Attentional Kernel Encoding Networks for Fine-Grained Visual Categorization

Yutao Hu, Yandan Yang, Jun Zhang, Xianbin Cao, *Senior Member, IEEE*, and Xiantong Zhen

***Abstract*— Fine-grained visual categorization aims to recognize objects from different sub-ordinate categories, which is a challenging task due to subtle visual differences between images. It is highly desired to identify discriminative regions while achieving highly non-linear compact representation for fine-grained visual categorization. However, existing methods either rely on manually defined part-based annotations to indicate the distinctive regions or operate on longitudinal vectors to capture the non-linear information, which may lose important spatial layout information. In this paper, we propose the Attentional Kernel Encoding Networks (AKEN) for fine-grained visual categorization. Specifically, the AKEN aggregates feature maps from the last convolutional layer of ConvNets to obtain a holistic feature representation. By Fourier embedding, it encodes features from both the longitudinal and transverse directions, which largely retains the spatial layout information. Moreover, we incorporate a Cascaded Attention (Cas-Attention) module to highlight local regions that distinguish among subordinate categories, enabling the AKEN to extract the most discriminative features. Working in conjunction with the attention mechanism, the proposed AKEN combines the strengths of ConvNets and kernels for non-linear feature learning, which can establish discriminative and descriptive feature representations for fine-grained image categorization. Experiments on three benchmark datasets show that the proposed AKEN delivers highly competitive performance, surpassing most existed methods and achieving state-of-the-art results.**

***Index Terms*— Fine-grained visual categorization, Kernel encoding, attention.**

I. Introduction

FINE-GRAINED visual categorization (FGVC) has attracted extensive research efforts in computer vision, and it aims to recognize objects from different subordinate categories, e.g., species of birds, models of cars, or different brands of planes. Nowadays, popular convolutional neural networks (ConvNets) have achieved great success in many tasks [1]–[6]. However, as for FGVC, it is still a challenging task and remains unsolved. The challenge is mainly caused by the subtle inter-class differences and large intra-class variations of fine-grained images [7]. On the one hand, as shown in Fig. 1, the three images in the first row belonging to the same subcategory called Least Auklet presenting different appearances due to views, poses and complex backgrounds. It is hard to recognize them even by human. On the other hand, images that belong to different subcategories may present similar appearance. For example, as shown in the second row of Fig. 1, the three images belong to three different subcategories: Brewer Blackbird, Rusty Blackbird and Red Winged Blackbird. However, they all have the black feather with only subtle differences. To take a closer look, we can find these three different birds have some unique parts with distinguishing colors on their wings. However, these differences are so subtle and even hard for a person without professional knowledge to figure out.

Fig. 1. Image examples from CUB-200-2011 dataset. The images in the first row are from the same subcategory, Least Auklet. The images in the second row are from three different subcategories, Brewer Blackbird, Rusty Blackbird and Red Winged Blackbird. Note that these images are even hard for human to recognize. Therefore, FGVC is a quite challenging task.

It is therefore usually desirable to localize the discriminative regions in the feature extraction. Previous work leveraged the manually defined part-based annotations to indicate the distinctive regions [8]. However, it is labor intensive to obtain manual annotations. Moreover, it is difficult even for a human to identify the really discriminative regions. Recently, the attention mechanism has been proposed and

Manuscript received September 18, 2019; revised November 23, 2019 and February 8, 2020; accepted February 27, 2020. Date of publication March 3, 2020; date of current version January 7, 2021. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 91738301 and Grant 61871016. This article was recommended by Associate Editor G.-J. Qi. *(Corresponding author: Xianbin Cao.)*

Yutao Hu and Yandan Yang are with the School of Electronics and Information Engineering, Beihang University, Beijing 100191, China.

Jun Zhang is with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China, and also with the National Engineering Laboratory for Comprehensive Transportation Big Data Application Technology, Beijing 100191, China.

Xianbin Cao is with the School of Electronics and Information Engineering, Beihang University, Beijing 100191, China, also with the Key Laboratory of Advanced Technologies for Near Space Information Systems, Ministry of Industry and Information Technology of China, Beihang University, Beijing 100191, China, and also with the Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beihang University, Beijing 100191, China (e-mail: xbcao@buaa.edu.cn).

Xiantong Zhen is with the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2020.2978115

shown convincing performance for extracting discriminative regions [9]–[11]. In FGVC, through the attention mechanism, we can highlight the regions that is significant for categorization. However, it is still a challenging problem to recognize the fine-grained image due to its subtle inter-class differences and large intra-class variations.

Moreover, in order to recognize those similar objects, a highly non-linear holistic representation is demanded. In fact, the importance of a high-quality discriminative feature representation is demonstrated in many previous work [12], [13]. In recent years, ConvNets have shown great effectiveness in extracting discriminative information from the image to produce a set of feature maps through a series of convolution operations. However, the holistic representation is simply obtained by average pooling of the feature maps from the last convolutional layer. These simple feature encoding methods discard crucial detailed information. Considering the challenges of FGVC, the feature maps produced by plain ConvNets are unsatisfactory for fine-grained image classification tasks.

To solve these problems, some encoding methods [12]–[17] have been proposed and achieved great successes. Encoding can be seen as an enhancement process that provides us more discriminative feature representation. A simple method called bilinear pooling [14] has produced state-of-the-art performance on a variety of fine-grained classification problems. The bilinear pooling collects second-order statistics of local features over a whole image to form a holistic representation for classification while higher-order statistics have also been explored in several vision tasks [13], [18], [19]. However, these encoding methods usually induce a high-dimensional holistic representation, which causes a heavy computational burden. Moreover, existing encoding methods mainly operate on longitudinal vectors, which do not fully capture the spatial layout information of images.

In this paper, we propose a novel deep learning architecture named Attentional Kernel Encoding Networks (AKEN) for fine-grained image categorization, which is illustrated in Fig. 2. The AKEN aggregates feature maps from the last convolutional layer of ConvNets into a holistic feature representation. Specifically, we propose applying the Fourier embedding to encode the feature maps into a holistic representation of images. By leveraging the strong non-linear learning ability of kernels, the Fourier embedding can capture more discriminative features for classification, which leads to a high-quality feature representation.

In contrary to previous encoding methods, we propose encoding along both longitudinal and transverse directions of the feature maps. As shown in Fig. 2, the two encoding modules are named Longitudinal Kernel Encoding and Transverse Kernel Encoding, respectively. The longitudinal vectors can be regarded as a batch of local feature vectors that describe the local response in each spatial location. If we regard each filter as a feature detector, the response in each feature map can be seen as the distribution of a special feature paradigm. Therefore, encoding from the longitudinal direction provides the feature paradigm in each spatial location. In the transverse direction, each feature map carries different aspects of features in the whole image. Encoding in this direction can obtain the

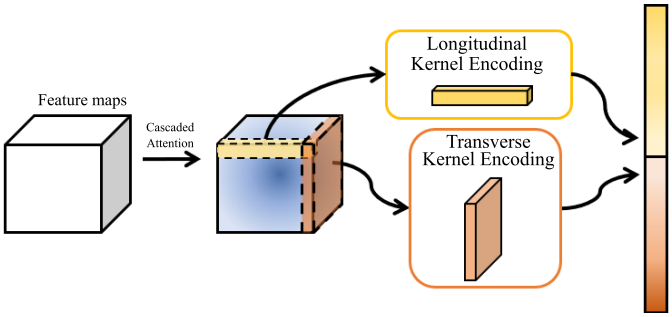


Fig. 2. This is the flow chart of our proposed Attentional Kernel Encoding Networks (AKEN). After feature extraction on the original image, we apply Cascaded Attention (Cas-Attention) module to highlight the discriminative regions. Then, we encode the feature maps to obtain the global feature representation. Differs from regular encoding methods that only concentrate on the longitudinal direction (Shown in yellow), we also encode in the transverse direction (Shown in orange). Finally, we concatenate two vectors produced by two directions encoding and get the final feature representation.

feature distribution of each specific feature paradigm in the original image, which is complementary to the information in the longitudinal direction. Therefore, our AKEN can well preserve these two sources of information that complements each other in classification. Additionally, the feature encoding via non-linear kernel encoding can be seamlessly injected into the convolutional learning architecture without forgoing the end-to-end training. More importantly, AKEN leverages the strength of ConvNets for feature extraction and kernels for non-linear learning, which can help to sufficient extract discriminative features from the input image.

