Conformer: Local Features Coupling Global Representations for Visual Recognition (Supplemental materials)

Zhiliang Peng[1] Wei Huang[1] Shanzhi Gu[3] Lingxi Xie[2] Yaowei Wang[3]
Jianbin Jiao[1] Qixiang Ye[1,3]
[1]University of Chinese Academy of Sciences, Beijing, China [2]Huawei Inc.
[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

1. Model Architectures

The architectures of Conformer-Ti/B are detailed in Tab. 5. Compared with Conformer-S, Conformer-Ti reduces channel number of the CNN branch by 1/4, and Conformer-B increases channel number in the CNN branch, head number of the multi-head attention module and the embedding dimensions in the transformer branch by 1.5.

2. Attention-based Sampling

We also design a down-sampling-up-sampling strategy based on the cross attention between feature maps and patch embeddings.

Let h, w and c respectively denote the height, width, channel of feature maps in a block (we omit the batch dimension here for simplicity), K and E respectively represent the number of patch embeddings (termed P_t) and channel dimension in the transformer branch. We split the feature maps into K patches ($e.g.$, 14×14), termed P_c. The dimension of each patch is $n \times c$. After aligning the channel dimension by 1×1 convolution, the shape of each patch is $n \times E$.

For down sampling, the fusion between patch i in P_c (denoted P_c^i) and patch j in P_t (denoted P_t^j) is formulated as

$$P_t^j = P_t^j + \text{Softmax}\left(\frac{(P_t^j W_q)(P_c^i W_k)^T}{\sqrt{E}}\right)(P_c^i W_v), \quad (1)$$

where $W_q, W_k, W_v \in \mathbb{R}^{E\times E}$ are learned linear transformations which map the input P_t^j to queries Q, keys K and values V, respectively.

For up sampling, we re-use the attention weights in Eq. 1 and formulate the process as

$$\tilde{P}_c^i = \tilde{P}_c^i + \text{Softmax}\left(\frac{(P_t^j W_q)(P_c^i W_k)^T}{\sqrt{E}}\right)^T \tilde{P}_t^j, \quad (2)$$

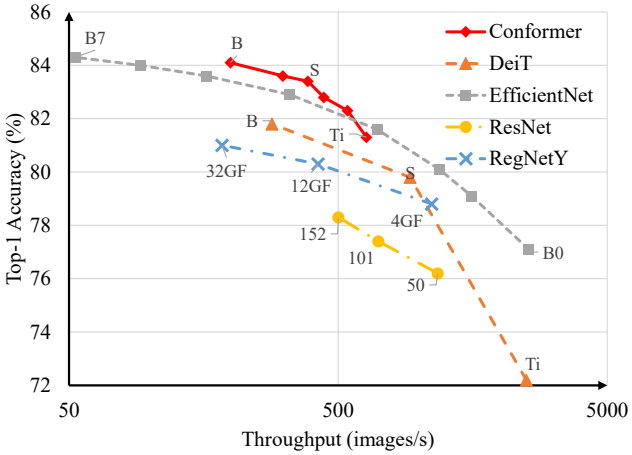

Figure 1: Throughput and accuracy on ImageNet of Conformer compared to DeiT [7], ResNet [2], RegNetY [5] and EfficientNet [6]. The throughput is measured as the number of images processed per second on a 32GB V100 GPU.

where $\tilde{P}_c^i$ and $\tilde{P}_c^i$ respectively denote that P_c^i is processed by convolution layers and P_t^j by a transformer block (Fig. 2 in the paper).

3. Inference Time

Classification. Following DeiT [7], we evaluate and compare the throughput of various methods in Fig. 1. One can see that our Conformer outperforms EfficientNet [6] under comparable throughput.

Object detection and instance segmentation. Similarly, we measure Frame Per Second (FPS) as the inference speed and show the comparison in the Tab. 1. Combining Tab.3 in the paper and Tab. 1 here, compared with ResNet-101 [2], Conformer-S/32 has the comparable parameters, GFLOPs and inference speed, but can outperform ResNet-101 by

Method	Backbone	#Params (M)	GLOPs	FPS
FPN	ResNet-50	41.5	215.8	20.2
	ResNet-101	60.5	295.7	15.9
	Conformer-S/32	55.4	288.4	13.5
	Conformer-S	54.2	404.6	8.2
Mask R-CNN	ResNet-50	44.2	268.9	13.2
	ResNet-101	63.2	348.8	11.5
	Conformer-S/32	58.1	341.4	10.9
	Conformer-S	56.9	457.7	7.1

Table 1: Comparison of inference time. FPS is measured on a 32GB V100 GPU with batchsize 1.

