantitative analysis, evaluation is accomplished with respect to the changes in the model performance via different approaches. [18] perturbs identified relevant units by zeroing their computed feature maps and tracking the possible reduction in the prediction accuracy. [7] zeros superpixels, corresponding to the clustered activation maps in the input examples. Then, it passes the perturbed images through the model and considers the output probability changes as the effectiveness of the clustered activation maps. [24] proposes the *ConceptShap* metric which includes a completeness score for measuring the contribution of each learned representation in the output probability. As can be seen, there is a disparity in the evaluation approaches. Some of the followed protocols are tailored to the inner-workings of the interpretation method they aim to validate. This makes a uniform quantitative comparison of interpretation methods problematic. To cope with such disparity, our proposed evaluation protocol considers the explanation heatmaps generated from relevant units identified/learned as part of the interpretation method.

Semantic interpretation feedback. Network Dissection [27] measures the alignment between activation maps computed in all the convolutional layers of a given base model w.r.t. pixel-level annotation masks. Then, the semantic label whose annotation masks have the highest overlap with the activation map is assigned to the filter that produced the activations. As a result, a list/histogram is produced indicating the semantic annotations from the dataset that were matched by the internal activations of a given base model. Net2Vec [6] extended this idea to consider a linear combination of activation maps encoding a semantic concept instead of using a single activation map. Excepting [27, 6], the other discussed works are unable to link the provided interpretation feedback with a semantic label.

[1] https://github.com/hamedbehzadi/CUB70-PartSegmentationDataset

In contrast to Network Dissection and Net2Vec, our method differs in the following ways. First, instead of just using intersection-over-union scores of the covered annotation, we take into account the amount of annotation examples being actually covered. This is achieved via the weight term (WAIoU) from Eq. 4, which helps to make the results from different methods comparable (Sec. 3.2). Second, we utilize the metric to measure the coverage rate between explanation heatmaps and annotation masks; while Network Dissection and Net2Vec aim to measure the coverage between annotation masks and activation maps. Third, Network Dissection and Net2Vec aim to compare different CNNs in terms of the number of semantic concepts internally learned by each filter. In contrast, the proposed evaluation protocol aims to address the fundamental question of *how effective is a given interpretation method at identifying the relevant latent features encoded in a base model?*

Heatmap binarization. [27, 6, 12] follow the common practice of using a fixed threshold, estimated on an example-per-example basis, for binarizing each individual heatmap. Different from these, we define these thresholds by considering sets of heatmaps (Sec. 3.3).

3. Proposed Evaluation Protocol

The proposed protocol, first, enhances the interpretation methods to generate a set of interpretation heatmaps. This is achieved via activation maps related to a given input example and interpretation elements provided by the interpretation method. Second, it generates an explanation heatmap per input sample using the produced interpretation heatmaps. Third, the protocol quantifies the explanation heatmaps by measuring the coverage rate between binarized explanation heatmaps and annotation masks. As a result of the coverage rate procedure, the protocol assigns a semantic label to the explanation heatmaps.

3.1. Interpretation with Explanation Capability

This section describes the procedure of generating visual explanation from provided interpretations. Consider $D{=}\{X^i, Y^i, M^i\}_{i=1}^n$ a dataset containing n image examples X^i, their corresponding class labels Y^i, and corresponding pixel-wise annotation masks $M^{i,j}{\in}\mathbb{R}^{w'\times h'}$ with j one of C pre-defined semantic concepts. Also, consider $A^i{\in}\mathbb{R}^{w\times h\times d}$ as the activation maps produced by the last convolutional layer resulting from pushing the example X^i into a CNN model. In the first step of the evaluation protocol, it is needed to enhance the considered interpretation method Z to generate visual explanations. These highlight parts of the input which led to high activation on the identified/learned relevant units. Hence, we aim to generate an explanation heatmap $H^i{\in}\mathbb{R}^{w\times h}$ for image X^i.

The procedure of generating explanation heatmaps is achieved by using the relevant components that have been identified/learned and use them as means of explanation (Sec. 4 explains the modifications done on each method).

