LFI-CAM: Learning Feature Importance for Better Visual Explanation

Kwang Hee Lee[1,*,**], Chaewon Park[1,*], Junghyun Oh[1,2,*] and Nojun Kwak[2]

[1]Boeing Korea Engineering and Technology Center(BKETC)
[2]Seoul National University

Abstract

*Class Activation Mapping (CAM) is a powerful technique used to understand the decision making of Convolutional Neural Network (CNN) in computer vision. Recently, there have been attempts not only to generate better visual explanations, but also to improve classification performance using visual explanations. However, previous works still have their own drawbacks. In this paper, we propose a novel architecture, LFI-CAM***(Learning Feature Importance Class Activation Mapping), which is trainable for image classification and visual explanation in an end-to-end manner. LFI-CAM generates attention map for visual explanation during forward propagation, and simultaneously uses attention map to improve classification performance through the attention mechanism. Feature Importance Network (FIN) focuses on learning the feature importance instead of directly learning the attention map to obtain a more reliable and consistent attention map. We confirmed that LFI-CAM is optimized not only by learning the feature importance but also by enhancing the backbone feature representation to focus more on important features of the input image. Experiments show that LFI-CAM outperforms baseline models' accuracy on classification tasks as well as significantly improves on previous works in terms of attention map quality and stability over different hyper-parameters.*

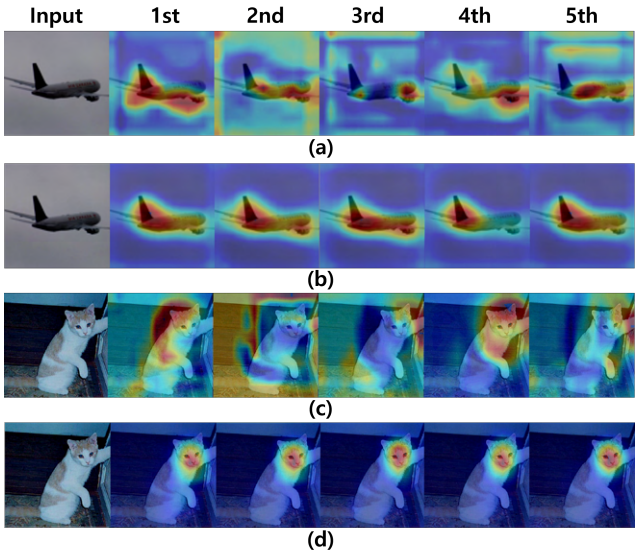

Figure 1. Examples of stability test on visual explanation. Each row displays CAM results of ABN or LFI-CAM models that were trained with various (5) hyper-parameters. As illustrated, ABN's CAM results are unreliable and inconsistent even for same test images despite the similar accuracies of the models. On the other hand, LFI-CAM results in much more consistent CAM images with better visual quality. (a)(c) ABN on STL10 (a) and Cat&Dog (c), (b)(d) LFI-CAM on STL10 (b) and Cat&Dog (d).

1. Introduction

As Convolutional Neural Network (CNN) models have become mainstream in computer vision tasks [1, 13, 6, 10, 17, 5, 8, 9], a rising need to understand the rationale behind models' decision has surfaced. Most deep neural networks are considered as black box due to the huge number of parameters and implicit non-linearity. We currently use metrics such as accuracy, precision, etc. for evaluation but these metrics can be misleading or inaccurate. To empower humans to trust the model, models should be equipped with the capability of providing human-comprehensible explanation on why it made certain decisions.

To address this need, several visual explanation methods have been proposed [20, 16, 15, 21, 2, 14, 12, 4, 18, 11] and are being widely used for various recognition tasks. These methods include, but are not limited to CAM [21] , Grad-CAM [15], Grad-CAM++ [2], LIME [14], RISE [12], ABN [4], and Score-CAM [18]. Broadly speaking, we

[*] indicates equal contribution.
[**] indicates corresponding author.
[***] https://github.com/TrustworthyAI-kr/LFI-CAM

can categorize the aforementioned methods into 4 categories: response-based, gradient-based, perturbation-based, and hybrid-based visual explanation.

Response-based. CAM [21] is a response-based visual explanation model which replaces the fully connected layer with Global Average Pooling (GAP) and projects weight matrix onto the channel-wise averaged feature maps. This method is restrictive as it requires architecture-sensitive changes in the original network, with degradation in classification accuracy compared to non-interpretable models.

Gradient-based. Grad-CAM [15] is a gradient-based visual explanation model that leverages the global average pooling of partial derivatives to capture the importance of a feature map for a target class. It fails to localize multiple occurrences of the same class and the entire region of the object. Grad-CAM++ [2] builds upon Grad-CAM's logic by capturing the weighted average of positive partial gradients to resolve the downsides of Grad-CAM. Both models require an extra back-propagation step during inference time.

Perturbation-based. LIME [14] applies perturbations on the input to learn a locally-weighted linear regression model that presents image regions as explanation that have the highest positive weight in approximating the true label. Though it is model-agnostic and simple, it requires additional regularization and is time-consuming. RISE [12] estimates the importance of input image regions as the predicted score by randomly sampling masks.