Moreover, in order to extract features from the most discriminative regions, we introduce the attention mechanism before the feature encoding. Specifically, we design the Cascaded Attention (Cas-Attention) module to highlight the regions that reflect the visual differences among different categories. Additionally, we incorporate the residual learning strategy into the attention module. More importantly, we also incorporate a cascaded structure. It generates attention maps from different scales of receptive field and locates the discriminative features with various scales. This multi-scale mechanism has been proved effective in feature extraction [20]–[22] and also plays an important role in our AKEN.

In conjunction with the attention mechanism, the proposed attentional kernel encoding network can not only identify the most discriminative features but also achieves the high non-linearity while compacting holistic representations for fine-grained image categorization. The main contributions of this paper can be summarized as in the following three points:

- We propose a new learning architecture, called Attentional Kernel Encoding Networks (AKEN), for fine-grained image categorization. It combines the respective advantages of convolutional neural networks and kernels for feature extraction and non-linear learning.
- We propose encoding feature maps from both longitudinal and transverse directions, which not only capture local discriminative features but also preserve spatial layout information, resulting in comprehensive representations.
- We design the Cas-Attention module to highlight significant regions in the feature maps, which enables the

network to encode the most discriminative features in different scales.

To evaluate the effectiveness of the proposed AKEN for fine-grained image classification, we conduct extensive experiments on three commonly-used benchmark datasets. The results have shown that our AKEN can achieve high performance that is competitive and even better than state-of-the-art methods. We also provide in-depth ablation studies to verify each component of our AKEN.

The rest of this paper is organized as follows. Sec.II reviews the related work. Sec.III introduces our AKEN in detail. Sec.IV presents the experimental results on three fine-grained datasets. At last, we conclude all of the paper in Sec.V.

II. Related Work

In this part, we will introduce some existing deep learning based methods for fine-grained image classification from three parts. First, we will introduce annotation based methods for FGVC in Sec.II-A. Second, we will present the attention based methods in Sec.II-B. Finally, we will review the encoding based methods in the recent literature in Sec.II-C.

A. Annotation Based Method

Since the differences among different sub-categories in FGVC are subtle and often exist in some specific local regions, a natural idea to improve the classification accuracy is finding and locating these significant regions. To achieve this goal, utilizing manual object part annotations directly is a straightforward way [8], [23], [24]. They can locate and crop the discriminative regions with the help of manmade bounding boxes to outstand them in the process of classification. These methods achieve impressive results and show the effectiveness of part localization. However, to obtain these annotations is quite costly, which requires a lot of time and workforce. This characteristic restricts the application of these methods. Therefore, we need a more flexible mechanism to reduce the heavy manual labeling work.

B. Attention Based Method

As mentioned in Sec.II-A, considering the difficulties faced in the annotation based methods, we need a more flexible mechanism to figure out and locate each discriminative regions. In fact, in ConvNets, different convolutional filters can be regarded as different feature detectors. Therefore, responses of different filters often indicate different feature regions, which usually indicate different parts in images. Inspired by this observation, the attention mechanism has been proposed in recent years. It can focus on a specific location and enhance representations of objects there. Meanwhile, attention mechanism can be easily inserted into ConvNets without losing the end-to-end architecture. Based on these advantages, attention mechanism has achieved great success in a broad range of visual tasks, e.g. image classification [25], video classification [9], [10], [26], image retrieval [27], person re-identification [28] and image segmentation [29].

In FGVC, many methods take advantage of the attention mechanism to highlight discriminative regions and reduce the heavy labeling work. Xiao *et al*. [30] proposed a two-level attention model to avoid the use of annotations. In their method, one object-level attention model selects relevant patches, and the part-level attention localizes discriminative parts. Liu *et al*. [31] presented a fully convolutional attention architecture to extract significant local regions. Jaderberg *et al*. [32] proposed a dynamic mechanism to actively transform an image according to the transverse information, including spatial correlation, expected appearance. Fu *et al*. [33] proposed a novel recurrent attention convolutional neural network (RA-CNN) to recursively locate the discriminative regions of original images. RA-CNN provides a series of sub-images at multiple scales and zooms in the important regions of images for classification. In [34], DFL-CNN captures the discriminative regions for different classes in an end-to-end manner. It regards a 1×1 convolutional filter as a feature detector, based on which the class-specific discriminative patches is localized. In [35], Sun *et al*. proposed the multi-attention multi-class constraint (MAMC). It designs multiple attention branches with a metric learning framework to locate different discriminative parts, which proves the effectiveness of taking the attention mechanism as a feature detector. In [36], TASN generates the attention map through self-trilinear product, and then produces a detail-preserved image and structure-preserved image through sampling that is guided by the attention map. In the sampling process, the discriminative regions will be highlighted with high resolution. Compared to the "crop" and "zoom-in" operation in [33], the highlight process in [36] is more flexible. All these attention methods have achieved great success in FGVC. In our AKEN, we also utilize the attention mechanism and design our Cas-Attention module to highlight the discriminative regions.

C. Encoding Based Method

As discussed in Sec.I, FGVC is quite a challenging task, which calls for a highly non-linear holistic feature representation. A Fine-grained Dictionary Learning (FDL) for image classification was proposed in [12]. It uses three hybrid dictionaries to encode the image and obtain a discriminative feature representation. The experimental results on fine-grained datasets show the powerful effects of FDL, which also demonstrates the importance of a high-quality feature representation for FGVC. Therefore, to get better performance, more and more concerns are given to obtain a discriminative feature representation. In recent years, ConvNets have been proven a powerful feature extractor that can provide discriminative features and achieved great success in many vision tasks. However, they are not enough for FGVC due to its extreme difficulties. Therefore, we urgently need an enhancement process to boost the discrimination of features.

Adding the encoding module after the convolutional layers in ConvNets has been proven an effective way to achieve this goal. The encoding module can fuse the feature maps and finally get a highly non-linearly holistic feature representation. Bilinear pooling is a classical encoding method. It was first

proposed by Tanenbaum and Freeman [37] to model two-factor variations. Lin *et al*. [14] introduced Bilinear pooling into ConvNets as a pooling layer to learn the channel's correlation, which uses element-wise square root normalization followed by L_2 normalization to normalize the covariance matrix. They took one step further and improved the original bilinear pooling by the novel matrix square root normalization in [15]. Additionally, in [38], Yu *et al*. developed Hierarchical Bilinear Pooling (HBP) network based on cross-layer bilinear modules to fuse the information from intermediate layers in ConvNets. However, bilinear pooling methods mentioned above have two drawbacks. On the one hand, bilinear pooling needs to operate the outer product, which is expensive in terms of computation. To solve this problem, two novel compact bilinear representations were proposed in [39]. They are produced by Random Maclaurin (RM) and Tensor Sketch (TS) projection respectively to approximate the outer product, which achieve a faster computing speed with no loss of classification accuracy. On the other hand, bilinear pooling only gathers 2nd order information of the input feature maps. It loses the 1st order information, which is believed to be important in holistic feature representation. To settle this problem, Wang *et al*. [16] proposed G^2DeNet with Gaussian embedding that combines 1st order information with 2nd order information together. However, G^2DeNet maps a Gaussian to a square rooted symmetric positive definite (SPD) matrix in a high dimension, which leads to extremely high dimensionality of the encoded features. Gou *et al*. [17] improved it by constructing a homogenous mapping (HM) layer to decompose the tensor product operator. Then, a compact fashion can be introduced to reduce the dimensionality with some mathematical tricks. Specifically, in [13], Cai *et al*. proposed a method to approximate polynomial kernel in CNNs. It encodes feature maps concatenated from the output of different convolutional layers and project them to a high-dimension space. We think our AKEN and [13] have the close inspiration that uses the non-linear learning ability of kernel to enhance the feature maps. The difference is that we use Fourier embedding to approximate the shift invariant kernel.

III. Our Method

In this section, we present our proposed Attentional Kernel Encoding Network (AKEN), as illustrated in Fig. 3. The end-to-end framework AKEN is composed of three parts:

1) A basic feature extraction module that extracts features from original input images, introduced in Sec. III-A.

2) An attention module to help the network focus on informative regions of features, introduced in Sec. III-B.

3) Two parallel kernel encoding modules that consider longitudinal and transverse information, respectively, explained in Sec. III-C.

At last, we explain how we integrate these three parts together and present the whole architecture of our AKEN in Sec. III-D.

A. Convolutional Module

Our AKEN does not rely on any specific convolutional architectures for feature extraction. We deploy the common convolutional neural network as the backbone for computational efficiency. Specifically, we keep the convolutional module of the original neural network and remove the remaining parts. The output of this module is a feature map in $N \times C \times W \times H$ dimensions, where N represents the batch size, W, H and C indicate the width, height and the number of channels of the feature maps respectively. Since fine-grained datasets are in relatively small scales, we pre-train the backbone on ImageNet in order to obtain a better parameter initialization.