Consider a set of m interpretation heatmaps $R^i{=}[r^1, r^2, ..., r^m]$ $(r^t{\in}\mathbb{R}^{w\times h}\ \forall t = 1...m)$ produced as a result of employing activation maps A^i (from the last convolutional layer) and a set of provided interpretation elements produced by identified/learned relevant components. Each interpretation heatmap r^t is a distribution in the activation space with width w, height h, and a value/response $r^t(u, v)$ at location (u, v). Next, we form a set S including m values computed from the aggregation of units in each heatmap r^t (Eq. 1).

$$S = \{S^t | S^t = \sum_{u\in w, v\in h} r^t(u, v)\ \forall t = 1...m\} \quad (1)$$

Each element S^t is considered as a score of its corresponding interpretation heatmap r^t. Afterwards, we select the top-k interpretation heatmaps with the highest score in the set S and concatenate them, that is $R^{i'}_{w\times h\times K}$. To generate an explanation heatmap the maximum response for each location (u, v) is selected across the K dimension (Eq. 2).

$$H^i = max_{u\in w, v\in h} R^{i'}(u, v) \quad (2)$$

Finally, to produce the explanation feedback, the resulting heatmap $H^i{\in}\mathbb{R}^{w\times h}$ is resized to the size of the image X^i and superimposed. This helps highlighting the regions in the input that determine the predicted output.

3.2. Quantifying the Explanation

This section aims at quantifying the semantic coverage of the explanation heatmap. To do so , the heatmap H^i is resized to the size of $M^{i,j}{\in}\mathbb{R}^{w'\times h'}$ and binarized to B^i using a threshold τ_{binary}, i.e. $B^i{=}b(H^i, \tau_{binary})$. We further discuss the selection of this threshold in Sec. 3.3. Then, the coverage between the binarized heatmap B^i and a given annotation mask $M^{i,j}$ is measured following Eq. 3.

$$IoU_j^Z = \left\{ \frac{|B^i \cap M^{i,j}|}{|B^i \cup M^{i,j}|} \Big|_{j=1}^{C} \right\} \quad (3)$$

where IoU_j^Z represents a list containing the Intersection-over-Union (IoU) values for an interpretation method Z with respect to annotated semantic concepts j. In order to compare the coverage of various interpretation methods, we measure the average of IoU per each annotation in IoU_j^Z. To compute this average, we only consider those values which exceed a threshold, namely τ_{iou}.

It should be noted that the number of selected IoU values for a given concept j might be different among the interpretation methods. Taking this into account, with the goal of making the estimated coverage performance value comparable across different methods, we compute a weight related to how well a given method Z covers a given semantic

concept j. Therefore, given a concept j, we give a higher weight to the interpretation method that has higher number of explanation heatmaps whose coverage w.r.t. concept j exceeds the threshold τ_{iou}. This is done following Eq. 4.

$$WAIoU_j^Z = \frac{|IoU_j^Z| * \frac{1}{|IoU_j^Z|}\Sigma_{p=0} IoU_j^z[p]}{\Sigma_{z=0}^Z |IoU_j^z|} \quad (4)$$

In this equation, p refers to each element in the IoU^Z list. $|\bullet|$ is the cardinality of a set. The right term of the numerator indicates the average IoU. The left factor in the numerator, indicates the weight applied to method z w.r.t the concept j. The denominator of the fraction shows the summation of the weight of the methods for the concept j. As evident, Eq. 4 can be simplified as follows.

$$WAIoU_j^Z = \frac{\Sigma_{p=0} IoU_j^z[p]}{\Sigma_{z=0}^Z |IoU_j^z|} \quad (5)$$

Following this process, each heatmap H^i has a IoU value (coverage accuracy) per concept j. Then, according to Eq. 6, the concept label j on which the heatmap H^i has the highest IoU value is selected as its semantic label.

$$L(H^i) = \arg \max_{j \in [1...C]} IoU(B^i, M^{i,j}) \quad (6)$$

3.3. Intensity Thresholding

This section describes the thresholding procedure followed to binarize the explanation heatmaps. In order to estimate optimal thresholds (τ_{binary}) values, we analyze different values based on the Cumulative Distribution Function (CDF) computed from heatmaps. Towards this goal, first, histograms are calculated from the intensity values of a given heatmap. Second, inspired by [27], a CDF is estimated from the histograms and linear interpolation is applied on the CDF. The output is an intensity distribution function that shows the intensity value for different proportion of points on the estimated function. To analyze different intensities as threshold values, we select 10 different proportions of the points from 0.01% up to 0.1% under the CDF curve and use their corresponding intensity values as thresholds to binarize the heatmaps.

We investigate two different scenarios to estimate these CDFs: In the first scenario, *IndivHM*, the CDF is estimated for each explanation heatmap individually.

