IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 23, 2020, accepted August 31, 2020, date of publication September 4, 2020, date of current version September 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021729

Enhanced Visual Attention-Guided Deep Neural Networks for Image Classification

CHIA-HUNG YEH[1,2], (Senior Member, IEEE), MIN-HUI LIN[1], PO-CHAO CHANG[1], AND LI-WEI KANG[2], (Member, IEEE)

[1]Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan
[2]Department of Electrical Engineering, National Taiwan Normal University, Taipei 106, Taiwan

Corresponding author: Li-Wei Kang (lwkang@ntnu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant NSC 102-2221-E-110-032-MY3, Grant MOST 103-2221-E-110-045-MY3, Grant MOST 103-2221-E-003-034-MY3, Grant MOST 105-2221-E-003-030-MY3, Grant MOST 108-2218-E-110-002-, Grant MOST 108-2218-E-003-002-, Grant MOST 109-2218-E-110-007-, Grant MOST 109-2218-E-003-002-, and Grant MOST 108-2221-E-003-027-MY3.

ABSTRACT A fully connected layer is essential for a CNN, i.e., convolutional neural network, which has been shown to be successful in classifying images in several related applications. A CNN begins with convolution and pooling operations for decomposing an input image into features. The result of this process is then fed into a fully connected neural network, driving the final classification decision for the input image. However, it has been found that the learned feature maps in a CNN are sometimes not good enough for being fed into the fully connected layers to get good classification results. In this article, a visual attention learning module is proposed to enhance the classification capability of the fully connected layers in a CNN. By learning better feature maps to emphasize salient regions and weaken meaningless regions, better classification performance can be obtained with integrating the proposed module into the fully connected layers. The proposed visual attention learning module can be imposed on any existed CNN-based image classification models to achieve incremental improvements with negligible overhead. Based on our experiments, the proposed method achieves the top-1 accuracies of 95.32%, 92.73%, and 66.50% on average, respectively, obtained on our collected Underwater Fish dataset, the public Animals-10 dataset, and the public Stanford Cars dataset.

INDEX TERMS Salient feature learning, deep learning, convolutional neural networks, image classification, object recognition.

I. INTRODUCTION

Image classification or visual object recognition is a fundamental problem [1]–[3] in several computer vision-based applications, such as fish species recognition for underwater exploration [4] and visual understanding-based autonomous driving [5]. Conventional image classification approaches usually apply the extraction of handcrafted features, e.g., [6], to analyze images. For example, a novel image retrieval framework was presented in [7] to retrieve digital images from huge databases based on texture analysis techniques for extracting discriminant features, including color and shape features. However, in recent years, based on the rapid development of deep learning techniques [8] with great success in numerous perceptual tasks, e.g., image classification [9]–[14] and image restoration [15]–[18], several CNN-based deep neural networks were presented for image classification. For example, a deep CNN, called AlexNet [9], was presented to perform image classification for the ImageNet dataset of 1.2 million high-resolution images into the 1000 different classes. In addition, a very deep CNN, called VGGNet [10], was proposed for large-scale image recognition. Its main contribution is to evaluate the network performance with increasing depth using an architecture with very small convolution filters. Moreover, a deep CNN architecture, called Inception or GoogleNet [11], was also presented for large scale visual recognition. The key is to allow for increasing the depth and width of the network while keeping the computational budget constant. Furthermore, a residual learning framework, call ResNet [12], was proposed to ease the

The associate editor coordinating the review of this manuscript and approving it for publication was Chang-Hwan Son.

training of deeper networks for image recognition. It explicitly reformulates the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. Moreover, an inverted residual network structure, called MobileNetV2 [13], was presented to improve the state of the art performance of mobile models on multiple tasks, including ImageNet classification.

On the other hand, to strengthen the representational power of a CNN, several approaches were presented recently by enhancing the quality of spatial encodings and/or recalibrating channel-wise feature responses. For example, an architectural unit, termed squeeze-and-excitation (SE) block [19], was proposed to model interdependencies between channels. The building blocks can be stacked and easily embedded into any CNN architectures, e.g., by insertion after the non-linearity operation following each convolution, for performance improvement. Furthermore, a convolutional block attention module (CBAM) [20] was presented by sequentially inferring attention maps along two separate dimensions, i.e., channel and spatial. CBAM can be also integrated into any CNN architectures for achieving better performance. For other recently developed deep attention models, a non-local neural model inspired by the classical non-local means method was presented in [21] for capturing long-range dependencies, e.g., successive video frames. In addition, a residual attention network built by stacking attention modules which can generate attention-aware features was proposed in [22]. Moreover, an efficient channel attention (ECA) module for deep CNNs was presented in [23], which captures cross-channel interaction in an efficient way.

For advanced applications of visual attention modules, a deep model was presented in [24] to consider the leaf spot attention mechanism. In addition, a deep architecture denoted by region-of-interest-aware deep CNN was proposed in [25] for making deep features more discriminative to increase classification performance.

In addition, by visualizing the process of a CNN [26], it has been shown that the final convolutional layer of a CNN usually dominates the decision resulted by the CNN [27]. That is, the last convolutional layer can produce a feature map or a coarse localization map highlighting the important spatial regions in the input image for predicting the concept. Therefore, in this article, we propose to only embed an enhanced visual attention layer after the final convolutional layer of a CNN for learning better feature maps to compromise between high-level semantics and detailed spatial information.

The main features and contributions of this article are three-fold: (i) the proposed enhanced visual attention module directly improves the last fully connected layer by enhancing the learned last feature maps of a CNN for better classification capability without needing extra convolutional layers; (ii) the proposed method uses the Huber loss function [28], [29] to guide the last learned feature map toward the corresponding ground truth, instead of using the MSE loss to avoid the effects from possible outlier samples; and (iii) the proposed module just needs to be embedded into a CNN only once and can fit any CNN architecture for image classification purpose with almost negligible extra overhead.

The rest of this article is organized as follows. Sec. II presents the proposed enhanced visual attention-guided deep neural networks for image classification. Experimental results are demonstrated in Sec. III, followed by concluding this article in Sec. IV.

II. PROPOSED ENHANCED VISUAL ATTENTION-GUIDED DEEP NEURAL NETWORKS FOR IMAGE CLASSIFICATION

A. OVERVIEW OF THE PROPOSED ENHANCED VISUAL ATTENTION MODULE

To strengthen the salient region feature maps and suppress the insignificant feature maps for an input image for image classification, we propose to separate the learned feature maps into one channel or map and the rest channels in the last convolutional layer. As illustrated in Fig. 1, through the channel separation process, the learned C feature maps of a CNN are split into the feature map of the C-th channel and the rest (C – 1) feature maps from the first to the (C – 1)-th channels. The selected map from the C-th channel is then fed into the leaned enhanced visual attention module (described later) for refining the feature map. The output enhanced feature map from the proposed enhanced visual attention module can be viewed as a weighting coefficient map used for refining all of the rest (C – 1) feature maps. The weighting coefficient map is then used to enhance each of the rest (C – 1) feature maps based on the element-wise multiplication. The refined feature maps can better capture the salient region or the main object region of the input image. The feature maps are then fed into the final fully connected layer of the CNN for generating the image classification result.