B. Cascaded Attention Module

In order to highlight the regions that are discriminative for classification, we design a Cas-Attention module to refine the feature maps before encoding. Specifically, as shown in the Cas-Attention module in Fig. 3, given the final feature maps output from the ConvNets, denoted as X, we generate two 3D attention maps $M_1(X)$ and $M_2(X)$ through two basic branches. Each basic branch produces the attention maps with a specific filter size through residual strategy. By the cascade connection of these basic branches, we obtain the multi-scale information in the attention module. Meanwhile, the obtained 3D attention maps are the same size as X. The 3D attention maps ensure that each pixel has its own corresponding weight.

We first generate $M_1(X)$ by using a 1×1 convolutional operation from the original feature maps. Then, we impose the attention map as a 3D mask on the output feature map using the element-wise multiplication. Instead of directly feeding attenuated feature maps $M_1(X) \otimes X$ as the input to the next layer, we borrow the idea of the residual learning and use element-wise summation.

We then expand the receptive field and generate $M_2(X)$ by using a 3×3 convolutional operation from the original feature maps. With the padding process, the 3D attention maps are also the same size as X. Then, we impose $M_2(X)$ as a 3D mask on the last output feature map using the element-wise multiplication. We merge it with original feature maps using element-wise summation.

The overall process can be presented as:

$$\tilde{X} = X + (X + X \otimes M_1(X)) \otimes M_2(X), \tag{1}$$

where $\otimes$ denotes element-wise multiplication. Since single attention only looks into the feature maps on one scale, it will ignore vital information in larger scale. Considering this, our Cas-Attention module highlights discriminative regions through two attention maps produced by two basic branches. Meanwhile, the residual strategy helps the gradient's propagation in back propagation [25] and leads to the better performance. The visualization results in Sec. IV-D show the great effectiveness of our Cas-Attention module for highlighting discriminative regions.

C. Kernel Encoding Module

We first provide a brief introduction of the kernel method and discuss kernel approximation with some theories in Sec.III-C1. Based on the given theories, we design a directional kernel encoding module, combining longitudinal kernel encoding and transverse kernel encoding, as explained in Sec.III-C2 and Sec.III-C3.

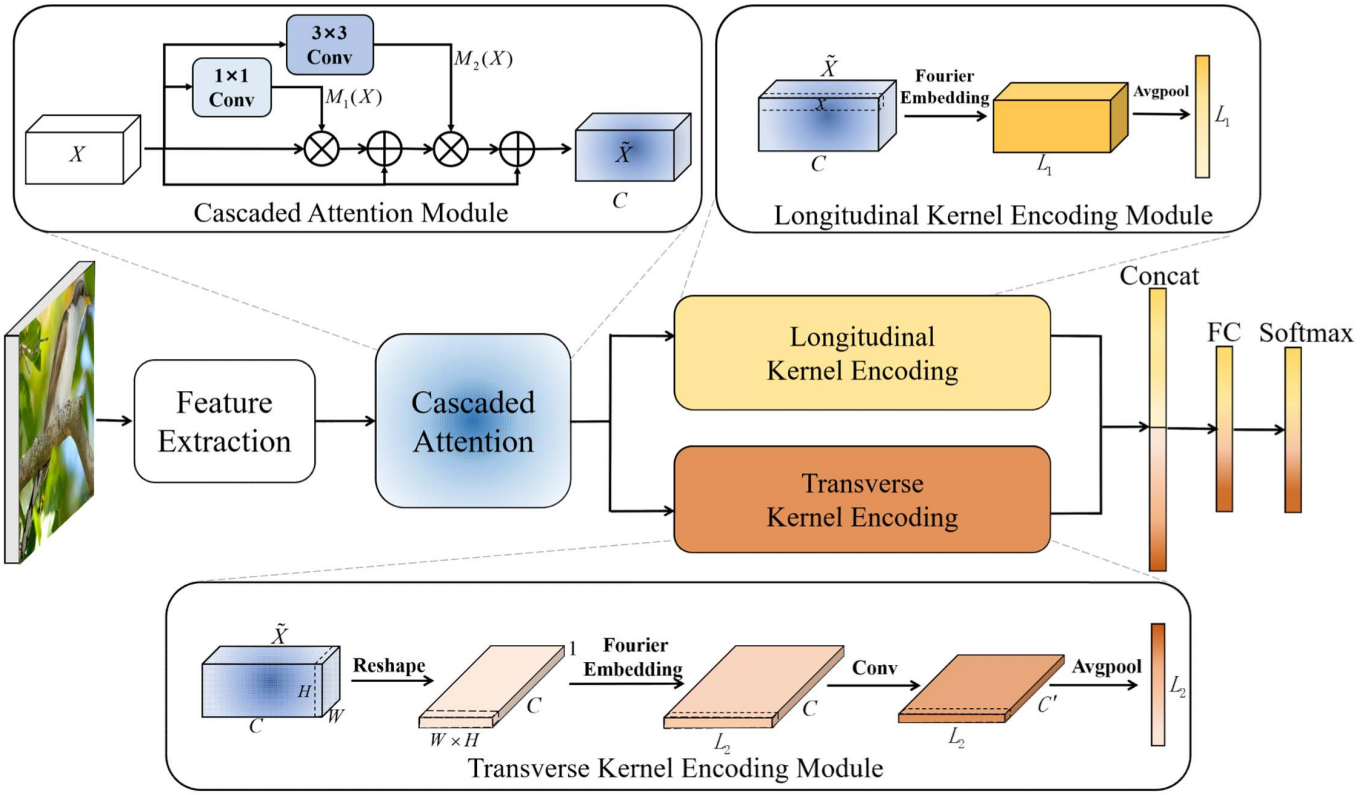


Fig. 3. **Diagram of AKEN**. It mainly contains three parts: feature extraction, Cas-Attention module, and directional kernel encoding module considering both longitudinal direction and transverse direction. We also illustrate the details of some modules in this diagram. As we can see, in the Transverse Kernel Encoding, we first reshape the feature map and exchange the dimension of transverse and longitude. Then we operate convolution and Fourier embedding on the reshaped feature map to encode the transverse points.

1) Fourier Embedding: Kernel methods have been widely explored in machine learning, showing high effectiveness in learning non-linearity in data. The great power of kernels has not been well explored in the scenarios of convolutional neural networks. Hereby, we introduce kernels into neural networks via Fourier transformation to encode feature maps into a holistic representation.

A kernel is a function that takes two input vectors in the original space and returns the dot product of the vectors. Formally, given input data $\mathbf{x}, \mathbf{y} \in \mathbf{X}$, and a mapping function $\phi(\cdot)$: $\mathbf{X} \rightarrow \mathcal{R}^N$, and a kernel function can be represented as

$$k(\mathbf{x}, \mathbf{y}) = < \phi(\mathbf{x}), \phi(\mathbf{y}) > \tag{2}$$

For a simple circumstance, it is easy to obtain lifting $\phi(\cdot)$, such as the linear transform. However, to obtain the non-linearity information, $\phi(\cdot)$ is non-linear and of high, even infinite dimensionality in most situations, which makes it hard to find the exact representation. To deal with this problem, the kernel method provides a shortcut, which skips the process of $\phi(\mathbf{x}), \phi(\mathbf{y})$ and calculates the kernel function $k(\mathbf{x}, \mathbf{y})$ directly. Kernel machines, e.g., the support vector machine, take advantage of the kernel method, where the inner product between lifted data points can be computed as $k(\mathbf{x}, \mathbf{y})$. However, the cost of this mechanism is that algorithms access the data only through evaluations of $k(\mathbf{x}, \mathbf{y})$ between every data pair, which brings large computation and storage costs when the training set is large.

Recently, kernel approximation has attracted increasing attention, and it is used to explicitly map the data to a low-dimensional inner product space using a randomized feature map $\mathbf{z}$: $\mathcal{R}^d \rightarrow \mathcal{R}^L$ such that

$$k(\mathbf{x}, \mathbf{y}) = < \phi(\mathbf{x}), \phi(\mathbf{y}) > \approx z(\mathbf{x})^T z(\mathbf{y}), \tag{3}$$

where L is the dimensionality of the approximated inner product space.

One of the most widely used approaches to kernel approximation is the one based on random Fourier features, which is derived from Bochner's theorem.