Hybrid-based. Score-CAM [18] is a hybrid of perturbation-based and response-based model. It uses attention maps as masks on the original image, and a forward-passing score on the target class is obtained and then aggregated as a weighted sum of score-based weights and attention maps. Though it achieves high accuracy and stable results compared to gradient-based methods, Score-CAM is very slow and time-consuming as it needs as many inferences as the number of feature maps to obtain CAM. Attention Branch Network (ABN) [4] is a hybrid model that uses a response-based model with the attention mechanism. It optimizes the loss term, which is the sum of attention loss and perception loss. ABN's limitation is that it often results in an unstable and suboptimal attention map for certain hyper-parameter settings (See Fig. 1).

Inspired by ABN and Score-CAM, we propose a novel architecture, LFI-CAM, which follows a similar structure to ABN. However, to constrain the attention map generation process as close as possible to the original CAM method, the LFI-CAM attention branch treats the feature maps as masks and obtains feature importance scores for each feature map to generate the attention map in a similar manner to Score-CAM. Unlike Score-CAM, LFI-CAM's Feature Importance Network (FIN) in the attention branch learns the feature importance for each feature map during training. Hence, LFI-CAM's attention map is generated much faster than Score-CAM during forward propagation.

LFI-CAM is composed of two parts: attention branch and perception branch. Attention branch plays an important role because it not only generates an attention map for visual explanation by learning the feature importance, but also uses attention maps to improve classification performance using the attention mechanism. The perception branch extracts feature maps and predicts a class through the attention mechanism using the feature map from the convolutional layer and attention map. LFI-CAM is trainable for image classification and visual explanation in an end-to-end manner and outputs more reliable and consistent attention maps with smaller model parameters than ABN. Our key contributions in this work are summarized as follows:

(1) We propose a new architecture LFI-CAM for image classification and visual explanation based on Class Activation Mapping with a simple but efficient learnable feature importance for each feature map.

(2) LFI-CAM learns the feature importance of attention maps in an intuitive and understandable way and leverages attention mechanism to improve classification performance and generate more reliable and consistent attention map simultaneously during forward propagation. When compared to Score-CAM, our model is equivalent in visual explanation quality but much faster in speed. Also, it results in better attention map quality and classification accuracy with smaller network parameters compared to ABN.

(3) As a gradient-free method, LFI-CAM bridges the gap between perturbation-based and CAM-based methods with much faster inference speed than Score-CAM.

(4) LFI-CAM is not architecture-sensitive and can be easily applied to various baseline models such as ResNet [6], DenseNet [8], ResNeXt [19] and SENet [7] by combining the baseline model with the Feature Importance Network and the attention mechanism.

2. Preliminary

2.1. Attention Branch Network (ABN)

Attention Branch Network (ABN) [4] was proposed not only to improve classification accuracy, but also to provide enhanced attention map for visual explanation simultaneously during inference time, by applying the attention mechanism. ABN is composed of the feature extractor, attention branch and perception branch. To create an attention map, the attention branch generates $K \times h \times w$ feature map through multiple 1×1 convolution layers, and integrates the feature map into one channel by applying a single 1×1 convolution. Finally, the sigmoid function is applied to a $1 \times h \times w$ feature map for normalization. Here K is the number of categories in dataset and also the number of channels, while h and w are the feature map's height and width.