More specifically, for each c-th, $c = 1, 2, \ldots, C$, learned feature map $X_c \in \mathbb{R}^{H\times W}$ from the last convolutional layer in a CNN, the weighting coefficient map $\omega \in \mathbb{R}^{H\times W}$ learned by the proposed enhanced visual attention module from enhancing the last channel X_c will be used to refine each X_c, $c = 1, 2, \ldots, C-1$. The refined version F_c is expressed as:

$$F_c = X_c \odot \text{abs}(\omega), \tag{1}$$

where $F_c \in \mathbb{R}^{H\times W}$ is the refined version of X_c, and H and W denote the height and the width of the feature map, respectively. The function abs( ) is used to set each coefficient of ω to its absolute value. The operation "$\odot$" means the element-wise multiplication. In our method, ω is used to highlight significant region in the input image and suppress insignificant information for image classification, as illustrated in Fig. 2.

B. TRAINING CONVOLUTIONAL NEURAL NETWORKS WITH THE PROPOSED ENHANCED VISUAL ATTENTION MODULE

The motivation for enhancing the last learned feature maps of a CNN in this article is mainly inspired by the fact that the final convolutional layer (immediately before the final fully connected layer) of a CNN usually captures higher sematic

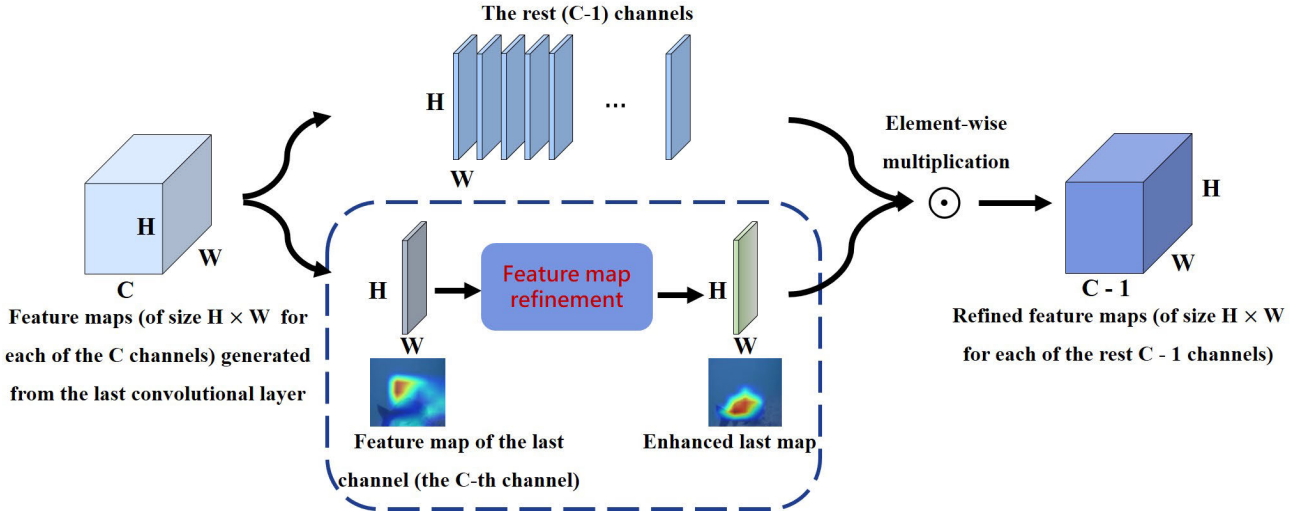

FIGURE 1. Illustration of the proposed enhanced visual attention module.

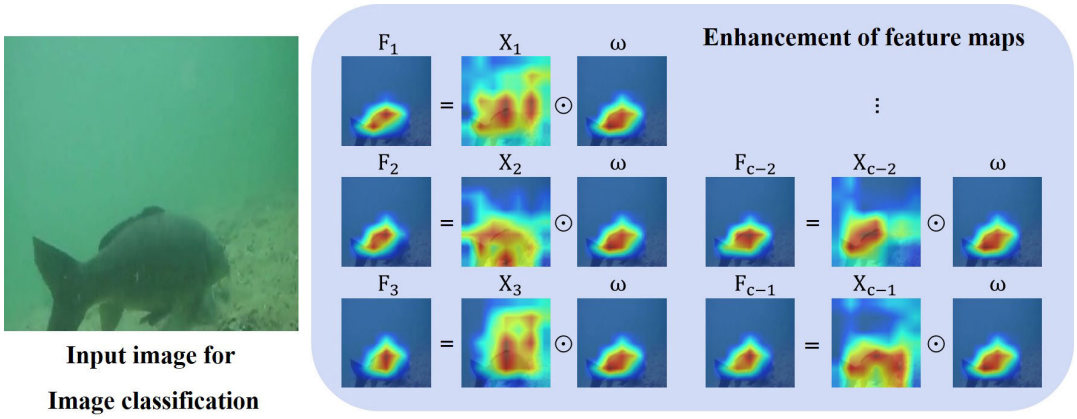

FIGURE 2. Illustration of the refinement for the feature maps based on the weighting coefficient map learned by the proposed enhanced visual attention module. The term X_c, c = 1, 2, ..., C-1, denotes the original learned feature map and ω denotes the weighting coefficient map (refined version of X_c, i.e., the C-th channel) learned by the proposed enhanced visual attention module, which is used to refine each X_c, c = 1, 2, ..., C-1, where $F_c = X_c \odot \text{abs}(\omega)$ and the abs() operation is omitted in this.

features for final decision output of the CNN. Moreover, based on the visualization of a CNN for classification purpose [27], the output can be semantically visualized by the weighted combination of the feature maps learned by the last convolutional layer, as illustrated in Fig. 3. Therefore, it is reasonable to refine the representational power of the feature maps learned from the last convolutional layer of a CNN for capturing richer semantic information and obtaining the better prediction result.

To realize this idea, this article proposes to embed an additional layer, called the enhanced visual attention module, into any existed CNN. This module will immediately follow the last convolutional layer of the CNN called the host CNN and enhances the feature maps generated from the last convolutional layer. To train the host CNN with the proposed enhanced visual attention module embedded, as shown in Fig. 4, we first simply split the C learned feature maps from the last convolutional layer of the host CNN into the C-th feature map and the rest (C – 1) feature maps. Our main goal is to refine the C-th feature map to enrich the significant information for image classification and suppress the insignificant information. Therefore, we calculate the feature loss between the C-th feature map and the corresponding ground truth (described later). On the other hand, the rest (C – 1) feature maps are also connected to the original fully connected layer called the 1st fully connected layer of the host CNN. Moreover, to guide the refined feature maps toward the correct prediction output, all of the refined feature maps are also connected to a fully connected layer called the 2nd fully connected layer, exactly the same as the 1st one.

