Attention Branch Network: Learning of Attention Mechanism for Visual Explanation

Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi
Chubu University
1200 Matsumotocho, Kasugai, Aichi, Japan
{fhiro@mprg.cs, hirakawa@mprg.cs, yamashita@isc, fujiyoshi@isc}.chubu.ac.jp

Abstract

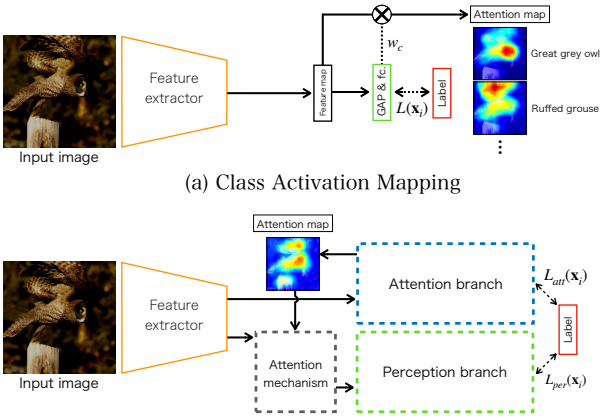
Visual explanation enables humans to understand the decision making of deep convolutional neural network (CNN), but it is insufficient to contribute to improving CNN performance. In this paper, we focus on the attention map for visual explanation, which represents a high response value as the attention location in image recognition. This attention region significantly improves the performance of CNN by introducing an attention mechanism that focuses on a specific region in an image. In this work, we propose Attention Branch Network (ABN), which extends a response-based visual explanation model by introducing a branch structure with an attention mechanism. ABN can be applicable to several image recognition tasks by introducing a branch for the attention mechanism and is trainable for visual explanation and image recognition in an end-to-end manner. We evaluate ABN on several image recognition tasks such as image classification, fine-grained recognition, and multiple facial attribute recognition. Experimental results indicate that ABN outperforms the baseline models on these image recognition tasks while generating an attention map for visual explanation. Our code is available [1].

1. Introduction

Deep convolutional neural network (CNN) [1, 17] models have been achieved the great performance on various image recognition tasks [25, 9, 7, 34, 8, 12, 18]. However, despite CNN models performing well on such tasks, it is difficult to interpret the decision making of CNN in the inference process. To understand the decision making of CNN, methods of interpreting CNN have been proposed [39, 41, 26, 4, 24, 3, 22].

"Visual explanation" has been used to interpret the decision making of CNN by highlighting the attention location in a top-down manner during the inference process. Visual explanation can be categorized into gradient-based or response-based. Gradient-based visual explanation typically use gradients with auxiliary data, such as noise [4] and class index [24, 3]. Although these methods can interpret the decision making of CNN without re-training and modifying the architecture, they require the backpropagation process to obtain gradients. In contrast, response-based visual explanation can interpret the decision making of CNN during the inference process. Class activation mapping (CAM) [41], which is a representative response-based visual explanation, can obtain an attention map in each category using the response of the convolution layer. CAM replaces the convolution and global average pooling (GAP) [20] and obtains an attention map that include high response value positions representing the class, as shown in Fig. 1(a). However, CAM requires replacing the fully-connected layer with a convolution layer and GAP, thus, decreasing the performance of CNN.

To avoid this problem, gradient-based methods are of-



(a) Class Activation Mapping

(b) Attention Branch Network

Figure 1. Network structures of class activation mapping and proposed attention branch network.

[1] https://github.com/machine-perception-robotics-group/attention_branch_network

ten used for interpreting the CNN. The highlight location in an attention map for visual explanation is considered an attention location in image recognition. To use response-based visual explanation that can visualize an attention map during a forward pass, we extended a response-based visual explanation model to an attention mechanism. By using the attention map for visual explanation as an attention mechanism, our network is trained while focusing on the important location in image recognition. The attention mechanism with a response-based visual explanation model can simultaneously interpret the decision making of CNN and improve their performance.

Inspired by response-based visual explanation and attention mechanisms, we propose $Attention\ Branch\ Network$ (ABN), which extends a response-based visual explanation model by introducing a branch structure with an attention mechanism, as shown in Fig 1(b). ABN consists of three components: feature extractor, attention branch, and perception branch. The feature extractor contains multiple convolution layers for extracting feature maps. The attention branch is designed to apply an attention mechanism by introducing a response-based visual explanation model. This component is important in ABN because it generates an attention map for the attention mechanism and visual explanation. The perception branch outputs the probabilities of class by using the feature and attention maps to the convolution layers. ABN has a simple structure and is trainable in an end-to-end manner using training losses at both branches. Moreover, by introducing the attention branch to various baseline model such as ResNet [9], ResNeXt [34], and multi-task learning [27], ABN can be applied to several networks and image recognition tasks.

Our contributions are as follows:

- ABN is designed to extend a response-based visual explanation model by introducing a branch structure with an attention mechanism. ABN is the first attempt to improve the performance of CNN by including a visual explanation.
- ABN is applicable to various baseline models such as VGGNet [14], ResNet [9], and multi-task learning [27] by dividing a baseline model and introducing an attention branch for generalizing an attention map.
- By extending the attention map for visual explanation to attention mechanism, ABN simultaneously improves the performance of CNN and visualizes an attention map during forward propagation.