We have observed that ABN outputs unreliable and in-

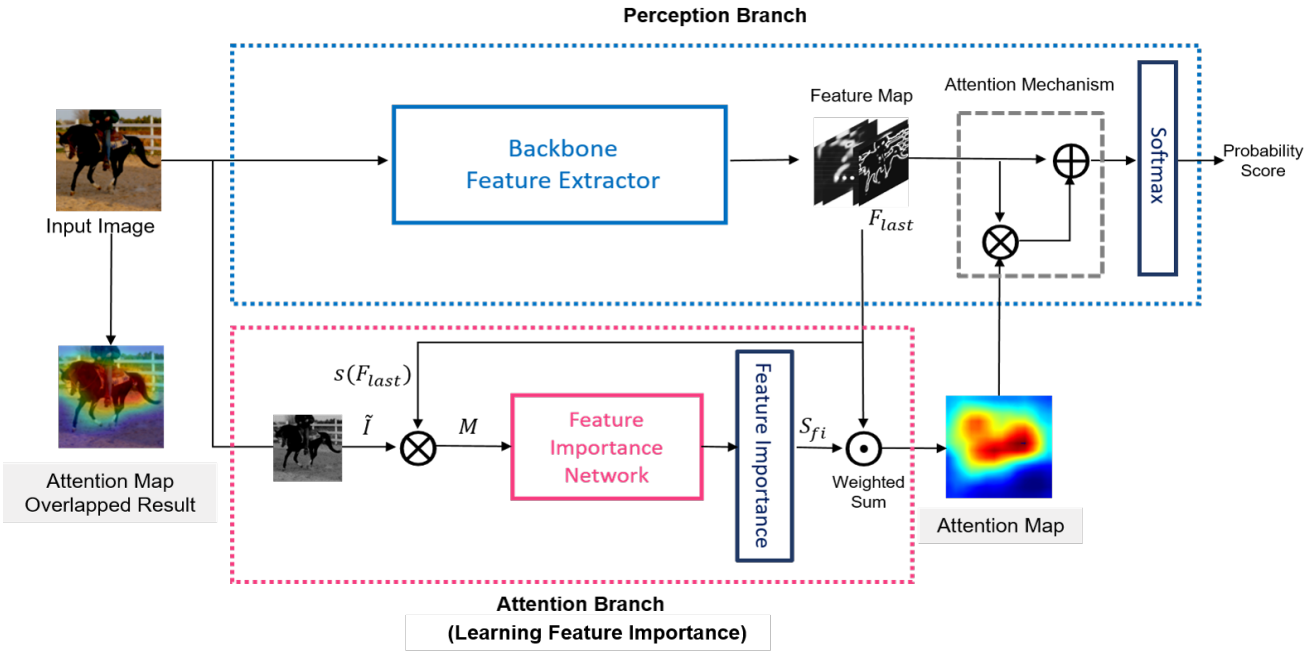

Figure 2. Overview of LFI-CAM. Feature Importance Network, Attention Branch and Learning Feature Importance are the same concept.

consistent attention maps through several experiments. We trained several ABN models with various hyper-parameters on the Cat&Dog dataset, and then compared CAMs of the same image from several models with similar accuracy. As shown in Fig. 1, CAM results for the exactly same test images are unreliable and inconsistent although the trained ABN models have similar accuracy.

We speculate this phenomenon is caused by two reasons:

(1) The K-channel feature map generated from the attention branch becomes very shallow if the dataset's number of categories K is small. The shallow feature map degrades attention map quality and makes attention map inconsistent.

(2) The attention branch of ABN aggregates the K-channel feature map to a one-channel feature map without considering the channel-wise feature importance. Although the attention map is trained by the attention mechanism, it is possible to generate various types of attention maps depending on the changes in the initial weight parameters or hyper-parameters due to high degree of freedom.

2.2. Score-CAM

Score-CAM [18] is based on CAM with a simple but efficient importance representation for each feature map. Unlike previous gradient-based visual explanation approaches such as Grad-CAM [15] and Grad-CAM++ [2], Score-CAM gets rid of the dependency on gradients by obtaining the weight of each feature map through its forward passing score on the target class. Ultimately, the final attention map is obtained as a weighted sum of feature maps. In order to obtain the class-discriminative attention map of Score-CAM, each feature map is first up-sampled to the original input size and normalized to range [0, 1]. To project highlighted areas in the feature map to the original input space, a masked image M^k is obtained by multiplying the normalized feature map A^k with the original input I.

$$M^k = A^k \otimes I \tag{1}$$

where $\otimes$ denotes element-wise multiplication and k denotes the k-th channel of the last convolution layer. For each masked image M^k, the output score S_k is obtained by the Softmax operation after forward computing $F(M^k)$.

$$S_k = Softmax(F(M^k)) \tag{2}$$

The score S_k^c on target class c represents the importance of the k-th feature map for target class c which is w_k^c.

$$w_k^c = S_k^c \tag{3}$$

The final class activation map is obtained by a linear weighted combination of all feature maps.

$$L_{Score-CAM}^c = ReLU(\sum_k w_k^c A^k) \tag{4}$$

Although Score-CAM achieves better visual performance with less noise and better stability than gradient-based approaches, multiple forward computing makes the generation of visual explanation very slow.

3. Proposed Method

In this section, we introduce LFI-CAM which is trainable for image classification and visual explanation in an end-to-end manner. LFI-CAM is composed of the attention branch and perception branch, as shown in Fig. 2. The perception branch extracts feature maps from the input image by passing it through multiple convolutional layers and predicts a class through the attention mechanism of the feature map from the convolutional layer and attention map. Meanwhile, the product between feature maps and down-sampled grayscaled input is fed into the attention branch, also denoted as the Feature Importance Network (FIN). FIN extracts feature importance of each feature map, and then the weighted sum between feature maps and extracted feature importance are calculated, generating the attention map. In this process, the attention branch not only generates attention map for visual explanation by learning feature importance but also leverages attention map to improve classification performance through the attention mechanism.

3.1. Motivation

We propose a novel architecture, LFI-CAM, which is inspired by ABN [4] and Score-CAM [18]. By leveraging the attention mechanism of ABN, LFI-CAM improves classification performance. However, the attention branch of ABN often generates unreliable and inconsistent visual explanation, due to ignoring the original CAM mechanism. To solve this issue, LFI-CAM's attention branch treats feature maps as masks and obtains feature importance scores for each feature map to generate the attention map in a similar manner to Score-CAM. In other words, as ABN learns the attention map itself without taking feature importance into consideration, we replaced ABN's attention branch with a new network architecture called 'Feature Importance Network (FIN)' which helps our model focus on learning the feature importance. Ultimately, LFI-CAM's attention map is generated by the weighted sum of feature maps from the last convolutional layer and the learned feature importance. Therefore, it generates more stable and reliable attention map. Since LFI-CAM's FIN in the attention branch learns the feature importance for each feature map during training unlike Score-CAM, our attention map is generated much faster than that of Score-CAM during forward propagation.

