TAME: Attention Mechanism Based Feature Fusion for Generating Explanation Maps of Convolutional Neural Networks

Mariano Ntrougkas
CERTH-ITI
Thessaloniki, Greece, 57001
ntrougkas@iti.gr

Nikolaos Gkalelis
CERTH-ITI
Thessaloniki, Greece, 57001
gkalelis@iti.gr

Vasileios Mezaris
CERTH-ITI
Thessaloniki, Greece, 57001
bmezaris@iti.gr

***Abstract*—The apparent "black box" nature of neural networks is a barrier to adoption in applications where explainability is essential. This paper presents TAME (Trainable Attention Mechanism for Explanations)[1], a method for generating explanation maps with a multi-branch hierarchical attention mechanism. TAME combines a target model's feature maps from multiple layers using an attention mechanism, transforming them into an explanation map. TAME can easily be applied to any convolutional neural network (CNN) by streamlining the optimization of the attention mechanism's training method and the selection of target model's feature maps. After training, explanation maps can be computed in a single forward pass. We apply TAME to two widely used models, i.e. VGG-16 and ResNet-50, trained on ImageNet and show improvements over previous top-performing methods. We also provide a comprehensive ablation study comparing the performance of different variations of TAME's architecture.[2]**

***Index Terms*—CNNs, Deep Learning, Explainable AI, Interpretable ML, Attention.**

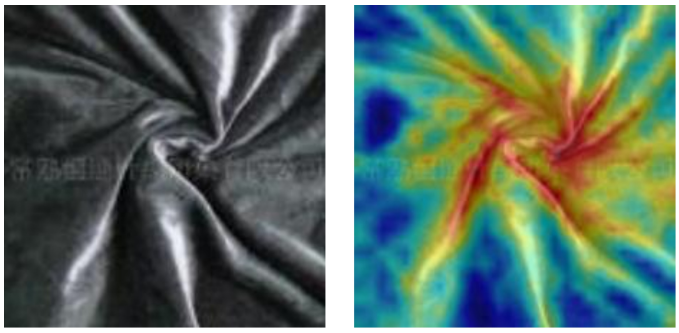

Fig. 1: An explanation produced by TAME. The input image belongs to the class "velvet", which cannot be localized. The produced explanation highlights the salient features of the image explaining the decision of the classifier.

I. Introduction

Convolutional neural networks (CNNs) [17] have achieved exceptional performance in many important visual tasks such as breast tumor detection [6], video summarization [3] and event recognition [10]. The trade-off between model performance and explainability, and the end-to-end learning strategy, leads to the development of CNNs that many times act as "black box" models that lack transparency [12]. This fact makes it difficult to convince users in critical fields, such as healthcare, law, and governance to trust and employ such systems, thus limiting the adoption of AI [2], [12]. Therefore, it is necessary to develop solutions that address these challenges.

Explainable artificial intelligence (XAI) is an active research area in machine learning. XAI focuses on developing explainable techniques that help users of AI systems to comprehend, trust and more efficiently manage them [4], [20]. For the image classification task, a diverse range of post-hoc explanation approaches exist that in a second step take the trained model and try to uncover its decision strategy [20]. These methods produce an explanation map, highlighting salient input features. We should note that these methods should not be confused with approaches targeting weakly supervised learning tasks such as weakly supervised object localization or segmentation [16], which also generate heatmaps as an intermediate step, and their goal is to locate the region of the target object rather than to explain the classifier's decision (e.g. see the example depicted in Fig. 1).

Gradient-based methods [5], [22] were probably among the first to appear in the XAI domain. These methods use gradient information to produce explanations, but they are strongly affected by noisy gradients, and the explanations contain high-frequency variations [1]. Perturbation-based methods [19], [28], perturb the input and observe changes in the output, thus do not suffer from gradient-based problems as above. Similarly, response-based methods [8], [21], [27] combine a model's intermediate representations, or features, to generate explanations. However, most methods of the two latter categories described above are computationally expensive because

This work was supported by the EU Horizon 2020 programme under grant agreement H2020-101021866 CRiTERIA.

[1]Source code is made publicly available at: https://github.com/bmezaris/TAME

each input requires many forward passes for an accurate explanation map to be produced.

To address the above limitation, L-CAM [11] trains an attention mechanism to combine feature maps from the last convolutional layer of a frozen CNN model and produce high quality explanations in one forward pass. However, L-CAM, by design, uses the feature maps of only the last convolutional layer, and thus, may not be able to adequately capture all the information contained in the CNN model. To this end, we propose TAME (Trainable Attention Mechanism for Explanations), which exploits intermediate feature maps extracted from multiple layers of any CNN model. These features are then used to train a multi-branch hierarchical attention architecture for generating class-specific explanation maps in a single forward pass. We provide a comprehensive evaluation study of the proposed method on ImageNet [7] using two popular CNN models (VGG-16 [23], ResNet-50 [13]) and popular XAI measures [5], demonstrating that TAME achieves improved explainability performance over other top-performing methods in this domain.

II. Related Work

In this section, we briefly survey the state-of-the-art XAI approaches that are mostly related to ours. For a more comprehensive review the interested reader is referred to [4], [20].

Most XAI approaches can be roughly categorized into response-, gradient- and perturbation-based. Gradient-based methods [5], [22] compute the gradient of a given input with backpropagation and modify it in various ways to produce an explanation map. Grad-CAM [22], one of the first in this category, uses global average pooling in the gradients of the target network's logits with respect to the feature maps to compute weights. The explanation maps are obtained as the weighted combination of feature maps and the computed weights. Grad-CAM++ [5] similarly uses gradients to generate explanation maps. These methods suffer the same issues as the gradients they use: neural network gradients can be noisy and suffer from saturation problems for typical activation functions such as ReLU and Sigmoid [1].