Variants	Conformer-S	w.o. Trans	w.o. Conv	w.o. FCU
Top-1 Acc.	83.4%	73.9%	79.8%	80.2%(-3.2%)

Table 2: Ablation study for different parts in Conformer-S.

Index	#Params (M)	MACs (G)	Accuracy (%)
1	8.6	9.2	73.9
2	37.0	10.8	80.8
3	22.1	4.6	79.8
4	28.9	6.0	80.2
Conformer-S	37.7	10.6	83.4

Table 3: Performance of Conformer sub-structures. Where the index 1, 2, 3 and 4 respectively represent the sub-structures shown in Figs. 3(b), (c), (d) and (e).

a significant margin on both object detection and instance segmentation tasks, which further demonstrates the potential to be a general backbone network.

4. More Ablation Studies

FCU's Effect. We compare three important baseline variants, Conformer w.o. Transformer branch, Conformer w.o Convolution branch, Conformer w.o FCU (just ensemble CNN and ViT results). The results are summarized in Tab. 2. In Tab. 2, one can see that without FCU the performance drops by 3.2%, which suggests that the proposed FCU performs vital effect on fusing local features and global representations.

Residual Structure. As shown in Fig. 3 in the paper, by considering FCUs as short connection we abstract Conformer with a dual structure to a serial structure with residual connections. In other words, under different residual connections, Conformer can degenerate to different sub-structures. We test some sub-structures and report the corresponding performance in Tab. 3. From Tab. 3, one can see that the proposed residual structure outperforms other sub-structures.

Number of heads in MHSA. We conducted ablation study with Conformer-Ti and concluded that the Top-1 accuracies on ImageNet val set respectively are 81.0%, 81.3%, and 81.8% when the head numbers are 3, 6, and 12.

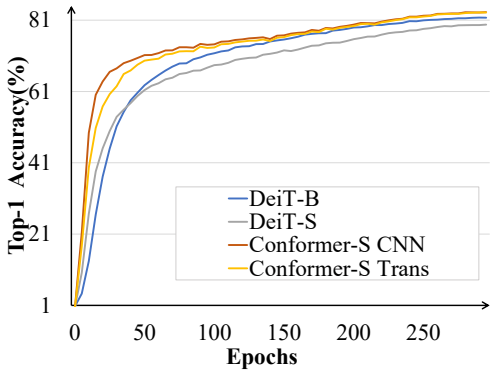

Figure 2: Training Accuracy on the val set.

Fusion Interval. In the paper, we proposed a Feature Coupling Unit to interact the local features and global representations in each block to progressively align the features to fill the semantic gap. To validate whether fusion should be done in each block, we conduct experiments on fusion intervals and report the performance on ImageNet in Tab. 4. From Tab. 4, one can see that smaller fusion intervals report higher performance, implying that frequent interaction facilities the representation learning.

Interval	#Params (M)	MACs (G)	Accuracy (%)
1	37.7	10.6	83.4
2	34.2	9.2	82.9
4	32.3	8.4	82.2

Table 4: Comparison of fusion intervals. 1, 2 and 4 respectively represent performing fusion every 1, 2 and 4 block(s).

5. Convergence speed

For the convolution operations introduced, Fig. 2, both the CNN branch and the transformer branch of Conformer-S significantly outperforms DeiT during the first 50 epochs. This demonstrates the inductive bias of convolution facilities the convergence of visual transformers.

6. Visualization

6.1. Feature Maps

Different from Fig. 1 in the paper, we summarize all the channels of feature maps or patch embeddings and show them in the Fig. 3. As shown in Fig. 3, Compared with ResNet, Conformer's CNN branch preserves clear information of foreground. Compared with DeiT, Conformer's transformer branch retains more local details while depressing the background.

