# IES: A Powerful Visual Feature Representation Network

Xiang Li, *Member, IAENG,* Xueqing Zhao, *Member, IAENG,*

***Abstract*—Effective visual information representation is significant for the feature extractability of deep neural networks. The transformation from images to feature maps realized by the stem, referring to the initial part of the neural network that processes input images, causes the loss of information owing to the heterogeneity of the color and structure information: the most prominent features. To solve this problem, we propose a powerful and effective image-embedding stem (IES) model with a color-embedding module (CEM), structure embedding module (SEM), and feature mixing module (FMM). Specifically, the red-green-blue (RGB) ternary color information is embedded into a high-dimensional vector space containing rich feature information through the CEM. Simultaneously, the SEM is used to explicitly encode multiscale structural information to enrich the detailed information in the feature maps. Finally, they are fused by the FMM to preserve more details. Comprehensive experiments demonstrate the efficacy of the IES in different visual tasks. It achieved $+1.2$ and $+0.5$ top-1 accuracy ratings on the ImageNet-100 dataset for the VanillaNet-5 and TinyViT-5m backbones, respectively, and obtained $+2.36$ and $+1.7$ mean intersection-over-union scores on the UTFPR-SBD3 dataset for PoolFormer and ConvNeXtV2 backbones, respectively. The code and models will be released soon.**

***Index Terms*—color embedding, deep learning, image-embedding stem, structure embedding, visual feature representation**

## I. Introduction

DEEP neural networks (DNNs) have developed rapidly in support of various computer vision tasks, such as classification [1], [2], object detection [3], [4], and segmentation [5], [6], [7]. DNNs are known for their remarkable generalizability in automatically learning patterns and features from visual images. Efficient and targeted structural designs [8], [9], [10], [11] can provide DNN backbone models with the powerful ability to extract visual semantic features. There are two key backbone types: convolutional neural networks (CNNs) [2], [9], [12], [5] and transformer-based methods [13], [14], [15], [16], [17], [18]. Both focus on extracting high-level semantic representations from input images and obtaining feature maps to track rich semantic information. Features can be represented by large parameters [14], [19] or complex structures [4], [3], [15]. Next, we outline these two methods and introduce our novel advancement.

Manuscript received September 9, 2023; revised February 23, 2024. This work was supported by the National Social Science Foundation of China Art Project (No.23EH232), and the Key Research and Development Program of Shaanxi Province in 2023 (No.2023-YBGY-404, No.2023-ZDLGY-48), and Research Center for Culture Sci-Tech Integration Innovation, Key Research Base of Humanities and Social Sciences of Hubei Province, and Shaanxi Province University Young Outstanding Talents Support Program.

Xiang Li is a graduate student of Computer Science Department, University of Xi'an Polytechnic, 58 Shaangu Avenue, Xi'an, Shaanxi, 710600, China. (e-mail: 210711018@stu.xpu.edu.cn)

Xueqing Zhao is an associate professor of Computer Science Department, University of Xi'an Polytechnic, 58 Shaangu Avenue, Xi'an, Shaanxi, 710600, China. (e-mail: zhaoxueqing@xpu.edu.cn)

### A. CNN-Based Models

CNNs have efficiently evolved from AlexNet [1] to ResNet [20] and ConvNeXtV2 [21] models, among these, the residual connection proposed in ResNet [20] has become the most widely used. The evolution of CNN architectures [22], [23], [24], [25] has further increased their popularity in a variety of vision tasks. To reduce model parameters and calculation complexity, MobileNet [24], [26], GhostNet [27], and Xception [28], among other backbones, leverage depthwise and group convolutions [29] to extract spatial features, which reduces the number of parameters without significantly reducing performance. This has influenced the designs of many subsequent models [5], [30], [31].

Alternatively a reasonable structural design can achieve twice the performance with half the effort [29], [4], [32], [33] by adopting multipath or multibranch structures to extract features at different scales. Other novel methods [2], [9], [12], [34] have further improved model performance using only pure convolutions while maintaining light weight. Considering the importance of the different dimensions of a feature map, new methods have dynamically weighted different channels [35], [36], [37], [38], [39] and applied spatial attention to enhance spatial feature extraction [40], [41]. In summary, CNNs have been developed to prioritize light weight and novelty to optimize performance. Studies are ongoing to further enhance CNN performance, such as by focusing on the extraction of advanced features from deep feature maps. Our research fills a gap in this field by improving the image-embedding stem to better characterize visual image features.

### B. Transformer-Based Models

The vision transformer (ViT) innovation [19] has ignited enthusiasm for the application of transformers in computer vision tasks. Many ViT variants [16], [17], [14], [15], [42] have achieved remarkable performance. Although they have stronger feature extraction ability than CNNs, the quadratic complexity within their pairwise attentions prohibitively restricts their computational efficiency [8], [14], [15]. Some studies have adopted hierarchical layouts [8], [17], [14], [15] and shift-invariant priors [43], [44], [45] to alleviate this limitation. One impressive model [46] refined the basic transformer architecture by replacing the self-attention module with pooling, convolution, or identity operations to achieve results comparable to or better than those of CNNs.

Transformer-based models have a global receptive field, which is one of the essential reasons for their superior performance. However, transformer methods still rely on complex architectures and numerous parameters to extract features, which fail to achieve superior primary representations of visual images using the initial part of the neural network

that processes input images (i.e., stems). By modifying the lowest structural stem, we propose a novel image-embedding stem (IES) model to overcome this limitation. The proposed IES model significantly improves the performance of existing backbones by strengthening their image representation.

### C. IES Module

Previous studies primarily focused on designing better feature-extraction backbones, whereas we focus on improving the network representation from the source by extracting information from the input images using the stem to provide a more meaningful feature map. By effectively embedding the structural texture and color information of the input image, the subsequent feature-extraction backbone obtains richer and more delicate feature representations, thereby making the model more expressive.

A significant performance improvement was achieved in comparative experiments using various state-of-the-art DNN models. IES achieved +1.2 and +0.5 top-1 accuracy ratings on the ImageNet-100 dataset for the VanillaNet-5 and TinyViT-5m backbones, respectively, and obtained +2.36 and +1.7 mean intersection-over-union (mIoU) scores on the UTFPR-SBD3 dataset for PoolFormer and ConvNeXtV2 backbones, respectively. The key contributions of this study are as follows:

- We summarize the common stem feature-extraction modules and identify their shortcomings.
- We provide a color-embedding module (CEM) and a structure-embedding module (SEM) to enhance the model's ability to extract different modes of information (i.e., color and texture) from visual images.
- Our proposed IES learns visual features from the input image rather than the original, which enhances the backbone's ability to mine deep-seated features and improve the overall performance.

