FEATURES DISENTANGLEMENT FOR EXPLAINABLE CONVOLUTIONAL NEURAL NETWORKS

Pasquale Coscia, Angelo Genovese, Fabio Scotti, Vincenzo Piuri

Department of Computer Science, Università degli Studi di Milano, Italy
{pasquale.coscia, angelo.genovese, fabio.scotti, vincenzo.piuri}@unimi.it

ABSTRACT

Explainable methods for understanding deep neural networks are currently being employed for many visual tasks and provide valuable insights about their decisions. While post-hoc visual explanations offer easily understandable human cues behind neural networks' decision-making processes, comparing their outcomes still remains challenging. Furthermore, balancing the performance-explainability trade-off could be a time-consuming process and require a deep domain knowledge. In this regard, we propose a novel auxiliary module, built upon convolutional-based encoders, which acts on the final layers of convolutional neural networks (CNNs) to learn orthogonal feature maps with a more discriminative and explainable power. This module is trained via a disentangle loss which specifically aims to decouple the object from the background in the input image. To quantitatively assess its impact on standard CNNs, and compare the quality of the resulting visual explanations, we employ metrics specifically designed for semantic segmentation tasks. These metrics rely on bounding-box annotations that may accompany image classification (or recognition) datasets, allowing us to compare both ground-truth and predicted regions. Finally, we explore the impact of various self-supervised pre-training strategies, due to their positive influence on vision tasks, and assess their effectiveness on our considered metrics.

Index Terms— Explainable AI (XAI), ResNet, self-supervised learning (SSL), disentanglement.

1. INTRODUCTION

Artificial Intelligence (AI) pervades our everyday life, and its role has been continually expanding and evolving [1]. Its impact on shaping future technologies and providing new capabilities also demands careful consideration, especially in critical domains. For instance, healthcare, autonomous vehicles, and environmental monitoring are critical sectors where analyzing the decision-making processes of deep neural networks is essential for identifying errors and saving lives [2,3]. Despite ongoing efforts, quantifying their fairness, trustworthiness, and transparency still remains challenging due to their complex and, to some extent, abstract definitions, as well as the heterogeneity of infrastructures and tools [4,5,6].

More specifically, explainable AI (XAI) [7] includes techniques to produce more transparent, trustable and fair models. Revealing biased data within datasets used to train deep learning models is crucial for ensuring fair decisions, or identifying vulnerabilities, and enhancing the security of AI systems against adversarial attacks. While attempts to explain deep neural networks have primarily focused on their inner inference processes (*e.g.*, on analyzing features or weights), visual explanations highlighting input regions responsible for their decision still remain one of the most effective and straightforward approach, particularly for convolutional-based neural networks. Class Activation Mapping (CAM) [8], for example, identifies discriminative regions using a weighted combination of activation maps immediately preceding a global pooling layer. Since many architectures may not follow this structure, Grad-CAM [9] still focuses on the last convolutional layers but computes the average gradients of the final feature maps. In fact, as the discriminative capability of the network increases towards its classification stage, last layers typically provide valuable insights to explain their predictions. Alternative methods focus on perturbing the original input [10], or overcome issues of gradient-based approaches, *e.g.*, saturation and false confidence [11, 12]. Nevertheless, these attempts in quantifying visual explanations have yet to reach maturity in effectively evaluating the quality of the provided explanations [13, 14] as the comparisons often rely on evaluation protocols that are not easily replicable, or on ad-hoc architectures. Most importantly, different cues can simultaneously be a valid visual explanation. As an example, the Content Heatmap (CH) [14], or the average drop/increase metrics [15, 16], can effectively assess the quality of coarse representations, but they may fail in capturing fine-grained details. To address these issues and measure both coarse and fine details, we focus on disentangling learned concepts and evaluating the networks'

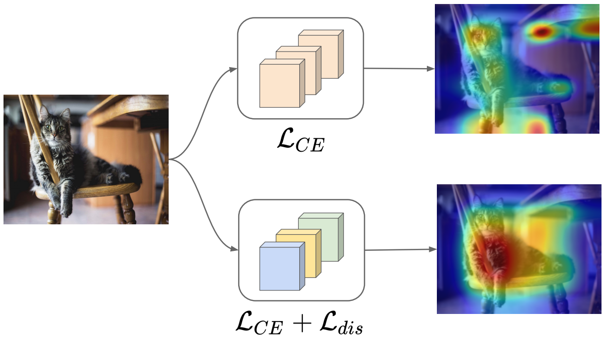

Fig. 1: Training deep neural networks for image classification using a standard cross-entropy loss may limit their capability in visually explaining decisions. The proposed features disentanglement towards final convolutional layers enhances this localization capability, which is fundamental for fostering more explainable methods.

This work was supported in part by the EC under project Chips JU EdgeAI (101097300), by the Italian MUR under PON project GLEAN, and by project SERICS (PE00000014) under the MUR NRRP funded by the EU - NextGenerationEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor Italian MUR can be held responsible for them.

localization capability.

Our paper introduces several key contributions aimed at enhancing the interpretability of CAM-based XAI methods. We propose an innovative auxiliary module based on two convolutional-based encoders, which uses the features from the final convolutional layer. We expect that each class has its own prototypical elements to be evaluated: the content and the background. Our encoders learn more discriminative features related to these two “concepts” while backpropagating by means of a disentangle loss which is employed in the auxiliary module to generate orthogonal feature maps. By doing so, we enhance the discriminative capacity of our model without interfering with standard architectures used for image classification (see Fig. 1). In addition, we leverage segmentation metrics to provide simple and effective quantitative measurements of visual explanations. Our approach is evaluated on datasets with annotated segmentation masks for each object in the input image, allowing us to assess the performance of CAM-based XAI methods in a more detailed manner. Furthermore, our study extends beyond the model architecture itself to explore the impact of various pre-training procedures. We analyze how different pre-training techniques affect the performance of our approach and demonstrate that their impact is network-dependent. Notably, our findings highlight the valuable contribution of orthogonal features obtained by our auxiliary module in achieving increased explainability in the context of pre-training procedures.