Perturbation-based methods [19], [28] alter the input and produce explanations based on the change in the confidence of the original prediction; thus, avoid problems related with noise gradients. For instance, RISE [19] utilizes Monte Carlo sampling to generate random masks, which are then used to perturb the input image and generate a respective CNN score. Using the generated scores as weights, the explanation is derived as the weighted combination of the various random masks. Thus, RISE, as most methods in this category, requires many forward passes through the network to generate an explanation, increasing the inference time considerably.

Finally, response-based methods [8], [11], [21], [27] use feature maps or activations of layers in the inference stage to interpret the decision-making process of a neural network. One of the earliest methods in this category, CAM [29], uses the output of the global average pooling layer as weights, and computes the weighted average of the features maps at the final convolutional layer. CAM requires the existence of such a global average pooling layer, restricting its applicability to only this type of architectures. SISE [21], and later Ada-SISE [27], aggregate feature maps in a cascading manner to produce explanation maps of any DCNN model. Similarly, Poly-CAM [8] upscales feature maps to the dimension of the largest spatial dimension feature map and combines them in a cascading manner. The above methods require many forward passes to produce an explanation. L-CAM [11] mitigates the above limitation using a learned attention mechanism to compute class-specific explanations in one forward pass. However, it can only harness the salient information of one set of feature maps. TAME also falls into the response-based category and operates in one forward pass, but contrarily to [11], it uses a trainable hierarchical attention module to exploit feature maps from multiple layers and generate explanations of higher quality.

We should also note that the methods of [9], [15] take a somewhat similar approach to ours in that they produce explanations using an attention module and multiple sets of feature maps. However, these methods jointly train the attention model with the CNN to improve the image classification task. In contrast, TAME does not modify the target model, which has been pretrained (and remains frozen); instead, TAME functions as a post-hoc method, exclusively optimizing the attention module in a supervised learning manner to generate visual explanations. Thus, no direct comparisons can be drawn with [9], [15] as they provide explanations for a different (i.e. not the initial pretrained one), concurrently trained classifier.

III. TAME

A. Problem formulation

Let f be a trained CNN for which we want to generate explanation maps,

$$f : \mathcal{I} \rightarrow \mathbb{R}^{Classes}, \tag{1}$$

where, $\mathcal{I}$ is the set of all possible input tensors $\mathcal{I} = \{\boldsymbol{I} \mid \boldsymbol{I} : \boldsymbol{C} \times \boldsymbol{W} \times \boldsymbol{H} \rightarrow \mathbb{R}\}$, $\boldsymbol{C} = \{1, \ldots, C\}$, $\boldsymbol{W} = \{1, \ldots, W\}$, $\boldsymbol{H} = \{1, \ldots, H\}$, $C, W, H \in \mathbb{R}$ are the input tensor dimensions [19], [21], and $Classes$ is the number of classes that f has been trained to recognize. E.g., for RGB images, $C = 3$, and the elements of a tensor instance are the image pixel values. Moreover, let $\mathcal{L}_i : \boldsymbol{C_i} \times \boldsymbol{W_i} \times \boldsymbol{H_i} \rightarrow \mathbb{R}$ be the feature map set corresponding to the ith layer of the CNN, where, C_i, W_i, H_i are the respective channel, width and height dimensions. We define a feature map set $\{\mathcal{L}\}^s$, where s is the set of layers for which we want to extract feature maps, i.e., $\{\mathcal{L}\}^s = \{\mathcal{L}_i \mid i \in s\}$.

Assume an attention module defined as in the following,

$$\text{AM} : \{\mathcal{L}\}^s \rightarrow \boldsymbol{E}, \tag{2}$$

where, the tensor $\boldsymbol{E}$ at the output of the attention module is the generated explanation map, $\boldsymbol{E} : \textbf{Classes} \times \boldsymbol{W_e} \times \boldsymbol{H_e} \rightarrow \{x \mid x \in \mathbb{R} \cap 0 < x < 1\}$, $W_e = \max\{W\}^s$ and $H_e = \max\{H\}^s$. Thus, explanation maps are class discriminative,

