Conformer: Local Features Coupling Global Representations for Visual Recognition

Zhiliang Peng\textsuperscript{1}  Wei Huang\textsuperscript{1}  Shanzhi Gu\textsuperscript{3}  Lingxi Xie\textsuperscript{2}  Yaowei Wang\textsuperscript{3}  Jianbin Jiao\textsuperscript{1}  Qixiang Ye\textsuperscript{1,3,*}

\textsuperscript{1}University of Chinese Academy of Sciences, Beijing, China  \textsuperscript{2}Huawei Inc.  \textsuperscript{3}Peng Cheng Laboratory, Shenzhen, China

\{pengzhiliang19, huangwei19\}@mails.ucas.ac.cn \{gushzh, wangyw\}@pcl.ac.cn
198808xc@gmail.com \{jiaojb, qxye\}@ucas.ac.cn

Abstract

Within Convolutional Neural Network (CNN), the convolution operations are good at extracting local features but experience difficulty to capture global representations. Within visual transformer, the cascaded self-attention modules can capture long-distance feature dependencies but unfortunately deteriorate local feature details. In this paper, we propose a hybrid network structure, termed Conformer, to take advantage of convolutional operations and self-attention mechanisms for enhanced representation learning. Conformer roots in the Feature Coupling Unit (FCU), which fuses local features and global representations under different resolutions in an interactive fashion. Conformer adopts a concurrent structure so that local features and global representations are retained to the maximum extent. Experiments show that Conformer, under the comparable parameter complexity, outperforms the visual transformer (DeiT-B) by 2.3% on ImageNet. On MSCOCO, it outperforms ResNet-101 by 3.7% and 3.6% mAPs for object detection and instance segmentation, respectively, demonstrating the great potential to be a general backbone network. Code is available at github.com/pengzhiliang/Conformer.

1. Introduction

Convolutional neural networks (CNNs) \cite{29, 37, 40, 19, 48, 22} have significantly advanced computer vision tasks such as image classification, object detection, and instance segmentation. This largely attributes to the convolution operation, which collects local features in a hierarchical fashion as powerful image representations. Despite of the advantage upon local feature extraction, CNNs experience difficulty to capture global representations, \textit{e.g.}, long-distance relationships among visual elements, which are often critical for high-level computer visual tasks. An intuitive solution is enlarging the receptive field, which however could require more intensive yet damaging pooling operations.

Recently, the transformer architecture \cite{42} has been introduced to visual tasks \cite{16, 47, 41, 51, 8, 9, 3, 55, 28}. The ViT method \cite{16} constructs a sequence of tokens by splitting each image to patches with positional embeddings and applies cascaded transformer blocks to extract parameterized vectors as visual representations. Thanks to the self-attention mechanism and Multilayer Perceptron (MLP) structure, the visual transformer reflects complex spatial transforms and long-distance feature dependencies, which constitute global representations. Unfortunately, visual transformers are observed ignoring local feature details which decreases the discriminability between background and foreground, Figs. 1(c) and (g). Improved visual transformers \cite{16, 51} have proposed a tokenization module or leveraged CNN feature maps as input tokens to capture feature neighboring information. Nevertheless, the problem about how to precisely embed local features and global representations to each other remains.

In this paper, we propose a dual network structure, termed Conformer, with the aim to couple CNN-based local features with transformer-based global representations for enhanced representation learning. Conformer consists of a CNN branch and a transformer branch which respectively follow the design of ResNet \cite{19} and ViT \cite{16}. The two branches form a comprehensive combination of local convolution blocks, self-attention modules, and MLP units. During training, the cross entropy losses are used to supervise both the CNN and transformer branches to couple CNN-style and transformer-style features.