In the second scenario, *SetHM*, we estimate the CDF by considering multiple explanation heatmaps. As mentioned in Sec. 3.1, the explanation heatmaps are produced by a combination of the K-top interpretation heatmaps. Hence, we consider the explanation heatmaps produced with the same indexes of the interpretation heatmaps (i.e., produced with same k-top indexes). Afterwards, the CDF estimation procedure is applied on them. This is different from the first scenario where the procedure is applied on each explanation heatmap individually.

4. Compared Methods

This section describes the extensions applied to the following compared interpretation methods (*VEBI* [18], *Topic-based interpretation* [24], and *ProtoPNet* [4]) for producing the response R^i (Sec. 3.1).

VEBI. We modify VEBI [18] as follows. First, VEBI considers the activations of all layers to interpret the base model. In contrast, Topic-based and ProtoPNet utilize the activations of the last convolutional layer from the base model. Hence, for the sake of comparison, we modify VEBI to just consider the activations of the last convolutional layer. Second, at test time, VEBI multiplies the aggregated activations from example X^i from class c by the indicator w_c. The indicator, originally, is selected based on the ground truth label c. We modify this step by considering the predicted class c' for selecting the indicator $w_{c'}$. The reason for such modification is explaining the model based on the predicted class instead of ground truth class. In the next step, the resulted non-zero responses with higher scores are selected as the relevant layer/filter elements. Finally, originally, this information is fed to a Deconvnet-based method with guided backpropagation to generate a heatmap visualization. Instead of Deconvnet-based method, we consider GradCAM with two reasons. First, guided backpropagation based method computes the gradient of the selected features with respect to the input features. Different from it, we aim to measure the influence of the identified features on the prediction made by the model. Therefore, we utilize GradCAM which computes the gradient of the predicted output with respect to the identified relevant filters. Second, due to the sanity check analysis done by [13], GradCAM provides more reliable visualizations in comparison to other well-know methods. Finally, the generated heatmaps by GradCAM pertaining to those identified filters by VEBI are considered as the set R^i in Sec. 3.1.

Topic-based Interpretation. We modify Topic-based interpretation [24] as follows. At test time, we push each example X^i to the base model to produce the activation maps $A^i \in \mathbb{R}^{w \times h \times d}$ from the last convolutional layer. Next, considering the learned topics with size $d \times T$, we compute a matrix product between activation maps and T topics. The obtained response represents the "closeness probability" of T topics to the activations from X^i. Also, the response is in the activation map spatial dimension (i.e., $h \times w \times T$) which means each slice of T topics can be superimposed into the input image to show highlighted part. As a result, we consider these T responses as the set R^i in Sec. 3.1.

ProtoPNet. [4] learns a set of $P_c = \{p_1, p_2, ..., p_Q\}$ prototypes for class c (c=1...N), where p_e(e=1...Q) is a 1D

vector, N indicates number of classes, and Q shows the number of prototypes per class. During the training phase, the method generates explanation heatmaps only for those patches of images in the training set whose activations have the lowest L^2 distance to the prototypes. Following this procedure, at the test time, we push each sample X^i through the base model to generate activation maps A^i. Then, the L^2 distance between the activation maps and each prototype set P_c pertaining to each class c is computed. The resulting response indicates the similarity of learned prototypes to the activations of the input sample. Also, it is a set of N elements such that each one has dimension $h \times w \times Q$. That is to say, for each of N classes we have a set of score matrices with dimension $h \times w \times Q$. Finally, we consider part of the response related to the predicted class c' as the set R^i in Sec. 3.1.

5. Evaluation

5.1. Datasets

Our method is validated in the following datasets: **CelebAMask-HQ [15].** has 30,000 512×512 face images, with 24183, 2824, and 2993 images used for training, validation and testing, respectively. Each image has 19 annotation masks of the facial attributes and the accessories corresponding to CelebA [17]. The labels of the masks are *skin, nose, eyes, eyebrows, ears, mouth, lips, hair, hat, eyeglass, earring, necklace, neck, cloth*, and *background*. We group them into three categories including: a) *texture*, b) *facial parts*, and c) *accessories*. The *texture* category includes two sub-classes *skin* (which contains *skin*, *neck*, and *ears* annotation masks), and *hair* (which contains *hair* and *eyebrows*); the *facial-parts* category includes annotation labels related to *nose*, *eyes*, *mouth*, and *lips*. The annotation masks *eyeglass*, *earring*, *necklace*, and *cloth* are included in the category *accessories*. In the rest of the paper, we refer to this dataset with the abbreviation CelebA.