	CNNTransformer-Ti					**CNNTransformer-B**				
stage	output	**CNN Branch**	**fcuc**	**Transformer Branch**		output	**CNN Branch**	**fcuc**	**Transformer Branch**	
c1	112×112	7×7, 64, stride 2				112×112	7×7, 64, stride 2			
	56×56	3×3 max pooling, stride 2				56×56	3×3 max pooling, stride 2			
c2	56 × 56,197	$\begin{bmatrix} 1\times1, 16 \\ 3\times3, 16 \\ 1\times1, 64 \end{bmatrix}$	-	4×4, 384, stride 4 $\begin{bmatrix} \text{MHSA-6, 384} \\ 1\times1, 1536 \\ 1\times1, 384 \end{bmatrix}$	×1	56 × 56,197	$\begin{bmatrix} 1\times1, 96 \\ 3\times3, 96 \\ 1\times1, 384 \end{bmatrix}$	-	4×4, 384, stride 4 $\begin{bmatrix} \text{MHSA-9, 576} \\ 1\times1, 2304 \\ 1\times1, 576 \end{bmatrix}$	×1
		$\begin{bmatrix} 1\times1, 16 \\ 3\times3, 16 \\ 1\times1, 64 \end{bmatrix}$	$[1 \times 1, 384] \longrightarrow$	$\begin{bmatrix} \text{MHSA-6, 384} \\ 1\times1, 1536 \\ 1\times1, 384 \end{bmatrix}$	×3		$\begin{bmatrix} 1\times1, 96 \\ 3\times3, 96 \\ 1\times1, 384 \end{bmatrix}$	$[1 \times 1, 576] \longrightarrow$	$\begin{bmatrix} \text{MHSA-9, 576} \\ 1\times1, 2304 \\ 1\times1, 576 \end{bmatrix}$	×3
		$\begin{bmatrix} 1\times1, 16 \\ 3\times3, 16 \\ 1\times1, 64 \end{bmatrix}$	$\longleftarrow [1 \times 1, 16]$				$\begin{bmatrix} 1\times1, 96 \\ 3\times3, 96 \\ 1\times1, 384 \end{bmatrix}$	$\longleftarrow [1 \times 1, 96]$		
c3	28 × 28,197	$\begin{bmatrix} 1\times1, 32 \\ 3\times3, 32 \\ 1\times1, 128 \end{bmatrix}$	$[1 \times 1, 384] \longrightarrow$	$\begin{bmatrix} \text{MHSA-6, 384} \\ 1\times1, 1536 \\ 1\times1, 384 \end{bmatrix}$	×4	28 × 28,197	$\begin{bmatrix} 1\times1, 192 \\ 3\times3, 192 \\ 1\times1, 768 \end{bmatrix}$	$[1 \times 1, 576] \longrightarrow$	$\begin{bmatrix} \text{MHSA-9, 576} \\ 1\times1, 2304 \\ 1\times1, 576 \end{bmatrix}$	×4
		$\begin{bmatrix} 1\times1, 32 \\ 3\times3, 32 \\ 1\times1, 128 \end{bmatrix}$	$\longleftarrow [1 \times 1, 32]$				$\begin{bmatrix} 1\times1, 192 \\ 3\times3, 192 \\ 1\times1, 768 \end{bmatrix}$	$\longleftarrow [1 \times 1, 192]$		
c4	14 × 14,197	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}$	$[1 \times 1, 384] \longrightarrow$	$\begin{bmatrix} \text{MHSA-6, 384} \\ 1\times1, 1536 \\ 1\times1, 384 \end{bmatrix}$	×3	14 × 14,197	$\begin{bmatrix} 1\times1, 384 \\ 3\times3, 384 \\ 1\times1, 1536 \end{bmatrix}$	$[1 \times 1, 576] \longrightarrow$	$\begin{bmatrix} \text{MHSA-9, 576} \\ 1\times1, 2304 \\ 1\times1, 576 \end{bmatrix}$	×3
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}$	$\longleftarrow [1 \times 1, 64]$				$\begin{bmatrix} 1\times1, 384 \\ 3\times3, 384 \\ 1\times1, 1536 \end{bmatrix}$	$\longleftarrow [1 \times 1, 384]$		
c5	7 × 7,197	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}$	$[1 \times 1, 384] \longrightarrow$	$\begin{bmatrix} \text{MHSA-6, 384} \\ 1\times1, 1536 \\ 1\times1, 384 \end{bmatrix}$	×1	7 × 7,197	$\begin{bmatrix} 1\times1, 384 \\ 3\times3, 384 \\ 1\times1, 1536 \end{bmatrix}$	$[1 \times 1, 576] \longrightarrow$	$\begin{bmatrix} \text{MHSA-9, 576} \\ 1\times1, 2304 \\ 1\times1, 576 \end{bmatrix}$	×1
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}$	$\longleftarrow [1 \times 1, 64]$				$\begin{bmatrix} 1\times1, 384 \\ 3\times3, 384 \\ 1\times1, 1536 \end{bmatrix}$	$\longleftarrow [1 \times 1, 384]$		
Parameters	**23.5 M**					**83.3 M**				
MACs	**5.2 G**					**23.3 G**				