To guide the selected feature map from the C-th channel to the corresponding ground truth map, the Huber loss [28], [29]

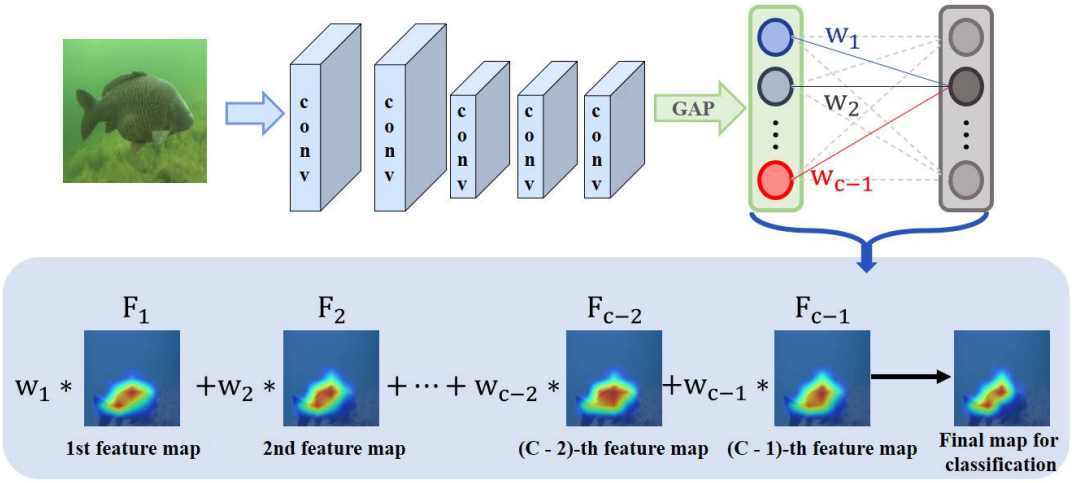

FIGURE 3. Illustration of the semantically visualized feature maps and their weighted combination for output prediction, where "Conv" means a convolutional layer, "GAP" means the global average pooling operation, and $w_1, w_2, \ldots, w_{C-1}$, mean the weighting coefficients for the corresponding feature maps. The term F_c, $c = 1, 2, \ldots, C\text{-}1$, denotes the refined version of the original learned feature map X_c.

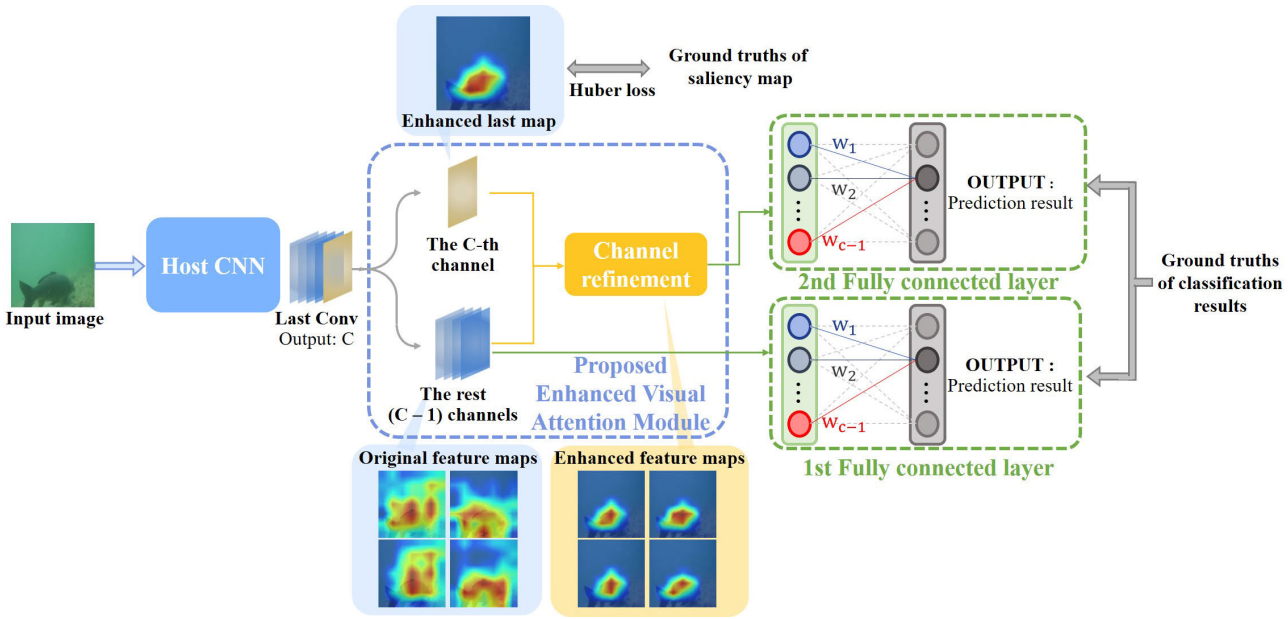

FIGURE 4. Illustration of the training process for the proposed framework.

is used as the loss function. The Huber loss function has been shown to be more robust to outlier than the generally used MSE (mean squared error) function. The Huber loss function for each pair of the selected feature map X_c and its corresponding ground truth Y_c is expressed as:

$$H(X_c, Y_c) = \frac{1}{H \times W} \sum_{a=1}^{H} \sum_{b=1}^{W} \text{Huber}\left(X_{c,a,b}, Y_{c,a,b}\right) \quad (2)$$

where

$$\text{Huber}\left(X_{c,a,b}, Y_{c,a,b}\right) = \begin{cases} \frac{1}{2}\left(X_{c,a,b} - Y_{c,a,b}\right)^2 & \text{if } \left|X_{c,a,b} - Y_{c,a,b}\right| \le \delta, \\ \delta\left|X_{c,a,b} - Y_{c,a,b}\right| - \frac{1}{2}\delta^2 & \text{otherwise,} \end{cases} \quad (3)$$

where $X_{c,a,b}$ and $Y_{c,a,b}$ denote the (a, b)-th element of X_c and Y_c, respectively. H and W are the height and the width of each feature map, respectively. The parameter δ is a threshold, empirically set to 0.7. During the training process, all of the element values of the feature maps are first normalized using the min-max normalization method [30], [31].