CUB-70. This dataset is a subset derived from the first 70 classes of the CUB-200 [22]. For this subset we manually produced pixel-wise annotation masks for 11 parts including *head, right eye, left eye, beak, neck, body, right wing, left wing, right leg, left leg, and tail*. Worth noting, there is no spatial overlap among the parts in both datasets.

Figure 2 presents some qualitative examples of the provided annotation masks. For training the base models, we split the dataset into two sets based on the distribution provided in the original dataset. The train set contains 70 classes with 30 images per class, and test set contains 1976 images in total for the 70 classes.

5.2. Base Models / Classifiers

We focus our assessment of the interpretation capabilities base models addressing two different classification problems: coarse-grained and fine-grained. Towards this goal, three well-known CNN architectures were selected, namely VGG19 [20], DenseNet121 [11], and ResNet50 [10]. For reproducibility, the specific hyperparameters used for each method can be found in the Supplementary Material. Table 1 shows the classification accuracy for each model. As can be seen, at the dataset level performance is quite comparable.

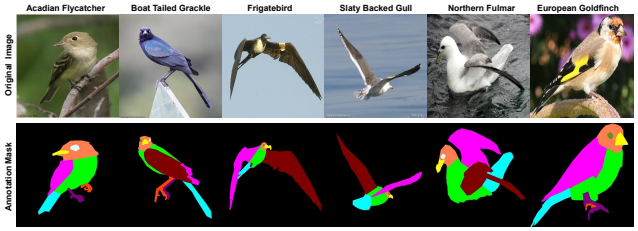

Figure 2. Qualitative examples of the generated annotation masks for the CUB-70 dataset.

Model	CelebA-HQ	CUB-70
VGG19	95.99	73.22
DenseNet121	95.50	78.21
ResNet50	93.44	78.11
Topic-based-VGG19	95.14	70.40
Topic-based-DenseNet121	96.06	74.48
Topic-based-ResNet50	94.44	70.45
ProtoPNet-VGG19	97.02	73.44
ProtoPNet-DenseNet121	92.26	72.36
ProtoPNet-ResNet50	90.85	72.10

Table 1. Test accuracy of the considered CNN models on the CelebA and CUB-70 datasets.

5.3. Qualitative Analysis:Visual Explanations

This section provides qualitative examples of the visual explanations produced by the modifications applied to each interpretation method proposed in Sec. 4.

As can be seen in Figure 3 the explanations of ProtoPNet and VEBI on the CNNs trained on CUB-70 highlight the regions of interest (i.e., different parts of the bird in the input image), while those of the Topic-based method on the VGG19 (Figure 3 (left)) could not detect any parts. In addition, the explanations of Topic-based interpretation on Densenet121 and Resnet50 (Figures 3 (middle) and (right)) could not highlight properly different parts of the birds in comparison to those of VEBI and ProtoPNet.

Different from what was observed in the CUB-70 dataset, Topic-based explanations in the CelebA dataset (Figures 4) highlight regions of interest with a better precision. Results in the CelebA dataset are qualitatively competitive with those from VEBI and ProtoPNet.

To sum up, the provided examples show that ProtoPNet

and VEBI are capable of producing explanation visualizations that focus on parts of the objects of interest. In contrast, Topic-based interpretation highlights the entire object in most cases instead of focusing on a smaller set of features related to the different parts.

5.4. Heatmap Thresholding

This section investigates the effect of the two thresholding scenarios discussed in Sec. 3.3. More specifically, we measure the change in coverage performance by either estimating thresholding parameters from each explanation heatmap independently (*IndivHM*), or from a set of explanation heatmaps generated with the same indexes of the interpretation heatmaps (*SetHM*).

To do so, as explained in Sec. 3.2, the explanation heatmaps are binarized using an intensity threshold. Then, we measure the coverage accuracy between the binarized heatmaps and each of the annotation masks, containing semantic concepts, in the dataset using the Intersection-over-Union (IoU) metric. Finally, the average of the coverage accuracy values is considered as the coverage performance of the interpretation method. Figure 5 shows the Average IoU computed in each of these two scenarios. The horizontal axis shows different proportions of units/intensities under the estimated CDF curves. For each percentage of units, there is an intensity value which is considered as intensity threshold. Hence, the vertical axis indicates the obtained coverage performance for considered intensity threshold.