Considering the feature misalignment between CNN and transformer features, the Feature Coupling Unit (FCU) is designed as the bridge. On the one hand, to fuse the two-
Shallow Layers
Ours-CNN
ResNet-101
DeiT-S
Ours-Transformer
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
Deep Layers
Ours-CNN
ResNet-101
DeiT-S
Ours-Transformer

Figure 1: Comparison of feature maps of CNN (ResNet-101) [19], Visual Transformer (DeiT-S) [41], and the proposed Conformer. The patch embeddings in transformer are reshaped to feature maps for visualization. While CNN activates discriminative local regions (e.g., the peacock’s head in (a) and tail in (e)), the CNN branch of Conformer takes advantage of global cues from the visual transformer and thereby activates complete object (e.g., full extent of the peacock in (b) and (f)). Compared with CNN, local feature details of the visual transformer are deteriorated (e.g., (c) and (g)). In contrast, the transformer branch of Conformer retains the local feature details from CNN while depressing the background (e.g., the peacock contours in (d) and (h) are more complete than those in (c) and (g)). (Best viewed in color)

style features, FCU leverages $1 \times 1$ convolution to align the channel dimensions, down/up sampling strategies to align feature resolutions, LayerNorm [2] and BatchNorm [25] to align feature values. On the other hand, since CNN and transformer branches tend to capture features of different levels (e.g., local vs. global), FCU is inserted into every block to consecutively eliminate the semantic divergence between them, in an interactive fashion. Such a fusion procedure can greatly enhance the global perception capability of local features and the local details of global representations.

The ability of Conformer in coupling local features and global representations is demonstrated in Fig. 1. While conventional CNNs (e.g., ResNet-101) tend to retain discriminative local regions (e.g., the peacock’s head or tail), the CNN branch of Conformer can activate the full object extent, Figs. 1(b) and (f). When solely using the visual transformers, for the weak local features (e.g., blurred object boundaries), it is difficult to distinguish the object from the background, Figs. 1(c) and (g). The coupling of local features and global representations significantly enhances the discriminability of transformer-based features, Figs. 1(d) and (h).

The contributions of this paper include:

- We propose a dual network structure, termed Conformer, which retains local features and global representations to the maximum extent.
- We propose the Feature Coupling Unit (FCU), to fuse convolutional local features with transformer-based global representations in an interactive fashion.
- Under comparable parameter complexity, Conformer outperforms CNNs and visual transformers by significant margins. Conformer inherits the structure and generalization advantages of both CNNs and visual transformers, demonstrating the great potential to be a general backbone network.

2. Related Work

CNNs with Global Cues. In the deep learning era, CNNs can be regarded as a hierarchical ensemble of local features with different reception fields. Unfortunately, most CNNs [29, 37, 19, 39, 48, 23, 44] are good at extracting local features but experience difficulty to capture global cues.

To alleviate such a limitation, one solution is to define larger receptive fields by introducing deeper architectures and/or more pooling operations [22, 21]. The dilated convolution methods [49, 50] increased the sampling step size, while deformable convolution [13] learned the sampling positions. SENet [22] and GENet [21] proposed to use global Avgpooling to aggregate global context and then used it to reweight feature channels, while CBAM [46] respectively used global Maxpooling and global Avgpooling to refine features independently in the spatial and channel dimensions.

The other solution is the global attention mechanism [45, 7, 4, 20, 38], which has demonstrated great advantage in capturing long-distance dependencies in natural language processing [42, 15, 5]. Inspired by the non-local means method [6], the non-local operation [45] was introduced to CNNs in a self-attention manner so that the response at each position is a weighted sum of the features at all (global) positions. Attention augmented convolutional networks [4] concatenated convolutional feature maps with self-attentional feature maps to augment convolution operations for capturing long-range interactions. Relation Net-
Figure 2: Network architecture of the proposed Conformer. (a) Up-sampling and down-sampling for spatial alignment of feature maps and patch embeddings. (b) Implementation details of the CNN block, the transformer block, and the Feature Coupling Unit (FCU). (c) Thumbnail of Conformer.

works [20] proposed an object attention module, which processes a set of objects simultaneously through interaction between their appearance feature and geometry.