Theorem 1 (Bochner [40]): A continuous function $g : \mathbb{R}^d \rightarrow \mathbb{C}$ is positive definite on $\mathbb{R}^d$ only if it is the Fourier transformation of a finite non-negative Borel measurement $\mu(\boldsymbol{\omega})$ on $\mathbb{R}^d$, i.e.,

$$g(\mathbf{x}) = \int_{\mathbb{R}^d} e^{-j\boldsymbol{\omega}^\top \mathbf{x}} d\mu(\boldsymbol{\omega}), \quad \forall \mathbf{x} \in \mathbb{R}^d, \tag{4}$$

where j denotes the imaginary unit.

Proposition 1: For shift invariant kernel $k(\mathbf{x} - \mathbf{y})$, Random Fourier Features of $\mathbf{x} \in R^d$ can be represented as

$$z(\mathbf{x}) = \frac{\sqrt{2}}{\sqrt{L}} [\cos(\omega_i^\top \mathbf{x} + b_i)]^L \in R^L, \tag{5}$$

where ω is sampled from Fourier transform of $k(\mathbf{x}-\mathbf{y})$ and b_i is drawn uniformly from $[0, 2\pi]$ [41].
The proof is given in the Appendix A. Proposition 1 ensures that the expectation of $z(\mathbf{x})^T z(\mathbf{y})$ is equal to $k(\mathbf{x}, \mathbf{y})$. However, this proposition does not ensure the convergence of the adopted kernel approximation and we therefore provide the following proposition to theoretically guarantee the convergence.

Proposition 2 (Convergence): For the feature mapping $z(\cdot)$ in proposition 1, $z(\mathbf{x})^\top z(\mathbf{y})$ converges to $k(\mathbf{x}, \mathbf{y})$, where larger L leads to faster convergence. Specifically,

$$Pr[|z(\mathbf{x})^\top z(\mathbf{y}) - k(\mathbf{x}, \mathbf{y})| \geq \epsilon] \leq 2\exp(-L\epsilon^2/8). \quad (6)$$

The proof is given in the Appendix B. Proposition 6 ensures the convergence of the approximation. With these two propositions, we can decouple the kernel into the inner product of $z(\mathbf{x})$ and $z(\mathbf{y})$, and regard $z(\cdot)$ as the replacement of the original lifting $\phi(\cdot)$.

However, Proposition 6 only guarantees an unbiased estimate when $L \to \infty$, which means that the computational overhead and storage cost go up greatly with the increase of dimensionality. Fortunately, implementing kernel approximation in ConvNets solves the problem. ConvNets are fully data-driven, which dynamically adjust parameters towards the minimum bias during the training process to reduce the gap between kernels and their corresponding approximations. From this perspective, integrating kernel approximation into ConvNets can alleviate the problem of computation and storage burden, without forgoing end-to-end training.

As discussed before, we need a holistic representation that can distinguish different sub-ordinate categories in the fine-grained task. Technically, we need to transform the feature maps refined by the attention module into a single vector. Thus we propose applying the Fourier embedding derived from Bochner's theorem in ConvNets to encode the feature maps into a holistic representation of images, by leveraging the strong non-linear learning ability of kernels.

With the motivations above, we start to design our encoding module. In order to extract features from different perspectives of views, we implement Fourier embedding in two directions. From a longitudinal perspective, the feature maps are a batch of local feature vectors that describe the local response associated with each spatial location. From the transverse perspective, each feature map carries a certain aspect of the whole image, if we regard each filter as a feature detector. Therefore, to well preserve these two sources of information that are complementary to each other in the feature maps, we apply the kernel encoding along both longitudinal and transverse directions.

2) Longitudinal Kernel Encoding: As shown in longitudinal kernel encoding module in Fig. 3, $\tilde{X} \in R^{W\times H\times C}$ contains the input feature maps, and each longitudinal vector extracted from $\tilde{X}$ is denoted as $\mathbf{x}_i \in R^C$, where $1 \leq i \leq W \times H$. According to Proposition 1, we now construct $z(\mathbf{x}_i)$ by the Fourier embedding, resulting in

$$z(\mathbf{x}_i) = \cos(W^\top \mathbf{x}_i + \mathbf{b}), \quad (7)$$

where $W \in \mathbb{R}^{C\times L_1}$ contains weight parameters that are trainable. We initially sample W from a Gaussian distribution, which is the Fourier transform of a Gaussian kernel, as a typical shift invariant kernel. The bias parameter is initialized by drawing uniformly from $[0, 2\pi]$. This turns out to be a non-linear layer with cosine activations, which can be seamlessly injected into the neural network without forgoing end-to-end training. After embedding all longitudinal vectors into a more compact lower-dimensional space, we aggregate them into a L_1-dimension single feature vector v_L by average pooling.

3) Transverse Kernel Encoding: The longitudinal vector contains local semantic features without spatial information. So we design the transverse kernel encoding module for aggregating the whole map in each channel. As shown in the transverse kernel encoding module in Fig. 3, we first reshape each feature map $\tilde{X}_j \in \mathbb{R}^{W\times H}$ in $\tilde{X}$ to a vector $\mathbf{y}_j$ of the length $W \times H$, where $1 \leq j \leq C$. Then, similarly to the longitudinal encoding, we apply Proposition 1 and embed $\mathbf{y}_i$ into a lower-dimensional space, resulting in

$$z(\mathbf{y}_j) = \cos(W^\top \mathbf{y}_j + \mathbf{b}), \quad (8)$$

where $W \in \mathbb{R}^{WH\times L_2}$ contains the parameters to be learned during the training stage. After embedding, we additionally employ a convolution layer to shrink the feature maps. Then we apply the average pooling to aggregate them into a L_2-dimension single feature vector v_T.

D. Integration

With the three modules introduced above, we obtain the entire framework, shown in Fig. 3. The convolutional module first extracts features from the input image and generates basic feature maps. Then, the Cas-Attention module highlights the discriminative regions of the previous feature maps. After this, the feature maps go through both longitudinal and transverse kernel encoding modules separately. The longitudinal kernel encoding module outputs a vector in the length of L_1, while the transverse kernel encoding module outputs a vector in the length of L_2. Then we concatenate these two vectors together into a vector in the dimension of $L_1 + L_2$. Finally, we employ a fully connected layer with a softmax operation to get the probability distribution for classification.

IV. Experiments

To evaluate the effectiveness of the proposed AKEN, we conduct experiments on three benchmark datasets, including CUB-200-2011 (Birds) [42], FGVC-Aircraft (Aircraft) [43], Stanford Cars (Cars) [44]. The experimental results have shown that AKEN achieves compromising performance on the widely used fine-grained recognition datasets. We have also conducted extensive ablation studies to show the effectiveness of the proposed attention module and direction kernel encoding module.

A. Datasets

We provide a brief description of the datasets used in our experiments associated with split settings and the statistics are

TABLE I

THE SPLIT OF THE DATASETS WE EMPLOY TO EVALUATE OUR NETWORK

Datasets	Class	Train	Test
CUB-200-2011 [42]	200	5994	5794
FGVC-Aircraft [43]	100	6667	3333
Stanford Cars [44]	196	8144	8041

Fig. 4. Here are a few samples of the datasets we use. The images from top to bottom are from CUB-200-2011, FGVC-Aircraft and Stanford Cars.

listed in Table I. We also show some samples of each dataset in Fig. 4.

1) CUB-200-2011 [42]: CUB-200-2011 consists of 11788 bird images from 200 bird categories. It is one of the most challenging fine-grained datasets due to the great similarities among different categories and huge variations in pose and viewpoints in each category. We use 5994 images for training and 5794 for testing, just like the split in [42].

FGVC-Aircraft

2) FGVC-Aircraft [43]: dataset contains 10000 images from 100 different aircraft models, giving 100 images for each model. Different models differ in appearance and structure. We adopt the training/testing split like [43] with 6667 for training and 3333 for testing.

3) Stanford Cars [44]: Stanford Cars contains 16185 images with 196 different car classes. We adopt the same split provided by [44] to perform our experiments, which use 8144 images for training and 8041 images for testing.