3.2. Feature Importance Network (Attention Branch)

In contrast to the previous method [4], which directly learns the class activation map in the attention branch, we replaced ABN's attention branch with a new network architecture, "Feature Importance Network (FIN)". FIN helps LFI-CAM learn the feature importance to generate better class activation map. Class activation map is generated by the weighted sum of the feature maps from the last convolutional layer and the learned feature importance vector.

To learn the feature importance, FIN follows a similar approach to Score-CAM [18]. However, unlike Score-CAM, we convert the original input into gray, which is downsampled to the feature map size. In addition, instead of conducting several forward computations, a concatenated masked image is fed as an input into the FIN. The feature importance for each feature map is outputted from the FIN.

In order to obtain the class activation map of LFI-CAM, the original input $I \in \mathbb{R}^{3\times w\times h}$ is first converted from RGB color space to a single gray scale space and down-sampled to a feature map with a size of $\mathbb{R}^{1\times m\times n}$. An example would be a conversion from $I \in \mathbb{R}^{3\times 224\times 224}$ to $\tilde{I} \in \mathbb{R}^{1\times 14\times 14}$ in Resnet18 [6] architecture.

$$\tilde{I} = Down(rgb2gray(I)) \tag{5}$$

Each feature map of the last convolutional layer, $F^k_{last} \in \mathbb{R}^{1\times m\times n}$ is normalized, where k denotes the channel index of the last convolutional layer. A masked image $M_k \in \mathbb{R}^{1\times m\times n}$ is obtained by multiplying the down-sampled gray input image $\tilde{I}$ with the normalized feature map.

$$M_k = \tilde{I} \otimes s(F^k_{last}) \tag{6}$$

where $s(\cdot)$ is a normalization function that maps each element in every feature map to range [0,1]. We generate a set of masked images $\{M_1, M_2, ..., M_N\}$ and concatenate them all, where N is the number of channels of the last convolutional layer of the model. Finally, we feed the concatenated masked image M into the FIN model $FIN(x)$ to conduct a forward propagation $FIN(M)$.

$$S_{fi} = FIN(M) \tag{7}$$

where $S_{fi} \in \mathbb{R}^N$ is the feature importance score vector. We take the k-th score S^k_{fi} as weight to represent the feature importance of the k-th feature map.

$$w_k = S^k_{fi} \tag{8}$$

The class activation map of the LFI-CAM is obtained by a weighted combination of all feature maps.

$$L_{LFI-CAM} = ReLU(\sum_{k=1}^{N} w_k F^k_{last}) \tag{9}$$

Similar to previous works [15, 18, 2], ReLU is applied to the linear combination of feature maps to remove features with negative influence.

3.3. Perception Branch

The perception branch takes the original input image as input and outputs the final probability of each class. The