## II. Related Work

In humans, visual information accounts for approximately 80% of all sensory inputs, and color and structural information are the main components of visual information. The stem is the first stage of visual image processing. Notably, a DNN's stem bridges the neural network and the input data, and it is usually located at the beginning of the network. It preprocesses the input visual data and performs preliminary feature extraction. As validated with ResNet [20], nearly all visual models contain stems, and they perform two key tasks. First, they extract shallow features containing a large amount of detailed information from the visual input, the so-called primary feature. Second, they reduce the feature map's resolution by adopting convolution or max-pooling layers with a stride greater than one.

Currently, the mainstream stem has three variants, as illustrated in Fig 1. The first type combines convolution with batch normalization (BN) and a rectified linear unit (ReLU) to provide a Conv+BN+ReLU structure [20], [47]. A convolution with a kernel size of seven and stride of two is adopted by this type to increase the receptive field. Subsequently, a MaxPool2d layer with a kernel size of three and stride of two is connected to reduce the resolution of the feature map and filter out irrelevant information. The

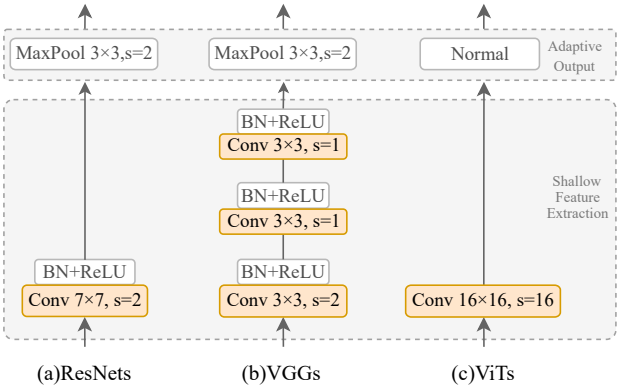



**Fig. 1:** Three stem variants. (a) a frequently employed single-layer stem as depicted in ResNet [20], (b) a multilayer stem exemplified by VGG [22], and (c) a distinctive stem known as Patch Embedding (PE) from ViT [19].

second type [23], [5] replaces the $7 \times 7$ convolution with a plurality of convolution layers with small kernel sizes, which reduces the number of model parameters while producing the same receptive field [23]. The third type, patch embedding (PE), combines convolution and normal layers to support ViT models [19], commonly setting the kernel size equal to its stride. Most visual models adopt one of these three base stems types.

Notably, these three types of stems do not fully consider the heterogeneity problem of the input information; therefore, they cannot effectively represent the image's feature information.

## III. IES

In this section, we explain how the stem functions in most DNNs. We then introduce the details of the proposed IES, including the new CEM, SEM, and feature mixing model (FMM) components.

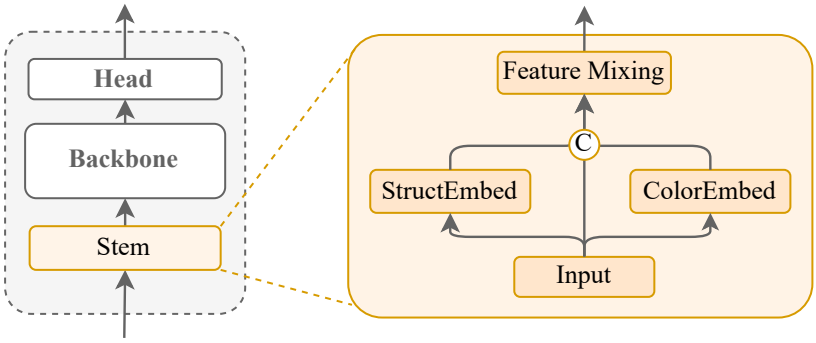
### A. Overview

During data processing, the input image must undergo stem processing before being input into the backbone. Given an input image, $X_{input} \in \mathbb{R}^{C\times H\times W}$, the processing is formulated as

$$X_f = Backbone(Stem(X_{input})) \tag{1}$$

where $Backbone(\cdot)$ represents the feature-extraction backbone network that will generate a feature map, $X_f$, and $Stem(\cdot)$ is the common primary feature-extraction module. For different modal features, common stems use the same processing method, which leads to information loss. As shown in Fig 2, we offer a novel and effective variant (i.e., the IES) composed of three parts, where $S_\psi(\cdot)$ is the SEM with $\psi$, $C_\phi(\cdot)$ is the CEM with $\phi$, and $F_\omega(\cdot)$ is the FMM with $\omega$. The given input image, $X_{input} \in \mathbb{R}^{C\times H\times W}$, is first fed into SEM and CEM to extract structural and color features. We describe the highly concise process as follows:

$$\begin{aligned} IES(X_{input}) = F_\omega(Concat(X_{input}, \\ C_\phi(X_{input}), \\ S_\psi(X_{input}))) \end{aligned} \tag{2}$$




(a)Architecture (b)Image Embedding Stem

**Fig. 2:** Overall architecture. (a) A widely used structure with stem. (b) The macro structure with our proposed IES which contains SEM, CEM, and a feature-mixing block FMM.

where $Concat(\cdot)$ represents the concatenation operation. We combine two feature extraction branches in parallel [4], [3], noting that different features can be efficiently fused by multiple paths. These are fully integrated using the FMM.

The proposed IES explicitly expresses the characteristics of different modes. Thus, the essence of visual images is captured and characterized more effectively. This hypothesis is verified using ablation and contrast experiments, and the results are listed in in Table II. The training matrices are shown in Fig 4, and a detailed analysis of the results is provided in Section IV-D.