2. Related work

2.1. Interpreting CNN

Several visual explanation, which highlight the attention location in the inference process, have been proposed [30, 39, 41, 26, 13, 4, 24, 3, 22]. There two types of visual explanation: gradient-based visual explanation, which uses a gradient and feed forward response to obtain an attention map, and response-based visual explanation, which only uses the response of a feed forward propagation. With gradient-based visual explanation, SmoothGrad [24] obtains sensitivity maps by adding noise to the input image iteratively and takes the average of these sensitivity maps. Guided backpropagation [13] and gradient-weighted class activation mapping (Grad-CAM) [4, 3], which are gradient-based visual explanation, have been proposed. Guided backpropagation and Grad-CAM visualize an attention map using positive gradients at a specific class in backpropagation. Grad-CAM and guided backpropagation have been widely used because they can interpret various pre-trained models using the attention map of a specific class.

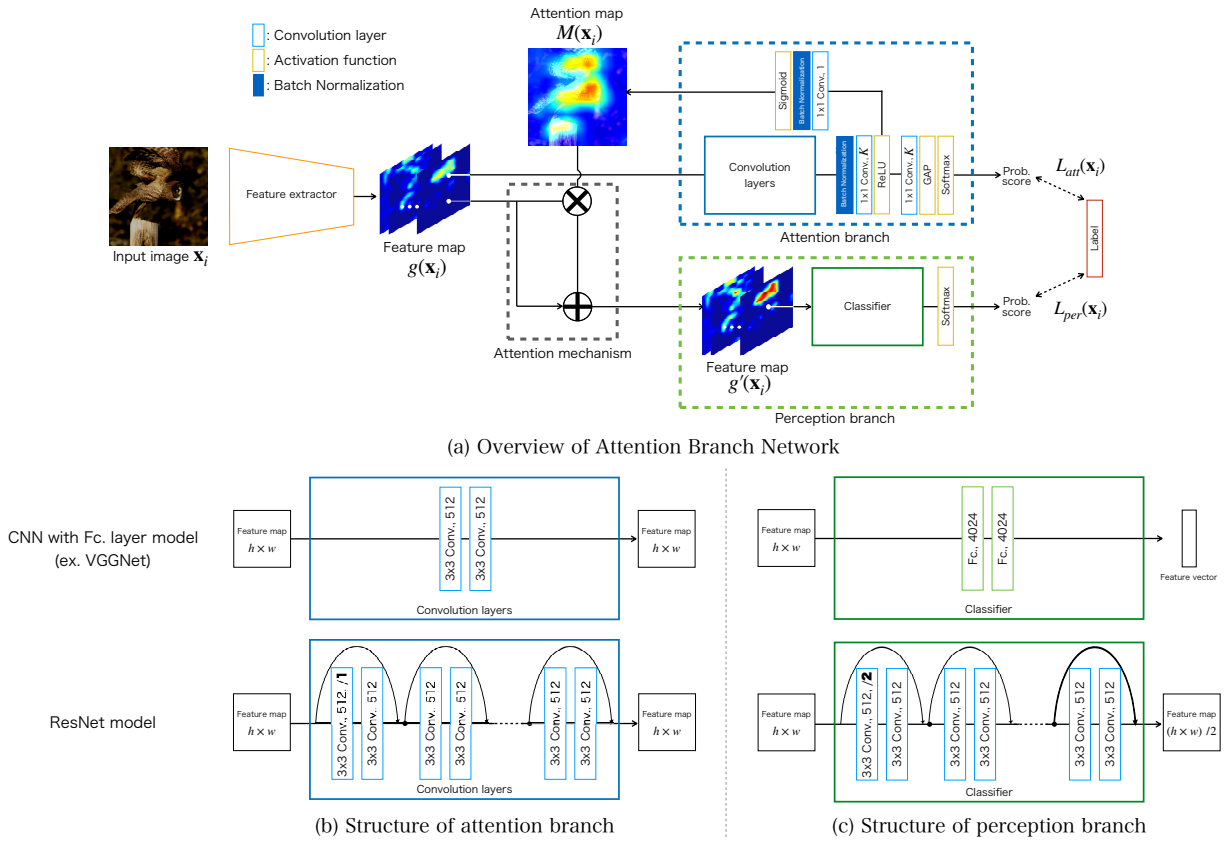
Response-based visual explanation visualizes an attention map using the feed forward response value from a convolution or deconvolution layer. While such models require re-training and modifying a network model, they can directly visualize an attention map during forward pass. CAM [41] can visualize an attention maps for each class using the response of a convolution layer and the weight at the last fully-connected layer. CAM performs well on weakly supervised object localization but not as well in image classification due to replacing fully-connected layers with convolution layers and passing through GAP.

We constract ABN by extending the CAM, which can visualize an attention map for visual explanation in feed forward propagation, to an attention mechanism. CAM is easily compatibles with an attention mechanism that directly weights the feature map. In contrast, gradient-based visual explanation is not compatible with ABN because it requires the back propagation process to obtain the gradients. Therefore, we use CAM as the attention mechanism for ABN.