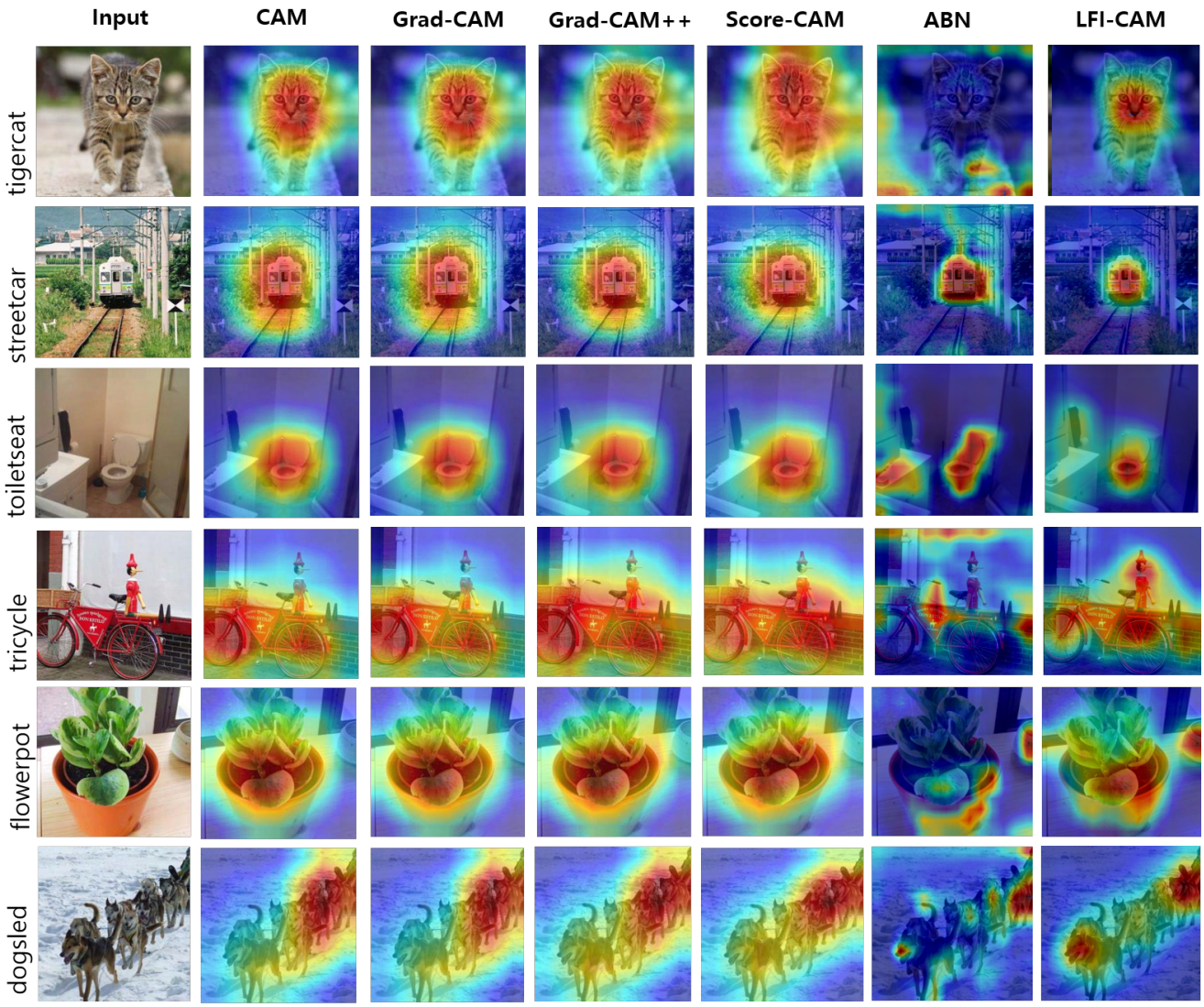

Figure 3. Visual explanation results of various methods on ImageNet. Notably, LFI-CAM always highlights the true class object correctly and in a more focused manner. For instance, LFI-CAM's tiger cat, streetcar, and toilet seat attention maps are tighter and focused on the salient features of the true class than any other methods. Additional results are provided in the supplementary material.

attention map generated from the attention branch (FIN) is overlapped onto the feature maps from intermediate convolutional layer by the attention mechanism. Unlike ABN, the LFI-CAM attention map is always generated using feature maps from the last convolutional layer, instead of the feature extractor. However, the attention mechanism can be applied to the feature map from any convolutional layer. We use the following attention mechanism formula from ABN [4].

$$F_l^{\acute{k}} = (1 + L_{LFI-CAM}) \otimes F_l^k \tag{10}$$

where F_l^k is the feature map at the l-th convolutional layer and $F_l^{\acute{k}}$ is the output of the attention mechanism. Note that k is the index of the channel and that $L_{LFI-CAM}$ is normalized to range [0,1] before being used in the attention mechanism. The attention mechanism helps the attention map improve the classification performance by highlighting the feature map at the location with a higher value of attention map while preventing the lower value region of the attention map from degrading to zero.

3.4. Training

LFI-CAM is trained in an end-to-end manner using training loss calculated as the combination of the Softmax function and cross-entropy at the perception branch in image classification task. The FIN is optimized by the attention mechanism of the perception branch to improve the classification accuracy without any additional loss function.

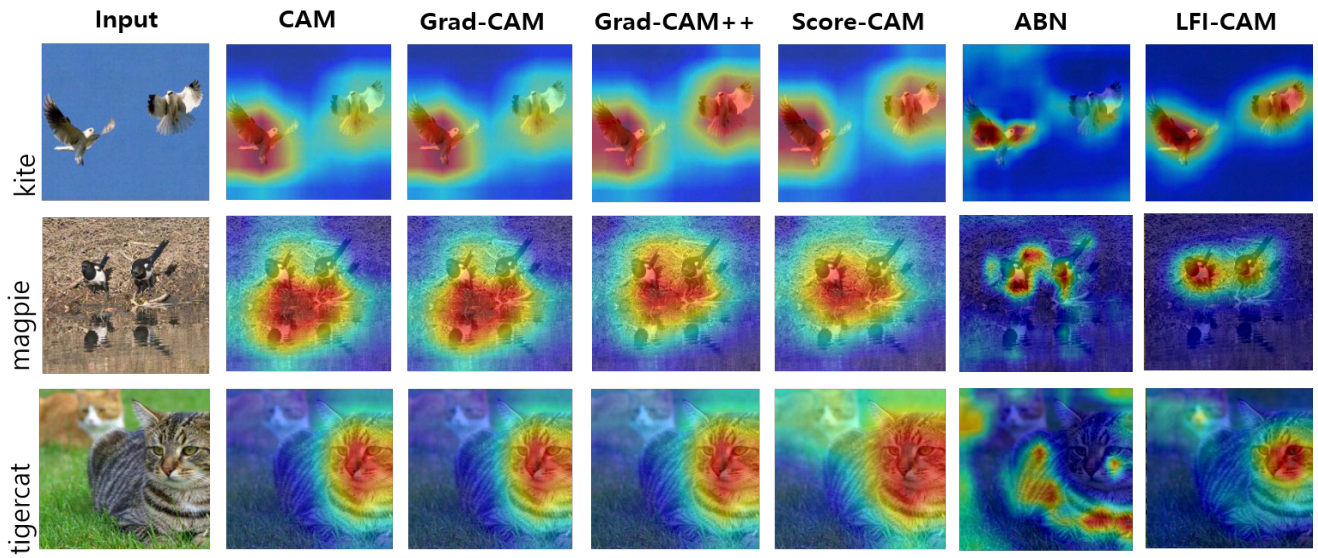

Figure 4. Visual explanation results of various methods for multi-target on ImageNet. More results are provided in supplementary material.