Despite of the progress, existing solutions that introduce global cues to CNNs have obvious disadvantages. For the first solution, larger receptive fields require more intensive pooling operations, which implies lower spatial resolution. For the second solution, if convolutional operations are not properly fused with attention mechanisms, local feature details could deteriorate.

Visual Transformers. As a pioneered work, ViT [16] validated the feasibility of pure transformer architectures for computer vision tasks. To leverage the long-distance dependencies, transformer blocks acted as independent architectures or were introduced to CNNs for image classification [47, 41, 51], object detection [8, 58, 3], semantic segmentation [55], image enhancement [9], weakly-supervised object localization [17] and image generation [11, 28]. However, the self-attention mechanism in visual transformers often ignores local feature details. To solve, DeiT [41] proposed using a distillation token to transfer CNN-based features to visual transformer while T2T-ViT [51] proposed using a tokenization module to recursively reorganize the image to tokens considering neighboring pixels. In object detection, DETR method [8, 58] fed local features extracted by CNN to the transformer encoder-decoder to model the global relationships between features in a serial fashion.

Different from existing works, Conformer defines the first concurrent network structure which fuses features in an interactive fashion. Such a structure not only naturally inherits the structure advantages of both CNN and transformers but also retains the representation capability of local features and global representations to the maximum extent.

3. Conformer

3.1. Overview

Local features and global representations are important counterparts, which have been extensively studied in the long history of visual descriptors. Local features and their descriptors [34, 27, 35], which are compact vector representations of local image neighborhoods, have been the building blocks of many computer vision algorithms. Global representations include, but not limited to, contour representations, shape descriptors, and object typologies at long-distance [32]. In the deep learning era, CNN collects local features in a hierarchical manner via convolutional operations and retains the local cues as feature maps. Visual transformer is believed to aggregate global representations among the compressed patch embeddings in a soft fashion by the cascaded self-attention modules.

In order to take advantage of local features and global representations, we design a concurrent network structure, as shown in Fig. 2(c), termed Conformer. Considering
Table 1: Architecture of Conformer-S, where MHSA-6 denotes the multi-head self-attention with heads 6 in transformer block and the fc layer is viewed as $1 \times 1$ convolution here. In FCU column, the arrows represent the flow of feature. And in output column, $56 \times 56$ respectively mean the size of feature map is $56 \times 56$ and the number of embedded patches is 197.