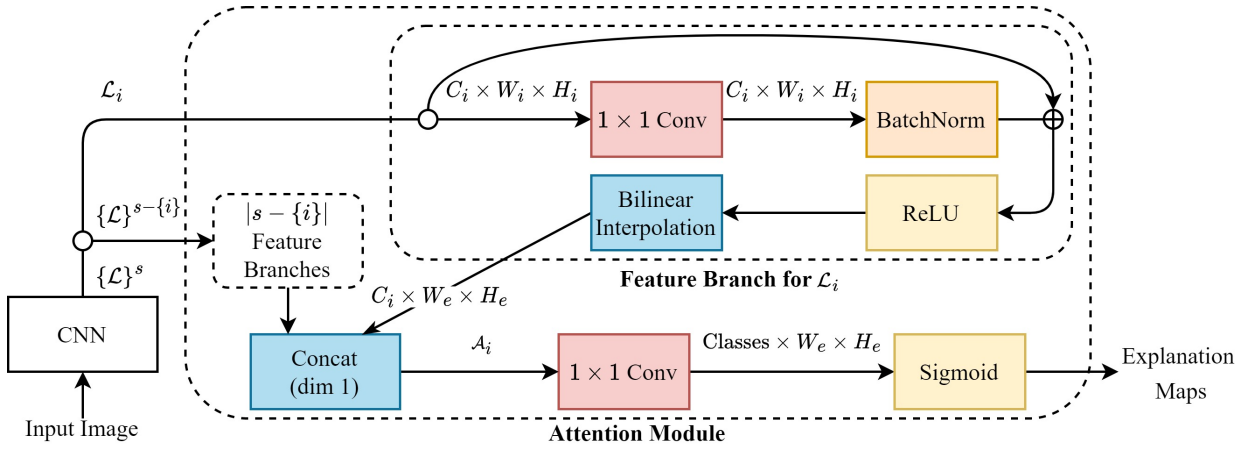

Fig. 2: TAME's attention module: Feature branches process feature maps to provide attention maps, which are concatenated and processed by the fusion branch (shown at the bottom of the attention module) to derive explanation maps.

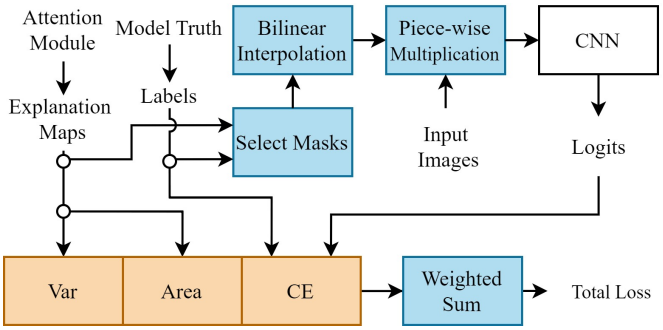

Fig. 3: TAME's training method. Var: Variation loss, Area: Area loss, CE: Cross-entropy loss. The explanation of an input image is first derived; it is then upscaled and piece-wise multiplied with the corresponding input image. Subsequently, the masked image does a second forward pass through the CNN to generate logits, which are used by the loss function to compute gradients and update the attention module's weights.

i.e., each slice of $\boldsymbol{E}$ along its first dimension corresponds to one of the classes that f has learned; moreover, the size of the spatial dimensions of these "class-specific" slices equal to the largest spatial dimensions in the set of feature maps.

Given the above formulation, the goal is to find an attention module architecture that can combine all the salient information contained in $\{\mathcal{L}\}^s$, and effectively train it.

B. Architecture

We propose the attention module architecture depicted in Fig. 2. In this architecture, there exists a separate feature branch for each feature map set that is included in $\{\mathcal{L}\}^s$ and one fusion branch. Each feature branch takes as input a single feature map set $\mathcal{L}_i$ and outputs an attention map set $\mathcal{A}_i$,

$$\text{FB} : \mathcal{L}_i \rightarrow \mathcal{A}_i, \tag{3}$$

where, $\mathcal{A}_i$ has the same channel and spatial dimensions as $\mathcal{L}_i$ and the final explanation map, respectively, i.e., $\mathcal{A}_i : \boldsymbol{C_i} \times \boldsymbol{W_e} \times \boldsymbol{H_e} \rightarrow \mathbb{R}$. The resulting attention maps are concatenated into an single attention map set $\{\mathcal{A}\}^s$, and forwarded into the fusion branch to generate the explanation map,

$$\text{FS} : \{\mathcal{A}\}^s \rightarrow \boldsymbol{E}. \tag{4}$$

The two branch types consist of different network components, as described in the following:

Feature branch: Each feature branch is a neural network that prepares the feature maps for the fusion branch. It consists of a 1×1 convolution layer with the same number of input and output channels, a batch normalization layer, a skip connection, a ReLU activation, and a bilinear interpolation that upscales the feature map to match the final explanation map's dimensions (the ablation study presented in Section IV assesses the importance of each part of the feature branch).

Fusion branch: It consists of a 1×1 convolutional layer that brings the number of the inputted channels to the number of classes that the CNN has been trained to recognize. Subsequently, a sigmoid activation function, $S(x) = \frac{1}{1-e^{-x}}$, is used to scale the attention map values to the range $(0, 1)$.

C. Training

The training procedure is shown in Fig. 3. An image is inputted to the CNN model, and the derived feature maps are forwarded to the attention module for generating the respective explanation maps and the model truth label. A model truth label is the CNN model's prediction of the input image's class, which may be different from the ground truth label. A single channel containing a class discriminative *explanation* is selected from the explanation map using the model truth label; this is used as the explanation of the input image with respect to the model truth class. The explanation is then upscaled to the dimensions of the input image using bilinear interpolation, and is piece-wise multiplied with the input image. The resulting masked image is then fed back into the CNN to generate logits. The logits, the original explanation maps, and the model truth labels are then used to compute the loss and through backpropagation update the weights of the

attention module, effectively training it. As already mentioned, the weights of the original CNN remain fixed to their original values for the whole training procedure.