4. Experiments

In this section, we evaluate LFI-CAM's classification performance and show its effectiveness. First, we describe experiment settings on image classification in Sec 4.1. Second, we qualitatively evaluate our approach via visualization on ImageNet in Sec 4.2. In Sec 4.3, we quantitatively evaluate LFI-CAM's image classification performance by comparing it with various baseline models. Finally, we measure the stability of LFI-CAM's visual explanation and compare it with the stability of ABN's visual explanation.

4.1. Experimental Settings on Image Classification

Datasets: We evaluate LFI-CAM using 5 different public datasets- CIFAR10, CIFAR100, STL10, Cat&Dog (Kaggle Cats and Dogs), and ImageNet [3]. Cat&Dog dataset has 2 classes, CIFAR10 and STL10 have 10 classes each, CIFAR100 has 100 classes, and ImageNet has 1,000 classes. Training and testing dataset sizes are as follows: CIFAR10 and CIFAR100 has 50,000 training images and 10,000 testing images, and ImageNet consists of 1,281,167 training images and 50,000 testing images. STL10 consists of 5,000 training images and 8,000 testing images. The Cat&Dog dataset has 8,007 training images and 2,025 testing images. The input image size of CIFAR10 and CIFAR100 is 32 x 32 pixels, for STL10 it is 96 x 96, for Cat&Dog dataset and ImageNet it is 224 x 224 pixels.

Base Models: In this experiment, CIFAR10 and CIFAR100 are evaluated via the CIFAR ResNet backbone (ResNet 20, 32, 44, 56, 110). STL10, Cat&Dog, and ImageNet is evaluated via the ImageNet ResNet backbone (ResNet 18, 34, 50, 101, 152). The CIFAR ResNet backbone is more lightweight than the ImageNet ResNet backbone, with fewer layers and parameters, which is suitable for relatively smaller sized input image. The CIFAR ResNet backbone uses standard data augmentation of zero-padding images with 4 pixels on each side and then randomly cropping to produce 32 x 32 pixels images. Subsequently, horizontal flip is applied at random. For ImageNet ResNet backbone, training images are randomly resized and cropped to 224 x 224 pixels and then horizontally mirrored at random. The validation images are resized to 256 x 256 and then center cropped to produce 224 x 224 sized images. LFI-CAM models are composed of perception branch (backbone) and attention branch (FIN), where the FIN is constructed with multiple convolutional layers. Further details on LFI-CAM architecture can be found in the supplementary material.

Optimizer and Hyper-parameters: We use the most standard optimizer which is stochastic gradient descent (SGD) with momentum. We set the total epoch hyper-parameter as follows: CIFAR10, CIFAR100, STL10, and Cat&Dog are 300 epochs, and ImageNet is 90 epochs. The learning rate is initialized with 0.1, and later on divided by 10 at 50 % and 75 % of the total number of training epochs. We used training batch size of 128 for CIFAR ResNet backbone and 256 for ImageNet ResNet backbone.

4.2. Visual Explanation Evaluation

We qualitatively compare the visual explanation generated by 5 state-of-the-art methods, namely CAM [21], Grad-CAM [15], Grad-CAM++ [2], ScoreCAM [18] and ABN [4]. While CAM, Grad-CAM, Grad-CAM++ and

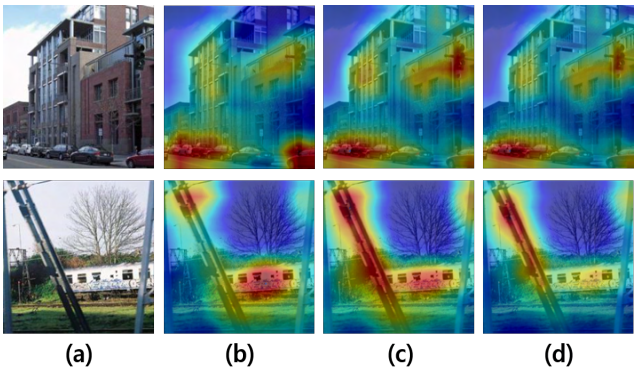

Figure 5. Visualization of the Feature Importance Network Effectiveness on ImageNet dataset. (a) Input image, (b) Pixel-wise mean feature map from the last convolutional layer of the LFI-CAM model trained without FIN, (c) Pixel-wise mean feature map from the last convolutional layer of the LFI-CAM model trained with FIN, (d) CAM generated from the LFI-CAM model.