The remainder of this paper is organized as follows. Section 2 presents related works. Section 3 describes our proposed approach and the main pre-training techniques used in this work. Section 4 presents our results and also discusses the main limitations of our approach. Finally, Section 5 concludes the paper.

2. RELATED WORK

Visual Explainability. An effective visual explanation should exhibit two crucial characteristics: it should be class-discriminative while simultaneously capturing fine-grained details [9]. Grad-CAM [9], for example, produces localization maps measuring the gradients flowing in the last convolutional layers. Contrastive approaches offer an alternative strategy for improving the consistency of visual explanations via positive and negative pairs [13]. However, a quantitative assessment of their explanatory capabilities is still in its early phase, also marked by the lack of a unified consensus within the community. Coarser metrics [14], for example, focus on the entire object, while finer metrics [17, 18], consider the spatial selectiveness of the method. Depending on the domain, these measures may not fully interpret the decisions thus requiring more investigation.

Feature Disentanglement. Deep neural networks commonly involve the cross-entropy loss for the image classification task which demonstrated to not properly focus on inter-class separability and intra-class compactness [19]. In this regard, several attempts focus on the learning strategy. For example, spherical, and geometric losses, confirmed to enhance the discriminative power of deep features [19, 20, 21]. In our work, we propose a novel loss to properly focus on two elements of the input image, *i.e.*, the content and the background, and increase the localization capability of a CNN.

Pre-training and fine-tuning. Many computer vision tasks, *e.g.*, image classification, object detection, or image segmentation, rely on the “pre-training and fine-tuning” paradigm [22] since ImageNet [23], JFT [24], and modern large-scale multi-modal datasets [25] can significantly boost the performance on downstream tasks. The zero-shot capability of recent deep learning models [25] demonstrate the impact of large-scale data. In some scenarios (*e.g.*, industrial, or medical) fine-grained annotations could be limited, and previously mentioned datasets may not provide useful representations due to the different characteristics of the domains. In these cases, to avoid time-consuming and expensive annotation processes, self-supervised methods [26, 27] permit to learn useful visual features. A common solution consists in considering pretext tasks, *e.g.*, image colorization [28], contrastive learning [29], image inpainting [30] or image jigsaw puzzle [31], that use pseudo-labels which can be automatically generated using some attributes of the input data. Siamese networks also demonstrated to be effective in learning suitable representations [32]. Our work also explores the impact of features disentanglement when employing self-supervised pre-training strategies.

3. METHOD

To effectively understand the decision-making process of convolutional neural networks, emphasis is often placed on the final convolutional layers due to their ability to retain both spatial information and high-level semantics. In this regard, we propose a novel auxiliary module that processes the last convolutional features. This aims to enforce the network to learn orthogonal and more discriminative representations, thereby enhancing its explainability capability. We investigate the common image classification setting wherein the network is composed of three distinct parts: a convolutional-based encoding, a global average pooling layer, and a final classification stage. In the following, we introduce our features disentanglement module, present the corresponding training loss, and finally discuss our investigated self-supervised pre-training strategies.

Features Disentanglement. To perform a feature disentanglement, we employ additional encoders to maximize the distance between the learned features. Our aim is to obtain low-dimensional representations of two elements in the image, the content and the background, and increase their distance. More specifically, our model-agnostic method depends upon two additional encoders, namely, $\varphi^B(\cdot)$ and $\varphi^C(\cdot)$, for background and content, respectively. These encoders reduce the depth dimension of the feature maps related to the N^{th} layer (see Fig. 2), from N_K to N_C, where N_K denotes the number of channels before the global average pooling layer and N_C denotes the number of classes in our domain, while their spatial size is not modified. The encoders convert the feature map $\hat{\mathbf{F}} \in \mathbb{R}^{W \times H \times N_K}$ to $\mathbf{F} \in \mathbb{R}^{W \times H \times N_C}$. Then, we transform each channel to a $d-$dimensional vector (with $d = W \times H$) and constrain it onto the surface of a hyper-sphere using L_2 normalization. Given the i^{th} $W \times H$ output channel of each encoder, *i.e.*, φ_i^B and φ_i^C, we learn disentangled representations considering two objectives based on a similarity measure defined by the inner product:

$$\text{sim}(\varphi_i^C, \varphi_i^B) = \langle \varphi_i^C, \varphi_i^B \rangle, \tag{1}$$

which also represents the cosine similarity between two our d-dimensional vectors. In our case, $\text{sim}(\cdot, \cdot) \in [0, 1]$. For the sake of simplicity, we drop from the notation the input of each encoder. Firstly, we consider an *intra-encoder* similarity, *i.e.*, a cosine similarity between all the feature pairs of each encoder:

$$\mathcal{L}_{dis}^1 = \frac{1}{N} \sum_{\substack{i \neq j \\ i,j \in \{1,\dots,N_C\}}} \left[\text{sim}(\varphi_i^B, \varphi_j^B) + \text{sim}(\varphi_i^C, \varphi_j^C) \right], \tag{2}$$

where N is a normalization factor employed in computing the mean value. Secondly, we consider an *inter-encoder* similarity, *i.e.*, a similarity measure among features associated with a specific channel in