2.2. Attention mechanism

Attention mechanisms have been used in computer vision and natural language processing [19, 15, 32, 12]. They have been widely used in sequential models [15, 36, 37, 2, 31] with recurrent neural networks and long short term memory (LSTM) [10]. A typical attention model on sequential data has been proposed by Xu $et\ al.$ [15]. The attention mechanism of their model is based on two types of attention mechanisms: soft and hard. The soft attention mechanism of Xu $et\ al.$ model is used as the gate of LSTM, and image captioning and visual question answering have been used [36, 37]. Additionally, the non-local neural network [33], which uses the self-attention approach, and the recurrent attention model [21], which controls the attention location by reinforcement learning, have been proposed.

The recent attention mechanism is also applied to single image recognition tasks [32, 12, 6]. Typical attention mod-



(a) Overview of Attention Branch Network

(b) Structure of attention branch

(c) Structure of perception branch

Figure 2. Detailed structure of Attention Branch Network.

els on a single image are residual attention network [32] and squeeze-and-excitation network (SENet) [12]. The residual attention network includes two attention components, i.e., a stacked network structure that consists of multiple attention components, and attention residual learning that applies residual learning [9] to an attention mechanism. SENet includes a squeeze-and-excitation block that contains a channel-wise attention mechanism introduced for each residual block.

ABN is designed to focus on the attention map for visual explanation that represents the important region in image recognition. Previous attention models extract a weight for an attention mechanism using only the response value of the convolution layers during feed forward propagation in an unsupervised learning manner. However, ABN easily extracts the effective weight for an attention mechanism in image recognition by generating the attention map for visual explanation on the basis of response-based visual explanation in a supervised learning manner.

3. Attention Branch Network

As mentioned above, ABN consists of three modules: feature extractor, attention branch, and perception branch, as shown in Fig. 1. The feature extractor contains multiple convolution layers and extracts feature maps from an input image. The attention branch outputs the attention location based on CAM to an attention map by using an attention mechanism. The perception branch outputs the probability of each class by receiving the feature map from the feature extractor and attention map.

ABN is based on a baseline model such as VGGNet [14] and ResNet [9]. The feature extractor and perception branch are constructed by dividing a baseline model between a specific layer. The attention branch is constructed after feature extractor on the basis of the CAM. ABN can be applied to several image recognition tasks by introducing the attention branch. We provide ABN for the several image recognition tasks such as image classification, fine-grained recognition, and multi-task learning.

3.1. Attention branch

CAM has a $K \times 3 \times 3$ convolution layer, GAP, and, fully-connected layer as last the three layers, as shown in Fig. 1(a). Here, K is the number of categories, and "$K \times 3 \times 3$ convolution layer" means a 3×3 kernel with K channels at the convolution layer. The $K \times 3 \times 3$ convolution

layer outputs a $K \times h \times w$ feature map, which represents the attention location for each class. The $K \times h \times w$ feature map is down-sampled to a 1×1 feature map by GAP and outputs the probability of each class by passing through the fully-connected layer with the softmax function. When CAM visualizes the attention map of each class, an attention map is generated by multiplying the weighted sum of the $K \times h \times w$ feature map by the weight at the last fully-connected layer.

CAM replaces fully-connected layers with 3×3 convolution layers. This restriction is also introduced into the attention branch. The fully-connected layer that connects a unit with all units at the next layer negates the ability to localize the attention area in the convolution layer. Therefore, if a baseline model contains a fully-connected layer, such as VGGNet, the attention branch replaces that fully-connected layer with a 3×3 convolution layer, similar with CAM, as shown at the top of Fig. 2(b) . ResNet models with ABN are constructed from the residual block at the attention branch, as shown at the bottom of Fig. 2(b). We set the stride of the first convolution layer at the residual block as 1 to maintain the resolution of the feature map.

To generate an attention map, the attention branch builds a top layer based on CAM, which consists of a convolution layer and GAP. However, CAM cannot generate an attention map in the training process because the attention map is generated using the feature map and weight at a fully-connected layer after training. To address this issue, we replace the fully-connected layer with a $K \times 1 \times 1$ convolution layer, as with CAM. This $K \times 1 \times 1$ convolution layer is imitated at the last fully-connected layer of CAM in a feed forward processing. After the $K \times 1 \times 1$ convolution layer, the attention branch outputs the class probability by using the response of GAP with the softmax function. Finally, the attention branch generates an attention map from the $K \times h \times w$ feature map. Then, to aggregate the K feature maps, these feature maps are convoluted by a $1 \times 1 \times 1$ convolution layer. By convoluting with a $1 \times 1 \times 1$ kernel, $1 \times h \times w$ feature map is generated. We use the $1 \times h \times w$ feature map normalized by the sigmoid function as the attention map for the attention mechanism.

3.2. Perception branch

The perception branch outputs the final probability of each class by receiving the attention and feature maps from the feature extractor. The structure of the perception branch is the same for conventional top layers from image classification models such as VGGNet and ResNet, as shown in Fig. 2(c). First, the attention map is applied to the feature map by the attention mechanism. We use one of two types of attention mechanisms, i.e., Eq. 1 and Eq. 2. Here, $g_c(\mathbf{x}_i)$ is the feature map at the feature extractor, $M(\mathbf{x}_i)$ is an attention map, and $g'_c(\mathbf{x}_i)$ is the output of the attention