### B. CEM

Color is one of the most essential features of visual images. Our CEM is a special multilayer perceptron construct, as shown in Fig 3(b). A remarkable difference from previous methods is that ours does not use a convolution kernel size greater than one; instead, we apply a point-wise convolution to encode the red-green-blue (RGB) ternary color information, which ensures that the same color obtains the same embedded representation in all feature maps. By embedding three-dimensional (3D) information into a higher dimension, the model obtains a more comprehensive and richer data representation. Although humans cannot visualize this phenomenon, neural networks can learn it effectively. Furthermore, in our model, the BatchNorm (BN) is replaced with LayerNorm [48]. BatchNorm can potentially compromise the coding of color information, as it applies a specific calculation method when using point-wise convolutions.

$$C_{\phi}(X_{input}) = L_n(\cdots L_1(LN2d(Conv_{1\times1}(X_{input})))) \quad (3)$$

where $L_n$, $n \in \{1, 2, \ldots\}$ indicates the layers. When $n = 1$, $L_1$ is composed of the Gaussian error linear unit (GELU) activation function and a $1 \times 1$ convolution with 16 input channels. When $n \geq 2$, $L_n$ is based on the $L_1$ with $(n-1)$ $1\times$ 1 conversion and LN2d inserted before it. In our experiments, we used $n = 2$. Specifically, a single-layer CEM is called when $n = 1$, and a double-layer CEM is called when $n = 2$.

This phenomenon is illustrated in Fig 3(b). The former (I) has a steep change in the number of channels from 3 to 16 in our experiment, whereas the latter (II) is buffered by an intermediate layer, and its channels change from 3 to 8 to 16.

We conducted ablation experiments on several different structural designs, and the results are shown in Table. II. The training metrics are shown in Fig 4. A detailed analysis of the results is presented in IV-D.

### C. SEM

The texture structure embodies the detailed information of visual images and is an integral part of image features. The effective representation of rich features is a critical challenge in visual tasks [5], [49], [31]. Hence, our SEM first captures detailed multiscale texture information from the input image using three convolutions with different kernel sizes. To fully preserve the details, we used a concatenation operation to obtain feature map $X_{ms}$.

$$\begin{aligned} X_{ms} = Concat(&f_{k1}(X_{input}),\\ &f_{k2}(X_{input}), f_{k3}(X_{input})) \end{aligned} \quad (4)$$

where $f_{ki}$ is the 2D convolution with kernel size $ki$, $i \in \{3, 7, 11\}$. Subsequently, the feature map, $X_{ms}$, is sent to the feature-mixing block for feature fusion and interactions at different scales.

$$S_{\psi}(X_{input}) = Conv_{1\times1}(\zeta(BN(Conv_{1\times1}(X_{ms})))) \quad (5)$$

where $Conv_{1\times1}(\cdot)$ is the convolution with kernel size 1 and stride 1, $\zeta(\cdot)$ represents the GELU [50] activation function, and BN denotes BatchNorm.

### D. FMM

The FMM includes a $1 \times 1$ convolution and a BatchNorm. The common practical $1 \times 1$ convolution is used to mix the feature maps of different branches to preserve the feature information to the greatest extent.

$$\begin{aligned} F_{\omega}(X_{input}) = BN(Conv_{1\times1}(Concat(&X_{input},\\ &X_{ce}, X_{se}))) \end{aligned} \quad (6)$$

where $Concat(\cdot)$ denotes the concatenate operation. $X_{input}$ represents the input image, and $X_{ce}$ and $X_{se}$ are feature maps from the color and structure embedding modules, respectively. The final feature map is sent to the backbone for further processing.

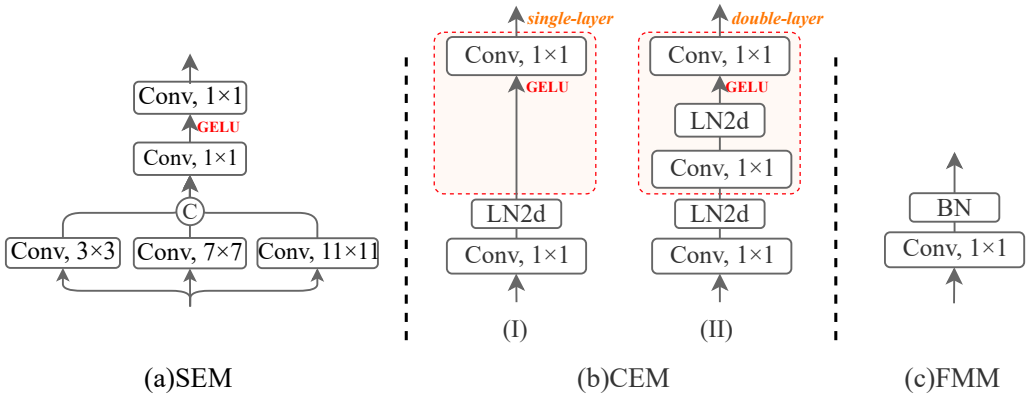



**Fig. 3:** Architecture of the proposed method. GELU [50] is the activation function, and LN2d is the special implementation of LayerNorm [48] without adjusting the order of dimension. $K \times K$ represents the kernel size.

## IV. Results And Discussion

This section presents the quantitative and qualitative experiments conducted to demonstrate the effectiveness and efficiency of the proposed IES. We conducted quantitative experiments on the ImageNet-100 [51] image classification dataset, UTFPR-SBD3 [52] semantic segmentation dataset, and CIHP [53] instance segmentation dataset. For the semantic and instance segmentation datasets, we used mIoU, mean accuracy (mAcc), and mean F-score (mFScore) to measure the performance of different baselines. Top-1 and Top-5 accuracies were applied to the classification datasets.

### A. Datasets

Imagenet-100 [51] is a subset of the ImageNet-1K dataset that includes 100 categories, each containing 1,300 images. The advantage of this dataset is that the number of categorical distributions is balanced, which avoids the long-tailed phenomenon. Owing to the limitations of hardware and parameters, and to quickly verify the performance of our plug-and-play IES module, we chose this dataset as our benchmark for image classification.

UTFPR-SBD3 [52] is a high-quality dataset intended for clothing segmentation tasks in the context of soft biometrics. It consists of 4,500 images manually annotated into 18 classes plus backgrounds, of which 1,003 are taken from the CCP dataset, 2,679 from the CFPD dataset, and 685 from the Fashionista dataset. Each class contains at least 100 instances, and all images were standardized to 400 × 600 pixels in RGB.

CIHP is a large-scale multi-person segmentation benchmark [53] that contains 38,280 images collected from real-world scenes, each with 19 semantic tags annotated at the pixel level. Each image contains three people on average, including challenging poses and viewpoints, severe occlusions, and wide resolutions. We divided this benchmark into three groups, 28,280 for training, 5,000 for verification, and 5,000 for testing.