Score-CAM can generate class activation map for each target class, ABN and LFI-CAM always generate a single class activation map for the class with the highest prediction probability. To make the comparison as fair as possible, we used the predicted output class of LFI-CAM as the target class for CAM, Grad-CAM, Grad-CAM++, and Score-CAM and selected examples with the same prediction result for both LFI-CAM and ABN. For CAM, Grad-CAM, Grad-CAM++, and ScoreCAM, we used ResNet18 model pretrained on ImageNet. For ABN and LFI-CAM, we used ResNet18 backbone trained on ImageNet.

4.2.1 Class Discriminative Visualization

As shown in Fig. 3, our method shows high-quality results beyond equivalence compared to CAM-variant methods, especially demonstrating less noisy and more focused results on target object area. Our approach can also generate more reliable visual explanation compared to ABN. More examples are provided in the supplementary material.

LFI-CAM shows better performance on locating multiple target objects than previous works as shown in Fig. 4. ABN often shows unreliable results where attention maps are generated improperly or in unrelated areas. Compared to other CAM-variants, LFI-CAM yields more focused and less noisy results as shown in single object experiments.

4.2.2 Effectiveness of Feature Importance Network

To evaluate the effectiveness of the proposed Feature Importance Network, we visualize the pixel-wise mean feature map from the last convolutional layer of the LFI-CAM model trained without and with the FIN. Then we compare them against the CAM generated from LFI-CAM model trained with FIN. Although we initially expected that the feature importance learned by the FIN plays an important role in generating reliable CAM, an interesting discovery is that the backbone network, FIN, and attention mechanism interact with each other during training. Therefore, we confirmed that LFI-CAM model is optimized not only by learning the feature importance but also by enhancing the backbone feature representation to focus more on important features to make decision for the input image. As shown in Fig. 5, after the FIN's feature importance is incorporated, our $L_{LFI-CAM}$ successfully focuses on the most distinguishable region of the target object. For example, as shown in the second row, the steel structure is highlighted

Table 1. Comparison of LFI-CAM and ABN's top-1 errors on CIFAR10 and CIFAR100 for ResNet110 and ResNeXt.

Model	CIFAR10	CIFAR100
ResNet110	6.43	24.14
ResNeXt [19]	3.84	18.32
ResNet110+ABN	**4.91**$_{(-1.52)}$	**22.82**$_{(-1.32)}$
ResNeXt+ABN	**3.8**$_{(-0.04)}$	**17.7**$_{(-0.62)}$
ResNet110+LFI-CAM	5.73$_{(-0.7)}$	23.33$_{(-0.81)}$
ResNeXt+LFI-CAM	4.27$_{(+0.43)}$	18.23$_{(-0.09)}$

Table 2. Comparison of LFI-CAM and ABN's top-1 errors on STL10 and Cat&Dog for ResNet18.

Model	STL10	Cat&Dog
ResNet18 +ABN	18.75	3.07
ResNet18+LFI-CAM	**18.16**$_{(-0.59)}$	**2.72**$_{(-0.35)}$

Table 3. Comparison of LFI-CAM and ABN' top-1 errors and model parameter size on ImageNet for ResNet18,34,50,101,152.

Model	Model Size	ImageNet
ResNet18	11.17M	30.24
ResNet34	21.28M	26.69
ResNet50	23.25M	23.87
ResNet101	42.51M	22.63
ResNet152	58.16M	21.69
ResNet18+ABN	21.61$M_{(+10.44)}$	28.98$_{(-1.26)}$
ResNet34+ABN	36.44$M_{(+15.16)}$	25.78$_{(-0.91)}$
ResNet50+ABN	43.58$M_{(+20.33)}$	23.1$_{(-0.77)}$
ResNet101+ABN	62.58$M_{(+20.07)}$	**21.8**$_{(-0.83)}$
ResNet152+ABN	78.22$M_{(+20.06)}$	**21.4**$_{(-0.29)}$
ResNet18+LFI-CAM	**17.47M**$_{(+6.3)}$	**27.75**$_{(-2.49)}$
ResNet34+LFI-CAM	**29.94M**$_{(+8.66)}$	**25.68**$_{(-1.01)}$
ResNet50+LFI-CAM	**43.05M**$_{(+19.8)}$	**22.71**$_{(-1.16)}$
ResNet101+LFI-CAM	**62.04M**$_{(+19.53)}$	21.84$_{(-0.79)}$
ResNet152+LFI-CAM	**77.68M**$_{(+19.52)}$	21.95$_{(+0.26)}$

prominently after applying the FIN because the LFI-CAM model classifies the input image as 'pole'.

4.3. Accuracy on Image Classification

For all tables, numbers in parentheses indicate difference in top-1 error and model parameter size from the baseline. Boldface indicates best performance among ABN and LFI.

Accuracy on CIFAR10 and CIFAR100: Table 1 shows top-1 errors on CIFAR10/100 using ResNet110, ResNeXt, ABN and LFI-CAM. Although LFI-CAM outperforms baseline models, ABN's top-1 errors tend to be slightly smaller than LFI-CAM's top-1 errors. However, we confirmed that LFI-CAM's CAM is much more reliable than ABN's CAM. The attention maps of ABN and LFI-CAM are provided in the supplementary material.