TABLE II

ACCURACY COMPARISON WITH DIFFERENT BACKBONES ON THREE CHALLENGING FINE-GRAINED DATASETS

Backbone	Birds	Aircraft	Cars
ResNet-50 [45]	86.2	93.3	92.6
ResNet-101 [45]	86.8	93.7	93.0
VGG-16 [46]	86.5	93.5	92.8
VGG-19[46]	**87.1**	**94.0**	**93.9**

TABLE III

ACCURACY COMPARISON WITH CLASSICAL METHODS ON CUB-200-2011 DATASET

Method	Anno.	Backbone	Accuracy
VGG-16 [46]		VGG-16	75.4
VGG-19 [46]		VGG-19	77.8
ResNet-50 [45]		ResNet-50	81.7
ResNet-101 [45]		ResNet-101	84.5
PA-CNN [47]	√	VGG	82.8
FCAN [31]	√	ResNet-50	84.7
SPDA-CNN [48]	√	VGG	85.1
B-CNN [14]	√	VGG-D,M	85.1
PN-CNN [49]	√	AlexNet	85.4
STN [32]		Inception	84.1
RA-CNN [33]		VGG-19	85.3
MA-CNN [50]		VGG-19	86.5
MAMC [35]		ResNet-101	86.5
DFL-CNN [34]		ResNet-50	**87.4**
B-CNN [14]		VGG-D,M	84.1
iB-CNN [15]		VGG-D	85.8
CB-CNN [39]		VGG-D	84.0
HIHCA [13]		VGG-16	85.3
MoNet [17]		VGG-16	86.4
DeepKSPD [51]		VGG-16	86.5
G^2DeNet [16]		VGG-16	87.1
PC [52]		DenseNet-161	86.8
HBP [38]		VGG-16	87.1
AKEN (ours)		VGG-19	87.1

B. Implementation Details

Our module can be inserted into many existing convolutional neural networks, such as AlexNet, VGGNet and ResNet. We compare different networks in Table II. According to the results, we finally choose VGG19 as the backbone of our AKEN. The VGG19 network is pretrained on the ImageNet classification dataset. We remove the last three fully-connected layers and insert our components in the framework. For the kernel encoding module, we set L_1, L_2 to be equal to 512, and we initialize the parameters by sampling from a Gaussian distribution. Note the initialization is corresponding to Proposition 1 in Sec. III-C1 and the Gaussian function is the Fourier transform of a Gaussian kernel. Parameters in each layer are all trainable. We train all the networks using a stochastic gradient descent with a batch size of 32, and the momentum of 0.9. The learning rate is initialized with 0.01 and is then annealed by 0.1 every 20 epochs. Meanwhile, we limit the minimum of learning rate not less than 0.0001. In other words,

TABLE IV
ACCURACY COMPARISON WITH CLASSICAL METHODS ON FGVC-AIRCRAFT DATASET

Method	Anno.	Backbone	Accuracy
VGG-16 [46]		VGG-16	82.4
VGG-19 [46]		VGG-19	84.8
ResNet-50 [45]		ResNet-50	88.5
ResNet-101 [45]		ResNet-101	90.3
MG-CNN [53]	√	VGG-19	86.6
BoT [54]	√	VGG-16	88.4
RA-CNN [33]		VGG-19	88.2
MA-CNN [50]		VGG-19	89.9
DFL-CNN [34]		VGG-16	92.0
B-CNN [14]		VGG-D	84.1
iB-CNN [15]		VGG-D	88.5
HIHCA [13]		VGG-16	88.3
MoNet [17]		VGG-16	89.3
DeepKSPD [51]		VGG-16	91.5
G^2DeNet [16]		VGG-16	89.0
PC [52]		DenseNet-161	89.2
HBP [38]		VGG	90.3
AKEN (ours)		VGG-19	**94.0**

TABLE V
ACCURACY COMPARISON WITH CLASSICAL METHODS ON STANFORD CARS DATASET

Method	Anno.	Backbone	Accuracy
VGG-16 [46]		VGG-16	82.2
VGG-19 [46]		VGG-19	84.9
ResNet-50 [45]		ResNet-50	89.8
ResNet-101 [45]		ResNet-101	91.9
FCAN [31]	√	ResNet-50	93.1
BoT [54]	√	VGG-16	92.5
PA-CNN [47]	√	VGG	92.8
RA-CNN [33]		VGG-19	92.5
MA-CNN [50]		VGG-19	92.8
MAMC [35]		ResNet-101	93.0
DFL-CNN [34]		VGG-16	93.8
B-CNN [14]		VGG-D,M	91.3
iB-CNN [15]		VGG-D	92.0
HIHCA [13]		VGG-16	91.7
MoNet [17]		VGG-16	91.8
DeepKSPD [51]		VGG-16	93.2
G^2DeNet [16]		VGG-D	92.5
PC [52]		ResNet-50	93.4
HBP [38]		VGG-16	93.7
AKEN (ours)		VGG-19	**93.9**

we only lower the learning rate twice. In order to mitigate overfitting, we carry out data augmentations, including random flip and rotation. Furthermore, instead of regularizing the image size directly, we add pixel points to the shorter edge with average value of this image before we resize it to 448×448. We perform all experiments using PyTorch on a server with NVIDIA Titan X GPUs.

C. Performance and Comparison

We compare our approach with various existing FGVC methods. The results on CUB-200-2011, FGVC-Aircraft and Stanford Cars are shown in Table III, Table IV and Table V respectively. Each table is split into five parts over the rows. The first part displays the fine-tuned baselines; the second part includes the annotation-based methods; the third part includes the unsupervised part-based methods; the forth part includes the encoding methods, and the last part is our AKEN. Additionally, in these three tables, "Anno." represents using bounding box or part annotation.

1) Results on CUB-200-2011 Dataset: The classification accuracy on CUB-200-2011 is displayed in Table III. We can see that our AKEN outperforms most classical methods with a recognition rate of 87.1%, only 0.3% lower than the state-of-the-art result. It is worth mentioning that our AKEN exceeds all of the existing encoding methods, which demonstrates the great effectiveness of the proposed kernel encoding method.

2) Results on FGVC-Aircraft: The classification accuracy on FGVC-Aircraft is shown in Table IV. We achieve the best performance and beat the second best method by a 2.17% improvement. Specifically, this performance largely surpasses all the encoding methods.

3) Results on Stanford Cars: The classification accuracy on Stanford Cars is displayed in Table V. Our method exceeds all of the comparison methods with an accuracy of 93.9%.

These results indicate that our proposed approach is powerful and demonstrate that the kernel encoding module in both longitudinal and transverse directions can improve the non-linear learning ability. Also, the impressive performance indicates the attention module can highlight significant regions and improve the performance to a large extent. We will show the power of the attention mechanism in Sec. IV-D and analyze the effectiveness of each module in detail in Sec. IV-E.

D. Visualization

To show the effectiveness of the Cas-Attention module, we visualize the attention maps M_2. Specifically, we compute the average value of M_2 across the channels to visualize the attention heat map, which is similar to the visualization method in [38]. The visualization results of some examples are illustrated in Fig. 5. From the visualization results, we can see that the attention map can highlight the discriminative regions from the complex background. Take the image in the second row and the fifth column for example, its attention map can clearly locate the bird inside the cluttered branches. Although the objects appear in multiple poses and views, even with complex backgrounds, the attention map can still well cover the main part of the target, which may further help to boost the performance of fine-grained classification. Meanwhile, we compare the attention map generated by different methods in Fig. 6, from which the effectiveness of our Cas-Attention is shown more clearly. We will give further analysis about Fig. 6 in Sec.IV-E1.

Additionally, we also visualize the feature maps before and after the attention module in Fig. 7. We show the input images in the first column, the feature maps before and after the attention module in the second and fourth column, the attention maps in the third column. We can see that the feature maps pay more attention to the target and the response

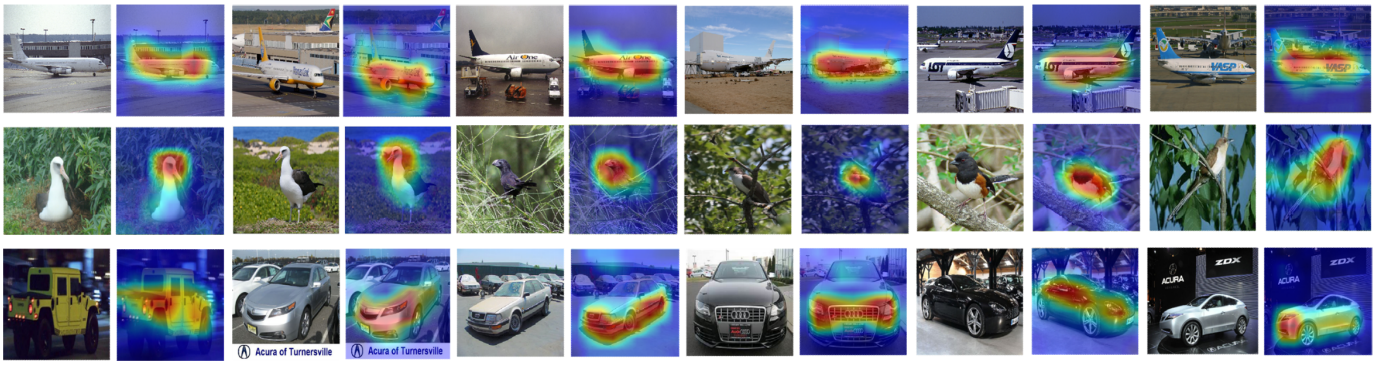


Fig. 5. The visualization of attention maps on three datasets, CUB-200-2011, FGVC-Aircraft and Stanford Cars respectively. For each dataset, we show original images in the odd rows and the corresponding heat maps on the right side. The visual results affirm that our attention module can effectively highlight the discriminative regions.