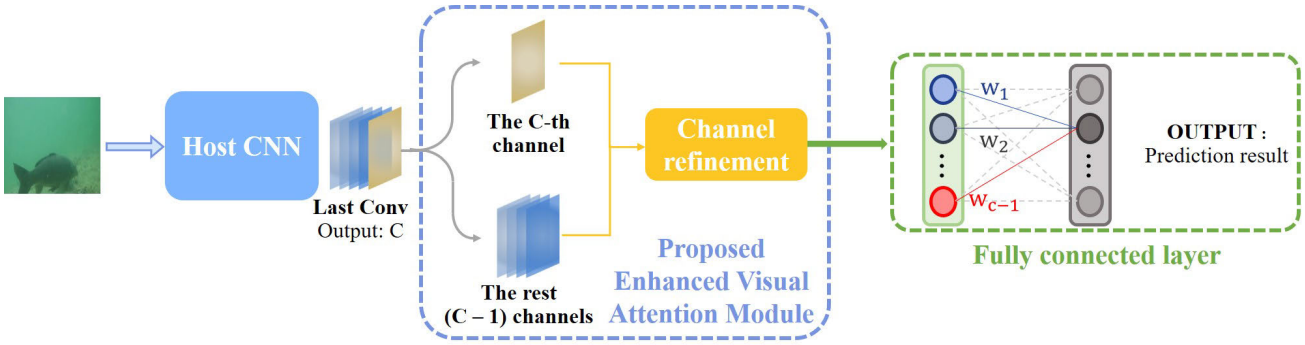

FIGURE 5. Illustration of the testing process for the proposed framework.

On the other hand, the loss functions used to guide the image classification outputs from the two fully connected layers to the corresponding ground truths are the generally used cross entropy functions. The used cross entropy loss functions for the 1st and 2nd fully connected layers are, respectively, expressed as:

$$\mathrm{CE}_1 = -\sum_{i=1}^{N}\sum_{j=1}^{C} \mathrm{Y}_{1,i,j}\log\left(\mathrm{P}_{1,i,j}\right), \tag{4}$$

$$\mathrm{CE}_2 = -\sum_{i=1}^{N}\sum_{j=1}^{C} \mathrm{Y}_{2,i,j}\log\left(\mathrm{P}_{2,i,j}\right), \tag{5}$$

where i, j, N, and C denote the serial number of the i-th training image, the serial number of the j-th class, the total number of the training images, and the total number of classes for image classification. The terms, $\mathrm{Y}_{1,i,j}$ and $\mathrm{Y}_{2,i,j}$, are two binary indicators (ground truths), respectively, for the 1st and 2nd fully connected layers. If the class label j is the correct prediction for the observation i, the indicator is 1. Otherwise, the indicator is 0. $\mathrm{P}_{1,i,j}$ and $\mathrm{P}_{2,i,j}$ are two predicted probabilities, respectively, generated by the 1st and 2nd fully connected layers for indicating the observation i belongs to the class j. The two fully connected layers used in the proposed training process are exactly the same and also with the same training data.

Based on the Huber loss defined in Eq. (2) and the cross entropy losses defined in Eqs. (4) and (5), the total loss function for training the host CNN with the proposed enhanced visual attention module embedded is expressed as:

$$\mathrm{loss}_{\mathrm{total}} = \lambda_1 \times \mathrm{CE}_1 + \lambda_2 \times \mathrm{CE}_2 + \lambda_3 \times \boldsymbol{H}, \tag{6}$$

where λ_1, λ_2, and λ_3 are the weighting coefficients to control the weight for each respective loss. Our guideline for empirically tuning the weighting coefficients are addressed as follows. The term CE_1 is used for guiding the original feature maps before refinement to the final results, and therefore its weighting coefficient λ_1 is set to be smaller. In addition, the term CE_2 is used for guiding the refined feature maps to the final results, and therefore its weighting coefficient λ_2 is set to be larger. Moreover, the term H is used to guide the selected feature map to the corresponding ground truth map for further feature refinement, and therefore, its weighting coefficient λ_3 is also set to be larger. As a result, based on the guideline, the three weighting coefficients λ_1, λ_2, and λ_3 are empirically set to 0.2, 0.4, and 0.4, respectively, where $\lambda_1 + \lambda_2 + \lambda_3 = 1.0$. Based on the proposed loss function defined in Eq. (6), in the training process, we aim at guiding both of the original learned feature maps and the refined feature maps toward the correct prediction output while guiding the selected feature map toward its corresponding ground truth salient map for refining the other feature maps. Therefore, the learned deep model would usually generalize well and be neither underfit nor overfit.

C. TESTING CONVOLUTIONAL NEURAL NETWORKS WITH THE PROPOSED ENHANCED VISUAL ATTENTION MODULE

In the testing process of the proposed method, different from the network structure used in the training stage, only one fully connected layer (used in the host CNN) is used. As illustrated in Fig. 5, in the testing stage, each input image for image classification is fed into the host CNN and goes through the deep network. After obtaining the feature maps generated from the last convolutional layer of the CNN, the proposed module splits the total C maps into the C-th map and the rest (C − 1) maps. The selected C-th channel is re-mapped to its refined version by our module, which is used as a weighting coefficient map. Then the weighting coefficient map is used to enhance each of the rest (C − 1) maps by the element-wise multiplication operation via Eq. (1). The enhanced feature maps are connected to the fully connected layer, which generates the final prediction output.

III. EXPERIMENTAL RESULTS

A. NETWORK TRAINING AND PARAMETER SETTINGS

To evaluate the performance of the proposed enhanced visual attention module, we selected four classic convolutional neural networks to form our host CNNs and embedded the proposed module into them. The four host CNNs are VGG16 [10], ResNet-50 [12], MobileNet V2 [13], and ShuffleNet V2 [14]. To train each host CNN with the proposed module embedded, we used three image datasets for