### B. Experimental Details

We used PyTorch, timm [54], mmpretrain [55], and mmsegmentation [56] libraries for classification and segmentation. In the classification experiments, all baselines utilized the same data augmentation methods and were implemented using mmpretrain [55], and a mixed-precision training strategy was added to save memory and shorten the training period. In particular, to ensure the fairness of comparisons and validate the effectiveness of our proposed IES, the baselines used their original training settings and model configurations to control variables.

### C. Ablation Studies

All ablation experiments were performed on the UTFPR-SBD3 [52] dataset using the PoolFormer backbone [46]. This model is lightweight and efficient, which allowed us to perform faster verifications with limited resources. It also uses a single convolution and BatchNorm in the stem, thereby ensuring that the ablation test obtains more evidence to validate efficiency improvements. We found that each part of the IES contributed to the final performance, as listed in Table I.

**TABLE I:** Ablation study of different components. Params denotes the number of parameters, measured in millions (M).

| CEM | SEM | Params(M) | mIoU |
|---|---|---|---|
| ✗ | ✗ | 15.653 | 49.16 |
| ✓ | ✗ | 15.696 | 51.18(+2.02) |
| ✗ | ✓ | 15.719 | 51.21(+2.05) |
| ✓ | ✓ | 15.761 | 51.52(+2.36) |

As shown in the Table I, ✗ and ✓indicate disabling and enabling the module, respectively. The base model achieved a 49.16 mIoU when neither the CEM nor SEM was used. However, with both, the performance improved by 2.36 to a 51.52 mIoU. When only the CEM was used, the performance improved by 2.02 over the baseline to a 51.18 mIoU. Similarly, the SEM improved the base model by 2.05 to a 51.21 mIoU. The experimental results strongly indicate that each component plays an important role in improving performance. The next two sections explore the impacts of the CEM and SEM more closely .

*1) CEM Experiments:* Noting that the number of MLP-like CEM layers influences model representability, past studies have adopted two full connection layers in a feed-forward network [14], [5], [17]. Hence, we designed four groups of experiments to determine the extent to which the number of CEM layers affect network representability based

on the same 42 output channels (see Table II). The first experimental group comprised four tests corresponding to the various model layers, incrementing the index from one to four. The second group comprised three tests for diversifying the output channels based on previous results. The channels in the one-layer variant were changed from $C_{in}$ to $C_{out}$, referring to input and output channels, respectively. In the two-layer variant, they were changed from $C_{in}$ to $\frac{1}{2}C_{out}$ then to $C_{out}$. For the other variants, the changes followed a similar pattern.

**TABLE II:** Ablation experimental results of CEM with various layers and output channels.

| layers | dims | Params(M) | mIoU |
|---|---|---|---|
| 1 | 42 | 15.696 | 51.18 |
| 2 | 42 | 15.697 | 51.16 |
| 3 | 42 | 15.699 | 51.03 |
| 4 | 42 | 15.702 | 50.09 |
| 2 | 18 | 15.695 | 50.64 |
| 2 | 30 | 15.696 | 50.48 |
| 2 | 54 | 15.699 | 49.86 |

In Table II, layers and dims mean the number of layers and the output channels in CEM. There was no significant difference in the experimental results (51.16 and 51.18 mIoU) for the two- and one-layer CEMs, respectively. However, based on the training curve illustrated in Fig 4, the two-layer variant achieved faster adaptability. That is, under the premise of not reducing performance, the smooth channel layer transformation had a higher convergence than the steep one. which allowed the model to fit the data distribution faster. However, when the number of layers exceeded two, the model performance declined. In the final version, we chose a two-layer variant with a higher fitting ability as the basic CEM architecture.

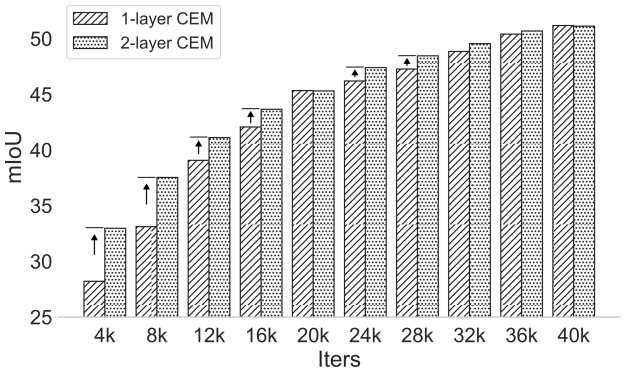



**Fig. 4:** Intermediate validation results for 1-layer and 2-layer CEM.

The number of feature map channels is crucial for model success; therefore, we further explored the influence of the final number of color-embedding channels on model representability. Based on the previous results, we adopted a two-layer architecture as the fundamental structure. Four groups of experiments were again performed in which the number of channels followed an arithmetic progression from 18, 30, 42, to 54 with interval steps of 12. From the results, the best performance occurred when the number of embedded channels was 42. With an increase in the number of color-embedding channels, the model performance first increased and then decreased. The mIoU ranged from 50.64 to 50.48 and 51.16 to 49.86. Because the CEM is located at the beginning of the full model, the gradient at this position has a larger range of change than the others, which can easily lead to gradient disappearance or explosions [20]. This instability makes training the CEM more difficult. Although BN and residual connections are helpful in resolving this problem, the changes are still noticeable.

**TABLE III:** Ablation experiment of various kernel size combinations and channels.

| index | kernels | dims | Params(M) | mIoU |
|---|---|---|---|---|
| $C_1$ | 3,7,11 | 32 | 15.707 | 50.69 |
| $C_2$ | 5,9,21 | 32 | 15.719 | **51.21** |
| $C_3$ | 5,11,21 | 32 | 15.720 | 51.16 |
| $C_4$ | 5,9,21 | 16 | 15.707 | 50.52 |
| $C_5$ | 5,9,21 | 48 | 15.734 | 51.01 |
| $C_6$ | 5,9,21 | 64 | 15.750 | 51.08 |