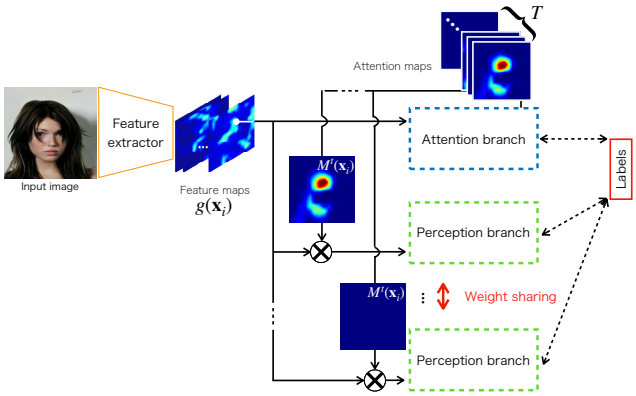


Figure 3. ABN for multi-task learning.

mechanism, as shown in Fig. 2(a). Note that $\{c|1, \ldots, C\}$ is the index of the channel.

$$g'_c(\mathbf{x}_i) = M(\mathbf{x}_i) \cdot g_c(\mathbf{x}_i) \quad (1)$$

$$g'_c(\mathbf{x}_i) = (1 + M(\mathbf{x}_i)) \cdot g_c(\mathbf{x}_i) \quad (2)$$

Equation 1 is simply a dot-product between the attention and feature maps at a specific channel c. In contrast, Eq. 2 can highlight the feature map at the peak of the attention map while preventing the lower value region of the attention map from degrading to zero.

3.3. Training

ABN can be trainable in an end-to-end manner using losses at both branches. Our training loss function $L(\mathbf{x}_i)$ is a simple sum of losses at both branches, as expressed by Eq. 3.

$$L(\mathbf{x}_i) = L_{att}(\mathbf{x}_i) + L_{per}(\mathbf{x}_i) \quad (3)$$

Here, $L_{att}(\mathbf{x}_i)$ denotes training loss at the attention branch with an input sample $\mathbf{x}_i$, and $L_{per}(\mathbf{x}_i)$ denotes training loss at the perception branch. Training loss for each branch is calculated by the combination of the softmax function and cross-entropy in image classification task. The feature extractor is optimized by passing through the gradients of the attention and perception branches during back propagation. If ABN is applied to other image recognition tasks, our training loss can adaptively change depending on the baseline model.

3.4. ABN for multi-task learning

ABN with a classification model outputs the attention map and final class probability by dividing the two branches. This network design can be applicable to other image recognition tasks, such as multi-task learning. In this section, we explain ABN for multi-task learning.

Conventional multi-task learning has units outputting the recognition scores corresponding to each task [27]. In training, the loss function defines multiple tasks using a single network. However, there is a problem with ABN for multi-task learning. In image classification, the relation between the numbers of inputs and recognition tasks is one-to-one. In contrast, the relation between the numbers of inputs and recognition tasks of multi-task learning is one-to-many. The one-to-one relation can be focused on the specific target location using a single attention map, but the one-to-many relation cannot be focused on multiple target locations using a single attention map. To address this issue, we generate multiple attention maps for each task by introducing multi-task learning to the attention and perception branches. Note that we use ResNet with multi-task learning as the baseline model.

To output multiple attention maps, we design the attention branch with multi-task learning, as shown in Fig. 3. First, a feature map at residual block 4 is convoluted by the $T\times1\times1$ convolution layer, and the $T\times14\times14$ feature map is output. The probability score during a specific task $\{t|1,\ldots,T\}$ is output by applying the 14×14 feature map at specific task t to GAP and the sigmoid function. In training, we calculated the training loss by combining the sigmoid function and binary cross-entropy loss function. We apply the 14×14 feature maps to the attention maps.

We introduce the perception branch to multi-task learning. Converting feature map $g_c'^t(\mathbf{x})$ is first generated using attention map $M^t(\mathbf{x})$ at specific task t and feature map $g(\mathbf{x})$ at the feature extractor, as shown in Eq. 4 in Sec. 3.2. After generating feature map $g_c'^t(\mathbf{x})$, the probability score at specific task t is calculated on perception branch $p_{per}(\cdot)$, which outputs the probability for each task by inputting feature map $g'^t(\mathbf{x})$.

$$g_c'^t(\mathbf{x}_i) = M^t(\mathbf{x}_i)\cdot g_c(\mathbf{x}_i) \quad (4)$$

$$\mathbf{O}(g_c'^t(\mathbf{x}_i)) = p_{per}(g_c'^t(\mathbf{x}_i)\,;\,\theta) \quad (5)$$

This probability matrix of each task $\mathbf{O}(g_c'^t(\mathbf{x}_i))$ on the perception branch consists of $T\times2$ components defined two categories classification for each task. The probability $\mathbf{O}^t(g_c'^t(\mathbf{x}_i))$ at specific task t is used when the perception branch receives the feature map $g_c'^t(\mathbf{x})$ that applies the attention map at specific task t, as shown in Fig. 3. These processes are repeated for each task.

