Supplementary Materials

1. Training Details

For STL10 dataset [3], we resize all input images to 128×128. All methods are trained from scratch for 300 epochs. Drop out rate is set to 0.2 and SGD with momentum equal to 0.9 is used as the optimizer. The initial learning rate is set to 0.01 which is divided by 5 at the 180^{th}, 240^{th} and 270^{th} epoch, respectively.

For both VWW [2] and ImageNet [4], the input images are resized to 224×224. For VWW, all the pooling methods are trained for 100 epochs, while for ImageNet, we train for 150 epochs. For VWW, we use the SGD optimizer (with momentum equal to 0.9) with an initial learning rate of 0.005, which is divided by 5 at the 70^{th} and 90^{th} epoch respectively. For ImageNet, we use the same SGD optimizer with an initial learning rate of 0.05 and cosine annealing.

For COCO dataset [9], we resize all input images to 1333×800 by keeping the original ratio and random flip each input image by 50% probability. Each backbone network is initialized by weight parameters pre-trained on ImageNet dataset [4] and all methods are trained for 12 epochs. SGD is used as the optimizer and the initial learning rate is set to 0.02. The momentum is set to 0.9 and weight decay rate is set to 0.0001.

2. Combine with other CNNs based Backbones

We combine the proposed and other state-of-the-art pooling methods with different CNNs based backbones, MobileNetV3 [7] and ResNeXt [14], to illustrate the robustness of our method, as shown in Table 1. Specifically, for MobileNetV3, we use the MobileNetV3_Small architecture as the backbone and keep the same stride settings with [7]. For ResNeXt, we use the ResNeXt18 architecture as the backbone and keep the same stride settings with [14]. For these two backbones, different pooling methods are both embedded into the backbones by inner stage pooling style. The proposed method still achieves state-of-the-art performance when combining with these two backbones.

Table 1. Comparison on STL10 dataset.

Metrics Methods	Top 1 (%)
Strided Conv.-MobileNetV3_Small	65.79
LIP-MobileNetV3_Small	67.33
GaussianPool-MobileNetV3_Small	62.90
Ours-MobileNetV3_Small	**69.46**
Strided Conv.-ResNeXt18	78.31
LIP-ResNeXt18	79.60
GaussianPool-ResNeXt18	79.53
Ours-ResNeXt18	**79.96**

Table 2. Comparison with Transformer based Methods.

Metrics	Top 1 (%)	Params	FLOPs (G)
LVT [15]	74.8	8.9M	0.900
MobileFormer [1]-26M	64.0	3.2M	0.026
MobileFormer-52M	68.7	3.5M	0.052
MobileFormer-96M	72.8	4.6M	0.096
MobileFormer-151M	75.2	7.6M	0.151
MobileFormer-214M	76.7	9.4M	0.214
MobileFormer-294M	77.9	11.4M	0.294
MobileNetV2	71.9	3.5M	0.303
Ours-MobileNetV2	72.88	3.8M	0.272
Ours-MobileNetV2-0.35x	60.92	0.31M	0.060

3. Comparison with Transformer based Methods

We compare the test accuracy, parameter count, and FLOPs of our proposed pooling technique with a few transformer-based models in Table 2, since both use the multi-head self-attention block to capture non-local features for image recognition. While these transformer-based models reduce the FLOPs and parameter count compared to traditional transformers [5], it is still difficult to deploy them in tiny micro-controllers, where the on-chip memory is typically constrained to few 100 KBs and the flash memory is typically constrained to only a few MBs. Moreover, our approach surpasses these transformer-based models in terms of accuracy-memory trade-off. For example, the Mobileformer-52M model achieves a top-1 accuracy of 68.7% with 3.5M parameters. With similar number of parameters, our approach achieves a top-1 accuracy of 72.88%. Our approach achieves 60.92% top-1 accuracy with only 0.3M parameters (memory footprint can be further reduced via $2\times$ down-sampling albeit with $\sim 9\%$ accuracy drop), while the smallest transformer-based model requires 3.2M parameters, which can never fit into tiny micro-controllers.