The loss function used for training the proposed attention module is the weighted sum of three individual loss functions,

$$L(\boldsymbol{\Psi}, \text{logits}, \text{labels}) = \lambda_1 CE(\text{logits}, \text{labels}) + \lambda_2 \text{Area}(\boldsymbol{\Psi}) + \lambda_3 \text{Var}(\boldsymbol{\Psi}), \quad (5)$$

where, $\boldsymbol{\Psi}$ is the slice of the explanation map $\boldsymbol{E}$ corresponding to the model truth class of the input image, $CE()$, Area(), Var() are the cross-entropy, area and variation loss, respectively, and λ_1, λ_2, λ_3, are the corresponding regularization parameters. The cross-entropy loss uses the logits generated from the CNN with the masked input image and the model truth label to compute a loss value. This term trains the attention module to focus on salient parts of the image. The variation loss is the sum of the squares of the partial derivatives of the explanation $\boldsymbol{\Psi}$ in the x and y direction. This term penalizes fragmentation in the generated heatmaps. For the partial derivatives, we use the forward difference approximation. To this end, in the x direction we have $\frac{\partial \boldsymbol{\Psi}[x_m, y_m]}{\partial x} \approx \boldsymbol{\Psi}[x_m + 1, y_m] - \boldsymbol{\Psi}[x_m, y_m]$. Thus, using the forward difference approximation the variation loss is defined as,

$$\text{Var}(\boldsymbol{\Psi}) = \sum_{x,y} \left[\left(\frac{\partial \boldsymbol{\Psi}[x,y]}{\partial x} \right)^2 + \left(\frac{\partial \boldsymbol{\Psi}[x,y]}{\partial y} \right)^2 \right]. \quad (6)$$

Finally, the area loss is the mean of the explanation map $\boldsymbol{E}$ to the Hadamard power of λ_4, i.e.:

$$\text{Area}(\boldsymbol{\Psi}) = \sum_{x,y} \boldsymbol{\Psi}[x,y]^{\lambda_4}. \quad (7)$$

This term forces the attention module to output heatmaps that emphasize small focused regions in the input image instead of arbitrarily large areas.

D. Inference

During inference, the final sigmoid activation function in the attention module (Fig. 2) is replaced with a min-max normalization operator, $m(x) = \frac{x - \min(\boldsymbol{\Psi})}{\max(\boldsymbol{\Psi}) - \min(\boldsymbol{\Psi})}$; the $\min()$ and $\max()$ operators return the smallest and largest element of the input tensor, respectively. This is done for consistency with other literature works, such as [5], [22], on how the final explanation maps are scaled in order to be evaluated. The test image is then forward-passed through the CNN, producing explanation maps, which are then upscaled to the input image size. The derived model truth label can then be used to provide an explanation concerning the decision of the classifier.

IV. Experiments

A. Datasets and CNNs

We evaluate TAME on two popular CNNs pretrained on ImageNet: VGG-16 [23] and ResNet-50 [13]. We choose these two models to test the generality of our method because there are significant differences in the VGG and ResNet architectures. We obtain these pretrained networks using the `torchvision.models` library.

We train the attention module of our method with the ImageNet ILSVRC 2012 dataset [7]. This dataset contains 1000 classes, 1.3 million and 50k images for training and evaluation, respectively. Due to the prohibitively high cost of executing the literature's perturbation-based approaches that we use in the experimental comparisons, we use only 2000 randomly-selected testing images for testing (the same as in [11] to allow a fair comparison) and a different 2000 randomly selected images as a validation set.

B. Evaluation measures

In the experimental evaluation, two frequently used evaluation measures, Increase in Confidence (IC) and Average Drop (AD) [5], are utilized,

$$\text{AD}(v) = \sum_{i=1}^{\Upsilon} \frac{\max(0, f(\boldsymbol{I}_i) - f(\boldsymbol{I}_i \odot \phi_v(\boldsymbol{\Psi}_i)))}{\Upsilon f(\boldsymbol{I}_i)} 100, \quad (8)$$

$$\text{IC}(v) = \sum_{i=1}^{\Upsilon} \frac{\text{sign}(f(\boldsymbol{I}_i \odot \phi_v(\boldsymbol{\Psi}_i)) > f(\boldsymbol{I}_i))}{\Upsilon} 100, \quad (9)$$

where, $\phi_v()$ is a threshold function to select the $v\%$ higher-valued pixels of the explanation map, sign() returns 1 when the input condition is satisfied and 0 otherwise, Υ is the number of test images, $\boldsymbol{I}_i$ is the ith test image and $\boldsymbol{\Psi}_i$ is the corresponding explanation produced by TAME or any other method under evaluation. Intuitively, AD measures how much, on average, the produced explanation maps, when used to mask the input images, reduce the confidence of the model. In contrast, IC measures how often the explanation masks, when used to mask the input images, increase the confidence of the model. We threshold the explanation maps to test how well the pixels of the explanation map are ordered based on importance. Thus, using a smaller threshold results in a much more challenging evaluation setup.

C. Experimental setup

TAME is applied to VGG-16 using feature maps from three different layers. The VGG-16 consists of five blocks of convolutions separated by 2×2 max-pooling operations, as shown in Fig. 4. We choose one layer from each of the last three blocks, namely the feature maps output by the max-pooling layers of each block. We have also experimented on the feature maps output by the last convolution layer of each block. On the other hand, ResNet-50 consists of five stages. In the experimental evaluation, we use the feature maps from the final three stages.