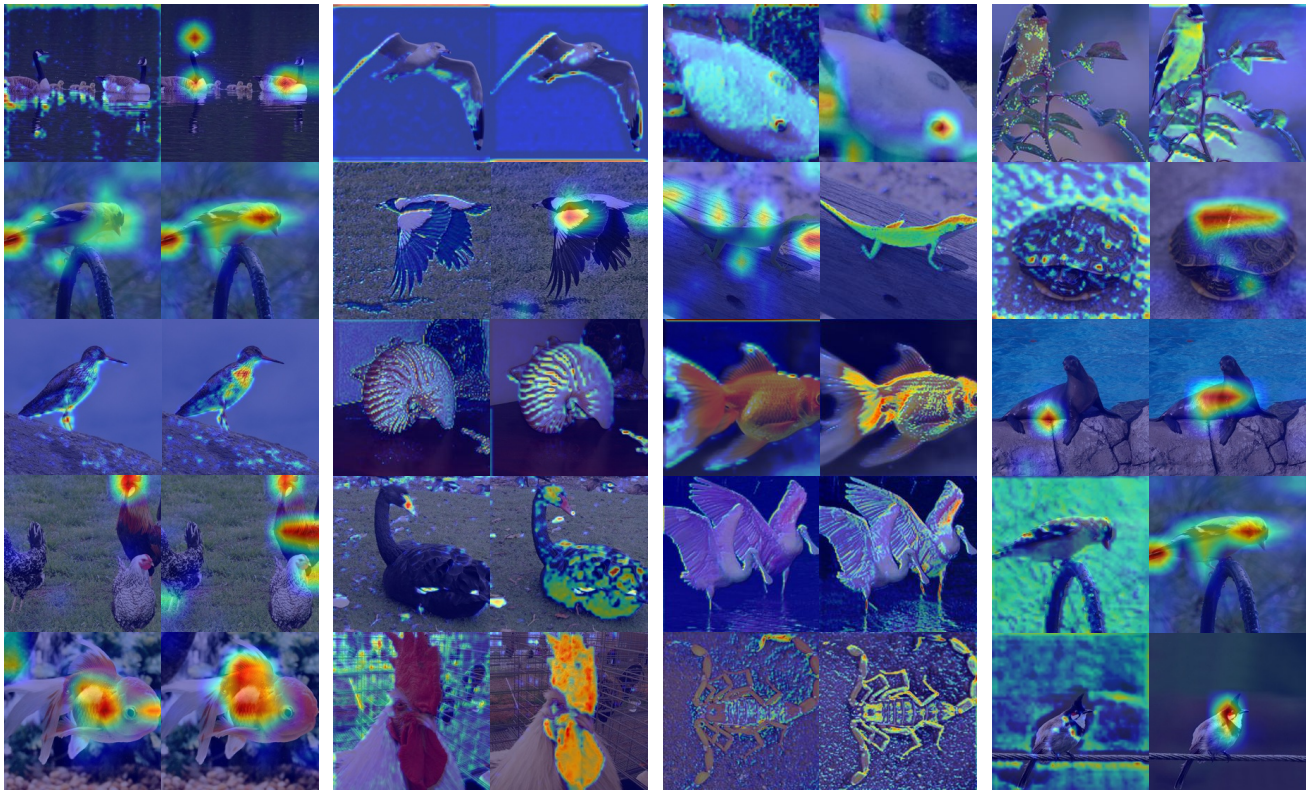
*2) SEM Experiments:* We extracted structural information of multiple scales through convolutions of different kernel sizes, including rich specific information. To determine the optimal combination and verify the effectiveness of our SEM, we conducted another set of ablation experiments.

In the upper part of Table III, mIoU is used as a measure of performance. Dims means the channels of output feature map. The index from $C_1$ to $C_3$ indicates different convolution kernel composition schemes: {3,7,11}, {5,9,21}, and {5,11,21}, respectively. This design was based on two considerations. First, we captured small-size detailed texture features using kernels sizes of three and five, medium-sized detailed features were captured using kernel sizes of seven and nine, and large-scale detailed texture features were captured with kernel sizes of 11 and 21. Second, according to the expressability of different convolution kernels, the feature extractors of the large, medium, and small receptive fields were cross-matched.

From the above considerations, three reasonable combinations were designed. Although larger kernel convolutions can increase model complexity, the small number of embedded channels mitigates these effects. The experimental results for the three convolution kernel combinations are summarized in Table III, where $C_2$ achieved the best performance with a 51.21 mIoU, and $C_3$ achieved a 51.16 mIoU, which was higher than that of $C_1$.

Based on the $C_2$ kernel size combinations, we conducted additional experiments to investigate the impact of varying the channel numbers on performance indices $C_4$ to $C_6$, as indicated in the lower part of Table III. In these experiments, we compared the results for channel numbers of 16, 32, 48, and 64, and the experimental results revealed the optimal performance at 32 channels.

Based on this analysis, we can conclude that a medium kernel size ($C_2$) provides the most effective feature extractor. When the number of output channels is 32, the generated feature maps exhibited the richest and most effective feature




(a) EdgeNext (b) TinyViT (c) MobileViT (d) VanillaNet

**Fig. 5:** Visualization of attention maps in the last norm layer of the different methods. It clearly shows that the SOTA baselines with IES can achieve better results than before.

information.

### *D. Comparisons with SOTA Methods*

*1) Attention Visualization:* To obtain a better and more intuitive understanding of the vital function of the stem and proposed IES, we further visualized the feature maps using GradCam [57], which utilizes the average gradient of the feature maps generated by the last layer in the original stem with respect to the specific class of our IES to generate a helpful visual explanation.

Fig 5 shows the visualization results for EdgeNext-xxs [58], TinyViT-5m [13], MobileViT-xxs [59], and VanillaNet [2] backbones. Red denotes high activations in the particular region and blue denotes weak network attention activations. We compared the results of the four methods. "w/ IES" indicates our proposed IES stem, and "w/o IES" indicates the original. As shown in Fig 5, our IES module captured more target feature information than the original in images with multiple targets, which benefited from the multiscale feature-extraction capability of the new SEM. The visualization results show that our proposed IES also had a strong characterization ability for textural details and played a beneficial role in images with obvious color characteristics. For example, it effectively represented the black color of a swan and the red color of a chicken, which is unusually precise for these types of models. From these results, we can clearly see that our proposed SEM and CEM play strong roles in improving overall model effectiveness.