4. Qualitative Results for Object Detection

The SSD [10] object detection results with the COCO dataset [9] evaluated on the ResNet18 [6] backbone for the different pooling techniques are shown in Fig. 2. Note that the ground truth bounding boxes have violet colored edges, while the predicted bounding boxes have orange colored edges. Our results indicate that our self-attentive pooling indeed has the best detection precision. Specifically, in the first, second, and third row, our method detects more people compared to other pooling techniques, which illustrates the

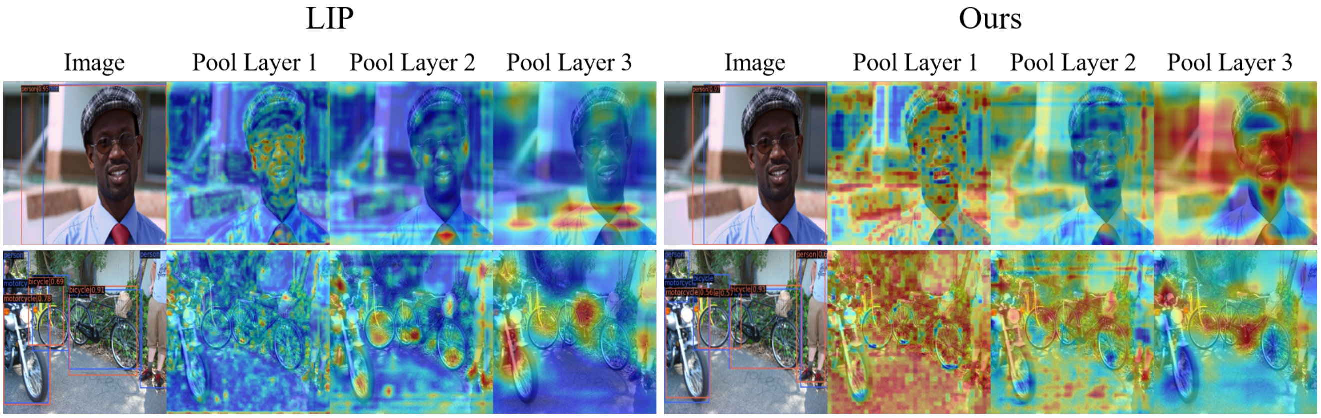

Figure 1. Visualization results on the COCO dataset of the SSD-ResNet18 network based on LIP pooling and the proposed pooling techniques. The left portion show the results of LIP, while the right portion show the results of the proposed methods. For each pooling method, the first column displays the detected results. The second, third and fourth columns render the heatmaps of pooling weights from different pooling layers. Specifically, Pool Layer 1, 2, 3 are from the shallow, middle and deep layer, respectively, i.e., the sensitive field becomes larger and larger from Pool Layer 1 to 3.

superiority of our method. In the fourth row and eighth row, only our pooling technique can detect the pedestrians with extremely small bounding boxes in the rail track and platform respectively. This might be because our method can model the long-range dependencies between different objects, such as between a person and a train. The non-local self-attention map *might pay more attention* to the relationship between these objects, even when they appear small, and thus retain their information when pooling. In the fifth, sixth, and seventh row, our method achieves higher intersection over union (IoU) compared to other methods, which also can be attributed to a similar line of reasoning.

We also render the heatmaps of pooling weights from different pooling layers with different sensitive fields, as shown in Fig. 1. Compared with locality based pooling method, our method is more concerned about contextual information in shallow pool layer. When the locality based pooling method drops the importance of most background pixels, the proposed method reserves the attention for most pixels. In middle and deep layer with larger sensitive fields, the proposed method mainly focuses on the relationship between different objects or between the objects with the background, while the locality based pooling method only focuses on some local regions inner the single object.