Table 5: Architecture of CNNTransformer-Ti and CNNTransformer-B, where MHSA-6/9 denotes the multi-head self-attention with heads 6/9 in transformer block and the fc layer is viewed as 1×1 convolution here. And in output column, 56×56,197 respectively mean the size of feature map is 56×56 and the number of embedded patches is 197.

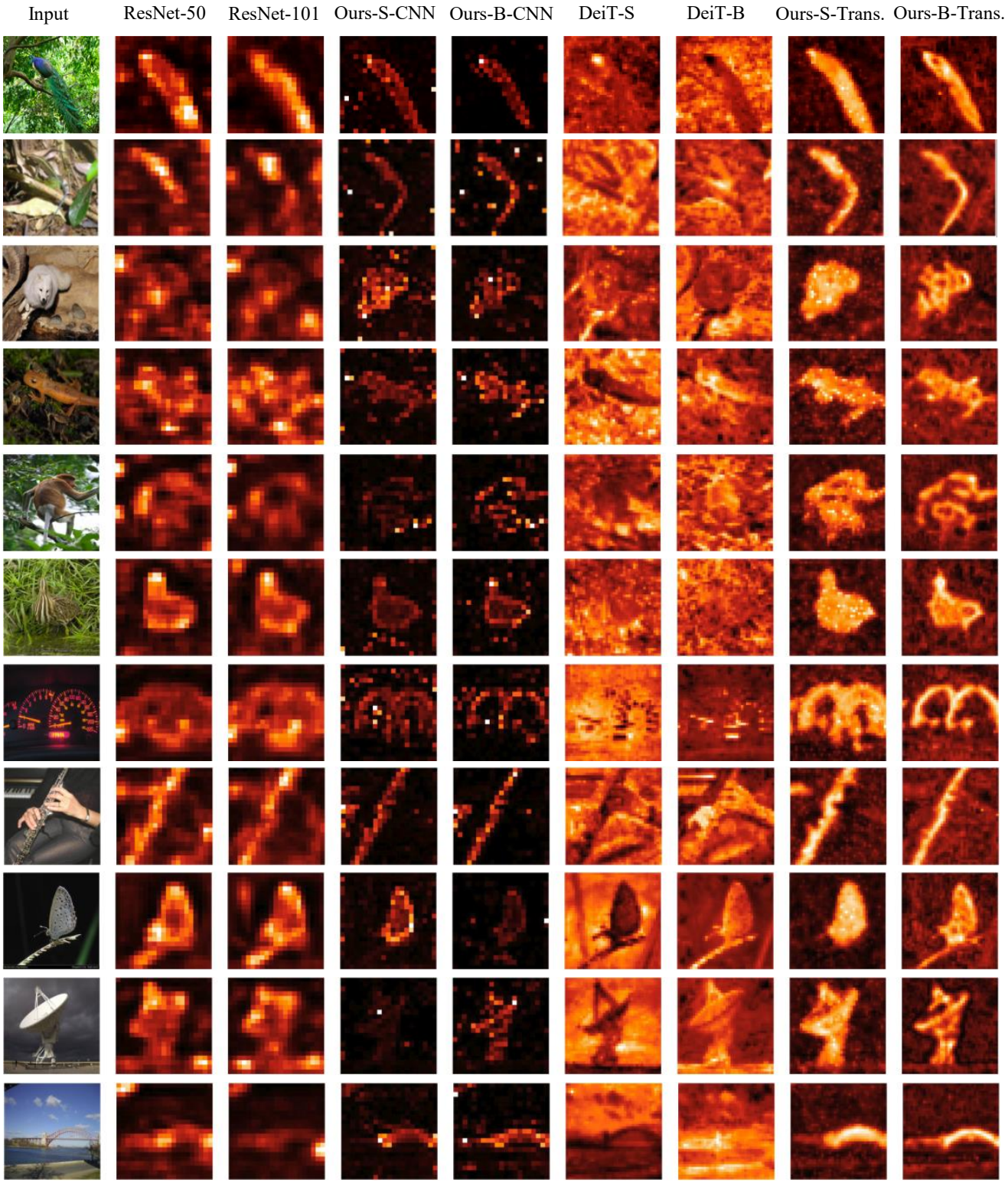