<table>
<thead>
<tr>
<th>stage</th>
<th>output</th>
<th>CNN Branch</th>
<th>FCU</th>
<th>Transformer Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>$112 \times 112$</td>
<td>$7 \times 7$, 64, stride 2</td>
<td>$56 \times 56$, 3 × 3 max pooling, stride 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 × 1, 64</td>
<td>MHSA-6, 384</td>
<td>-</td>
<td>4 × 4, 384, stride 4</td>
</tr>
<tr>
<td></td>
<td>3 × 3, 64</td>
<td>$1 \times 1$, 1536</td>
<td>-</td>
<td>$1 \times 1$, 1536</td>
</tr>
<tr>
<td></td>
<td>1 × 1, 256</td>
<td>$1 \times 1$, 1536</td>
<td>-</td>
<td>$1 \times 1$, 1536</td>
</tr>
<tr>
<td>c2</td>
<td>$56 \times 56, 197$</td>
<td>$1 \times 1, 64$</td>
<td>$1 \times 1, 128$</td>
<td>MHSA-6, 384</td>
</tr>
<tr>
<td></td>
<td>3 × 3, 64</td>
<td>$1 \times 1, 384$</td>
<td>$1 \times 1, 384$</td>
<td>$1 \times 1, 128$</td>
</tr>
<tr>
<td></td>
<td>1 × 1, 256</td>
<td>$1 \times 1, 1536$</td>
<td>$1 \times 1, 1536$</td>
<td>$1 \times 1, 1536$</td>
</tr>
<tr>
<td>c3</td>
<td>$28 \times 28, 197$</td>
<td>$1 \times 1, 128$</td>
<td>$1 \times 1, 128$</td>
<td>MHSA-6, 384</td>
</tr>
<tr>
<td></td>
<td>3 × 3, 128</td>
<td>$1 \times 1, 384$</td>
<td>$1 \times 1, 384$</td>
<td>$1 \times 1, 128$</td>
</tr>
<tr>
<td></td>
<td>1 × 1, 512</td>
<td>$1 \times 1, 1536$</td>
<td>$1 \times 1, 1536$</td>
<td>$1 \times 1, 1536$</td>
</tr>
<tr>
<td>c4</td>
<td>$14 \times 14, 197$</td>
<td>$1 \times 1, 256$</td>
<td>$1 \times 1, 256$</td>
<td>MHSA-6, 384</td>
</tr>
<tr>
<td></td>
<td>3 × 3, 256</td>
<td>$1 \times 1, 384$</td>
<td>$1 \times 1, 384$</td>
<td>$1 \times 1, 256$</td>
</tr>
<tr>
<td></td>
<td>1 × 1, 1024</td>
<td>$1 \times 1, 1536$</td>
<td>$1 \times 1, 1536$</td>
<td>$1 \times 1, 1536$</td>
</tr>
<tr>
<td>c5</td>
<td>$7 \times 7, 197$</td>
<td>$1 \times 1, 256$</td>
<td>$1 \times 1, 256$</td>
<td>MHSA-6, 384</td>
</tr>
<tr>
<td></td>
<td>3 × 3, 256</td>
<td>$1 \times 1, 384$</td>
<td>$1 \times 1, 384$</td>
<td>$1 \times 1, 256$</td>
</tr>
<tr>
<td></td>
<td>1 × 1, 1024</td>
<td>$1 \times 1, 1536$</td>
<td>$1 \times 1, 1536$</td>
<td>$1 \times 1, 1536$</td>
</tr>
<tr>
<td>classifier</td>
<td>$1 \times 1, 1$</td>
<td>global pooling</td>
<td>-</td>
<td>class token</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
<td>$1 \times 1,1000$</td>
<td>-</td>
<td>$1 \times 1,1000$</td>
</tr>
<tr>
<td>MACs</td>
<td></td>
<td>37.7 M</td>
<td>-</td>
<td>10.6 G</td>
</tr>
</tbody>
</table>

3.2. Network Structure

**CNN Branch.** As shown in Fig. 2(b), the CNN branch adopts feature pyramid structure, where the resolution of feature maps decreases with network depth while the channel number increases. We split the whole branch into 4 stages, as described in Tab. 1(CNN Branch). Each stage is composed of multiple convolution blocks and each convolution block contains $n_c$ bottlenecks. Following the definition in ResNet [19], a bottleneck contains a $1 \times 1$ down-projection convolution, a $3 \times 3$ spatial convolution, a $1 \times 1$ up-projection convolution, and a residual connection between the input and output of the bottleneck. In experiments, $n_c$ is set to be 1 in the first convolution block and satisfies $\geq 2$ in the subsequent $N - 1$ convolution blocks.

Visual transformers [16, 41] project an image patch into a vector through a single step, causing the lost of local details. While in CNNs, convolution kernels slide over feature maps with overlap, which provides the possibility to preserve fine-detailed local features. Consequently, the CNN branch is able to consecutively provide local feature details for the transformer branch.

**Transformer Branch.** Following ViT [16], this branch contains $N$ repeated transformer blocks. As shown in Fig. 2(b), each transformer block consists of a multi-head self-attention module and an MLP block (contains a up-projection fc layer and a down-projection fc layer). LayerNorms [2] are applied before each layer and residual connections in both the self-attention layer and MLP block. For tokenization, we compress the feature maps generated by the stem module into $14 \times 14$ patch embeddings without overlap, by a linear projection layer, which is a $4 \times 4$ convolution with stride 4. A class token is then pretended to
Structure Analysis. By considering the FCU as a short connection, we can abstract the proposed dual structure into the special serial residual structure, as shown in Fig. 3(a). Under different residual connection units, Conformer can implement different depths combinations of bottlenecks (as in ResNet, Fig. 3(b)) and transformer blocks (as in ViT, Fig. 3(d)), implying that Conformer inherits the structural advantages of both CNNs and visual transformers. Furthermore, it achieves different permutations of bottlenecks and transformer blocks at different depths, including but not limited to Figs. 3(c) and (e). This greatly enhances the representation capacity of the network.