TAME is trained using the loss function defined in (5) with the SGD (Stochastic Gradient Descent) algorithm. The biggest batch size that can fit in the graphics card's memory is used, as recommended in [25]. The learning rate is varied using the OneCycleLR policy described in [26]. The maximum learning rate used by the OneCycleLR policy is chosen using the LR finder test defined in [24]. The hyperparameters of the loss

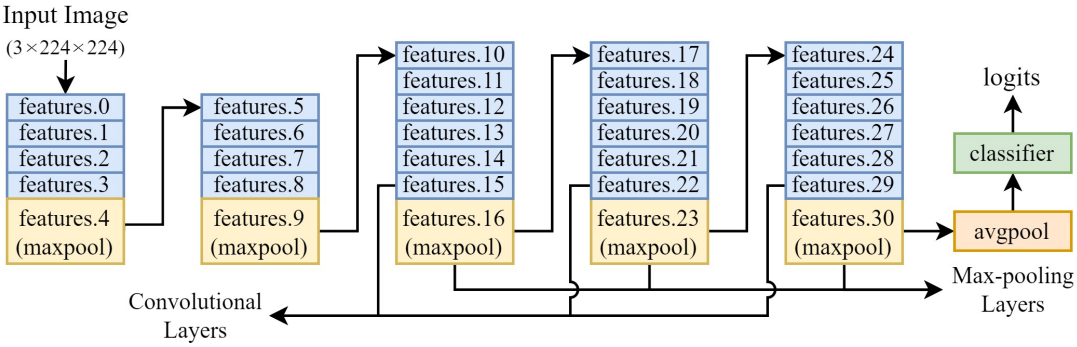

Fig. 4: The layers from which feature map sets are extracted on VGG-16. We denote by "Convolutional Layers" the three layers before the last three max-pooling layers. In the case of VGG-16, the layer before a max-pooling layer is the ReLU activation function. We use the same layer naming as the library `torchvision.models.feature_extraction`.

function ((5), (7)) are empirically chosen using the validation dataset, as: $\lambda_1 = 1.5, \lambda_2 = 2, \lambda_3 = 0.01, \lambda_4 = 0.3$.

We train the attention module for eight epochs in total and select the epoch for which the attention module achieved the best IC(15%) and AD(15%) in the validation set. That is, in this model selection procedure we opt for the measures at the 15% threshold because they are the most challenging measures to improve upon and provide more focused explanation masks.

During training, each image is transformed in the same way as with the original CNN, i.e., its smaller spatial dimension is resized to 256 pixels, random-cropped to dimensions $W = H = 224$, and normalized to zero mean and unit variance. The same is done during testing, except that center-cropping is used. The feature maps are extracted from the CNN using `torchvision.models.feature_extraction` library.

D. Quantitative Evaluation

The proposed method is compared against the top-performing approaches in the visual explanation domain for which source code is publicly available i.e. Grad-CAM [22], Grad-CAM++ [5], Score-CAM [28], RISE [19] and L-CAM [11]. The performance is measured using AD(v) and IC(v) on three different thresholds v of increasing difficulty, i.e., $v = 100\%, 50\%$ and 15%, similarly to the evaluation protocol of [11]. An ablation study is also conducted, to assess the importance of the different architecture components for VGG-16 and ResNet-50, as well as to showcase the effect of different layer selections in the VGG-16 model.

1) Comparison with the State-Of-The-Art: In Table I we highlight with bold letters the best result and underline the second best result for each measure, separately for each base model. We can see that TAME outperforms the gradient-based methods, and is competitive to the perturbation-based methods, obtaining the best results for the more demanding 15% measures while requiring only one forward pass.

2) Ablation Study: In Table II we highlight with bold letters the best results and underline the second best result for each measure in each model and layer selection. For the VGG-16 model, inspired from similar works in the literature suggesting that the last layers of the network provide more salient features [15], we report two sets of experiments, one that uses features extracted from the three last max-pooling layers and one where features are extracted from the layers exactly before the last three max-pooling layers (Fig. 4). There is a difference in the spatial dimensions of the explanation maps generated using the former or the latter layers for feature extraction, i.e., 28×28 versus 56×56, since the dimension of the explanation maps obtained by TAME is dictated by the largest feature map set (as explained in Section III). For the ResNet-50 model, we extract features from the outputs of the final three stages, resulting to an explanation map of 28×28 spatial dimensions. We examine the following variants of the proposed architecture:

No skip connection: It has been shown that the skip connection promotes a smoother loss landscape [18], thus contributing to training very deep neural networks. Even for shallower neural networks, such as the proposed attention module, we can benefit from using a skip connection. We see that by omitting the skip connection, we get worse results in ResNet-50. Similarly, for both baseline models we report worse performance for the harder 50% and 15% measures.

No skip + No batch norm: Batch normalization is used in CNNs for speeding up training and combating internal covariate shift [14]. Compared to the proposed architecture, we see that this variant generally performs better in the 100% measures, but this does not hold for the other measures. We compare the masks produced by this variant in Fig. 5.

Sigmoid in feature branch: In this variant we replace the ReLU function with the sigmoid function, which squeezes the input from $(-\infty, \infty)$ to the output $(0, 1)$. It is well known that the sigmoid function in deeper neural networks causes the vanishing gradient problem, making it more difficult to train the early layers of the CNN. We see again that the proposed architecture prevails for the more challenging 15% measures.