4. Experiments

4.1. Experimental details on image classification

First, we evaluate ABN for an image classification task using the CIFAR10, CIFAR100, Street View Home Number (SVHN) [23], and ImageNet [5] datasets. The input image size of the CIFAR10, CIFAR100, SVHN datasets is 32×32 pixels, and that of ImageNet is 224×224 pixels. The number of categories for each dataset is as follows: CIFAR10 and SVHN consist of 10 classes, CIFAR100 consists of 100 classes, and ImageNet consists of 1,000 classes. During training, we applied the standard data augmentation. For CIFAR10, CIFAR100, and SVHN, the images are first zero-padded with 4 pixels for each side then randomly cropped to again produce 32×32 pixels images, and the images are then horizontally mirrored at random. For ImageNet, the images are resized to 256×256 pixels then randomly cropped to again produce 224×224 pixels images, and the images are then horizontally mirrored at random. The numbers of training, validation, and testing images of each dataset are as follows: CIFAR10 and CIFAR100 consist of 60,000 training images and 10,000 testing images, SVHN consists of 604,388 training images (train:73,257, extra:531,131) and 26,032 testing images, and ImageNet consists of 1,281,167 training images and 50,000 validation images.

We optimize the networks by stochastic gradient descent (SGD) with momentum. On CIFAR10 and CIFAR100, the total number of iterations to update the parameters is 300 epochs, and the batch size is 256. The total numbers of iterations to update the networks is as follows: CIFAR10 and CIFAR100 are 300 epochs, SVHN is 40 epochs, and ImageNet is 90 epochs. The initial learning rate is set to 0.1, and is divided by 10 at 50 % and 75 % of the total number of training epochs.

Table 1. Comparison of the top-1 errors on CIFAR100 with attention mechanism.

	$g(\mathbf{x})$	$g(\mathbf{x})\cdot M(\mathbf{x})$	$g(\mathbf{x})\cdot(1+M(\mathbf{x}))$
ResNet20	31.47	30.61	**30.46**
ResNet32	30.13	28.34	**27.91**
ResNet44	25.90	**24.83**	25.59
ResNet56	25.61	24.22	**24.07**
ResNet110	24.14	23.28	**22.82**

4.2. Image classification

Analysis on attention mechanism We compare the accuracies of attention mechanisms Eq. 1 and Eq. 2. We use ResNet {20, 33, 44, 56, 110} models on CIFAR100.

Table 1 shows the top-1 errors of attention mechanisms Eq. 1 and Eq. 2. The $g(\mathbf{x})$ is conventional ResNet. First, we compare ABN with $g(\mathbf{x})\cdot M(\mathbf{x})$ attention mechanism at Eq. 1 and conventional ResNet $g(\mathbf{x})$. Attention mechanism $g(\mathbf{x})\cdot M(\mathbf{x})$ has suppressed the top-1 errors than conventional ResNet. We also compare the accuracy of both $g(\mathbf{x})\cdot M(\mathbf{x})$ and $g(\mathbf{x})\cdot(1+M(\mathbf{x}))$ attention mechanisms. Attention mechanism $g(\mathbf{x})\cdot(1+M(\mathbf{x}))$ is slightly more accurate than attention mechanism $g(\mathbf{x})\cdot M(\mathbf{x})$. In residual attention network, which includes the same attention mechanisms, accuracy decreased with attention mechanism $g(\mathbf{x})\cdot M(\mathbf{x})$ [32]. Therefore, our attention map re-

Table 2. Comparison of top-1 errors on CIFAR10, CIFAR100, SVHN, and ImageNet dataset.

Dataset	CIFAR10	CIFAR100	SVHN [23]	ImageNet [5]
VGGNet [14]	–	–	–	31.2
VGGNet+BN	–	–	–	26.24^{*}
ResNet [9]	6.43	24.14^{*}	2.18^{*}	22.19^{*}
VGGNet+CAM [41]	–	–	–	33.4
VGGNet+BN+CAM	–	–	–	$27.42^{*}_{(+1.18)}$
ResNet+CAM	–	–	–	$22.11^{*}_{(-0.08)}$
WideResNet [38]	4.00	19.25	2.42^{*}	21.9
DenseNet [11]	4.51	22.27	2.07^{*}	22.2
ResNeXt [34]	3.84^{*}	18.32^{*}	2.16^{*}	22.4
Attention [32]	3.90	20.45	–	21.76
AttentionNeXt [32]	–	–	–	21.20
SENet [12]	–	–	–	21.57
VGGNet+BN+ABN	–	–	–	$25.55_{(-0.69)}$
ResNet+ABN	$4.91_{(-1.52)}$	$22.82_{(-1.32)}$	$\mathbf{1.86}_{(\mathbf{-0.32})}$	$21.37_{(-0.82)}$
WideResNet+ABN	$\mathbf{3.78}_{(\mathbf{-0.22})}$	$18.12_{(-1.13)}$	$2.24_{(-0.18)}$	–
DenseNet+ABN	$4.17_{(-0.34)}$	$21.63_{(-0.64)}$	$2.01_{(-0.06)}$	–
ResNeXt+ABN	$3.80_{(-0.04)}$	$\mathbf{17.70}_{(\mathbf{-0.62})}$	$2.01_{(-0.15)}$	–
SENet+ABN	–	–	–	$\mathbf{20.77}_{(\mathbf{-0.80})}$

$*$ *indicates results of re-implementation accuracy*

sponds to the effective region in image classification. We use attention mechanism $g(\mathbf{x}) \cdot (1 + M(\mathbf{x}))$ at Eq. 2 as default manner.