Feature Analysis. We visualize the feature maps in Fig. 1, class activation maps and attention maps in Fig. 4. Compared with ResNet [19], with the coupled global representations, the CNN branch of Conformer tends to activate larger regions rather than local areas, suggesting enhanced long-distance feature dependencies, which are significantly demonstrated in Figs. 1(f) and 4(a). Thanks to the fine-detailed local features progressively provided by the CNN branch, the patch embeddings of the transformer branch in the Conformer retain important detailed local features.
Table 2: Top-1 accuracy for image classification on the ImageNet validation set.

<table>
<thead>
<tr>
<th>Model</th>
<th>Image size</th>
<th>#Params MACs</th>
<th>Top-1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50 [19]</td>
<td>224^2</td>
<td>25.6 4.1</td>
<td>76.2</td>
</tr>
<tr>
<td>ResNet-101 [19]</td>
<td>224^2</td>
<td>44.5 7.8</td>
<td>77.4</td>
</tr>
<tr>
<td>ResNet-152 [19]</td>
<td>224^2</td>
<td>60.2 11.6</td>
<td>78.3</td>
</tr>
<tr>
<td>RegNet-Y-4.0GF [36]</td>
<td>224^2</td>
<td>20.6 4.0</td>
<td>78.8</td>
</tr>
<tr>
<td>RegNet-Y-12.0GF [36]</td>
<td>224^2</td>
<td>51.8 12.1</td>
<td>80.3</td>
</tr>
<tr>
<td>RegNet-G-32.0GF [36]</td>
<td>224^2</td>
<td>145.0 32.3</td>
<td>81.0</td>
</tr>
<tr>
<td>ViT-B [16]</td>
<td>384^2</td>
<td>86 55.5</td>
<td>77.9</td>
</tr>
<tr>
<td>ViT-L [16]</td>
<td>384^2</td>
<td>307 191.1</td>
<td>76.5</td>
</tr>
<tr>
<td>T2T-ViT-14 [51]</td>
<td>224^2</td>
<td>21.5 5.2</td>
<td>80.7</td>
</tr>
<tr>
<td>T2T-ViT-19 [51]</td>
<td>224^2</td>
<td>39.0 8.4</td>
<td>81.4</td>
</tr>
<tr>
<td>T2T-ViT-24 [51]</td>
<td>224^2</td>
<td>64.1 13.2</td>
<td>82.2</td>
</tr>
<tr>
<td>DeiT-S [41]</td>
<td>224^2</td>
<td>22.1 4.6</td>
<td>79.8</td>
</tr>
<tr>
<td>DeiT-B [41]</td>
<td>224^2</td>
<td>86.6 17.6</td>
<td>81.8</td>
</tr>
<tr>
<td>Conformer-Ti</td>
<td>224^2</td>
<td>23.5 5.2</td>
<td>81.3</td>
</tr>
<tr>
<td>Conformer-S</td>
<td>224^2</td>
<td>37.7 10.6</td>
<td>83.4</td>
</tr>
<tr>
<td>Conformer-B</td>
<td>224^2</td>
<td>83.3 23.3</td>
<td>84.1</td>
</tr>
</tbody>
</table>

4. Experiments

4.1. Model Variants

By tuning the parameters of the CNN and transformer branches, we have the model variants, termed Conformer-Ti, -S, and -B, respectively. The details of Conformer-S are described in Tab. 1, and those of Conformer-Ti/B are in the Supplementary material. Conformer-S/32 splits the feature maps to 7×7 patches, i.e., the patch size is 32×32 in the transformer branch.