The curves with the empty square markers refer to the *IndivHM* scenario, while those with the circle markers represent the *SetHM* scenario. As can be seen, for most of the curves, namely ProtoPNet in VGG19-CUB70, VEBI and ProtoPNet in Densenet121-CUB70 and Resnet50-CUB70, VEBI in Densenet121-CelebA, as well as ProtoPNet in VGG19-CelebA, the scenario *SetHM* leads to higher coverage of annotated concepts. The reason for such trends is that in the scenario *IndivHM* there is a variety of distributions among individual explanation heatmaps. As a consequence, the Average IoU metric is sensitive to the intensity of each explanation heatmap. Consequently, for the visualizations on which a wider range of the area-wide features (i.e., low-relevance features) are highlighted, the resulting binary explanation heatmap has poor alignment w.r.t. the annotation mask. This results in a lower IoU value/coverage. In contrast, by estimating thresholding parameters from sets of explanation heatmaps, the scenario *SetHM* is capable of conducting thresholding operations in a more uniform manner. Thus, having a more uniform means to adjust the focus (binary masks) of the produced explanation heatmaps.

Interestingly, the two curves of the Topic-based method are almost overlapping. This shows that in this method there are far fewer number of the explanation heatmaps generated with the same index of interpretation heatmaps. As a result, the CDFs used for estimating thresholding parameters in both scenarios are quite similar which in turn leads to similar performance.

To sum up, most of the results show the higher coverage performance obtained in the *SetHM* scenario. To ensure a more uniform and representative approach to binarize the produced heatmaps, we follow the *SetHM* scenario for the rest of the experiments.

5.5. Visual Explanations with High / Low Coverage

Here we analyze visual explanations which have the highest and lowest IoU values/coverage following the *SetHM* scenario (Figure 6). We focus on explanations from interpretation methods on the VGG19-CUB70, Resnet-CUB70, and Densenet121-CelebA models. See the supplementary material for results from other models.

As can be seen in Figure 6, the highlighted parts in the images with the highest IoU values cover the entire birds or different parts of the birds. In contrast, the images with the lowest IoU values focus on other objects or parts of the background. For example, the foliage of the trees and plants are common parts highlighted in these images. This reveals that these areas are important for the classification task. However, since the annotation masks cover just the birds, the explanation heatmaps of these images have a lower coverage rate with the annotation masks.

A similar trend is observed on CelebA. For instance, according to the Topic-based interpretation results from Densenet121-CelebA, images with high IoU values highlight the entire face, while the images with lowest IoU values show the entire background as the important features.

5.6. Dataset-Level Coverage Rate

In this section, we aim to compare the performance of the methods with each other at the dataset level. Eq. 4 compares the methods in terms of coverage rate for each concept j (i.e. semantic part j). Accordingly, we can aggregate the results obtained from the equation for all the semantic parts for each interpretation method Z. The resulting value indicates the dataset coverage rate of the interpretation method Z w.r.t other methods in different thresholds (Figure 7).

As can be seen in Figure 7, VEBI outperforms ProtoPNet and Topic-based interpretation on models Densenet121-CUB70, Resnet50-CUB70, and Resnet50-CelebA in all intensity thresholds. As mentioned in Sec. 5.4, for each proportion of units under the estimated CDF, there is an intensity threshold. In contrast, ProtoPNet outperforms just VEBI and Topic-based interpretation on the VGG19-CUB70 for all intensity thresholds.

This can also be observed in Figures 3, 4, and 6. Considering the visualizations from the Resnet50-CelebA and Resnet50-CUB70 models, VEBI generates more localized visual explanations, which highlight different smaller parts

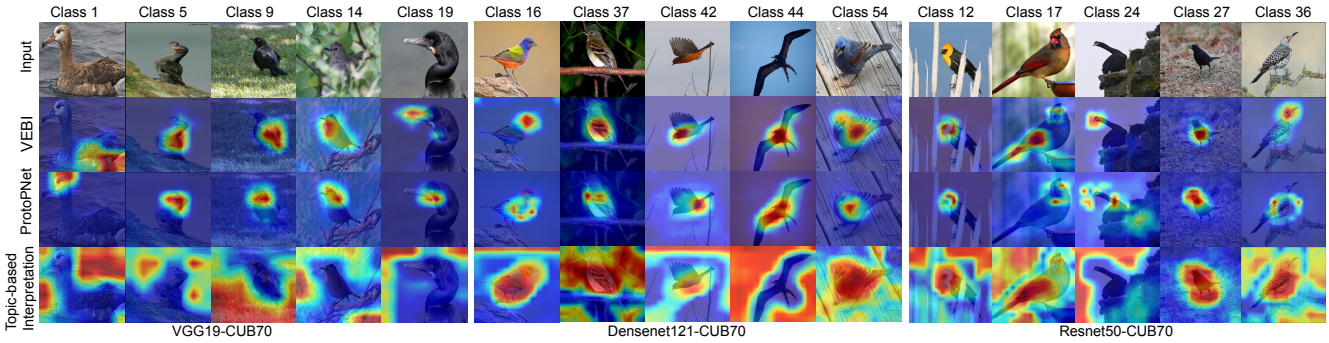

Figure 3. Visual explanations of the investigated interpretation methods over the CNNs trained on the CUB-70 dataset.