Accuracy on CIFAR and SVHN Table 2 shows the top-1 errors on CIFAR10/100, SVHN, and ImageNet. We evaluate these top-1 errors using various baseline models, CAM, and ABN regarding image classification. These errors are an original top-1 error at referring paper [14, 9, 41, 38, 11, 34, 32, 32, 12] or top-1 errors of our model, and the '$*$' indicates the results of re-implementation accuracy. The numbers in brackets denote the difference in the top-1 errors from the conventional models at re-implementation. On CIFAR and SVHN, we evaluate the top-1 errors by using the following ResNet models as follows: ResNet (depth=110), DenseNet (depth=100, growth rate=12), Wide ResNet (depth=28, widen factor=4, drop ratio=0.3), ResNeXt (depth=28, cardinality=8, widen factor=4). Note that ABN is constructed by dividing a ResNet model at residual block 3.

Accuracies of ResNet, Wide ResNet, DenseNet and ResNeXt are improved by introducing ABN. On CIFAR10, ResNet and DenseNet with ABNs decrease the top-1 errors from 6.43 % to 4.91 % and 4.51 % to 4.17 %, respectively. Additionally, all ResNet models are decrease the top-1 errors by more 0.6 % on CIFAR100.

Accuracy on ImageNet We evaluate the image classification accuracy on ImageNet as shown in Table 2 in the same manner as for CIFAR10/100 and SVHN. On ImageNet, we evaluate the top-1 errors by using the VGGNet (depth=16), ResNet (depth=152), and SENet (ResNet152 model). First, we compare the top-1 errors of CAM. The performance of CAM slightly decreased with a specific baseline model because of the removal of the fully-connected layers and adding a GAP [41]. Similarly, the performance on VGGNet+BatchNormalization (BN) [29] with CAM decreases even in re-implementation. In contrast, the performance of ResNet with CAM is almost the same as that of baseline ResNet. The structure of the ResNet model that contains GAP and a fully-connected layer as the last layer resembles that in CAM. ResNet with CAM can be easily constructed by stacking on the $K \times 1 \times 1$ convolution layer at the last residual block, which sets the stride to 1 at the first convolution layer. Therefore, it is difficult to decrease the performance of ResNet with CAM due to removal of the fully-connected layer and adding GAP. On the other hand, ABN outperforms conventional VGGNet and CAM and performs better than conventional ResNet and CAM.

We compare the accuracy of a conventional attention models. By introducing the SE modules to ResNet152, SENet reduces the top-1 errors from 22.19% to 21.90%. However, ABN reduces the top-1 errors from 22.19 % to 21.37%, indicating that ABN is more accurate than SENet. Moreover, ABN can introduce the SENet in parallel. SENet with ABN reduces the top-1 errors from 22.19 % to 20.77 % compared to the ResNet152. Residual attention network results in the same amount of top-1 errors from the size of the input image, which is 224×224, as fol-

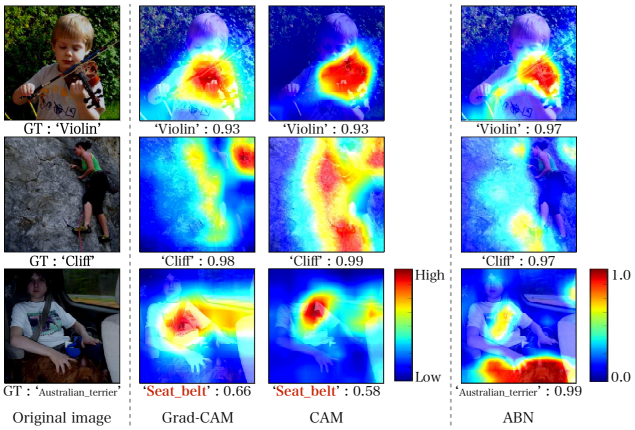


Figure 4. Visualizing high attention area with CAM, Grad-CAM, and our ABN. CAM and Grad-CAM are visualized attention maps at top-1 result.

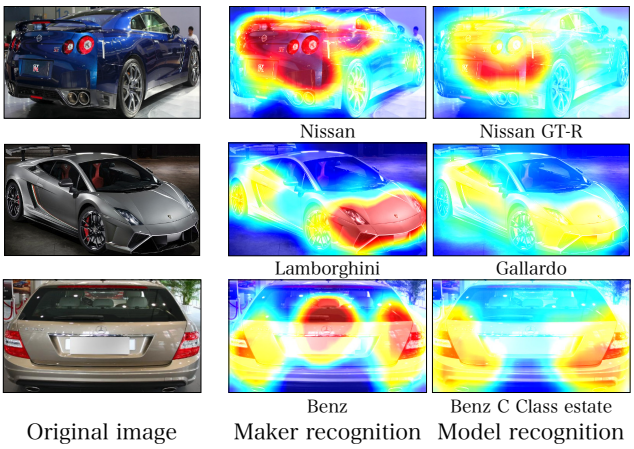


Figure 5. Visualizing attention map on fine-grained recognition.

Table 3. Comparison of car model and maker accuracy on CompCars dataset

task	model [%]	maker [%]
VGG16	85.9	90.4
ResNet101	90.2	90.1
VGG16+ABN	90.7	92.9
ResNet101+ABN	**97.1**	**98.1**

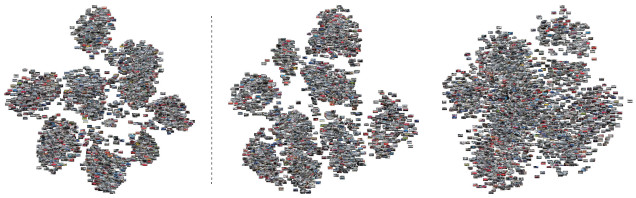


Figure 6. Comparison of distribution maps at residual block 4 by t-SNE. **Left :** distribution of baseline ResNet101 model. **Center and Right :** distribution of ABN. **Center** did not apply the attention map.

lows: ResNet is 21.76%, and ResNeXt is 21.20%. Therefore, ResNet152+SENet with ABN indicates more accurate than these residual attention network models.