5. Model Architectures & Frameworks

MobileNetV2 [13]: A lightweight depthwise convolution neural network that has gained significant traction for being deployed on resource-constrained edge devices, such as mobile devices. It consists of 7 stages with total 17 inverted residual blocks. The proposed method is combined with MobileNetV2 by inner stage pooling as described in Section 5. We keep the same pooling settings except the first pooling layer with MobileNetv2 [13], that strides for each stage are $(1, 2, 2, 2, 1, 2, 1)$, respectively. Specifically, we use strides $(s1, 2, 2, 2, 1, 2, 1)$, where $s1 \in \{1, 2, 4\}$. We also evaluate the pooling methods on MobileNetV2-0.35x [12], which shrinks the output channel count by 0.35× to satisy the compute budget of 30M floating point operations (FLOPs) representing standard micro-controllers.

ResNet18 [6]: A deep convolutional neural network widely used as backbone for feature extraction on image recognition and object detection tasks. It consists of 4 stages with total 8 residual blocks. The proposed method is combined with ResNet18 by outer stage pooling, as described in Section 5. We keep the same pooling settings except the first pooling layer with ResNet18 [6], that strides for each stage are $(1, 2, 2, 2)$, respectively. Specifically, the pooling strides used by us are $(s1, 2, 2, 2)$, where $s1 \in \{1, 2, 4\}$. Note that the three different values of $s1$ simulate the different amount of down-sampling in the initial activation maps, capturing models with significantly different memory footprints.

SSD [10]: A single shot and end-to-end framework with an excellent memory-accuracy trade-off for standard detection benchmarks. In this work, we use MobileNetV2 [13] and ResNet18 as the backbone networks of SSD.

Faster R-CNN [11]: A two-stage object detection framework that consists of a feature extraction, a region proposal, and a RoI pooling module. For our experiments, we use ResNet18 as the backbone network for feature extraction, since MobileNetV2 significantly degrades the test mAP.

6. Dataset Details

STL-10 [3]: The STL-10 dataset is an image recognition dataset with the same 10 classes as CIFAR-10 [8], but