4.2. Image Classification

**Experimental Setting.** Conformer is trained on the ImageNet-1k [14] training set with 1.3M images and tested upon the validation set. The Top-1 accuracy is reported in Tab. 2. To make the transformer converge to a reasonable performance, we follow the data augmentation and regularization techniques in DeiT [41]. These techniques include Mixup [53], CutMix [52], Erasing [56], RandAugment [12] and Stochastic Depth [24]. The model is trained for 300 epochs with the AdamW optimizer [33], batchsize 1024 and weight decay 0.05. The initial learning rate is set to 0.001 and decay in a cosine schedule.

**Performance.** Under similar parameters and computational budgets, Tab. 2, Conformers outperform both CNN and visual transformers. For example, Conformer-S (with 37.7M parameters and 10.6G MACs) respectively outperforms ResNet-152 (with 60.2M parameters and 11.6G MACs) by 4.1%((83.4% vs. 78.3%) and DeiT-B (with 86.6M parameters and 17.6G MACs) by 1.6% (83.4% vs. 81.8%). Conformer-B, with comparable parameters and moderate MAC cost, outperforms DeiT-B by 2.3% (84.1% vs. 81.8%). Beyond its superior performance, Conformer converges faster than the visual transformers.

4.3. Object Detection and Instance Segmentation

To verify Conformer’s versatility, we test it on instance-level tasks (e.g., object detection) and pixel-level tasks (e.g., instance segmentation) on the MSCOCO dataset. Conformer, as the backbone, is migrated without extra design, and the relative accuracy and parameter comparison is included in Tab. 2. With the CNN branch, we can use the output feature maps of [c2, c3, c4, c5] as side-output to construct the feature pyramid.

**Experimental Setting.** As is common practice, the models are trained on the MSCOCO training set and tested on the MSCOCO minival set. In Tab. 3, we report APbbox (APseg), APbbox (APseg), APbbox (APseg), APbbox (APseg), and APbbox (APseg) averaged over IoU thresholds, small, medium and large objects of box (mask), respectively. Unless explicitly specified, we use the batch size 32, with a learning rate 0.0002, optimizer AdamW [33], weight decay 0.0001 and max epoch 12. The learning rate decays at the 8-th and 11-th epoch by a magnitude.

**Performance.** As shown in Tab. 3, Conformer significantly boosts the APbbox and APseg. For object detection, the mAP of Conformer-S/32 (55.4 M & 288.4 GFLOPs) is 3.7% higher than that of the FPN baseline (ResNet-101, 60.5 M & 295.7 GFLOPs). For instance segmentation, the mAP of Conformer-S/32 (58.1 M & 341.4 GFLOPs) is 3.6% higher than that of the Mask R-CNN baseline (ResNet-101, 63.2 M & 348.8 GFLOPs). This demonstrates the importance of global representations for high level tasks and suggests the great potential of Conformer to be a general backbone network.

4.4. Ablation Studies

**Number of Parameters.** The parameters of the proposed Conformer are combinations of the CNN and transformer branches. The parameter proportion of the two branches is a hyper-parameter to be experimentally determined. In Tab. 4, we evaluate performance of the two branches under different parameter settings. For the CNN branch, we tune the parameters of the CNN branch by changing the channels.
Table 3: Performance for object detection and instance segmentation on the MSCOCO minival set. † means the results are reported by the mm检测 library [10].

Table 4: Performance under different parameter proportions. $E$ and $d_h$ respectively denote the embedding dimensions and the head in the multi-head attention module in the transformer branch. $C$ and $n_e$ respectively represent the channels of $c_2$ and the bottleneck number within each convolution block in the CNN branch. $p_p$ is the proportion of CNN (including stem and FCUs) and transformer branch parameters.

and the number of bottlenecks, which respectively control the width and depth of the CNN branch. For the transformer branch, we tune the parameters by changing the numbers of embedding dimensions and heads. From Tab. 4, one can see that the accuracy is improved by increasing either parameters of the CNN or the transformer branch. More CNN parameters bring greater improvement while the computational cost overhead is lower.