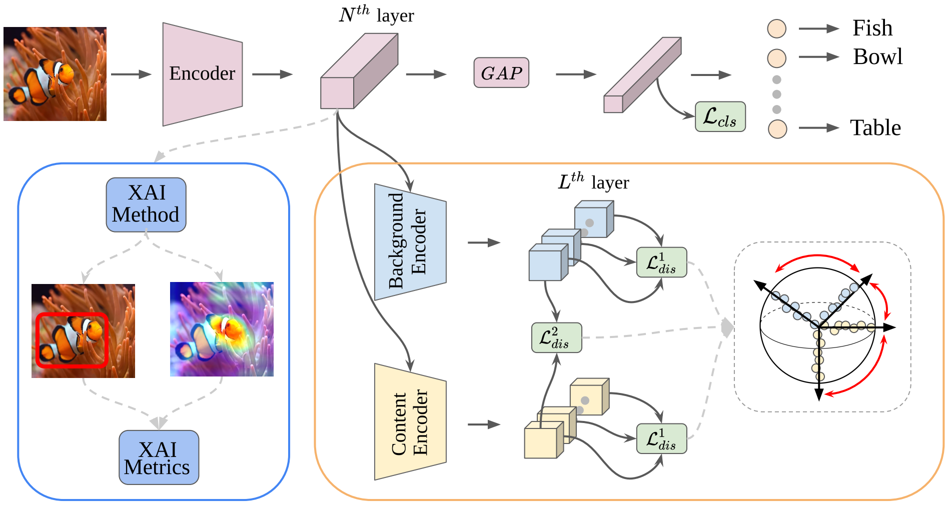

Fig. 2: We propose an auxiliary module to learn orthogonal feature maps extracted from two convolutional-based encoders (content and background encoders). Each feature map in the L^{th} layer is normalized to represent a unit vector on a d-dimensional sphere. Our disentangle loss reduces the similarity among the feature maps of each encoder ($\mathcal{L}^1_{dis}$) as well as between the corresponding feature maps of both encoders ($\mathcal{L}^2_{dis}$) by increasing the angle between the feature maps (shown in red).

each encoder:

$$\mathcal{L}^2_{dis} = \frac{1}{N}\sum_{i=1}^{N_C} \text{sim}(\varphi_i^B, \varphi_i^C), \tag{3}$$

where N is defined similarly as in the previous case. Decreasing both $\mathcal{L}^1_{dis}$ and $\mathcal{L}^2_{dis}$ allows the network to encourage diversity in the representations. The optimization procedure involves using a standard cross-entropy loss along with our disentangle loss, which we aim to minimize:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_{dis}\mathcal{L}_{dis} = \mathcal{L}_{CE} + \lambda_{dis}(\mathcal{L}^1_{dis} + \mathcal{L}^2_{dis}). \tag{4}$$

Self-supervised Pre-training. Our aim is also to investigate the role of pre-training procedures on representation learning for effectively visualizing the decision-making processes behind the decisions of convolutional neural networks. Specifically, we consider the general pipeline of a self-supervised learning procedure, *i.e.*, after applying a self-supervised pre-training method, we transfer the learned parameters to a standard convolutional neural network to solve the downstream image classification task. In this case, we firstly train the network using a pretext task, and then fine-tune it on our dataset additionally considering the disentangle loss. As pre-training procedures, we consider the following techniques:

- *Image colorization* [28]: we perform color regression on the L*a*b space. To enforce colors, we consider grayscale inputs and introduce Conv2d-BatchNorm2D-ReLU layers using 3×3 kernels, and Upsampling layers, to double the size of the feature maps, and restore the original image size. After colorization pre-training, the convolutional filters in the first layer are replaced by randomly initialized filters to handle RGB images, while the up-sampling layers are discarded;
- *Image rotation* [33]: this strategy consists in learning image representations by recognizing the rotation applied to the input image and training the network using a $4-$way classification task;
- *SimCLR* [29]: this method consists in a contrastive learning of visual representations. SimCLR adopts a composition of transformations applied to each input image for obtaining two correlated views and maximize their agreement.

We additionally consider ImageNet [23] as pre-training procedure, since it represents the main dataset for pre-training visual models and could act as the most effective pre-training strategy. Except for the ImageNet pre-training, we consider the 5 closest classes to our selected ones using WordNet similarity [34] for the other methods.

Explainable Metrics. We rely on an image classification dataset containing bounding-box annotations which are transformed into binary segmentation masks highlighting the image regions the network should focus on. After applying different post-hoc visual explanation methods, we also binarize their outcomes using a threshold α. A higher value of α corresponds to increased confidence in the network. Finally, we assess the pixel-level comparison between the ground-truth and predicted masks using the following metrics: Intersection over Union (IoU), or Jaccard Index, precision, recall, accuracy and Dice coefficient.

4. RESULTS

Implementation Details. Each encoder implements 3 Conv-BatchNorm2d-ReLU layers that halve the depth size at each step in order to output N_C channels. Both width and height of the processed feature maps are not altered using stride and padding appropriately. We train each network for 250 epochs, and use as many epochs for fine-tuning. Input images are upsampled to 224×224 pixels. We set the batch size to 128, the learning rate to 0.0005 and use the Adam optimizer and the “1cycle” learning rate policy [35]. We adopt an L_2 regularization for the weights. To implement the SimCLR augmentation, we employ a temperature parameter $\tau = 0.07$ and a 128-d representation for the *latent* vector. To obtain segmentation masks from an explainabile method, we set the threshold α to 0.5.