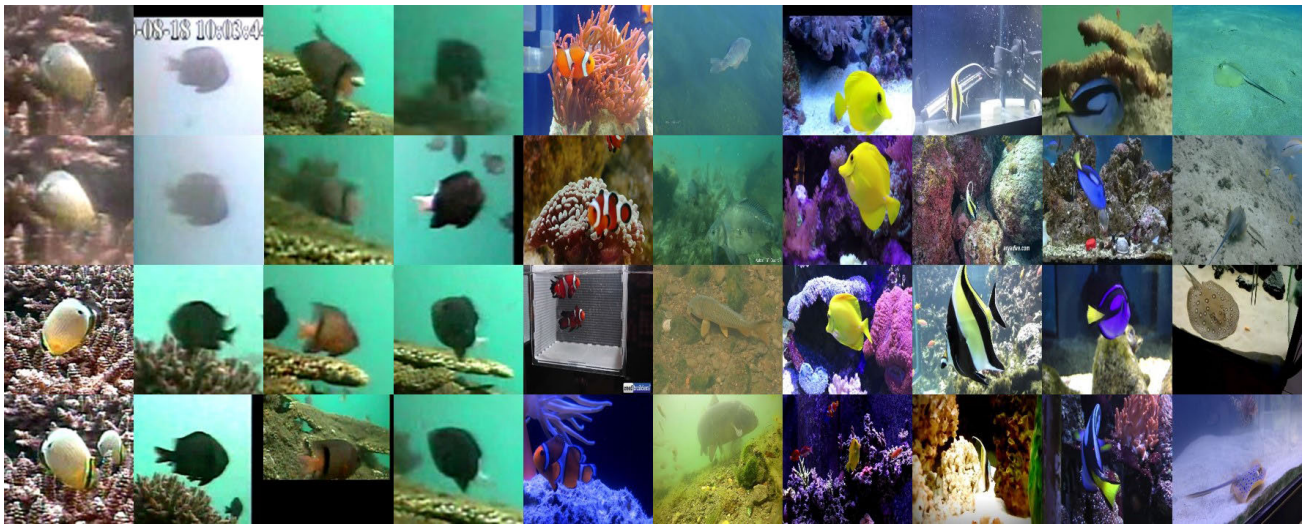

FIGURE 6. Some examples of our collected underwater fish dataset of 10 classes. Each column indicates one class, where the upper two images are for training and the lower two images are for testing.

FIGURE 7. Some examples of the public Animals-10 dataset of 10 classes [33]. Each column indicates one class, where the upper two images are for training and the lower two images are for testing.

image classification. The used three datasets are our collected Underwater Fish dataset, where some images in this dataset are collected from [32] of 10 classes (Fig. 6), the public Animals-10 dataset of 10 classes (Fig. 7) [33], and the public Stanford Cars dataset of 196 classes (Fig. 8) [34]. In all our experiments presented in this article, all used images are in the RGB color space with the number of input channels set to 3. Moreover, based on the fact that it is not easy to get the ground truths of the feature maps learned by the last convolutional layer of a CNN, in our experiments, we applied the PoolNet [35] and BASNet [36] to generate the ground truths of feature maps for our training images, as examples shown in Fig. 9. Both of the two deep networks [35], [36] are mainly designed for salient object detection with the corresponding salient map generated. The numbers of training images, testing images, and ground truth salient maps for the three datasets are summarized in Table 1.

It should be noted that using the ground truths of saliency maps to guide the last learned feature map of a CNN in the training process indeed introduces richer information than only using the classification labels for network learning. However, the applied existed saliency detection models [35], [36] may generate wrong saliency maps, and therefore, we applied the Huber loss function to reduce the effects of possible outliers. On the other hand, we also introduced the other two terms based on the classification labels for guiding the two fully connected layers into our total loss function.

In addition, to train each host CNN with embedding all the evaluated attention modules, we used the RMSprop, i.e., root mean square propagation, optimizer [37] with the momentum

FIGURE 8. Some examples of the public stanford cars of 196 classes [34]. Each column indicates one class (only 10 classes are shown), where the upper two images are for training and the lower two images are for testing.

FIGURE 9. Some examples of the ground truths of the feature/saliency maps and the generated feature/saliency maps by our method. The former five columns are training images (upper) and the corresponding ground truths of the feature/saliency maps (lower). The latter five columns are testing images (upper) and the corresponding feature/saliency maps (the enlarged versions for the original size of 7 × 7) obtained by our method.

TABLE 1. Numbers of training images, testing images, and ground truth salient maps for the three used datasets.

Dataset	No. of Classes	No. of Training Images	No. of Testing Images	No. of Ground Truth Salient Maps
Underwater Fish	10	21783	5884	21783
Animals-10 [33]	10	18324	7855	18324
Stanford Cars [34]	196	8144	8041	8144

TABLE 2. Parameter settings.

Dataset	Host CNN	Initial Learning Rate	Weight Decay	Epoch
Our Underwater Fish	VGG16 [10]	0.001	5e-4	150
	ResNet-50 [12]	0.010	1e-4	150
	MobileNetV2 [13]	0.045	4e-5	150
	ShuffleNetV2 [14]	0.500	4e-5	150
Animals-10 [33]	VGG16 [10]	0.005	5e-4	200
	ResNet-50 [12]	0.020	1e-4	200
	MobileNetV2 [13]	0.045	4e-5	200
	ShuffleNetV2 [14]	0.010	4e-5	200
Stanford Cars [34]	VGG16 [10]	0.001	5e-4	400
	ResNet-50 [12]	0.020	1e-4	400
	MobileNetV2 [13]	0.045	4e-5	400
	ShuffleNetV2 [14]	0.050	4e-5	400

set to 0.9, the learning rate decay set to 0.98 per epoch, and the input image size set to 224 × 224. All the evaluated attention modules denote the four compared state-of-the-art modules [19], [20], [22], [23], descried in Sec. III.B, and the proposed module. The other parameter settings are summarized in Table 2. In Table 2, for each host CNN trained on a dataset, all the parameters are the same as those used for this host CNN with embedding each attention module.

Moreover, the selection of the datasets used for model training and testing in our experiments mainly depends on the following three principles. First, the proposed framework focuses on image classification for images with dominated objects inside, and therefore, prefers to datasets consisting of images with clear objects and suitable labels. Second, the selected datasets should be representative and popular in the image processing and computer vision community. Third, the selected datasets may be useful for our recently executed project for the applications of unmanned underwater

TABLE 3. Quantitative results in terms of the Top-1 and Top-5 accuracies, the number of parameters, denoted by params (in M or Mega), the FLOPs (in G or Giga), and the average run time per image (in milliseconds) obtained by each host CNN with/without embedding the compared attention modules and the proposed module conducted on our underwater fish dataset. For each term of params and FLOPs, only the increment from the corresponding value of the corresponding host CNN is shown.

Host CNN with/without Attention Module	Top-1 (%)	Top-5 (%)	Params (M)	FLOPs (G)	Run Time per Image (Milliseconds)	Host CNN with/without Attention Module	Top-1 (%)	Top-5 (%)	Params (M)	FLOPs (G)	Run Time per image (Milliseconds)
VGG16 [10]	89.36	98.28	134.31	15.6265	7.59	MobileNetV2 [13]	92.47	99.56	2.2366	0.31287	19.28
SE [19]	89.50	98.30	+0.2318	+0.0140	12.87	SE [19]	92.35	99.80	+2.2674	+0.00683	29.90
CBAM [20]	87.00	98.37	+0.2300	+0.0147	36.81	CBAM [20]	92.10	99.58	+0.5734	+0.00454	52.47
RAN [22]	85.08	97.55	+10.950	+3.3626	41.83	RAN [22]	92.83	99.29	+0.0824	+0.01238	39.87
ECA [23]	89.62	99.39	+0.0001	+0.0136	13.28	ECA [23]	89.77	99.01	+0.0001	+0.00058	25.90
Proposed	**95.36**	**99.88**	**+0.0000**	**+0.0004**	**7.84**	Proposed	**96.11**	**99.90**	**+0.0000**	**+0.00006**	**19.29**
ResNet-50 [12]	92.11	99.76	23.529	4.10951	22.14	ShuffleNetV2 [14]	90.33	99.46	1.2639	0.14780	23.70
SE [19]	91.55	99.52	+2.5144	+0.01055	29.23	SE [19]	89.82	99.03	+0.6882	+0.00219	34.06
CBAM [20]	91.20	99.18	+2.5320	+0.01134	47.03	CBAM [20]	85.38	98.52	+0.1721	+0.00116	54.54
RAN [22]	91.93	99.37	+22.265	+4.41079	54.05	RAN [22]	89.00	99.61	+1.7637	+0.16937	71.99
ECA [23]	91.04	98.78	+0.0001	+0.00562	26.69	ECA [23]	90.69	**99.76**	+0.0001	+0.00084	30.09
Proposed	**95.38**	**99.77**	**+0.0000**	**+0.00010**	**23.25**	Proposed	**94.44**	99.71	**+0.0000**	**+0.00005**	**24.03**