Figure 3: Feature maps. The feature maps are obtained by summarizing of all channels of the last convolutional layer or transformer block. Ours-S-CNN denotes the CNN branch of Conformer-S model, and Ours-S-Trans. denotes the transformer branch of Conformer-S. (Best viewed in color)

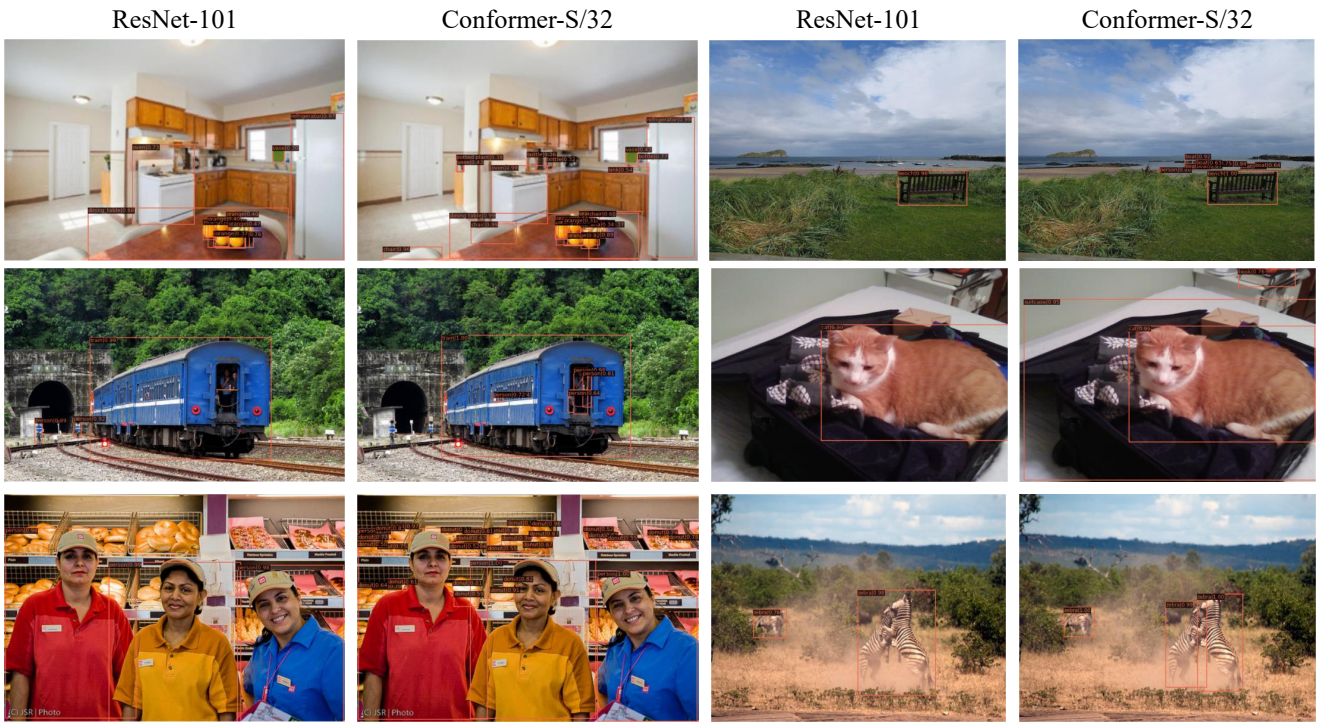

Figure 4: Object detection examples on the minival set of MSCOCO [4], based on FPN [3].