Dataset. For our experiments, we use the TinyImageNet dataset [36], a modified subset of the popular ImageNet [23] dataset, obtained considering 200 classes. It provides 500 training images per class, 50 validation images per class and 10000 unlabeled images for testing. Image dimension is 64 × 64 pixels. For our experiments, we randomly select 5 classes, divide the training images into training and validation sets using an 80-20 split, and use the validation images as test set. As training classes, we include *European fire salamander, Sea slug, Tabby, Sports car* and *Flagpole*. On the other hand, *Cougar, Drumstick, Lion, Syringe*, and *Projectile* are chosen as pre-training classes.

Post-hoc Methods. To extract visual explanations, we consider the following methods: GradCAM [9], GradCAM++ [15], LayerCAM [12], and ScoreCAM [11].

Quantitative Results. Table 1 reports the accuracies of our networks with different pre-training strategies. The best result is achieved by the ResNet-101 architecture with features disentanglement, which also presents the greatest number of parameters among the residual-based networks. While ImageNet pre-training improves the classification results in most cases, notably, the other techniques appears more effective when the number of parameters is limited. It is worth mentioning that these pre-training methods have been demonstrated to effectively learn the main features of the domain, especially when a large quantity of data is available. Specifically, if only a standard classification loss is considered, the ImageNet pre-training clearly obtains the best performance, except for the ResNet-18 network since no pre-training, or the SimCLR pre-training, appear sufficient for a better learning. While the self-supervised pre-training strategies are beneficial for both ResNet-18/50 networks, we note a detrimental impact on the ResNet-101 network, in which the accuracy is reduced by $\sim$ 2-10 %. Including our disentangle loss, we observe a significant improvement of the accuracy by $\sim$ 2-5 %, especially for the ResNet-18 network.

In Table 2, we report our explainable metrics training the networks from scratch and using the ImageNet pre-training. Almost all the metrics benefit from our disentanglement. We note a remarkable gain for both the recall and Dice coefficient metrics confirming a more powerful localization capability for the objects in the input images, which, in turn, also improves the accuracy metric. On the other hand, precision values remain almost stable, or decrement. We also point out that the considered post-hoc visualization methods show similar trends. Furthermore, we note a limited impact of our approach on the GradCAM method. This limitation could be indicative of the challenges it faces in identifying discriminative regions, especially in scenarios involving a huge gap between convolutional and classification features. These results are also confirmed in Table 3 where our disentangle loss increases the explainable metrics by large margin compared to a standard classification loss. Despite the limited number of considered pre-training images, our networks benefits from a pre-training procedure.

Qualitative Results. Fig. 4 shows our qualitative results using a ResNet-18 network trained from scratch, and pre-trained used the self-supervised image colorization procedure. Regardless of the visual explanation method considered, a standard classification loss focuses more on the frontal region showing the plate of the car, while a colorization pre-training allows the network to also highlight objects, *i.e.*, other cars, present in the background. When the disentangle loss is included, more parts of the car are taken into account, leading to decisions that rely more heavily on the entirety of the object. In fact, more parts of the car are highlighted. In this example, the GradCAM method appears to suffer more from the lack of pre-training, and the disentangle loss shifts the learned concepts of the network to a less interpretable representation. Fig. 5, instead, depicts the results of a ResNet-101 network pre-trained using SimCLR and the ImageNet dataset. In the first case, the network struggles in learning appropriate characteristics of this specific class (sea slug) since most of the highlighted regions appear to belong to the background. The ImageNet pre-training clearly improves the discrimination capabilities and the different post-hoc visualization methods focus more on the main characteristics of this class. By contrast, our disentangle loss increases the discriminative power of the network since it focuses more on stripes and shadows which represent typical elements of the images of this class.

Pre-training **Method**	**ResNet-18** (11.7 M)	**ResNet-50** (25.5 M)	**ResNet-101** (44.5 M)
-	92.44 / **94.06**	90.84 / **93.65**	90.86 / **93.67**
ImageNet [23]	91.70 / **92.11**	**95.31** / 94.08	96.06 / **98.01**
Colorization [28]	89.63 / **91.60**	88.93 / **90.92**	88.48 / **90.45**
Rotation [33]	87.64 / **92.00**	**91.25** / 85.32	81.72 / **86.14**
SimCLR [29]	**92.11** / 90.80	**91.70** / 90.06	88.12 / **88.51**

Table 1: **Top@1 Accuracy (%)**. The values on the left correspond to the training procedure using only the classification loss, while those on the right refer to the additional disentangle loss. We underline the best value for each network.

Ablation Studies. To compute the weight λ_{dis} for our augmented loss, we conduct an ablation study on the ResNet-18 architecture, testing it with 4 different values without considering any pre-training method, as presented in Table 4. The introduction of this loss leads to a slight increase in accuracy by 1.19% when λ_{dis} is set to 10^{-1}. Notably, among all the metrics, both precision and dice coefficient outperform their previous values by a considerable margin. The remaining metrics demonstrate improved values regardless of the lambda parameter. These results also reveal a weak correlation between our metrics and the network's accuracy.