TABLE 4. Quantitative results in terms of the Top-1 and Top-5 accuracies, the number of parameters, denoted by params (in M or Mega), the FLOPs (in G or Giga), and the average run time per image (in milliseconds) obtained by each host CNN with/without embedding the compared attention modules and the proposed module conducted on the Animals-10 dataset [33]. For each term of params and FLOPs, only the increment from the corresponding value of the corresponding host CNN is shown.

Host CNN with/without Attention Module	Top-1 (%)	Top-5 (%)	Params (M)	FLOPs (G)	Run Time per Image (Milliseconds)	Host CNN with/without Attention Module	Top-1 (%)	Top-5 (%)	Params (M)	FLOPs (G)	Run Time per image (Milliseconds)
VGG16 [10]	91.78	98.97	134.31	15.6265	7.78	MobileNetV2 [13]	92.36	99.08	2.2366	0.31287	19.09
SE [19]	93.21	99.17	+0.2318	+0.0140	14.72	SE [19]	91.02	99.15	+2.2674	+0.00683	33.63
CBAM [20]	92.45	99.21	+0.2300	+0.0147	37.12	CBAM [20]	91.83	99.06	+0.5733	+0.00454	52.28
RAN [22]	86.25	97.85	+10.950	+3.3626	42.60	RAN [22]	91.64	**99.35**	+0.0824	+0.01238	39.18
ECA[23]	93.10	99.21	+0.0001	+0.0136	11.73	ECA [23]	92.48	99.07	+0.0001	+0.00058	25.97
Proposed	**94.98**	**99.40**	**+0.0000**	**+0.0004**	**7.96**	Proposed	**93.62**	99.33	**+0.0000**	**+0.00006**	**19.75**
ResNet-50 [12]	91.99	98.98	23.529	4.10951	21.50	ShuffleNetV2 [14]	87.05	**98.79**	1.2639	0.14780	21.94
SE [19]	92.18	98.98	+2.5145	+0.01055	28.44	SE [19]	85.65	98.29	+0.6882	+0.00219	31.21
CBAM [20]	91.46	98.93	+2.5320	+0.01134	46.74	CBAM [20]	82.71	98.00	+0.1721	+0.00116	53.45
RAN [22]	90.26	98.92	+22.265	+4.41079	69.10	RAN [22]	84.81	98.33	+1.7637	+0.16937	69.53
ECA [23]	91.06	99.03	+0.0001	+0.00562	30.26	ECA [23]	85.58	98.64	+0.0001	+0.00084	28.98
Proposed	**94.54**	**99.53**	**+0.0000**	**+0.00010**	**22.57**	Proposed	**87.79**	98.66	**+0.0000**	**+0.00005**	**23.80**

vehicles, i.e., UUVs. The selection for our collected Underwater Fish dataset mainly depends on the first and the third principles. In addition, one of our data sources [32] for forming our Underwater Fish dataset is also popular in recently related research works, e.g., [38], [39]. On the other hand, the selections of both the public Animals-10 dataset [33], also used in recent works, e.g., [40], [41], and the public Stanford Cars dataset [34], also used in recent works, e.g., [42], [43], are mainly based on the former two principles.

B. QUANTITATIVE RESULTS

To evaluate the image classification performance for each selected host CNN with the proposed enhanced visual attention module embedded, we reported the top-1 and top-5 accuracies obtained on the respective dataset. Moreover, we also compared the SE (squeeze-and-excitation) module [19], CBAM (convolutional block attention module) [20], RAN (residual attention network) module [22], and ECA (efficient channel attention) module [23] with the proposed module by embedding the respective attention module into the four selected host CNNs. To get significant performance improvement compared with each original host CNN, the compared attention modules might be usually embedded into the host CNN multiple times, for example, to be embedded after each convolutional layer. Different from these approaches, the proposed module just needs to be embedded once after the last convolutional layer of each host CNN. Tables 3-5, respectively, show the top-1 and top-5 accuracies (suggested by [44]) obtained by the four host CNNs, VGG16 [10], ResNet-50 [12], MobileNet V2 [13], and ShuffleNet V2 [14], with and without embedding the SE [19], CBAM [20], RAN [22], ECA [23], and the proposed modules, respectively, on our Underwater Fish, the Animals-10 [33], and the Stanford Cars [34] datasets. It can be found from Tables 3-5, embedding the proposed module into the host CNNs can significantly improve the top-1 and top-5 accuracies, compared with those obtained by the original host CNNs and those obtained by the host CNNs with embedding the compared attention modules. That is, the proposed enhanced visual attention module can be widely embedded

TABLE 5. Quantitative results in terms of the Top-1 and Top-5 accuracies, the number of parameters, denoted by params (in M or Mega), the FLOPs (in G or Giga), and the average run time per image (in milliseconds) obtained by each host CNN with/without embedding the compared attention modules and the proposed module conducted on the stanford cars dataset [34]. For each term of params and FLOPs, only the increment from the corresponding value of the corresponding host CNN is shown.