Two layers and *One layer*: In this case, the proposed attention module architecture is employed with fewer feature maps. The results when using just one layer, i.e., omitting the two earlier layers in the CNN pipeline (Fig. 4), are very similar to the L-CAM-Img method (as shown in Table I), which also

TABLE I: Comparison of TAME with other methods

Model	Measure	Grad-CAM [22]	Grad-CAM++ [5]	Score-CAM [28]	RISE [19]	L-CAM-Img [11]	TAME
VGG-16	AD(100%)	32.12	30.75	27.75	**8.74**	12.15	9.33
	IC(100%)	22.1	22.05	22.8	**51.3**	40.95	50
	AD(50%)	58.65	54.11	45.6	42.42	37.37	**36.5**
	IC(50%)	9.5	11.15	14.1	17.55	20.25	**22.45**
	AD(15%)	84.15	82.72	75.7	78.7	74.23	**73.29**
	IC(15%)	2.2	3.15	4.3	4.45	4.45	**5.6**
	Forward Passes (Inference)	**1**	**1**	512	4000	**1**	**1**
ResNet-50	AD(100%)	13.61	13.63	11.01	11.12	11.09	**7.81**
	IC(100%)	38.1	37.95	39.55	46.15	43.75	**54**
	AD(50%)	29.28	30.37	**26.8**	36.31	29.12	27.88
	IC(50%)	23.05	23.45	24.75	21.55	24.1	**27.5**
	AD(15%)	78.61	79.58	78.72	82.05	79.41	**78.58**
	IC(15%)	3.4	3.4	3.6	3.2	3.9	**4.9**
	Forward Passes (Inference)	**1**	**1**	2048	8000	**1**	**1**

TABLE II: Ablation study of TAME

Model	Feature Extraction	Architecture Variant	AD(100%)	IC(100%)	AD(50%)	IC(50%)	AD(15%)	IC(15%)
VGG-16	Max-pooling layers	Proposed Architecture	9.33	50	36.5	22.45	73.29	5.6
		No skip connection	10.09	45.25	36.44	20.65	74.85	5.15
		No skip + No batch norm	**5.92**	**57.9**	34.49	**24.2**	74.58	5.15
		Sigmoid in feature branch	7.22	55.65	38.4	21.6	79	4.85
		Two layers	10.72	45.45	**34.48**	23.05	**71.94**	**5.75**
		One layer	12.1	42.1	35.81	20.8	74.19	4.85
	Convolutional layers	Proposed Architecture	9.07	51.1	40.72	**20.9**	77.05	4.8
		No skip connection	**6.22**	58.85	41.47	**20.9**	79.12	3.8
		No skip + No batch norm	6.62	56.6	**40.48**	20.75	77.84	**4.95**
		Sigmoid in feature branch	6.8	**60**	42.17	19.75	80.73	4.1
		Two layers	10.99	45.85	40.89	19.55	**76.66**	4.8
		One layer	13.09	39.65	42.3	17.7	78.02	3.8
ResNet-50	Stage Outputs	Proposed Architecture	7.81	54	27.88	**27.5**	78.58	**4.9**
		No skip connection	**5.7**	**62.65**	46.58	18.25	89.32	2.3
		No skip + No batch norm	9.29	50.25	29.43	25.95	79.81	3.95
		Sigmoid in feature branch	9.11	53.3	45.68	18.1	86.95	3.15
		Two layers	9.48	47.05	**27.83**	25	**77.95**	4.25
		One layer	11.32	43.45	29.85	24.25	79.59	3.55

uses just one feature map set. All measures are improved when utilizing a second feature map set, i.e., excluding only the earliest layer in the CNN pipeline; however, the case is not the same clear when going from the two to three feature map sets, which are used in the proposed architecture. These mixed results could be attributed to the extra noise of feature maps taken earlier in a CNN pipeline.

We note that by omitting both the skip connection and the batch normalization in the feature branch architecture, we obtain generally better results in the case of the VGG-16 model, but this is not the case for the same architecture applied to the ResNet-50 model. In addition, all architectures struggle with the more difficult 15% measures compared to the proposed architecture. Although every architecture varies between models, the proposed architecture generalizes best across different models. Thus, our goal of finding an effective architecture across radically different models is achieved through the proposed architecture.

Fig. 5: Qualitative comparison between the proposed attention module and the 'no skip + no batch norm' variant, applied to VGG-16. We observe that for the 'no skip + no batch norm' architecture, the produced explanation map is more spread out, showing that even if it performs well on the 100% measures, it fails to precisely identify the salient regions in the image.

E. Qualitative Analysis

An extensive qualitative analysis is also performed using the ILSVRC 2012 ImageNet dataset in order to gain insight of the

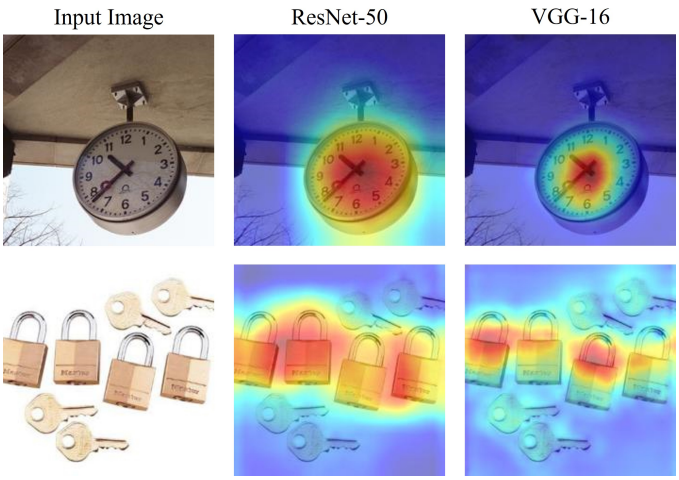

Fig. 6: TAME applied to ResNet-50 and VGG-16 for the ground truth class (Top image: "analog clock" (406), Bottom image: "padlock" (695). The explanation masks produced using the VGG-16 are more focused in comparison to the ones of ResNet-50.

proposed approach and appreciate its usefulness in real-world applications, e.g., understanding why an image was correctly classified or misclassified. The examples used in this study are depicted in Figs. 5, 6 and 7.