*2) Image Classification:* We compared our method to extant SOTA methods on the ImageNet-100 classification dataset. For fairness and credibility of comparison, we performed two offsetting configurations. First, apart from the stem module, the two models shared the same code and settings. Second, the same data augmentation methods were adopted as the backbone for classification purposes. We adopted common data augmentation methods including RandomResizedCrop [23] with a crop size of 224, RandomFlip with a crop size of 0.5, RandAugment by Timm [54], and RandomErasing with a crop size of 0.25 [61]. We adopted the AdamW optimizer with an initial learning rate of 0.001, a weight decay of 0.05, betas of 0.9 and 0.999, and an eps of 1e-8. We trained all the baselines for 100 epochs using a batch size of 256, and a warm-up learning rate scheduler [20] was adopted for the first 10 epochs. The remaining epochs were adjusted using a cosine annealing scheduler. The results of the comparative trials of the ImageNet-100 classification experiments are summarized in Table IV. We can see that the baselines armed with our proposed IES achieved a very competitive performance. Different baselines had different numbers of embedded channels in their stems; therefore, the increased parameter quantity changed slightly after replacing the respective baseline with our IES.

To evaluate the effectiveness of the proposed IES on diverse model architectures, experiments were conducted using CNN-, transformer-, and CNN+transformer-based models. For the CNN-based model, our IES helped the VanillaNet [2] achieve a +1.52 top-1 accuracy at a computing cost

**TABLE IV:** Results on ImageNet-100 classification dataset. Type indicates the architecture of models. The notation $w/o$ indicates that the model does not adopt IES, while $w/$ signifies the opposite.

| Model | IES | Type | Params(M) | Top-1 Acc(%) |
|---|---|---|---|---|
| EMO[9] | $w/o$ | CNN | 5.097 | 84.78 |
| | $w/$ | | 5.146 | 86.03(+1.25) |
| EfficientFormer[60] | $w/o$ | CNN | 5.286 | 83.51 |
| | $w/$ | | 5.316 | 84.65(+1.14) |
| VanillaNet5[2] | $w/o$ | CNN | 17.649 | 81.20 |
| | $w/$ | | 17.780 | 82.72(+1.52) |
| EdgeNeXt[58] | $w/o$ | CNN-Trans | 1.175 | 81.14 |
| | $w/$ | | 1.205 | 81.60(+0.46) |
| TinyVit-5m[13] | $w/o$ | Trans | 5.104 | 84.98 |
| | $w/$ | | 5.133 | 85.5(+0.52) |
| TinyVit-11m[13] | $w/o$ | Trans | 10.593 | 85.48 |
| | $w/$ | | 10.618 | 85.80(+0.32) |
| MobileVit-xxs[59] | $w/o$ | CNN-Trans | 0.983 | 79.40 |
| | $w/$ | | 1.010 | 80.18(+0.78) |
| MobileVit-s[59] | $w/o$ | CNN-Trans | 5.002 | 84.56 |
| | $w/$ | | 5.025 | 84.98(+0.42) |

**TABLE V:** Comparison with state-of-the-art methods on CIHP segmentation dataset. SS means single-scale.

| Model | IES | Params(M) | mIoU(SS) | mAcc | mFscore |
|---|---|---|---|---|---|
| SegNeXt[5] | $w/o$ | 4.297 | 48.27 | 60.59 | 59.17 |
| | $w/$ | 4.318 | 50.13(+1.86) | 61.94(+1.35) | 61.08(+1.91) |
| ConvNeXtV2[21] | $w/o$ | 3.760 | 33.35 | 43.34 | 45.23 |
| | $w/$ | 3.796 | 35.14(+1.79) | 45.08(+1.74) | 47.28(+2.05) |
| SegFormer[62] | $w/o$ | 3.612 | 51.32 | 61.91 | 62.61 |
| | $w/$ | 3.634 | 52.70(+1.38) | 63.53(+1.62) | 64.12(+1.51) |
| HRNet[32] | $w/o$ | 9.641 | 59.40 | 71.11 | 72.68 |
| | $w/$ | 9.675 | 59.46(+0.06) | 71.69(+0.58) | 72.68(+0.00) |
| PoolFormer[46] | $w/o$ | 15.653 | 49.32 | 61.27 | 63.33 |
| | $w/$ | 15.761 | 51.06(+1.74) | 63.00(+1.73) | 65.13(+1.80) |
| VAN[31] | $w/o$ | 7.967 | 51.93 | 63.39 | 65.63 |
| | $w/$ | 8.030 | 53.27(+1.34) | 65.24(+1.85) | 66.89(+1.26) |
| MobileNetV3[26] | $w/o$ | 1.141 | 29.89 | 40.64 | 43.72 |
| | $w/$ | 1.162 | 30.32(+0.43) | 40.61(-0.03) | 44.65(+0.93) |

of 0.131M parameters. For the transformer-based TinyVit method [13], we performed a comparative test on two versions, 5M and 11M, respectively, achieving consistent performance improvements (85.5 vs. 84.98 and 85.8 vs. 85.48 top-1 accuracies). For the CNN+transformer-based approaches (i.e., EdgeNext-xxs [58] and MobileVit [59]), the former obtained a +0.46 top-1 accuracy with the addition of our IES, and the latter gained +0.78 and +0.42 values for $xxs$ and $s$ versions over the originals.

In conclusion, compared with all SOTA methods, the proposed IES significantly improved their performance. For example, Vaillanet [2] gained a +1.52 top-1 accuracy improvement. More importantly, the proposed IES exhibited stable performance for distinct baseline versions, demonstrating the excellent ability of the proposed IES to extract image features.

*3) Image Segmentation:* To further evaluate the generalizability of IES, we conducted experiments on the challenging CIHP and UTFPR-SBD3 datasets for instance and semantic segmentation tasks. We compared our method's results on SOTA ConvNeXtV2-atto [21], HRNet-18 [32], PoolFormer-s12 [46], VAN-b0 [31], SegFormer-B0 [62], SegNeXt-T [5], and MobileNet-V3 [26] baselines. We adopted the AdamW optimizer and set the initial learning rate to 0.0001, weight decay to 0.01, and betas to 0.9 and 0.999. For the CIHP and UTFPR-SBD3 datasets, images were cropped to 512 × 512 and 608 × 416, respectively. We trained the SOTA baselines for 96K iterations using their default settings and a batch size of eight. The warm-up scheduler [20] was adopted for the first 1.5K iterations, and the rest were adjusted using a

**TABLE VI:** Comparison with state-of-the-art networks on SBD3 validation set. The number of parameters is measured on the image size of $512 \times 512$.