Host CNN with/without Attention Module	Top-1 (%)	Top-5 (%)	Params (M)	FLOPs (G)	Run Time per Image (Milliseconds)	Host CNN with/without Attention Module	Top-1 (%)	Top-5 (%)	Params (M)	FLOPs (G)	Run Time per image (Milliseconds)
VGG16 [10]	56.35	80.84	135.07	15.6280	7.61	MobileNetV2 [13]	66.53	86.69	2.4749	0.31335	18.59
SE [19]	56.26	81.01	+0.2338	+0.0140	13.66	SE [19]	68.00	87.30	+2.2674	+0.00683	32.12
CBAM [20]	49.88	76.35	+0.2300	+0.0147	38.65	CBAM [20]	56.87	80.36	+0.5734	+0.00453	53.59
RAN [22]	51.35	77.62	+10.950	+3.3627	39.72	RAN [22]	66.98	86.94	+0.0824	+0.01238	38.24
ECA[23]	56.37	80.56	+0.0001	+0.0136	12.56	ECA [23]	65.69	86.47	+0.0001	+0.00058	25.77
Proposed	**58.49**	**83.20**	**+0.0000**	**+0.0004**	**7.68**	Proposed	**70.82**	**89.73**	**+0.0000**	**+0.00006**	**18.68**
ResNet-50 [12]	62.96	85.33	23.909	4.11027	18.65	ShuffleNetV2 [14]	61.53	83.87	1.4545	0.14819	23.21
SE [19]	64.67	85.52	+2.5156	+0.01055	28.47	SE [19]	61.26	83.68	+0.6882	+0.00219	33.69
CBAM [20]	61.02	83.43	+2.5330	+0.01135	55.35	CBAM [20]	55.78	80.14	+0.1722	+0.00115	52.63
RAN [22]	64.72	85.35	+22.266	+4.41079	54.41	RAN [22]	59.06	82.04	+1.7637	+0.16936	71.83
ECA [23]	61.20	83.10	+0.0001	+0.00562	25.87	ECA [23]	59.54	82.65	+0.0001	+0.00084	32.28
Proposed	**72.11**	**90.42**	**+0.0000**	**+0.00010**	**21.57**	Proposed	**64.57**	**86.53**	**+0.0000**	**+0.00043**	**23.97**

TABLE 6. The Top-1 and Top-5 image classification accuracies obtained by MobileNetV2 and MobileNetV2 with proposed module embedded (denoted by + proposed) on our collected underwater fish dataset in terms of input image sizes of 224 × 224, 112 × 112, and 56 × 56.

Input image size	Evaluated Frameworks	Top-1 (%)	Top-5 (%)
224×224	MobileNetV2	92.47	99.56
	+Proposed	**96.11**	**99.90**
112×112	MobileNetV2	88.48	98.93
	+Proposed	**93.63**	**99.29**
56×56	MobileNetV2	81.41	98.61
	+Proposed	**84.18**	**98.74**

into any existed CNN architectures, enhance the features learned by the last convolutional layers, and be generalized to many datasets for image classification.

On the other hand, to evaluate the image classification accuracies obtained by the proposed method in terms of different input image sizes, we reported the related results in Table 6. Table 6 shows the top-1 and top-5 accuracies obtained by embedding the proposed module into MobileNetV2 in terms of the image sizes of 224 × 224, 112 × 112, and 56 × 56, respectively. As shown in Table 6, larger input image size will lead to better classification accuracy. However, even if the input image size is relatively small, the proposed method still achieves acceptable results.

C. NETWORK COMPLEXITY ANALYSIS

The proposed method was implemented in Python programming language with Pytorch [45] on a personal computer equipped with Intel ®Core ™i7-4790 CPU, 3.6 GHz, 16 GB memory, and NVIDIA GeForce RTX 2080 Ti GPU. To analyze the complexities of the evaluated host CNNs with the proposed module embedded, we reported the numbers of network parameters, the FLOPs (floating point operations) for network testing, and the average run time per image (in milliseconds). Tables 3-5 shows the numbers of network parameters, the FLOPs for network testing, and the average run time per image (in milliseconds) for the four evaluated host CNNs with and without embedding the SE [19], CBAM [20], RAN [22], ECA [23], and the proposed modules, respectively, conducted on the three datasets. Based on Tables 3-5, the additional burden of network complexity induced by embedding the proposed module is almost negligible. The main reason is that the proposed module is only required to be embedded once into each host CNN, where only one additional element-wise multiplication is required for enhancing each feature map. It can be also observed from Tables 3-5 that the run time for testing an image based on all the host CNNs with embedding the proposed module is lower than those obtained by embedding the compared state-of-the-art deep attention modules [19], [20], [22], [23]. Therefore, the proposed enhanced visual attention module can be easily embedded into any CNN architectures with negligible extra burden.

IV. CONCLUSION

In this article, we have proposed an enhanced visual attention module for being embedded into any existed CNNs for image classification purpose. By enhancing the features learned from the last convolutional layer, which can capture richer semantic information for image classification while suppressing insignificant information, of a CNN, the CNN with the proposed module embedded achieves significant improvement in classification performance with negligible extra overhead. For future works, it is expected to extend our module for enhancing CNNs of different purposes, such as image regression.

REFERENCES

[1] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward," *Comput. Vis. Image Understand.*, vol. 117, no. 8, pp. 827–891, Aug. 2013.

[2] J. Seo and H. Park, "Object recognition in very low resolution images using deep collaborative learning," *IEEE Access*, vol. 7, pp. 134071–134082, Sep. 2019.

[3] H. M. Bui, M. Lech, E. Cheng, K. Neville, and I. S. Burnett, "Object recognition using deep convolutional features transformed by a recursive network structure," *IEEE Access*, vol. 4, pp. 10059–10066, 2016.

[4] F. Storbeck and B. Daan, "Fish species recognition using computer vision and a neural network," *Fisheries Res.*, vol. 51, no. 1, pp. 11–15, Apr. 2001.

[5] X. Liu, M. Neuyen, and W. Q. Yan, "Vehicle-related scene understanding using deep learning," in *Proc. Asian Conf. Pattern Recognit.*, 2019, pp. 61–73.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[7] F. Tajeripour, M. Saberi, and S. F. Ershad, "Developing a novel approach for content based image retrieval using modified local binary patterns and morphological transform," *Int. Arab J. Inf. Technol.*, vol. 12, no. 6, pp. 574–581, 2015.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[14] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[15] C.-H. Yeh, Z.-T. Zhang, M.-J. Chen, and C.-Y. Lin, "HEVC intra frame coding based on convolutional neural network," *IEEE Access*, vol. 6, pp. 50087–50095, Sep. 2018.

[16] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[17] N. Sun and H. Li, "Super resolution reconstruction of images based on interpolation and full convolutional neural network and application in medical fields," *IEEE Access*, vol. 7, pp. 186470–186479, Dec. 2019.

[18] C.-H. Yeh, C.-H. Huang, and L.-W. Kang, "Multi-scale deep residual learning-based single image haze removal via image decomposition," *IEEE Trans. Image Process.*, vol. 29, no. 1, pp. 3153–3167, Dec. 2020.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[20] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[21] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7794–7803.

[22] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6450–6458.

[23] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 11534–11542.

[24] H.-J. Yu and C.-H. Son, "Leaf spot attention network for apple leaf disease identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Seattle, WA, USA, Jun. 2020, pp. 229–237.

[25] H.-J. Yu, C.-H. Son, and D. H. Lee, "Apple leaf disease identification through region-of-interest-aware deep convolutional neural network," *J. Imag. Sci. Technol.*, vol. 64, no. 2, pp. 20507-1–20507-10, Mar. 2020.