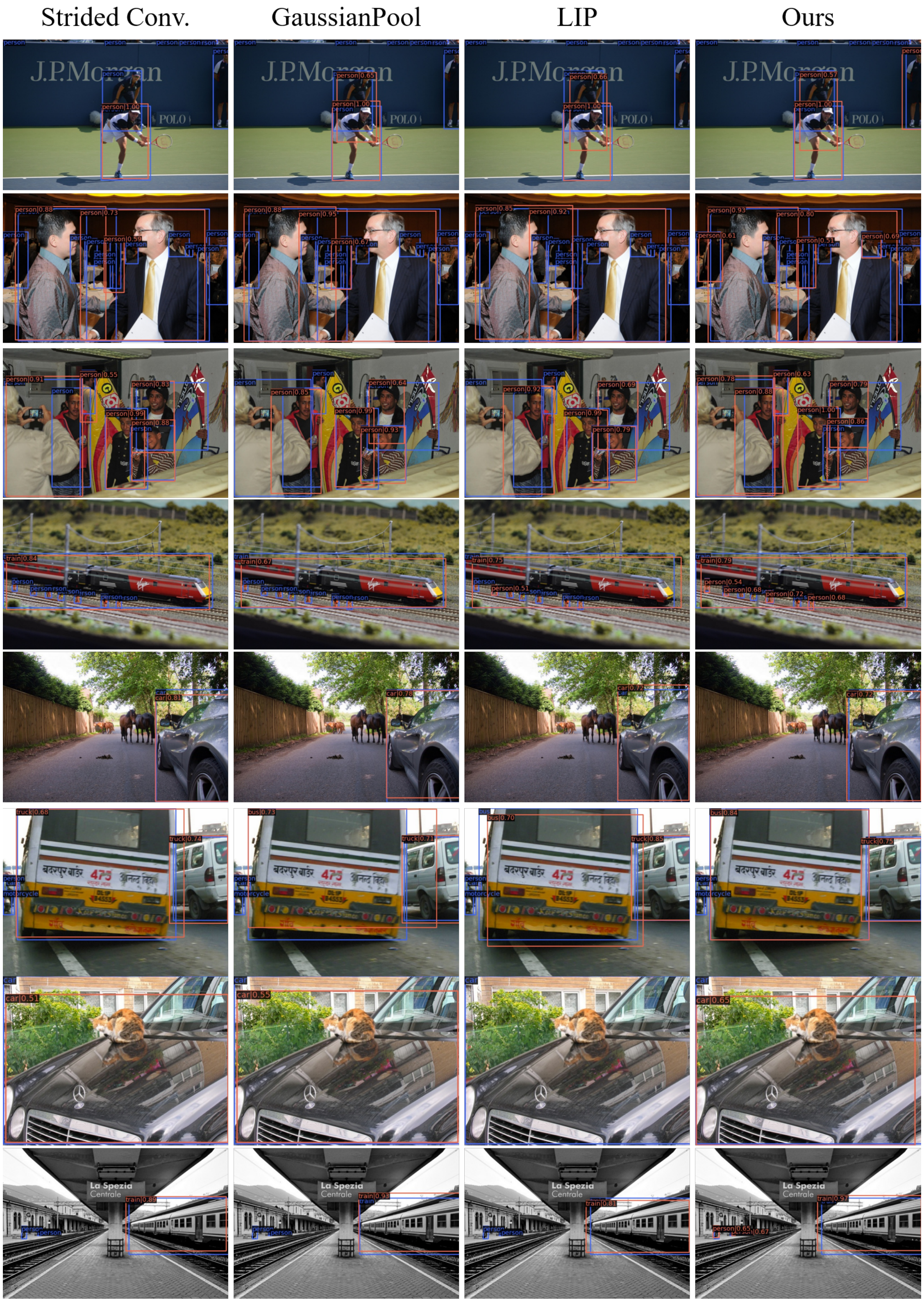

Figure 2. The results on COCO dataset of the SSD-ResNet18 network based on different pooling techniques. Blue boxes indicate the ground-truth bounding boxes. Red boxes indicate the detection results.

each class has fewer labeled training examples than CIFAR-10. In this work, we use an image resolution of 128×128 (instead of the traditional 96×96) to evaluate the aggressively down-sampling of the initial activation maps, so that the spatial dimensions do not vanish before applying the classifier layer(s).

VWW [2]: The Visual Wake Words (VWW) dataset consists of high resolution images that include visual cues to "wake-up" AI-powered resource-constrained home assistant devices that require real-time inference. The goal of the VWW challenge is to detect the presence of a human in the frame (a binary classification task with 2 labels) with very little resources - close to 250KB peak RAM usage and model size, which is only satisfied by MobileNetV2-0.35x, and hence, used in our experiments. [2]. In this work, we use a VWW image resolution of 224×224 [2].

ImageNet [4]: The ILSVRC-2012 ImageNet [4] is an image recognition dataset with 1k classes and 1.3M images, which is widely used as a benchmark to pre-train backbone networks for various down stream tasks, such as object detection. In this work, we use an image resolution of 224×224 for ImageNet, the same as used in the original paper [4]. To emulate a memory-constrained platform, we use MobileNetV2-0.35x to evaluate the pooling methods.

References

[1] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5270–5279, 2022.

[2] Aakanksha Chowdhery, Pete Warden, Jonathon Shlens, Andrew Howard, and Rocky Rhodes. Visual wake words dataset. *arXiv preprint arXiv:1906.05721*, 2019.

[3] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 248–255, 2009.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[8] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[9] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[12] Oindrila Saha, Aditya Kusupati, Harsha Vardhan Simhadri, Manik Varma, and Prateek Jain. RNNPool: Efficient non-linear pooling for RAM constrained inference. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20473–20484. Curran Associates, Inc., 2020.

[13] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[14] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[15] Chenglin Yang, Yilin Wang, Jianming Zhang, He Zhang, Zijun Wei, Zhe Lin, and Alan Yuille. Lite vision transformer with enhanced self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11998–12008, 2022.