| Model | IES | Params.(M) | mIoU(SS) | mAcc | mFscore |
|---|---|---|---|---|---|
| SegNeXt[5] | $w/o$ | 4.297 | 51.06 | 62.49 | 64.91 |
| | $w/$ | 4.318 | 52.67(+1.61) | 64.02(+1.53) | 66.61(+1.70) |
| ConvNeXtV2[21] | $w/o$ | 3.760 | 49.56 | 61.34 | 63.85 |
| | $w/$ | 3.796 | 51.26(+1.70) | 63.57(+2.23) | 65.30(+1.45) |
| SegFormer[62] | $w/o$ | 3.612 | 52.53 | 62.17 | 65.36 |
| | $w/$ | 3.634 | 54.03(+1.50) | 63.85(+1.68) | 67.01(+1.55) |
| HRNet[32] | $w/o$ | 9.641 | 59.70 | 71.74 | 73.18 |
| | $w/$ | 9.675 | 60.11(+0.41) | 71.92(+0.18) | 73.43(+0.25) |
| PoolFormer[46] | $w/o$ | 15.653 | 49.16 | 60.63 | 62.92 |
| | $w/$ | 15.761 | 51.52(+2.36) | 63.39(+2.76) | 65.75(+2.83) |
| VAN[31] | $w/o$ | 7.967 | 52.22 | 63.89 | 66.10 |
| | $w/$ | 8.030 | 53.07(+0.85) | 64.84(+0.95) | 66.53(+0.43) |
| MobileNetV3[26] | $w/o$ | 1.141 | 51.03 | 62.02 | 64.85 |
| | $w/$ | 1.162 | 51.87(+0.84) | 63.19(+1.17) | 65.92(+1.07) |

poly-learning rate decay policy, with its power set to 1.0. For a fair comparison, the training configurations remained the same except for the stem, which was replaced by the IES in the comparison models.

**TABLE VII:** Semantic segmentation on ADE20K validation set. The performance is measured by single-scale mIoU.

| Model | IES | Type | mIoU(SS) |
|---|---|---|---|
| Uniformer-S[63] | $w/o$ | CNN-Trans | 46.6 |
| | $w/$ | | 46.67(+0.07) |
| DeiT-S[64] | $w/o$ | Trans | 44.0 |
| | $w/$ | | 44.12(+0.12) |
| PVT-S[17] | $w/o$ | Trans | 39.8 |
| | $w/$ | | 40.01(+0.21) |
| MogaNet-S[4] | $w/o$ | CNN | 47.7 |
| | $w/$ | | 47.88 (+0.18) |
| SLaK-S[49] | $w/o$ | CNN | 49.4 |
| | $w/$ | | 49.55 (+0.15) |
| Swin-S[14] | $w/o$ | Trans | 41.5 |
| | $w/$ | | 41.67 (+0.17) |

On the CIHP dataset, the mIoU, mAcc, and mFscore were again used for the SOTA methods, as listed in Table V. In the experiments, the IES+ConvNextV2-atto gained a +1.79 mIoU (35.14 vs. 33.35), a +1.74 mAcc (45.08 vs. 43.34), and a +2.05 mFscore (47.28 vs. 45.23) over the ConvNextV2-atto. The IES+PoolFormer-s12, with similar parameters, surpassed the PoolFormer-s12 by 1.74 mIoU (51.06 vs. 49.32), by 1.73 mAcc (63.0 vs. 61.27), and by 1.8 mFscore (65.13 vs. 63.33). Moreover, the IES+VAN-b0 yielded a +1.34 mIoU improvement (53.27 vs. 51.93) over the original VAN-b0. The results showed that the SOTA methods achieved remarkable performance improvements with the addition of the proposed IES.

For the UTFPR-SBD3 dataset, as listed in Table VI, we again report the mIoU, mAcc, and mFscore. While maintaining similar parameters, the fpn-PoolFormer-s12 improved by 2.36 mIoU, by 2.76 mAcc, and by 2.83 mFscore. Similarly, ConvNeXtV2-atto [21] increased by 1.7 mIoU, 2.23 mAcc, and 1.45 mFscore. Similarly, MobileNet-V3 [26], VAN-b0 [31], and HRNet-18 [32] increased by 0.84, 0.85, and 0.41 mIoU, respectively. This reflects both excellent feature representability and superior generalizability for the IES in that the same SOTA models gained significant performance improvements with different datasets.

We also evaluate our IES with different methods on ADE20K [65]. The experimental results are presented in Table VII. In all the reference models, performance improvements were achieved when using IES.

## V. Conclusion

We present our novel IES stem, which consists of a CEM, SEM, and FMM. This combination of advancements significantly increased the performance of existing backbone methods while requiring fewer parameters. Preliminary feature extraction was realized by effectively fusing the color information captured by the CEM and the detailed texture information captured by the SEM, which provided higher-quality feature maps for feature extraction. Our experiments used various popular visual datasets to demonstrate and validate the effectiveness of the proposed IES. Based on the results, it is now possible to improve the performance of extant SOTA CNNs and transformers by improving their visual image representation baselines. We expect that this advancement will provide the groundwork for additional improvements and real-world implementation. In the future, we plan to continue perfecting our model for more challenging visual tasks, such as video classification and key point detection.

## ACKNOWLEDGMENT

Throughout the writing of this dissertation, we have received a great deal of support and assistance. In particular, we would like to thank Editage (www.editage.cn) for English language editing, which has significantly enhanced the clarity

and coherence of our manuscript. Their meticulous attention to detail and expertise in language refinement have been instrumental in ensuring the overall quality of our research work.

## REFERENCES


[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012. I, I-A

[2] H. Chen, Y. Wang, J. Guo, and D. Tao, "Vanillanet: the power of minimalism in deep learning," *Advances in Neural Information Processing Systems*, 2023. I, I-A, IV-D1, IV-D2, IV, IV-D2

[3] C. Si, W. Yu, P. Zhou, Y. Zhou, X. Wang, and S. Yan, "Inception transformer," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23495–23509, 2022. I, III-A

[4] S. Li, Z. Wang, Z. Liu, C. Tan, H. Lin, D. Wu, Z. Chen, J. Zheng, and S. Z. Li, "Efficient multi-order gated aggregation network," *International Conference on Learning Representations*, 2022. I, I-A, III-A, VII

[5] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-M. Hu, "Segnext: Rethinking convolutional attention design for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1140–1156, 2022. I, I-A, II, III-C, IV-C1, V, IV-D3, VI

[6] Y. Rao, W. Zhao, Y. Tang, J. Zhou, S. N. Lim, and J. Lu, "Hornet: Efficient high-order spatial interactions with recursive gated convolutions," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10353–10366, 2022. I