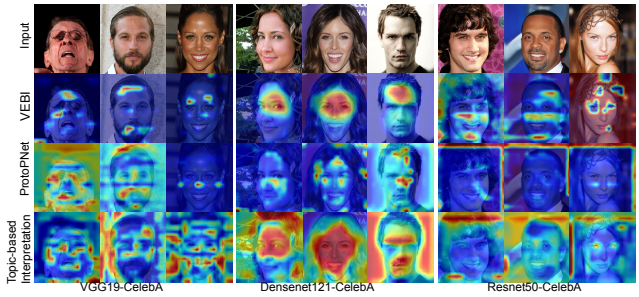

Figure 4. Visual explanations of the investigated interpretation methods over the CNNs trained on the CelebA dataset.

Method	VGG19	Densenet121	Resnet50
VEBI	0.01/0.07	0.02/0.05	0.01/0.1
ProtoPNet	0.04/0.09	0.01/0.07	0.09/0.01
Topic-based	0.01/0.01	0.01/0.02	0.01/0.01

Table 2. The optimal obtained proportion of units (in the form (CelebA/CUB-70)) for each pairs interpretation method and CNN.

such as facial parts or bigger parts such as skin and hair, compared to ProtoPNet. Similarly, this is also more evident in Densenet121-CUB70 where the visualizations from VEBI have higher precision than those of Topic-based interpretation (Figure 3 (middle)).

From the dataset point of view, VEBI and ProtopNet are competitive with each other. However, Topic-based interpretation has poor performance in CUB-70 models and is only competitive with others in models trained on CelebA. This is also evident in Figures 3, 4, and 6 where Topic-based interpretation has visual explanations with better coverage in the CelebA models in comparison to those of CUB-70 models. This shows that VEBI and ProtoPNet can explain properly both types of coarse-grained (i.e., CelebA) and fine-grained (i.e., CUB-70) datasets, while Topic-based interpretation has better performance on coarse-grained tasks as that from CelebA.

To sum up, the quantitative analysis at the dataset level

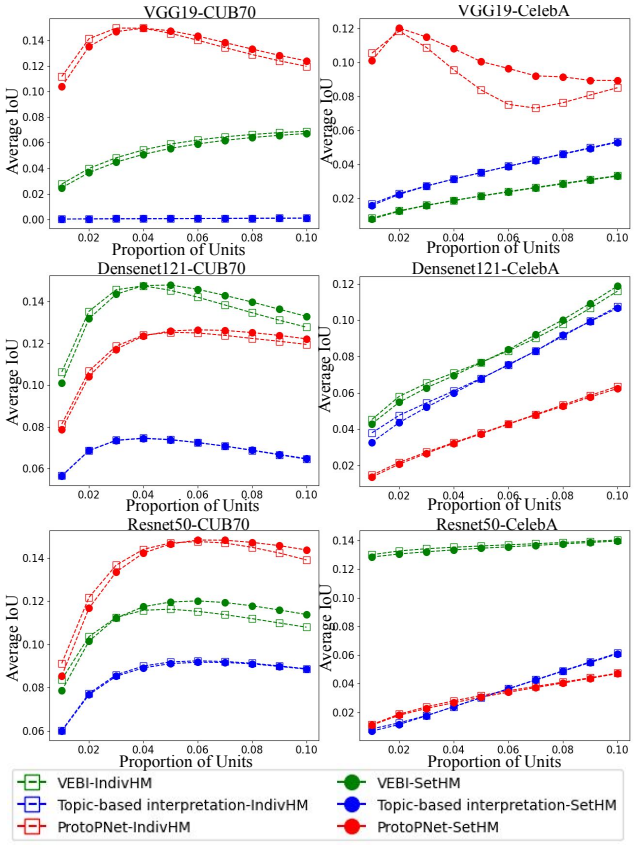

Figure 5. Comparison between two binarization scenarios over CNNs trained on two datasets CelebA and CUB-70.

shows that each method has a better coverage rate at different thresholds in comparison to each other. Hence, for each method we consider the proportion of units indicated in Table 2 which leads to the highest coverage rate. Considering these thresholds, VEBI outperforms ProtoPNet and Topic-based Interpretations on models trained on CUB-70 and CelebA datasets, such as Densenet121-CUB70, ResNet50-CUB70, Resnet50-CelebA, and VGG19-CelebA.