Furthermore, to evaluate the effect of the threshold α for binarizing the heatmaps, we measure our metrics with $\alpha \in [0.1, 0.9]$. The results are shown in Fig. 6 for the ResNet-18 architecture, and highlight that the ImageNet pre-training is clearly the most effective pre-training method in focusing on the correct parts of the input image. While rotation pre-training appears useful for learning more discriminative features, colorization pre-training has a detrimental impact instead. A high threshold clearly increases the precision of the network by $\sim 20\%$ compared with a low threshold value, but also decreases the recall metric due to less identified correct regions. The best accuracy is achieved with $\alpha = 0.3$ while a low IoU metric for all the values demonstrates that the predicted binary masks poorly align with the provided bounding boxes. A less penalization is shown by the Dice coefficient. When our disentangle loss is added, we note a regularization effect which notably reduces the gap between all the pre-training methods. More specifically, IoU and Recall metrics increases almost by 15% at different threshold values while, for the self-supervised pre-training strategies, the achieved results reach the ones obtained by a supervised pre-training (*e.g.*, ImageNet). In some cases, both colorization and rotation pre-training, obtain superior performance. Finally, we observe that, for low threshold values, IoU and recall metrics increase with our disentangle loss, while precision and accuracy exhibit contrasting behavior.

Failure Cases. Finally, we also show in Fig. 7 some failures cases where our disentangle loss produces a shift in the learned concepts.

Limitations. Our proposed loss function, while enhancing explainable metrics, introduces an additional number of trainable param-

Network	IoU	Precision	Recall	Accuracy	Dice Coeff.	Avg.
GradCAM [9]						
ResNet-18	0.17 / **0.20**	**0.69** / 0.52	0.22 / **0.31**	**0.53** / 0.50	0.28 / **0.30**	**0.38** / 0.37
ResNet-50	**0.20** / 0.03	**0.67** / 0.43	**0.24** / 0.04	**0.52** / 0.47	**0.31** / 0.06	**0.39** / 0.21
ResNet-101	0.21 / **0.34**	**0.66** / 0.58	0.27 / **0.49**	0.53 / **0.59**	0.33 / **0.46**	0.40 / **0.50**
GradCAM++ [15]						
ResNet-18	0.20 / **0.50**	**0.70** / 0.65	0.25 / **0.76**	0.54 / **0.65**	0.32 / **0.64**	0.40 / **0.64**
ResNet-50	0.20 / **0.49**	**0.64** / **0.64**	0.25 / **0.73**	0.51 / **0.63**	0.31 / **0.62**	0.38 / **0.62**
ResNet-101	0.19 / **0.50**	0.61 / **0.65**	0.25 / **0.74**	0.50 / **0.65**	0.30 / **0.64**	0.37 / **0.64**
ScoreCAM [11]						
ResNet-18	0.22 / **0.49**	**0.67** / 0.66	0.27 / **0.72**	0.53 / **0.65**	0.34 / **0.63**	0.41 / **0.63**
ResNet-50	0.21 / **0.37**	**0.64** / 0.56	0.28 / **0.51**	0.51 / **0.60**	0.33 / **0.48**	0.39 / **0.50**
ResNet-101	**0.25** / 0.21	0.64 / **0.70**	**0.31** / 0.28	0.52 / **0.58**	**0.36** / 0.27	**0.42** / 0.41
LayerCAM [12]						
ResNet-18	0.21 / **0.50**	**0.70** / 0.64	0.27 / **0.77**	0.54 / **0.64**	0.39 / **0.64**	0.42 / **0.64**
ResNet-50	0.24 / **0.51**	**0.68** / 0.65	0.30 / **0.78**	0.54 / **0.65**	0.36 / **0.64**	0.42 / **0.65**
ResNet-101	0.23 / **0.50**	0.65 / **0.66**	0.30 / **0.75**	0.54 / **0.66**	0.36 / **0.64**	0.42 / **0.64**

(a) *From Scratch*

Network	IoU	Precision	Recall	Accuracy	Dice Coeff.	Avg.
GradCAM [9]						
ResNet-18	**0.32** / 0.15	**0.72** / 0.53	**0.41** / 0.19	**0.60** / 0.49	**0.46** / 0.23	**0.50** / 0.32
ResNet-50	**0.32** / 0.26	**0.73** / 0.56	0.41 / **0.43**	**0.59** / 0.49	**0.46** / 0.39	**0.50** / 0.43
ResNet-101	**0.36** / 0.10	**0.73** / 0.39	**0.46** / 0.13	**0.62** / 0.50	**0.51** / 0.15	**0.54** / 0.25
GradCAM++ [15]						
ResNet-18	0.41 / **0.49**	**0.72** / 0.65	0.54 / **0.75**	**0.64** / **0.64**	0.56 / **0.63**	0.57 / **0.63**
ResNet-50	0.37 / **0.49**	**0.73** / 0.65	0.49 / **0.75**	0.62 / **0.64**	0.51 / **0.63**	0.54 / **0.63**
ResNet-101	0.40 / **0.51**	**0.74** / 0.67	0.52 / **0.76**	0.64 / **0.66**	0.55 / **0.65**	0.57 / **0.65**
ScoreCAM [11]						
ResNet-18	0.40 / **0.49**	**0.73** / 0.65	0.51 / **0.75**	0.64 / **0.65**	0.54 / **0.64**	0.56 / **0.64**
ResNet-50	0.35 / **0.39**	**0.71** / 0.58	0.45 / **0.61**	**0.61** / 0.58	0.49 / **0.51**	0.52 / **0.53**
ResNet-101	0.36 / **0.50**	**0.70** / 0.58	0.47 / **0.83**	**0.61** / 0.60	0.50 / **0.63**	0.53 / **0.63**
LayerCAM [12]						
ResNet-18	0.43 / **0.48**	**0.71** / 0.65	0.56 / **0.71**	**0.65** / 0.64	0.57 / **0.62**	0.58 / **0.62**
ResNet-50	0.39 / **0.50**	**0.71** / 0.65	0.52 / **0.76**	0.62 / **0.64**	0.53 / **0.63**	0.55 / **0.64**
ResNet-101	0.42 / **0.50**	**0.71** / 0.68	0.55 / **0.74**	0.63 / **0.67**	0.56 / **0.64**	0.57 / **0.65**

(b) *ImageNet*

Table 2: Explainable metrics with (a) no pre-training and (b) ImageNet pre-training. The left values refer to the $\mathcal{L}_{CE}$ loss while the right values additionally consider the $\mathcal{L}_{dis}$ loss. We underline the best value for each metric.