Accuracy on STL10 and Cat&Dog: We evaluate the image classification accuracy on STL10 and Cat&Dog as shown in Table 2 with the same method used for CIFAR10/100. We evaluate the top-1 errors for ResNet18 with ABN and LFI-CAM. On STL10, ResNet with LFI-CAM decreases the top-1 errors by 0.59 compared to ResNet18 with ABN. On Cat&Dog, LFI-CAM also decreases the top-1 errors by 0.35 compared to ResNet18 with ABN.

Accuracy on ImageNet: We evaluate image classification accuracy on ImageNet as shown in Table 3. We tested ResNet18, 34, 50, 101, 152 models with ABN and LFI-CAM. The table shows LFI-CAM is on par with ABN in terms of classification accuracy for each backbone model, even with less parameters. Also, LFI-CAM generates much more reliable CAM than ABN as shown in Sec 4.2.1.

4.4. Stability Evaluation of Visual Explanation

The stability of visual explanation is an important measure of CAM-related algorithm's performance and real world applicability. Researchers have observed instability of visual explanation from several previous works [15, 18], and one recent work, Attention Branch Network [4], shows significant instability for datasets with fewer number of classes, such as Cat&Dog, CIFAR10, STL10. Hence, we evaluated stability of visual explanation of LFI-CAM and other relevant models with those datasets. As shown in Fig. 1, we observed that LFI-CAM shows stable visual explanations unlike previous works such as ABN. To measure stability, we used IoU (Intersection of Union) between visual explanations on all the test data generated by 6 models, which were trained fully and individually on the same dataset using slightly different learning rate sampled from [0.07, 0.08, 0.09, 0.1, 0.11, 0.12]. First, we select one model with the highest classification accuracy from the 6 models as baseline. Then, we compare IoU between the visual explanation generated from the other 5 models with the baseline. Since the image area where the model gives more attention will have higher temperature, we used visual explanations with reasonably high temperature ($\geq$127, value range [0, 256]) for IoU calculation. As seen in Fig. 6, when comparing areas with high attention, LFI-CAM shows 60% or more overlap on average, but ABN shows 30%. The backbone used for stability evaluation was ResNet18 for Cat&Dog and STL10, and ResNet110 for Cifar10.

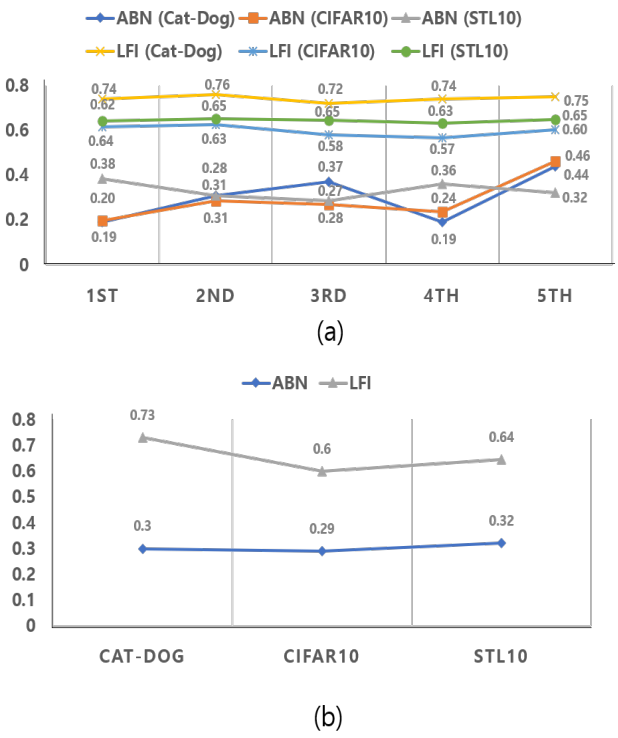

Figure 6. Stability evaluation of visual explanation. (a) IoU between models per dataset, (b) Average IoU per dataset.

5. Conclusion and Future Work

In this paper, we proposed LFI-CAM, which is trainable for image classification and produces better visual explanation in an end-to-end manner. We replaced ABN's attention branch with a new network architecture called "Feature Importance Network (FIN)" which helps our model focus on learning the feature importance to generate more stable and reliable attention map. In other words, LFI-CAM's attention map is generated by the weighted sum of the feature map from the last convolutional layer and the learned feature importance, while ABN learns the attention map itself without taking the feature importance into consideration. Throughout the paper, we evaluated the classification performance and visual explanation quality of LFI-CAM, and we concluded that LFI-CAM is on par with ABN in terms of classification accuracy and outmatches ABN in terms of attention map quality. Future work is planned to apply LFI-CAM's FIN to other tasks such as object detection, semantic segmentation and multi-task learning.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018. 1, 2, 3, 4, 6

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[4] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10705–10714, 2019. 1, 2, 4, 5, 6, 8

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[6] 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. 1, 2, 4

[7] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[8] 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. 1, 2

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[11] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018. 1

[12] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018. 1, 2

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 1

[14] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 1, 2

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 2, 3, 4, 6, 8

[16] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 1

[17] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015. 1

[18] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020. 1, 2, 3, 4, 6, 8

[19] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 2, 7

[20] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1

[21] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 1, 2, 6