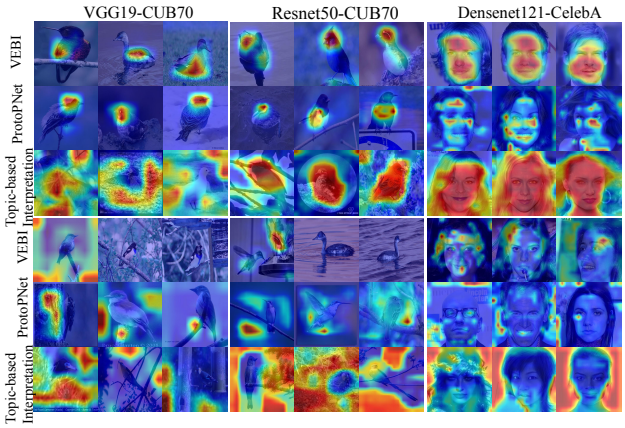

Figure 6. Visual explanations with the highest (top) and lowest (bottom) IoU coverage.

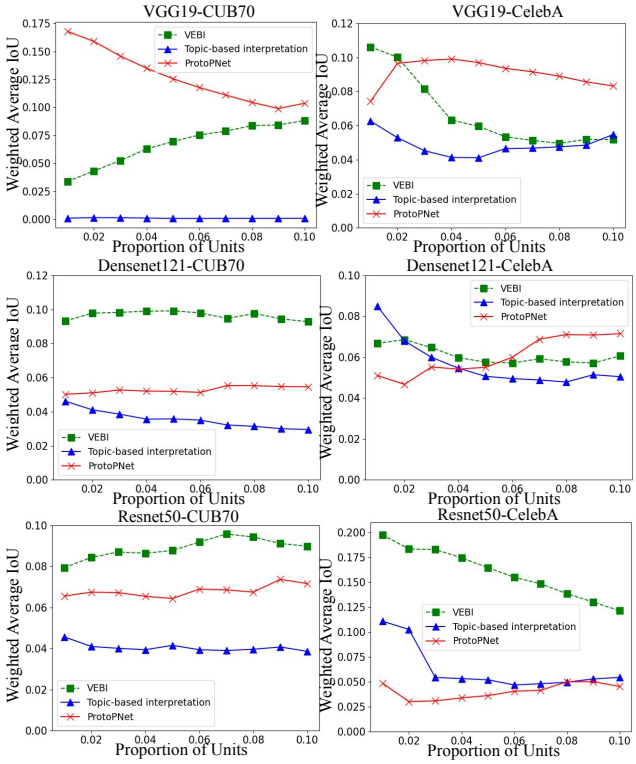

Figure 7. Dataset-level comparison among interpretation methods over CNNs trained on the CelebA and CUB-70 datasets.

5.7. Part-Level Coverage Rate

In this section, we consider the obtained thresholds (corresponding to the proportion of units in Table 2) for comparing the interpretation methods in terms of semantic part coverage rate. To do so, we illustrate in Figure 8 the semantic part coverage accuracy computed by Eq. 4 according to the obtained optimum thresholds.

As can be seen, VEBI has the higher coverage rate in the higher number of semantic parts in models trained on both

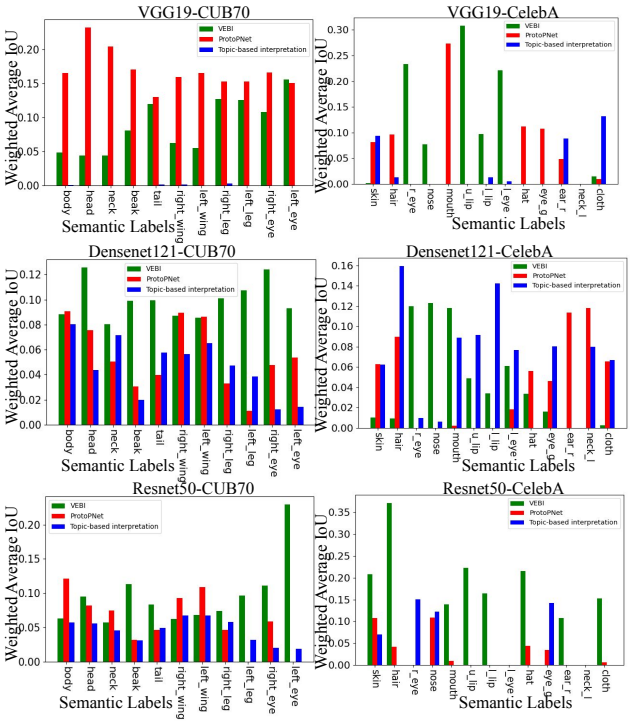

Figure 8. Semantic part-level comparison among interpretation methods over CNNs trained on the CelebA and CUB-70 datasets.