TABLE VI
ACCURACY COMPARISON WITH DIFFERENT ATTENTION MAPS

Attention Maps	Accuracy
M_1	86.0%
$\boldsymbol{M_1}$+$\boldsymbol{M_2}$	**87.1**%
M_1+M_2+M_3	86.8%
M_1+M_2+M_3+M_4	86.4%

TABLE VII
ACCURACY COMPARISON WITH FOUR DIFFERENT ATTENTION DESIGNS

Method		Accuracy
Conv-Attention	1×1	86.0%
	$1 \times 1 + 3 \times 3$	86.2%
Res-Attention	1×1	86.1%
	$1 \times 1 + 3 \times 3$	86.3%
Non-Local	1×1	86.5%
	$1 \times 1 + 3 \times 3$	86.8%
Cas-Attention	$1 \times 1 + 3 \times 3$	**87.1**%
Baseline(without attention)		85.7%

of background that may be less significant in classification is much lower. These results prove the effectiveness of our Cas-Attention module in highlighting discriminative regions.

E. Ablation Study

In this section, we show the effectiveness of our Cas-Attention module and kernel encoding module by extensive ablation study. We first evaluate the effectiveness of our Cas-Attention module in Sec.IV-E1. Then, we operate experiments to demonstrate the capability of Fourier embedding in Sec.IV-E2. Afterwards, we evaluate the effectiveness of LKE and TKE in Sec.IV-E3 and Sec.IV-E4 respectively. At last, in Sec.IV-E5, we try to combine the output of LKE and TKE in different ways, and compare their performance. Specifically, if not specified, we use the CUB-200-2011 dataset and adopt VGG19 [46] as the backbone in the following experiments.

1) Effectiveness of Cascaded Attention: To evaluate the effectiveness of the Cas-Attention module, we first operate experiments to discuss the compact of the number of attention maps in our Cas-Attention module. Based on the M_1 and M_2 shown in Fig. 3, we add two branches to obtain M_3 and M_4 with 5×5 and 7×7 convolution respectively. Meanwhile, we also test the performance when removing the M_2 and only keep the M_1. In this situation, our Cas-Attention degenerate to Res-Attention structure. The experimental results are illustrated in Table VI. From the results, we can see that the best performance is achieved with M_1 and M_2. When more attention maps are used, the accuracy is decreased. We think it is caused by two factors. First, more attention maps means more parameters, which may lead to the overfitting. Second, it will lose some important detailed information if the filter size is too large.

Then, we compare our Cas-Attention with different methods. On the one hand, we compare the quality of attention map generated by our Cas-Attention with two baselines, convolutional attention (Conv-Attention) and residual attention (Res-Attention). They are all attention-map based methods, in which attention map is utilized to highlight the discriminative regions and plays an important role. Specifically, Conv-Attention is a 3D attention that measures the discrimination of each point in the feature map. Res-Attention adds residual mechanism based on Conv-Attention, and our Cas-Attention adds an extra branch with attention maps on different scales based on the Res-Attention. The visualization results are shown in Fig. 6. It is clear that our Cas-Attention can highlight the discriminative region. Especially for images with the challenging background, the performance of our Cas-Attention is obviously better than the baseline. For example, as shown in the first column, the bird is occluded and surrounded by the branches. Moreover, the color of the bird is not salient in the environment either. In this challenging situation, only our Cas-Attention localizes the bird precisely and highlight the discriminative region. This demonstrates the effectiveness of our proposed Cas-Attention.

On the other hand, we compare the classification accuracy of different methods. Here, we take the non-local method [55] into account. The non-local operation calculates the relationship between each pair of spatial points through the similarity matrix, which captures the long-range dependencies across the whole image. It can be inserted into the ConvNets as a

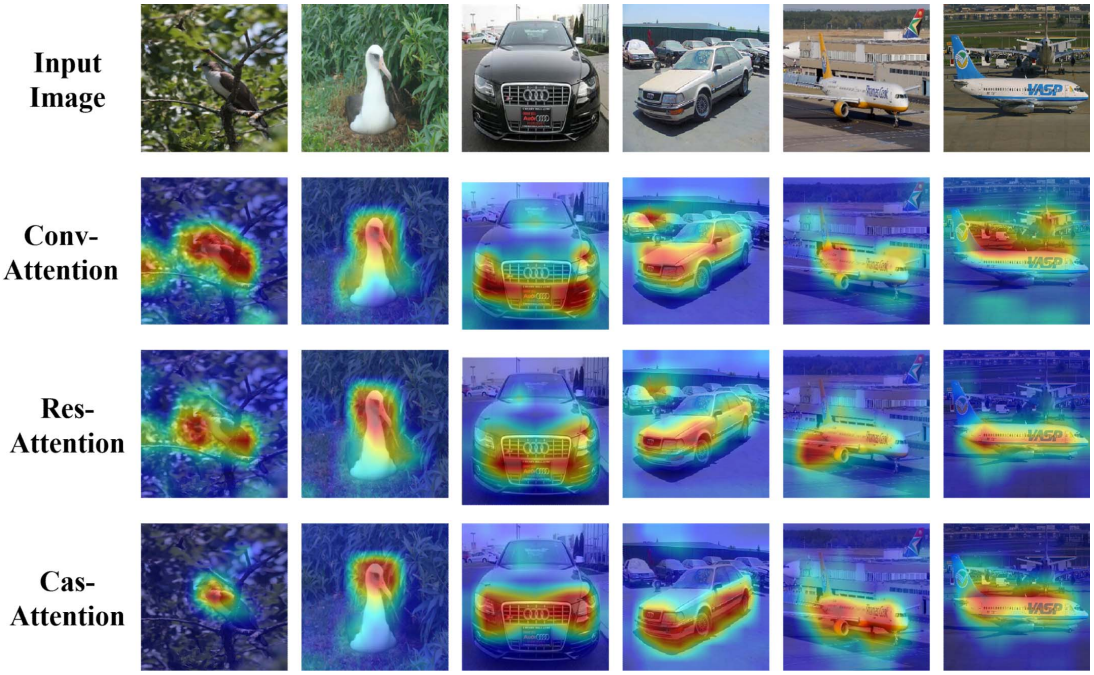


Fig. 6. The visualization of attention map from different methods. The first row shows the input image. The second to fourth row depicts the attention map from Conv-Attention, Res-Attention and Cas-Attention respectively. We can find that our proposed Cas-Attention can precisely localize the discriminative regions, especially in the challenging situation (*e.g.* images in the first column).

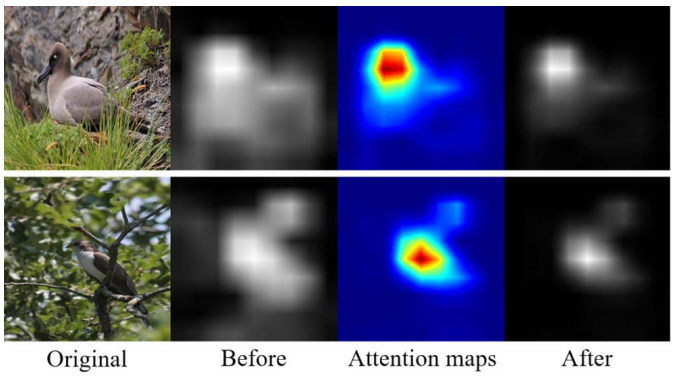


Fig. 7. The visualization of feature maps before and after the Cas-Attention module. The first to fourth columns are the input images, the feature maps before the attention module, the attention heat maps, the feature maps after the attention module respectively. It is apparent that the feature maps become more focused on the target and the response of background, which may be insignificant in classification is much lower.

non-local block with residual strategy. Notably, to make a full comparison with our Cas-Attention, as shown in Table VII, we directly operate two Conv-Attention (or Res-Attention) with different filter sizes (1×1 and 3×3 convolutions respectively) in sequence. Meanwhile, for non-local block, we also add an extra block with 3×3 convolutions in non-local operation. From the experimental results illustrated in Table VII, we observe that all the attention methods can improve the accuracy, compared with the baseline (without attention), while Cas-Attention shows the best performance. It is worth mentioning that, compared with the other three methods, our Cas-Attention still has a better performance when both operate two successive block with 1×1 and 3×3 convolutions respectively. This demonstrates that the performance of Cas-Attention is not relying on the parameters. We think it benefits from its cascaded multi-branch architecture.