Fig. 5 compares TAME generated explanation maps with explanations generated by the "No skip + No batch norm" architecture examined in Section IV-D2. The improved ability of TAME to identify the salient image regions highlights the importance of evaluating the method using the AD and IC measures on multiple thresholds (Table II), and particularly the significance of the 15% measures over the 100% and 50% ones in determining the quality of generated explanations.

The differences between explanations produced using TAME on ResNet-50 and VGG-16 are examined in Fig. 6. We observe that explanations produced for the ResNet-50 model are generally more activated, and, in general, explanations produced for the two different CNN types attend different areas of the image. This suggests that ResNet-50 and VGG-16 classify images in fundamentally different ways, focusing on different features of an input image to make their predictions.

In Fig. 7, we provide class-specific explanation masks referring to the ground truth class but also to an erroneous but closely related class, for both ResNet-50 and VGG-16 models. The first image of Fig. 7a, depicts a spoonbill, a bird similar to the flamingo. Two significant differences between the spoonbill and the flamingo are the characteristic bill and the darker pink stripe on the wing of the spoonbill. We can see in the explanation maps of both models, that when choosing the class flamingo, there is no significance attributed to the bill, but, on the other hand, when the spoonbill class is chosen, the bill area is gaining significant attention. By comparing the explanation maps for adversarial classes, we can gain insight into important features which characterize a specific object against similar ones, and possibly gain new insight *from* the classifier. The second image in Fig. 7a is a similar case.

The examples of Fig. 7b demonstrate the potential of the explanation maps to be used for explaining multiple different classes contained in a single image, i.e., the "english fox-hound" and "soccer ball" image, and the "head cabbage" and "butternut squash" image.

Finally, in Fig. 7c we provide two cases of images that have been miscategorized, and use the explanations to understand what has gone wrong. The first image of Fig. 7c belongs to the "dingo" class (273) but is evidently misclassified as "timber wolf" from both CNN models. Using the explanations, we can identify important features on the image for each class and CNN model. The second image depicts a lighthouse. VGG-16 misclassifies this image as a "sundial". Again, using the explanations generated by TAME we can understand which features led the model to produce a wrong decision. For instance, in this case, we see that for both models the "sundial" explanations focus on the lighthouse roof, which might resemble a sundial, explaining the erroneous classification decision of VGG-16.

V. Conclusions

We proposed TAME, a novel method for generating visual explanations for various CNNs. This is accomplished by training a hierarchical attention module to extract information from feature map sets of multiple layers. Experimental results verified that TAME outperforms gradient-based methods and competes with perturbation-based, while, in contrast to them, requires only a single forward pass to generate explanations. Further research is needed to discover the limits of the proposed approach, e.g., generalizing it to non-CNN architectures.

References

[1] J. Adebayo, J. Gilmer, et al. Sanity checks for saliency maps. In *Proc. NIPS*, page 9525–9536, Montréal, Canada, 2018.

[2] J. Amann, A. Blasimme, et al. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC Medical Informatics and Decision Making*, 20(1):1–9, 2020.

[3] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras. Summarizing videos using concentrated attention and considering the uniqueness and diversity of the video frames. In *Proc. ICMR*, pages 407–415, New York, NY, USA, June 2022.

[4] A. B. Arrieta, N. Díaz-Rodríguez, et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[5] A. Chattopadhay, A. Sarkar, et al. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *Proc. IEEE WACV*, pages 839–847, 2018.

[6] J.-Y. Chiao, K.-Y. Chen, et al. Detection and classification the breast tumors using mask R-CNN on sonograms. *Medicine*, 98(19), 2019.

[7] J. Deng, W. Dong, et al. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE CVPR*, pages 248–255, Miami, USA, June 2009.

[8] A. Englebert, O. Cornu, and C. de Vleeschouwer. Backward recursive class activation map refinement for high resolution saliency map. In *Proc. ICPR*, 2022.

[9] H. Fukui, T. Hirakawa, et al. Attention branch network: Learning of attention mechanism for visual explanation. In *Proc. IEEE CVPR*, pages 10705–10714, 2019.

[10] N. Gkalelis, A. Goulas, D. Galanopoulos, and V. Mezaris. ObjectGraphs: Using objects and a graph convolutional network for the bottom-up recognition and explanation of events in video. In *Proc. IEEE CVPRW*, pages 3375–3383, June 2021.