datasets, such as Densenet121-CUB70, ResNet50-CUB70, Resnet50-CelebA, and VGG19-CelebA.

To sum up, taking the quantitative analysis presented in Sec. 5.6 and this section, VEBI outperforms Topic-based interpretation and ProtoPNet in terms of coverage rate in both datasets and semantic parts levels. This reveals that explanation heatmaps generated from interpretations provided by VEBI can highlight uniformly the variety of semantic parts with high coverage rate in comparison to those of ProtoPNet and Topic-Based interpretation methods.

6. Conclusion

We propose an evaluation protocol for assessing the effectiveness of interpretation methods in identifying the relevant latent features encoded in a base model. To do so, we enhanced the compared methods to be capable of generating visual explanations. In addition, we examined two intensity thresholding scenarios to determine the best strategy for obtaining an optimal intensity threshold. Our experiments suggest that VEBI outperforms others in two datasets CUB-70 and CelebA on the considered CNN architectures. Moreover, in contrast to estimating binarization parameters in a per-heatmap basis, our results suggest that higher performance can be achieved when considering sets of heatmaps.

Acknowledgements: This work is supported by the UAntwerp BOF DOCPRO4-NZ Project (id 41612) "Multimodal Relational Interpretation for Deep Models".

References

[1] Pei An, Junxiong Liang, Kun Yu, Bin Fang, and Jie Ma. Deep structural information fusion for 3d object detection on lidar–camera system. *Computer Vision and Image Understanding*, 214:103295, 2022.

[2] Hamed Behzadi-Khormouji and Habib Rostami. Fast multiresolution occlusion: a method for explaining and understanding deep neural networks. *Applied Intelligence*, 51(4):2431–2455, 2021.

[3] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and survey of explanation methods for black box models. *CoRR*, abs/2102.13076, 2021.

[4] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

[5] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[6] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8730–8738, 2018.

[7] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

[8] Felix Grün, Christian Rupprecht, Nassir Navab, and Federico Tombari. A taxonomy and library for visualizing learned features in convolutional neural networks. *In International Conference on Machine Learning (ICML) Workshops*, 2016.

[9] Bin Guan, Jinkun Yao, Shaoquan Wang, Guoshan Zhang, Yueming Zhang, Xinbo Wang, and Mengxuan Wang. Automatic detection and localization of thighbone fractures in x-ray based on improved deep learning method. *Computer Vision and Image Understanding*, page 103345, 2022.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[12] Vidhya Kamakshi, Uday Gupta, and Narayanan C Krishnan. Pace: Posthoc architecture-agnostic concept extractor for explaining cnns. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[13] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

[14] Okan Köpüklü, Stefan Hörmann, Fabian Herzog, Hakan Cevikalp, and Gerhard Rigoll. Dissected 3d cnns: Temporal skip connections for efficient online video processing. *Computer Vision and Image Understanding*, 215:103318, 2022.

[15] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020.

[16] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[18] José Antonio Oramas Mogrovejo, Kaili Wang, and Tinne Tuytelaars. Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks. *In International Conference on Learning Representations (ICLR)*, 2019.

[19] Gesina Schwalbe and Bettina Finzel. XAI method properties: A (meta-)study. *CoRR*, abs/2105.07190, 2021.

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

[21] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *In International Conference on Learning Representations (ICLR) Workshops*, 2015.

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. (CNS-TR-2011-001), 2011.

[23] Sandareka Wickramanayake, Wynne Hsu, and Mong-Li Lee. Towards fully interpretable deep neural networks: Are we there yet? *CoRR*, abs/2106.13164, 2021.

[24] Chih-Kuan Yeh, Been Kim, Sercan Ömer Arik, Chun-Liang Li, Pradeep Ravikumar, and Tomas Pfister. On concept-based explanations in deep neural networks. *CoRR*, abs/1910.07969, 2020.

[25] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[26] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[27] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018.