[26] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, Sep. 2014, pp. 818–833.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 618–626.

[28] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, Mar. 1964.

[29] R. Matsuoka, S. Ono, and M. Okuda, "Transformed-domain robust multiple-exposure blending with Huber loss," *IEEE Access*, vol. 7, pp. 162282–162296, Nov. 2019.

[30] S. Jain, S. Shukla, and R. Wadhvani, "Dynamic selection of normalization techniques using data complexity measures," *Expert Syst. Appl.*, vol. 106, pp. 252–262, Sep. 2018.

[31] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, "Mixture of activation functions with extended min-max normalization for forex market prediction," *IEEE Access*, vol. 7, pp. 183680–183691, Dec. 2019.

[32] B. J. Boom, P. X. Huang, J. He, and R. B. Fisher, "Supporting ground-truth annotation of image datasets using clustering," in *Proc. Int. Conf. Pattern Recognit.*, Tsukuba, Japan, Nov. 2012, pp. 1542–1545.

[33] *Animals-10-Animal Pictures of 10 Different Categories Taken From Google Images*. Accessed: Aug. 2020. [Online]. Available: https://www.kaggle.com/alessiocorrado99/animals10

[34] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Sydney, NSW, Australia, Dec. 2013, pp. 554–561.

[35] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 3917–3926.

[36] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7479–7489.

[37] S. Ruder, "An overview of gradient descent optimization algorithms," Jun. 2017, *arXiv:1609.04747*. [Online]. Available: http://arxiv.org/abs/1609.04747

[38] M.-C. Chuang, J.-N. Hwang, and K. Williams, "A feature learning and object recognition framework for underwater fish images," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1862–1872, Apr. 2016.

[39] H. Qin, X. Li, J. Liang, Y. Peng, and C. Zhang, "DeepFish: Accurate underwater live fish recognition with a deep architecture," *Neurocomputing*, vol. 187, pp. 49–58, Apr. 2016.

[40] D. Kang, A. Mathur, T. Veeramacheneni, P. Bailis, and M. Zaharia, "Jointly optimizing preprocessing and inference for DNN-based visual analytics," Jul. 2020, *arXiv:2007.13005*. [Online]. Available: http://arxiv.org/abs/2007.13005

[41] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, J. Santamaría, Y. Duan, and S. R. Oleiwi, "Towards a better understanding of transfer learning for medical imaging: A case study," *Appl. Sci.*, vol. 10, no. 13, p. 4523, Jun. 2020.

[42] D. Chang, Y. Ding, J. Xie, A. K. Bhunia, X. Li, Z. Ma, M. Wu, J. Guo, and Y.-Z. Song, "The devil is in the channels: Mutual-channel loss for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 29, pp. 4683–4695, Feb. 2020.

[43] M. Ye and J. Shen, "Probabilistic structural latent representation for unsupervised embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 5457–5466.

[44] N. Hor and S. Fekri-Ershad, "Image retrieval approach based on local texture information derived from predefined patterns and spatial domain information," *Int. J. Comput. Sci. Eng.*, vol. 8, no. 6, pp. 246–254, Nov-Dec. 2019.

[45] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2019, pp. 8024–8035.

CHIA-HUNG YEH (Senior Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan, in 1997 and 2002, respectively. He was an Assistant Professor, from 2007 to 2010, an Associate Professor, from 2010 to 2013, and a Professor, from 2013 to 2017, with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. He is currently a Distinguished Professor with National Taiwan Normal University, Taipei, Taiwan. He has coauthored more than 250 technical international conferences and journal papers and held 47 patents in the USA, Taiwan, and China. His research interests include multimedia, video communication, 3-D reconstruction, video coding, image/video processing, and big data. He was a recipient of the 2007 Young Researcher Award of NSYSU, the 2011 Distinguished Young Engineer Award from the Chinese Institute of Electrical Engineering, the 2013 Distinguished Young Researcher Award of NSYSU, the 2013 IEEE MMSP Top 10% Paper Award, the 2014 IEEE GCCE Outstanding Poster Award, the 2015 APSIPA Distinguished Lecturer, the 2016 NARLabs Technical Achievement Award: Superior Achievement Award, the 2017 IEEE SPS Tainan Section Chair, the 2017 Distinguished Professor Award of NTNU, and the IEEE Outstanding Technical Achievement Award (the IEEE Tainan Section). He is an Associate Editor of the *Journal of Visual Communication and Image Representation* (JVCI), the *EURASIP Journal on Advances in Signal Processing*, and the *APSIPA Transactions on Signal and Information Processing*. He has been on the Best Paper Award Committee of JVCI and APSIPA.

MIN-HUI LIN received the B.S. degree from the Department of Electrical Engineering, National University of Kaohsiung, Taiwan, in 2016. She is currently pursuing the Ph.D. degree in electrical engineering with National Sun Yat-sen University, Taiwan. Her research interests include deep learning for computer vision, 3-D reconstruction, and multimedia applications.

PO-CHAO CHANG received the B.S. degree from the Department of Electrical Engineering, Yuan Ze University, Taiwan, in 2019. He is currently pursuing the master's degree in electrical engineering with National Sun Yat-sen University, Taiwan. His research interests include deep learning for image processing, computer vision, and multimedia applications.

LI-WEI KANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from National Chung Cheng University, Chiayi, Taiwan, in 1997, 1999, and 2005, respectively. Since August 2019, he has been with the Department of Electrical Engineering, National Taiwan Normal University, Taipei, Taiwan, as an Associate Professor. Before that, he worked for the Graduate School of Engineering Science and Technology-Doctoral Program and the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Douliu, Taiwan, as an Associate Professor, from August 2016 to July 2019, and an Assistant Professor, from February 2013 to July 2016. He also worked with the Institute of Information Science, Academia Sinica, Taipei, as an Assistant Research Scholar, from 2010 to 2013, and a Postdoctoral Research Fellow, from 2005 to 2010. His research interests include multimedia content analysis and multimedia communications. He has been an Organizing Committee Member of the IEEE ICCE-TW, since 2015, and a Multimedia Systems and Applications Technical Committee (MSATC) Member of the IEEE Circuits and Systems Society, since September 2020. He received the Top 10% Paper Award from the IEEE MMSP 2013 and the Best Performance Award from the Social Media Prediction (SMP) Challenge of the ACM MM 2019. He served as an Editorial Board Member and a Guest Editor for the *International Journal of Distributed Sensor Networks*, a Guest Editor for the *APSIPA Transactions on Signal and Information Processing*, the Special Session Co-Chair of the APSIPA ASC 2012, the Registration Co-Chair of the APSIPA ASC 2013, and the Ph.D. Forum Committee Member of the APSIPA ASC 2018.