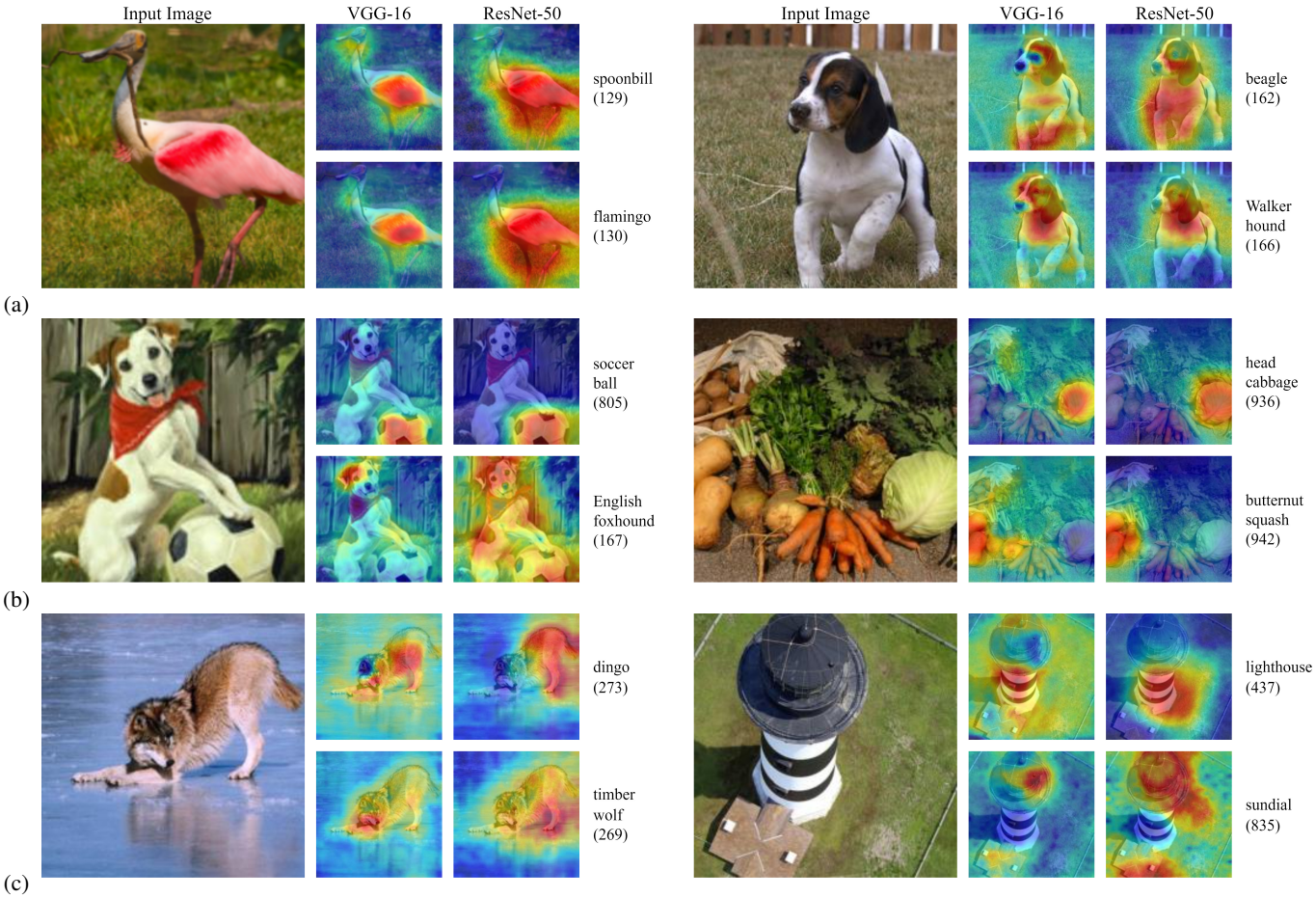

Fig. 7: Explanation for six input images. In each case we display four class-specific explanations, i.e., of the true (top) and an erroneous (bottom) class prediction of the input image, for both ResNet-50 and VGG-16. Figs. 7a, 7b depict examples of images with similar classes and with images containing multiple classes, respectively. In Fig. 7c two cases of misclassification are provided: dataset misclassification (left side example) and model misclasssification (right side example).

[11] I. Gkartzonika, N. Gkalelis, and V. Mezaris. Learning visual explanations for DCNN-based image classifiers using an attention mechanism. In *Proc. ECCV, Workshop on Vision with Biased or Scarce Data (VBSD)*, Oct. 2022.

[12] R. Hamon, H. Junklewitz, I. Sanchez, et al. Bridging the gap between AI and explainability in the GDPR: Towards trustworthiness-by-design in automated decision-making. *IEEE Computational Intelligence Magazine*, 17(1):72–85, 2022.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, pages 770–778, 2016.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, pages 448–456, 2015.

[15] S. Jetley, N. A. Lord, et al. Learn to pay attention. In *Proc. ICLR*, Vancouver, BC, Canada, May 2018.

[16] P.-T. Jiang, C.-B. Zhang, et al. LayerCAM: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021.

[17] A. Krizhevsky, I. Sutskever, et al. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, editors, *Proc. NIPS*, volume 25, 2012.

[18] H. Li, Z. Xu, et al. Visualizing the loss landscape of neural nets. *Proc. NIPS*, 31, 2018.

[19] V. Petsiuk, A. Das, and K. Saenko. RISE: randomized input sampling for explanation of black-box models. In *Proc. BMVC*, Newcastle, UK, September 2018.

[20] W. Samek, G. Montavon, et al. Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE*, 109(3):247–278, 2021.

[21] S. Sattarzadeh, M. Sudhakar, et al. Explaining convolutional neural networks through attribution-based input sampling and block-wise feature aggregation. In *Proc. AAAI*, volume 35, pages 11639–11647, 2021.

[22] R. R. Selvaraju, M. Cogswell, et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proc. IEEE ICCV*, pages 618–626, 2017.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, San Diego, CA, USA, May 2015.

[24] L. N. Smith. Cyclical learning rates for training neural networks. In *Proc. IEEE WACV*, pages 464–472, 2017.

[25] L. N. Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[26] L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Proc. Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386, 2019.

[27] M. Sudhakar, S. Sattarzadeh, et al. Ada-SISE: adaptive semantic input sampling for efficient explanation of convolutional neural networks. In *Proc. IEEE ICASSP*, pages 1715–1719, 2021.

[28] H. Wang, Z. Wang, et al. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proc. IEEE CVPRW*, pages 24–25, 2020.

[29] B. Zhou, A. Khosla, et al. Learning deep features for discriminative localization. In *Proc. IEEE CVPR*, pages 2921–2929, 2016.