TABLE VIII
THE EFFECTIVENESS OF FOURIER EMBEDDING

Activation	Accuracy
ReLu	86.3%
Sigmoid	86.6%
Cosine	**87.1%**

2) Effectiveness of Fourier Embedding: Given the refined features after the Cas-Attention module, we explore an effective method to get a highly non-linear holistic feature representation through kernel encoding. Here, we use the kernel approximation based on random Fourier features. To evaluate the effectiveness of Fourier Embedding, we use ReLu and Sigmoid activation to replace the cosine activation in Equation 7 and Equation 8. The results are shown in Table VIII. It is apparently that AKEN has higher accuracy when activated by cosine. Specifically, the structure of the network is exactly the same in these three situations. Therefore, we believe the performance of kernel encoding does not rely on the number of parameters, but benefits from the non-linearity of Fourier Embedding.

3) Effectiveness of Longitudinal Kernel Encoding: Based on the Fourier embedding, we apply kernel encoding along both longitudinal and transverse directions of feature maps. Here, we first discuss the effectiveness of LKE, and the experimental

TABLE IX
ACCURACY COMPARISON ABOUT LONGITUDINAL KERNEL ENCODING (LKE) AND TRANSVERSE KERNEL ENCODING (TKE)

Method	Accuracy
VGG19 + Attention	83.2%
VGG19 + LKE	84.5%
VGG19 + TKE	84.2%
VGG19 + LKE + TKE	85.7%
VGG19 + Attention + LKE	85.9%
VGG19 + Attention + TKE	86.1%
VGG19 + Attention + LKE + TKE	**87.1%**

TABLE X
ACCURACY COMPARISON OF DIFFERENT COMBINATION WAYS OF LKE AND TKE

Activation	Accuracy
Element-wise Weighted Multiplication	86.8%
Element-wise Weighted Sum	87.0%
Concatenation	87.1%

results are presented in Table IX. We can find that, compared with the baseline, LKE can improve the accuracy a lot. This result is consistent with our inspiration. Since the longitudinal vector indicates different feature responses in a local area, LKE provides us with the relationship of different feature patterns in a non-linear space through kernel encoding, which brings more discriminative information.

4) Effectiveness of Transverse Kernel Encoding: Since the LKE takes each longitudinal vector as input, it only computes the inter-channel relationship. Therefore, we propose TKE to take the spatial layout of the image into consideration. From Table IX, we can find that TKE can also improve the performance. Moreover, combining both LKE and TKE achieves a better result than each individual one, which shows the great complementarity of LKE and TKE. Specifically, it is worth mentioning that, compared to the first line in Table IX (without kernel encoding), the three results in Table VIII all have improvements. We owe this performance to the effectiveness of our two-directional structure. In our AKEN, LKE and TKE can capture the inter-channel and intra-channel relationship respectively. LKE provides the relationship of different feature pattern in each spatial point and TKE provides the spatial distribution of each feature pattern. Therefore, when equipped with our two directional kernel encoding, three activations in Table VIII can all improve the accuracy. However, due to the effectiveness of Fourier embedding, the cosine activation achieves the best performance among all activation functions.

Additionally, we also test the Cas-Attention module with different kernel encoding. The results show that combining Cas-Attention with LKE and TKE produces the highest accuracy that is better than without the attention module. The experimental results indicate that our Cas-Attention module plays an essential role in assisting non-linear feature extraction.

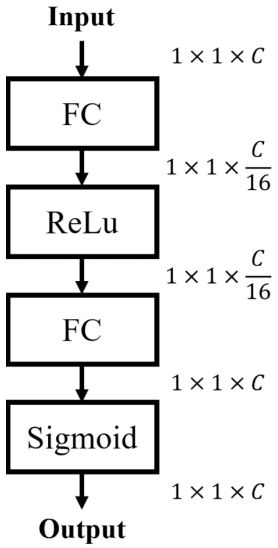


Fig. 8. The illustration of the Weights Generation Module (WGM). "Fc" means fully connected layer. ReLu and Sigmoid are two activations. The size of vector before and after each step is denoted accordingly.

5) Combination of LKE and TKE: Here, we attempt to combine the output of LKE (v_L) and TKE (v_T) in different ways and make a comparison. Specifically, except concatenation, we also try element-wise weighted sum and element-wise weighted multiplication.

In element-wise weighted sum, given two vectors v_L and v_T, we firstly utilize a Weights Generation Module (WGM) to generate corresponding weights. The detail of WGM is illustrated in Fig. 8. It takes v_L (v_T) as input and outputs w_L (w_T). Then, the combination through element-wise weighted sum can be written as:

$$v_{all} = w_L \times v_L + w_T \times v_T \tag{9}$$

where "$\times$" denotes the element-wise multiplication and "$+$" denotes the element-wise sum.

In element-wise weighted multiplication, similar to the element-wise weighted sum, we firstly utilize WGM to generate w'_L and w'_T for v_L and v_T respectively. Then, the combination can be written as:

$$v'_{all} = \left(w'_L \times v_L\right) \times \left(w'_T \times v_T\right) \tag{10}$$

The experimental results are listed in Table X. We can find the best performance is produced when we concatenate v_L and v_T. On the other hand, in fact, the classification performances of three combination ways are not significantly different, which shows our AKEN has great robustness and is not sensitive to different combination ways.

With these experiments above, we evaluate the effectiveness of each part in our AKEN. The results demonstrate the Cas-Attention, LKE and TKE all play important roles in obtaining a high-quality discriminative feature representation, which is consistent with our motivation.

V. CONCLUSION

In this work, we have presented the attentional kernel encoding network (AKEN), which offers a new deep feature

encoding architecture to generate highly discriminative feature representations for fine-grained visual categorization. We introduce the kernel approximation to the deep convolutional networks for non-linear feature encoding, which is implemented in both longitudinal and transverse directions. To enhance the feature encoding module, we further design a Cas-Attention module with the residual mechanism to highlight local regions that can distinguish different categories. Our AKEN leverages the strengths of both ConvNets for feature extraction and kernels for non-linear learning. Experimental results on three benchmark datasets demonstrate that our proposed AKEN delivers highly competitive performance, surpassing most previous methods.

APPENDIX A
PROOF OF PROPOSITION 1

Proof: From the Bochner's theorem, we have that a continuous kernel $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}-\mathbf{y})$ on $\mathcal{R}^d$ is positive definite if and only if $k(\delta)$ is the Fourier transform of a non-negative measure.

Now if a shift invariant kernel $k(\delta)$ is properly scaled, Bochner's theorem guarantees that its Fourier transform $p(\omega)$ is a probability distribution, as the integral of $p(\omega)$ is scaled to 1. The kernel can be written as:

$$\begin{aligned} k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}-\mathbf{y}) &= \int_{R^d} p(\omega) e^{j\omega^\top(\mathbf{x}-\mathbf{y})} d\omega \\ &= \int_{R^d} p(\omega)\cos(\omega^\top \mathbf{x} - \omega^\top \mathbf{y}) d\omega \end{aligned} \tag{11}$$

Therefore, we have $\mathbb{E}_\omega \cos(\omega^\top(\mathbf{x}-\mathbf{y})) = k(\mathbf{x}, \mathbf{y})$, where $\omega \sim p(w)$. And we have:

$$\begin{aligned} &\cos(\omega^\top \mathbf{x} + b)\cos(\omega^\top \mathbf{y} + b) \\ &\quad = \frac{1}{2}\cos(\omega^\top(\mathbf{x}-\mathbf{y})) + \frac{1}{2}\cos(\omega^\top(\mathbf{x}+\mathbf{y}) + 2b) \end{aligned} \tag{12}$$

Since $\mathbb{E}_b \cos(\omega^\top(\mathbf{x}+y) + 2b) = 0$, with $b \sim \text{Unif}_{[0,2\pi]}$, we could finally have:

$$\begin{aligned} k(\mathbf{x}, \mathbf{y}) &= \mathbb{E}_\omega \mathbb{E}_b[\sqrt{2}\cos(\omega^\top \mathbf{x} + b) \cdot \sqrt{2}\cos(\omega^\top \mathbf{y} + b)] \\ &\approx \sum_{i=1}^{L}[\frac{\sqrt{2}}{\sqrt{L}}\cos(\omega_i^\top \mathbf{x} + b_i) \cdot \frac{\sqrt{2}}{\sqrt{L}} cos(\omega_i^\top \mathbf{y} + b_i)] \\ &= z(\mathbf{x})^T z(\mathbf{y}) \end{aligned} \tag{13}$$