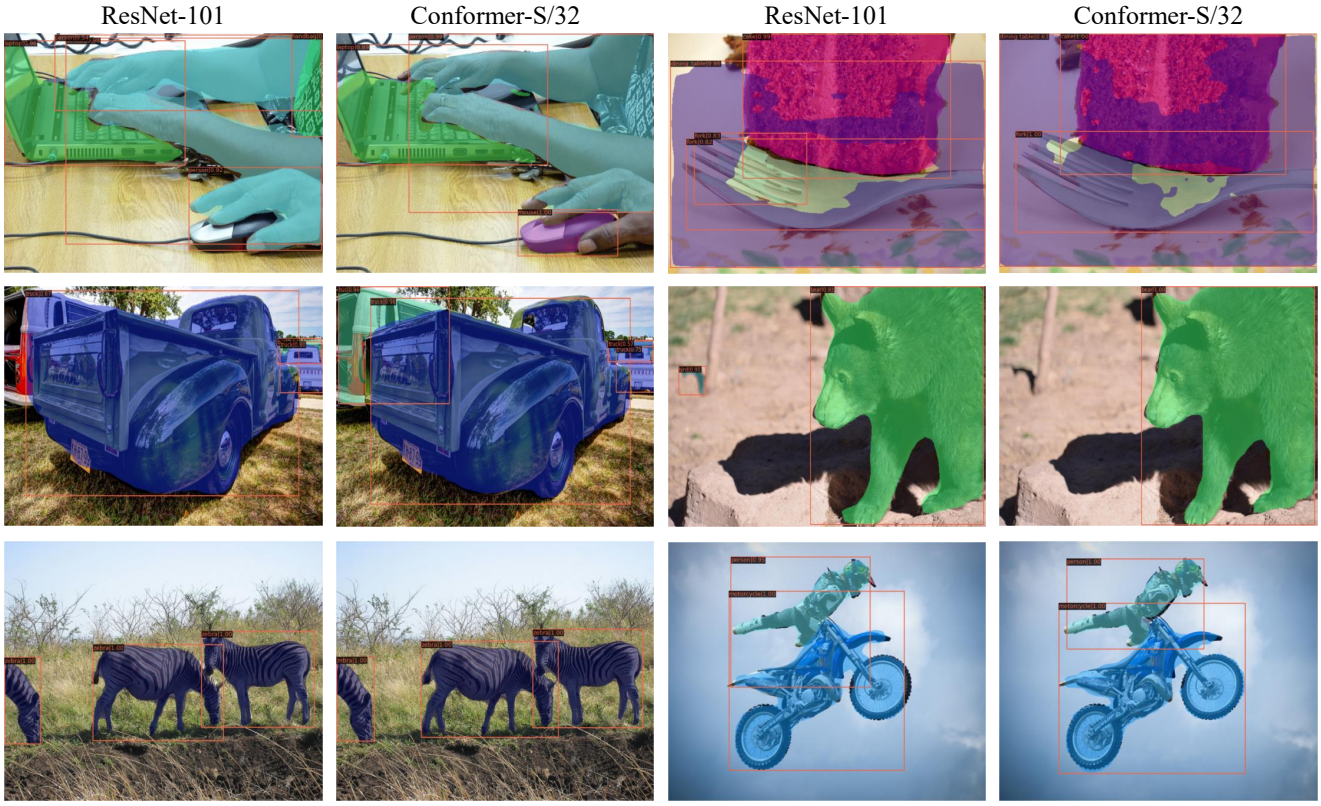

Figure 5: Instance segmentation examples on the minival set of MSCOCO [4], based on Mask R-CNN [1].

6.2. Object Detection and Instance Segmentation

We respectively visualize object detection examples and instance segmentation examples in Fig. 4 and Fig. 5. One can see that Conformer, by taking advantage of global representations, reports better results on large and/or slender objects.

References

[1] Kaiming He, Georgia Gkioxari, Piotr Dollár, and B. Ross Girshick. Mask r-cnn. In *IEEE ICCV*, pages 386–397, 2017. 5

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016. 1

[3] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE CVPR*, pages 936–944, 2017. 5

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 5

[5] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. *arXiv preprint arXiv:2003.13678*, 2020. 1

[6] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019. 1

[7] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1