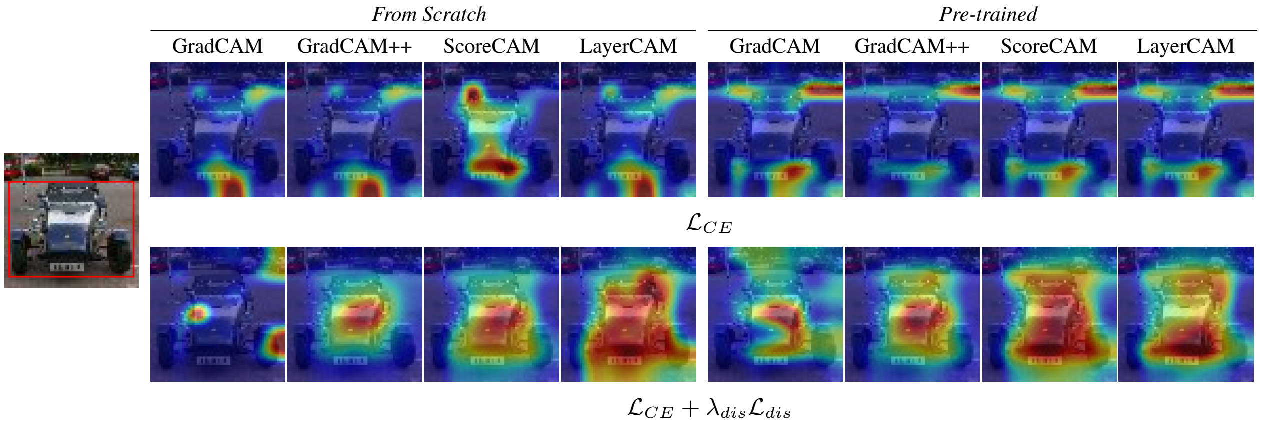

Fig. 4: Results for different post-hoc visualization methods using the ResNet-18 network trained from scratch (left) and pre-trained using image colorization [28] (right) for a sample of the *car* class. The 1^{st} row only uses the classification loss while the 2^{nd} row adds our disentangle loss.

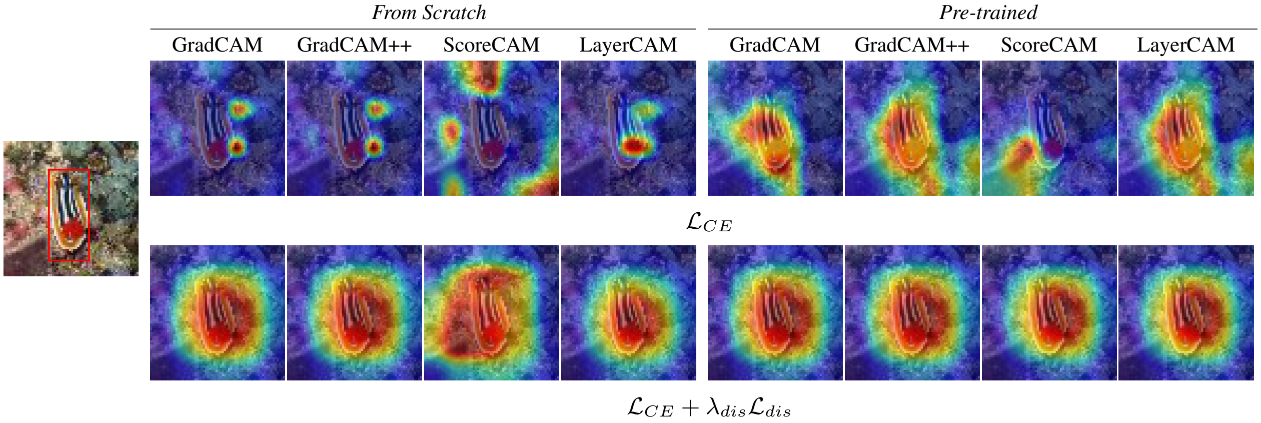

Fig. 5: Results for different post-hoc visualization methods using the ResNet-101 network pre-trained using the SimCLR [29] method (left) and pre-trained on ImageNet [23] (right) for a sample of the *sea slug* class. The 1^{st} row only uses the classification loss while the 2^{nd} row adds our disentangle loss.