[7] L. Themyr, C. Rambour, N. Thome, T. Collins, and A. Hostettler, "Full contextual attention for multi-resolution transformers in semantic segmentation," pp. 3224–3233, 2023. I

[8] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," pp. 4804–4814, 2022. I, I-B, I-B

[9] J. Zhang, X. Li, J. Li, L. Liu, Z. Xue, B. Zhang, Z. Jiang, T. Huang, Y. Wang, and C. Wang, "Rethinking mobile block for efficient neural models," *Computer Vision and Pattern Recognition*, 2023. I, I-A, IV

[10] L. Fu, H. Tian, X. B. Zhai, P. Gao, and X. Peng, "Incepformer: Efficient inception transformer with pyramid pooling for semantic segmentation," *Computing Research Repository*, vol. abs/2212.03035, 2022. I

[11] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24261–24272, 2021. I

[12] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: Chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12021–12031, 2023. I, I-A

[13] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, and L. Yuan, "Tinyvit: Fast pretraining distillation for small vision transformers," in *European Conference on Computer Vision*, pp. 68–85, 2022. I, IV-D1, IV, IV-D2

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021. I, I-B, I-B, I-B, IV-C1, VII

[15] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12124–12134, 2022. I, I-B, I-B, I-B

[16] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 558–567, 2021. I, I-B

[17] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578, 2021. I, I-B, I-B, IV-C1, VII

[18] N. Tian and W. Zhao, "East: Extensible attentional self-learning transformer for medical image segmentation.," *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 1021–1030, 2023. I

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *International Conference on Learning Representations*, vol. abs/2010.11929, 2020. I, I-B, 1, II

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. I-A, I-A, II, II, 1, IV-C1, IV-D2, IV-D3

[21] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16133–16142, 2023. I-A, V, IV-D3, VI, IV-D3

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. I-A, 1

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015. I-A, II, IV-D2

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *Computing Research Repository*, 2017. I-A, I-A

[25] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022. I-A

[26] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019. I-A, V, IV-D3, VI, IV-D3

[27] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1580–1589, 2020. I-A

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, 2017. I-A

[29] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1492–1500, 2017. I-A, I-A

[30] W. Yu, P. Zhou, S. Yan, and X. Wang, "Inceptionnext: When inception meets convnext," *Computing Research Repository*, 2023. I-A

[31] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *Computational Visual Media*, vol. 9, no. 4, pp. 733–752, 2023. I-A, III-C, V, IV-D3, VI, IV-D3

[32] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020. I-A, V, IV-D3, VI, IV-D3

[33] C. Chen, B. Wu, and H. Zhang, "An image recognition technology based on deformable and cbam convolution resnet50.," *IAENG International Journal of Computer Science*, vol. 50, no. 1, pp. 274–281, 2023. I-A

[34] Y. Li, Q. Hou, Z. Zheng, M.-M. Cheng, J. Yang, and X. Li, "Large selective kernel network for remote sensing object detection," pp. 16748–16759, 2023. I-A

[35] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11534–11542, 2020. I-A

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018. I-A

[37] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," *British Machine Vision Conference*, 2018. I-A

[38] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 510–519, 2019. I-A

[39] Y. Ding and L. Wang, "Research on the application of improved attention mechanism in image classification and object detection.," *IAENG International Journal of Computer Science*, vol. 50, no. 4, pp. 1174–1182, 2023. I-A

[40] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018. I-A

[41] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019. I-A

[42] W. Xue, Y. Zhang, X. Wang, and S. Ge, "Msf-net: Multi-level semantic feature network extractor for paraphrase identification.," *IAENG Inter-*

*national Journal of Computer Science*, vol. 50, no. 4, pp. 1391–1400, 2023. I-B
[43] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," *Advances in Neural Information Processing Systems*, 2021. I-B
[44] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3965–3977, 2021. I-B
[45] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, 2021. I-B
[46] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10819–10829, 2022. I-B, IV-C, V, IV-D3, VI
[47] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2019. II
[48] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016. III-B, 3
[49] S. Liu, T. Chen, X. Chen, X. Chen, Q. Xiao, B. Wu, M. Pechenizkiy, D. Mocanu, and Z. Wang, "More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2022. III-C, VII
[50] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016. III-C, 3
[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Ieee, 2009. IV, IV-A
[52] A. D. S. Inacio and H. S. Lopes, "Epynet: Efficient pyramidal network for clothing segmentation," *IEEE Access*, vol. 8, pp. 187882–187892, 2020. IV, IV-A, IV-C
[53] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin, "Instance-level human parsing via part grouping network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 770–785, 2018. IV, IV-A
[54] R. Wightman *et al.*, "Pytorch image models," 2019. IV-B, IV-D2
[55] M. Contributors, "Openmmlab's pre-training toolbox and benchmark," 2023. IV-B, IV-B
[56] Contributors, MMSegmentation, "Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark," 2020. IV-B
[57] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017. IV-D1
[58] M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S. W. Zamir, R. M. Anwer, and F. Shahbaz Khan, "Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications," in *European Conference on Computer Vision*, pp. 3–20, Springer, 2022. IV-D1, IV, IV-D2
[59] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," *Transactions on Machine Learning Research*, 2022. IV-D1, IV, IV-D2
[60] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "Efficientformer: Vision transformers at mobilenet speed," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12934–12949, 2022. IV
[61] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13001–13008, 2020. IV-D2
[62] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12077–12090, 2021. V, IV-D3, VI
[63] K. Li, Y. Wang, J. Zhang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao, "Uniformer: Unifying convolution and self-attention for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. VII
[64] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*, pp. 10347–10357, PMLR, 2021. VII
[65] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, pp. 302–321, 2019. IV-D3

**Xiang Li** received the B.S. degree in Intelligent Science and Technology from Qingdao University. He is currently working toward the M.S. degree in Computer Science and Technology, Xi'an Polytechnic University. His research area is in the field of Image Processing, and Deep Learning.

**Xueqing Zhao** is currently an associate professor and tutor of the School of Computer Science of Xi'an Polytechnic University, and has been a visiting scholar to Peking University, Graz University of Technology in Austria, "Young Outstanding Talents" in Shaanxi Province Universities, "Young Talents Entrusted Talents" of Shaanxi Higher Education Science Association, a member of ACM, and the CCF Information System Committee. Her main research areas are intelligent computing, big data and blockchain.