Thus, according to Equation 13, if we draw L samples $\omega_1, \omega_2, \ldots, \omega_L$ from $p(\omega)$, and $b_1, b_2, \ldots, b_L$ from $Unif_{[0,2\pi]}$, then the random Fourier features can be represented as $z(\mathbf{x}) = \frac{\sqrt{2}}{\sqrt{L}}[\cos(\omega_i^\top \mathbf{x} + b)]^L \in R^L$. □

APPENDIX B
PROOF OF PROPOSITION 2

Proof: According to Proposition 1,

$$z(\mathbf{x})^\top z(\mathbf{y}) = \frac{1}{L}\sum_{i=1}^{L}[\sqrt{2}\cos(\omega_i^\top \mathbf{x} + b_i) \cdot \sqrt{2}\cos(\omega_i^\top \mathbf{y} + b_i)] \tag{14}$$

Here $\sqrt{2}\cos(\omega_i^\top \mathbf{x} + b_i) \cdot \sqrt{2}\cos(\omega_i^\top \mathbf{y} + b_i)$, where $i \in \{1, 2, \ldots, L\}$, can be seen as L independent random variables bounded by the interval $[-2,2]$. And $z(\mathbf{x})^\top z(y)$ is the empirical mean of these variables.

Then by the aid of Hoeffding's Inequality [56], we can educe

$$Pr[|z(\mathbf{x})^T z(\mathbf{y}) - k(\mathbf{x}, \mathbf{y})| \geq \epsilon] \leq 2\exp(-L\epsilon^2/8) \tag{15}$$

This indicates that more samples taken from the Fourier features promise better performance in kernel approximation. □

REFERENCES

[1] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[2] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[4] X. Jiang *et al.*, "Crowd counting and density estimation by trellis encoder-decoder networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6133–6142.

[5] J. Zheng, X. Cao, B. Zhang, X. Zhen, and X. Su, "Deep ensemble machine for video classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 553–565, Feb. 2019.

[6] P. Li, A. Zhang, L. Yue, X. Zhen, and X. Cao, "Multi-scale aggregation network for direct face alignment," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 2156–2165.

[7] Y. Zhao, Z. Jin, G.-J. Qi, H. Lu, and X.-S. Hua, "An adversarial approach to hard triplet generation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 501–517.

[8] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 834–849.

[9] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," 2015, *arXiv:1511.04119*. [Online]. Available: http://arxiv.org/abs/1511.04119

[10] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 34–45.

[11] Y. Tang, X. Wang, E. Dellandrea, and L. Chen, "Weakly supervised learning of deformable part-based models for object detection via region proposals," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 393–407, Feb. 2017.

[12] X. Shu, J. Tang, G.-J. Qi, Z. Li, Y.-G. Jiang, and S. Yan, "Image classification with tailored fine-grained dictionaries," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 28, no. 2, pp. 454–467, Feb. 2018.

[13] S. Cai, W. Zuo, and L. Zhang, "Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 511–520.

[14] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1449–1457.

[15] T.-Y. Lin and S. Maji, "Improved bilinear pooling with CNNs," 2017, *arXiv:1707.06772*. [Online]. Available: http://arxiv.org/abs/1707.06772

[16] Q. Wang, P. Li, and L. Zhang, "G2DeNet: Global Gaussian distribution embedding network and its application to visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2730–2739.

[17] M. Gou, F. Xiong, O. Camps, and M. Sznaier, "MoNet: Moments embedding network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3175–3183.

[18] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 533–544, Feb. 2010.

[19] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, "Kernel pooling for convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2921–2930.

[20] G.-J. Qi, "Hierarchically gated deep networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2267–2275.

[21] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[23] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked CNN for fine-grained visual categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1173–1182.

[24] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep LAC: Deep localization, alignment and classification for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1666–1674.

[25] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.

[26] J. Yang *et al.*, "Neural aggregation network for video face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4362–4371.

[27] L. Jin, X. Shu, K. Li, Z. Li, G.-J. Qi, and J. Tang, "Deep ordinal hashing with spatial attention," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2173–2186, May 2019.

[28] C. Shen *et al.*, "Sharp attention network via adaptive sampling for person re-identification," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 29, no. 10, pp. 3016–3027, Oct. 2019.

[29] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3640–3649.

[30] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 842–850.

[31] X. Liu, T. Xia, J. Wang, Y. Yang, F. Zhou, and Y. Lin, "Fully convolutional attention networks for fine-grained recognition," 2016, *arXiv:1603.06765*. [Online]. Available: http://arxiv.org/abs/1603.06765

[32] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[33] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4438–4446.

[34] Y. Wang, V. I. Morariu, and L. S. Davis, "Learning a discriminative filter bank within a CNN for fine-grained recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4148–4157.

[35] M. Sun, Y. Yuan, F. Zhou, and E. Ding, "Multi-attention multi-class constraint for fine-grained image recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 805–821.

[36] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5012–5021.

[37] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models," *Neural Comput.*, vol. 12, no. 6, pp. 1247–1283, Jun. 2000.

[38] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 595–610.

[39] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 317–326.

[40] W. Rudin, *Fourier Analysis on Groups*. New York, NY, USA: Interscience, 1962.

[41] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1177–1184.

[42] P. Welinder, S. Branson, C. Wah, F. Schroff, S. Belongie, and P. Perona, *Caltech-UCSD Birds 200*. Pasadena, CA, USA: California Institute of Technology, 2010.

[43] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," in *Proc. HAL-INRIA*, 2013, pp. 1–6.

[44] 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*, Dec. 2013, pp. 554–561.

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

[46] 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

[47] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5546–5555.

[48] H. Zhang *et al.*, "SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1143–1152.

[49] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," 2014, *arXiv:1406.2952*. [Online]. Available: http://arxiv.org/abs/1406.2952

[50] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5209–5217.

[51] M. Engin, L. Wang, L. Zhou, and X. Liu, "DeepKSPD: Learning kernel-matrix-based SPD representation for fine-grained image recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 612–627.

[52] A. Dubey, O. Gupta, P. Guo, R. Raskar, R. Farrell, and N. Naik, "Pairwise confusion for fine-grained visual classification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 70–86.

[53] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang, "Multiple granularity descriptors for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2399–2406.

[54] Y. Wang, J. Choi, V. I. Morariu, and L. S. Davis, "Mining discriminative triplets of patches for fine-grained classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1163–1172.

[55] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

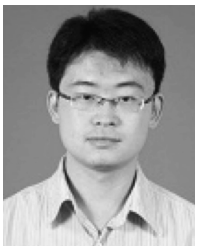
[56] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Publications Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.

Yutao Hu received the B.S. degree in electronics and information engineering from Beihang University, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree with the National Key Laboratory of CNS/ATM, School of Electronics and Information Engineering. His research interests include machine learning and computer vision.

Yandan Yang received the B.S. degree from the Shenyuan Honors College, Beihang University, Beijing, China, in 2018, where she is currently pursuing the M.S. degree with the National Key Laboratory of CNS/ATM, School of Electronics and Information Engineering. Her research interests include machine learning and computer vision.

Jun Zhang received the B.S., M.S., and Ph.D. degrees in communications and electronic systems from Beihang University, Beijing, China, in 1987, 1991, and 2001, respectively. He was a Professor with Beihang University and also the Dean of the School of Electronic and Information Engineering, the Vice President, and the Secretary of the Party Committee, Beihang University. He is currently a Professor and the President of the Beijing Institute of Technology. His research interests are networked and collaborative air traffic management systems, covering signal processing, integrated and heterogeneous networks, and computer vision. He is a member of the Chinese Academy of Engineering. He was a recipient of the awards for Science and Technology in China many times.

Xianbin Cao (Senior Member, IEEE) received the Ph.D. degree in information science from the University of Science and Technology of China, Beijing, China, in 1996. He is currently the Dean and a Professor of the School of Electronic and Information Engineering, Beihang University, Beijing. His current research interests include intelligent transportation systems, airspace transportation management, and intelligent computation.



Xiantong Zhen received the B.S. and M.E. degrees from Lanzhou University, Lanzhou, China, in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K., in 2013. He worked as a Post-Doctoral Fellow at the University of Western Ontario, London, ON, Canada, and the University of Texas at Arlington, Arlington, TX, USA, from 2013 to 2017. He was an Associate Professor at the School of Electronic and Information Engineering, Beihang University, Beijing, China, from 2017 to 2018. He is currently a Senior Scientist with the Inception Institute of Artificial Intelligence, UAE. His research interests include machine learning and computer vision.