**Dual Structure.** Conformer is a dual model, which is totally different from the serial hybrid ViT (CNN $\rightarrow$ Transformer) [16]. In Tab. 5, ResNet-26/50d & DeiT-S is a hybrid model which consists of ResNet-26/50d [19] and DeiT-S [41], where DeiT-S forms tokens upon the feature maps extracted by ResNet-26/50d. With comparable computational cost overhead, Conformer-S/32 outperforms the serial hybrid model although ResNet-26/50d can retain more local information within the stem stage.

**Positional Embeddings.** Considering that the CNN branch encodes both local features and spatial location information, the positional embeddings are assumed no longer required for Conformer. In Tab. 6, when the positional embedding is removed, the accuracy of DeiT-S decreases 2.4%, while that of Conformer-S decreases marginally (0.1%).

**Sampling Strategies.** In FCU, to make CNN-based feature maps coupling with Transformer-based patch embeddings, up/down-sampling operations are used to align the spatial scale. In Tab. 7, we compare different up/down-sampling strategies including Maxpooling, Avgpooling, convolution and attention-based sampling. Compared with Max/Avgpooling sampling, convolution and attention-based sampling methods use more parameters and computation cost but achieve comparable accuracy. We thereby choose the Avgpooling strategy.

3Refer to Supplementary material for detailed attention-based sampling.
Comparison with Ensemble Models. Conformer is compared with the ensemble models combining the outputs of CNN and transformer. For fair comparison, we use the same data augmentation and regularization strategies and the same training epochs (300) to train ResNet-101 [19], and combine it with the DeiT-S [41] model to form an ensemble model, and report the accuracy in Tab. 8. The accuracies of the CNN branch, the transformer branch, and the Conformer-S respectively reach 83.3%, 83.1%, and 83.4%. In contrast, the ensemble model (DeiT-S+ResNet-101) archives 81.8%, which is 1.6% lower than that of Conformer-S (83.4%), although it uses significantly more parameters and MACs.

4.5. Quantitative Evaluation

In Figs. 1 and 4, we qualitatively show that Conformer can learn global information by weakly-supervised object localization (WSOL), which can reflects the localization capability of models. The experiment is conducted on CUB-200-2011 dataset [43]. In Tab. 9, the localization performance of TS-CAM [17] with Conformer-S significantly outperforms those of CNN-based RCAM [54] and Transformer-based DeiT-S, which supports the claim that Conformer can better learn full object extent.

4.6. Generalization Capability

Rotation Invariance. To verify the generalization capability of the model in terms of rotation, we rotate test images by $0^\circ$, $60^\circ$, $120^\circ$, $180^\circ$, $240^\circ$ and $300^\circ$ and evaluate the performance of models trained under same data augmentation settings. As shown in Fig. 5(a), all models report comparable performance for images without rotation ($0^\circ$). For the rotated test images, the performance of ResNet-101 drops significantly. In contrast, Conformer-S reports higher performance, which implies stronger rotation invariance.

Scale Invariance. In Fig. 5(b), we compare the scale adaptation ability of Conformer with those of visual transformers (DeiT-S) and CNN (ResNet). We interpolate the positional embeddings of DeiT-S to adapt it to input images of different resolutions during inference. When the size of input images reduces from 224 to 112, DeiT-S’s performance drops by 25% and that of ResNet-50/152 drops by 15%. In contrast, the performance of Conformer drops only by 10%, demonstrating higher scale invariance of the learned feature representations.

5. Conclusion

We propose Conformer, the first dual backbone to combining CNN with visual transformer. Within Conformer, we leverage the convolution operators to extract local features and the self-attention mechanisms to capture global representations. We design the feature coupling unit to fuse local and the self-attention mechanisms to capture global representations. We demonstrate the potential to be a simple yet effective backbone network.
References


[52] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junskul Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In IEEE ICCV, pages 6023–6032, 2019. 6