Visualizing attention maps We compare the attention maps visualized using Grad-CAM, CAM, and ABN. Grad-CAM generates an attention map by using ResNet152 as a baseline model. CAM and ABN are constructed using ResNet152 as a baseline model. Figure. 4 shows the attention maps for each model on ImageNet dataset.

This Fig. 4 shows that Grad-CAM, CAM and ABN highlights a similar region. For example in the first column in Fig. 4, these models classify the "Violin", and highlight the "Violin" region on the original image. Similarly, they classify "Cliff" in the second column and highlight the "Cliff" region. For the third column, this original image is a typical example because multiple objects, such as "Seat belt" and "Australian terrier", are included. In this case, Grad-CAM (conventional ResNet152) and CAM failes, but ABN performs well. When visualizing the attention maps in the third column, the attention map of ABN highlights each object. Therefore, this attention map can focus on a specific region when multiple objects are in an image.

4.3. Fine-grained recognition

We evaluate the performance of ABN for the fine-grained recognition on the comprehensive cars (CompCars) dataset [35], which has 36,451 training images and 15,626 testing images with 432 car models and 75 makers. We use VGG16 and ResNet101 as baseline model and optimized these models by SGD with momentum. The total number of update iterations is 50 epochs, and the mini-batch size is 32. The learning rate starts from 0.01 and is multiplied by 0.1 at 25 and 35 epochs. The input image is resized to 323×224 pixels. The image size is calculated by taking the average of the bounding box aspect ration from the training data. This resizing process prevents the collapse of the car shape.

Table 3 shows the car model and maker recognition accuracy on the CompCars dataset. The car model recognition accuracy of ABN improves by 4.9 and 6.2 % with VGG16 and ResNet101, respectively. Moreover, maker recognition accuracy improves by 2.0 and 7.5 %, respectively. These results indicate that ABN is also effective for fine-grained recognition. We visualize the attention maps for car model or maker recognition, as shown in Fig. 5. From these visualizing results, training and testing images are the same for

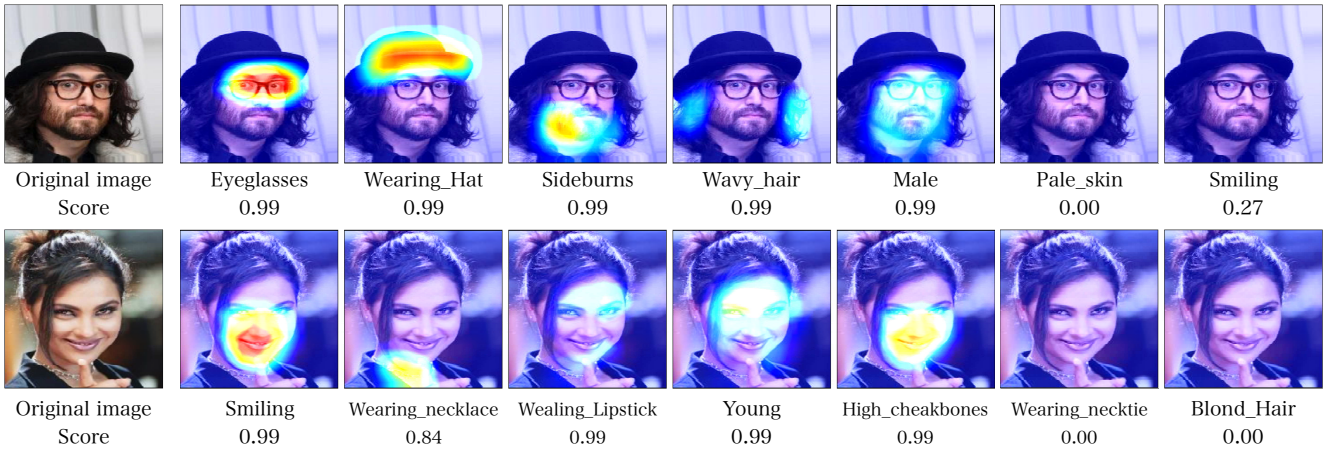


Figure 7. Visualizing attention maps on multiple facial attributes recognition. These scores are final recognition scores at the perception branch.

Table 4. Comparison of multiple facial attribute recognition accuracy on CelebA dataset

Method	Average of accuracy [%]	Odds
FaceTracer [16]	81.13	40/40
PANDA-l [40]	85.43	39/40
LNet+ANet [42]	87.30	37/40
MOON [28]	90.93	29/40
ResNet101	90.69	27/40
ABN	**91.07**	–

car model and maker recognition, however, our attention maps differ depending on the recognition task.