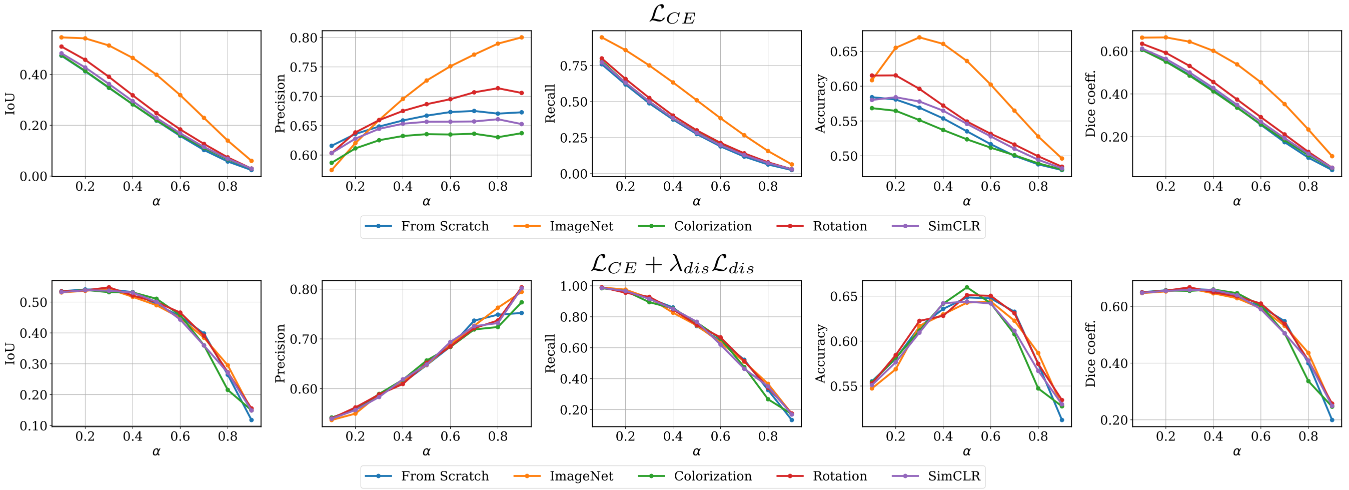

Fig. 6: Explainable metrics using different values in the range $[0.1, 0.9]$ for binarizing the heatmaps of the ScoreCAM [11] method. These metrics refer to the ResNet-18 model. The 2^{nd} row show the results using our disentangle loss, which acts both as a regularizer and increases the explainable metrics.

Network	IoU	Precision	Recall	Accuracy	Dice Coeff.	Avg.
			Colorization [28]			
ResNet-18	0.20 / **0.50**	0.60 / **0.66**	0.27 / **0.75**	0.51 / **0.65**	0.32 / **0.64**	0.38 / **0.64**
ResNet-50	0.21 / $\underline{\textbf{0.51}}$	0.63 / **0.64**	0.27 / $\underline{\textbf{0.79}}$	0.52 / **0.65**	0.34 / $\underline{\textbf{0.65}}$	0.39 / **0.65**
ResNet-101	0.26 / **0.49**	0.64 / **0.66**	0.32 / **0.74**	0.55 / **0.65**	0.38 / **0.63**	0.43 / **0.63**
			Rotation [33]			
ResNet-18	0.24 / $\underline{\textbf{0.51}}$	**0.73** / 0.65	0.30 / **0.77**	0.56 / $\underline{\textbf{0.66}}$	0.37 / $\underline{\textbf{0.65}}$	0.44 / **0.65**
ResNet-50	0.23 / $\underline{\textbf{0.51}}$	0.64 / **0.65**	0.28 / **0.78**	0.54 / **0.65**	0.35 / **0.64**	0.41 / **0.65**
ResNet-101	0.20 / **0.50**	0.66 / **0.67**	0.25 / **0.75**	0.53 / $\underline{\textbf{0.66}}$	0.31 / **0.64**	0.39 / **0.64**
			SimCLR [29]			
ResNet-18	0.21 / **0.50**	0.65 / **0.66**	0.25 / **0.74**	0.54 / **0.65**	0.32 / **0.63**	0.27 / **0.64**
ResNet-50	0.24 / **0.50**	**0.70** / 0.66	0.30 / **0.75**	0.55 / **0.65**	0.37 / **0.64**	0.43 / **0.64**
ResNet-101	0.20 / $\underline{\textbf{0.51}}$	$\underline{\textbf{0.74}}$ / 0.66	0.25 / **0.77**	0.54 / $\underline{\textbf{0.66}}$	0.32 / $\underline{\textbf{0.65}}$	0.41 / **0.65**

Table 3: Explainable metrics using different self-supervised pre-training strategies for the LayerCAM [12] method. The values on the left refer to the $\mathcal{L}_{CE}$ loss, while the values on the right also consider the $\mathcal{L}_{dis}$ loss. We underline the best value for each metric.

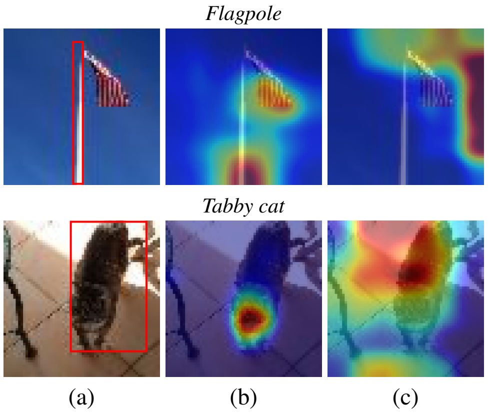

Fig. 7: **Failure cases.** Results for the ScoreCAM [11] method applied to the ResNet-101 pre-trained on ImageNet (1^{st} row) and pre-trained with the self-supervised image rotation strategy [33] (2^{nd} row). (a) shows the ground-truth annotation, (b) uses the $\mathcal{L}_{CE}$ loss, while, in column (c), the $\mathcal{L}_{CE} + \lambda_{dis}\mathcal{L}_{dis}$ loss is applied.