We compare the feature representations of the ResNet101 and ResNet101 with ABN. We visualize distributions by t-distributed stochastic neighbor embedding (t-SNE) [30] and analyze the distributions. We use the comparison feature maps at the final layer on residual block 4. Figure 6 shows the distribution maps of t-SNE. We use 5,000 testing images on the CompCars dataset. The feature maps of ResNet101 and the feature extractor in the attention branch network are clustered by car pose. However, the feature map applying the attention map is split distribution by car pose and detail car form.

4.4. Multi-task Learning

For multi-task learning, we evaluate for multiple facial attributes recognition using the CelebA dataset [42], which consists of 202,599 images (182,637 training images and 19,962 testing images) with 40 facial attribute labels. The total number of iterations to update the parameters is 10 epochs, and the learning rate is set to 0.01. We evaluate the accuracy rate using FaceTracer [16], PANDA-l [40], LNet+ANet [42], mixed objective optimization network (MOON) [28], and ResNet101.

Table 4 shows that ABN outperforms all conventional methods regarding the average recognition rate and number of facial attribute tasks. Note that the numbers in the third column in Table 4 are the numbers of winning tasks when we compare the conventional models with ABN for each facial attribute. The accuracy of a specific facial attribute task is described in the appendix. When we compare ResNet101 and ABN, ABN is 0.38% more accurate. Moreover, the accuracy of 27 facial tasks is improved. ABN also performs better than conventional facial attribute recognition models, i.e., FaceTracer, PANDA-l, LNet+ANet, MOON. ABN outperforms these models for difficult tasks such as "arched eyebrows", "pointy nose", "wearing earring", and "wearing necklace". Figure 7 shows the attention map of ABN on CelebA dataset. These attention maps highlights the specific locations such as mouth, eyes, beard, and hair. These highlight locations correspond to the specific facial task, as shown in Fig. 7. It is conceivable that these highlight locations contributed to performance improvement of ABN.

5. Conclusion

We propose an Attention Branch Network, which extends a response-based visual explanation model by introducing a branch structure with an attention mechanism. ABN can be simultaneously trainable for visual explanation and improving the performance of image recognition with an attention mechanism in an end-to-end manner. It is also applicable to several CNN models and image recognition tasks. We evaluated the accuracy of ABN for image classification, fine-grained recognition, and multi-task learning, and it outperforms conventional models for these tasks. We plan to apply ABN to reinforcement learning that does not include labels in the training process.

References

[1] K. Alex, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, pages 1097–1105. 2012.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2016.

[3] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *arXiv preprint arXiv:1710.11063*, 2017.

[4] S. Daniel, T. Nikhil, K. Been, B. V. Fernanda, and W. Martin. Smoothgrad: removing noise by adding noise, 2017.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, 2009.

[6] L. Drew, S. Dan, E. Sven, and S. Thomas. Global-and-local attention networks for visual recognition. *arXiv*, abs/1805.08819, 2018.

[7] H. Emily M. and C. Rama. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *Association for the Advancement of Artificial Intelligence*, 2017.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *International Conference on Computer Vision*, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] H. Jie, S. Li, and S. Gang. Squeeze-and-excitation networks. *Computer Vision and Pattern Recognition*, 2017.

[13] S. Jost, Tobias, D. Alexey, B. Thomas, and R. Martin. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations*. 2015.

[14] S. Karen and Z. Andrew. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[15] X. Kelvin, B. Jimmy, K. Ryan, C. Kyunghyun, C. Aaron, S. Ruslan, Z. Rich, and B. Yoshua. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[16] N. Kumar, P. N. Belhumeur, and S. K. Nayar. Facetracer: A search engine for large collections of images with faces, October 2008.

[17] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[18] C. Liang-Chieh, Z. Yukun, P. George, S. Florian, and A. Hartwig. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 2018.

[19] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

[20] L. Min, C. Qiang, and Y. Shuicheng. Network in network. *International Conference on Learning Representations*, 2014.

[21] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu. Recurrent models of visual attention. In *Neural Information Processing Systems*, pages 2204–2212. 2014.

[22] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[23] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems*, 2011.

[24] S. Ramprasaath, R., C. Michael, D. Abhishek, V. Ramakrishna, P. Devi, and B. Dhruv. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision*, pages 618–626, 2017.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems*, pages 91–99. 2015.

[26] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[27] C. Richard. Multitask learning: A knowledge-based source of inductive bias. In *International Conference on Machine Learning*, pages 41–48, 1993.

[28] E. Rudd, M. Gunther, and T. Boult. Moon: A mixed objective optimization network for the recognition of facial attributes. In *European Conference on Computer Vision*. 2016.

[29] I. Sergey and S. Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[30] M. L. Van, Der and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, A. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, pages 5998–6008. 2017.

[32] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *Computer Vision and Pattern Recognition*, 2017.

[33] W. Xiaolong, G. Ross, G. Abhinav, and H. Kaiming. Non-local neural networks. *Computer Vision and Pattern Recognition*, 2018.

[34] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5995, 2017.

[35] L. Yang, P. Luo, C. C. Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *Computer Vision and Pattern Recognition*, pages 3973–3981, 2015.

[36] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *Computer Vision and Pattern Recognition*, pages 21–29, 2016.

[37] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. *Computer Vision and Pattern Recognition*, pages 4651–4659, 2016.

[38] S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

[39] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014.

[40] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1644, 2014.

[41] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *Computer Vision and Pattern Recognition*, 2016.

[42] L. Ziwei, L. Ping, W. Xiaogang, and T. Xiaoou. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, 2015.