	Top@1 Accuracy (%)				
	$\mathcal{L}_{CE}$	$\mathcal{L}_{CE} + \lambda_{dis}\mathcal{L}_{dis}$			
		$\lambda_{dis} = 0.01$	$\lambda_{dis} = 0.1$	$\lambda_{dis} = 1$	$\lambda_{dis} = 5$
	92.44	93.22	**93.63**	92.85	91.31
	XAI Metrics				
IoU	0.22	0.47	**0.49**	**0.49**	**0.49**
Precision	**0.67**	0.63	0.66	0.65	0.64
Recall	0.27	0.73	0.72	**0.76**	**0.76**
Accuracy	0.53	0.62	**0.65**	0.64	0.64
Dice Coeff.	0.34	0.61	**0.63**	**0.63**	**0.63**
Avg.	0.41	0.61	**0.63**	**0.63**	**0.63**

Table 4: Ablation study related to the λ_{dis} parameter for the ResNet-18 network with no pre-training. We report the accuracies for different values of λ_{dis} parameter along with the explainable metrics for the ScoreCAM [11] method.

eters. Challenges also arise from the availability and reliability of ground-truth annotations, impeding direct comparisons. Furthermore, like any data-driven approach, our method may inadvertently capture biases inherent in the training data.

5. CONCLUSION

Our paper proposes a robust methodological framework aimed at improving the effectiveness of post-hoc visual explanations. We disentangle feature maps within the last convolutional layers using a novel encoder which focuses on discerning between background and content, as the main elements contributing to the decision-making process. We also design a loss function, based on the cosine similarity distance, to facilitate the learning of more discriminative features. Finally, our study systematically examines the impact of self-supervised pre-training strategies on the proposed method, which is also beneficial in the case of pre-training procedures. We demonstrate that our approach helps in understanding and improving AI models to enhance transparency and implement security measures, enabling the detection and correction of biased decisions.

6. REFERENCES

[1] I. Ahmed, G. Jeon, and F. Piccialli, "From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where," *IEEE Trans. Ind. Inform.*, 2022.

[2] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE Trans. Neural Netw. Learn. Syst.*, 2021.

[3] P. Wang and N. Vasconcelos, "A generalized explanation framework for visualization of deep learning model predictions," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[4] O. Vermesan, V. Piuri, F. Scotti, A. Genovese, R. Donida Labati, and P. Coscia, "Explainability and interpretability concepts for edge ai systems," in *Advancing Edge Artificial Intelligence: System Contexts*, 2023.

[5] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du, "Explainability for large language models: A survey," *ACM Trans. Intell. Syst. Technol.*, 2024.

[6] A. Genovese, V. Piuri, and F. Scotti, "Towards explainable face aging with generative adversarial networks," in *ICIP*, 2019.

[7] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, and R. Ranjan, "Explainable ai (xai): Core ideas, techniques, and solutions," *ACM Comput. Surv.*, 2023.

[8] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016.

[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.

[10] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *ICCV*, 2017.

[11] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *CVPRW*, 2020.

[12] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "Layercam: Exploring hierarchical class activation maps for localization," *IEEE TIP*, 2021.

[13] V. Pillai, S. A. Koohpayegani, A. Ouligian, D. Fong, and H. Pirsiavash, "Consistent explanations by contrastive learning," in *CVPR*, 2022.

[14] V. Pillai and H. Pirsiavash, "Explainable models with consistent interpretations," in *AAAI*, vol. 35, 2021.

[15] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *WACV*, 2018.

[16] S. Poppi, M. Cornia, L. Baraldi, and R. Cucchiara, "Revisiting the evaluation of class activation mapping for explainability: A novel metric and experimental analysis," in *CVPRW*, 2021.

[17] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *IJCV*, 2018.

[18] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," in *BMVC*, 2018.

[19] H. Choi, A. Som, and P. Turaga, "Amc-loss: Angular margin contrastive loss for improved explainability in image classification," in *CVPRW*, 2020.

[20] F. Pernici, M. Bruni, C. Baecchi, and A. D. Bimbo, "Regular polytope networks," *IEEE TNNLS*, 2022.

[21] B. Barz and J. Denzler, "Deep learning on small datasets without pre-training using cosine loss," in *WACV*, 2020.

[22] K. He, R. Girshick, and P. Dollar, "Rethinking imagenet pretraining," in *ICCV*, 2019.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[24] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *ICCV*, 2017.

[25] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "LAION-5b: An open large-scale dataset for training next generation image-text models," in *NeurIPS*, 2022.

[26] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE TPAMI*, 2021.

[27] C. J. Reed, X. Yue, A. Nrusimha, S. Ebrahimi, V. Vijaykumar, R. Mao, B. Li, S. Zhang, D. Guillory, S. Metzger, K. Keutzer, and T. Darrell, "Self-supervised pretraining improves self-supervised pretraining," in *WACV*, 2022.

[28] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *CVPR*, 2017.

[29] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[30] Y. Zeng, Z. Lin, J. Yang, J. Zhang, E. Shechtman, and H. Lu, "High-resolution image inpainting with iterative confidence feedback and guided upsampling," in *ECCV*, 2020.

[31] G. Camporese, E. Izzo, and L. Ballan, "Where are my neighbors? exploiting patches relations in self-supervised vision transformer," in *BMVC*, 2022.

[32] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021.

[33] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.

[34] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, 1995.

[35] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019.

[36] Y. Le and X. S. Yang, "Tiny imagenet visual recognition challenge," URL http://cs231n.stanford.edu/tiny-imagenet-200.zip, 2015.