TREE-STRUCTURED ATTENTION MODULE FOR IMAGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies in attention modules have enabled higher performance in computer vision tasks by capturing global contexts and accordingly attending important features. In this paper, we propose a simple and highly parametrically efficient module named *Tree-structured Attention Module* (TAM) which recursively encourages neighboring channels to collaborate in order to produce a spatial attention map as an output. Unlike other attention modules which try to capture long-range dependencies at each channel, our module focuses on imposing non-linearities between channels by utilizing point-wise group convolution. This module not only strengthens representational power of a model but also acts as a gate which controls signal flow. Our module allows a model to achieve higher performance in a highly parameter-efficient manner. We empirically validate the effectiveness of our module with extensive experiments on CIFAR-10/100 and SVHN datasets. With our proposed attention module employed, ResNet50 and ResNet101 models gain 2.3% and 1.2% accuracy improvement with less than 1.5% parameter overhead. Our PyTorch implementation code is publicly available.

1 INTRODUCTION

Advancements in attention modules have boosted up the performance where they are employed over broad fields in deep learning such as machine translation, image generation, image and video classification, object detection, segmentation, etc(Vaswani et al., 2017; Wang et al., 2017; Hu et al., 2018a;b;c; Park et al., 2018; Woo et al., 2018; Wang et al., 2018; Cao et al., 2019; Zhang et al., 2019). In the fields of computer vision tasks, numerous attention modules have been proposed in a way that one can attach it to a backbone network obtaining an efficient trade-off between additional parameters of the attached attention module and the model's performance. SENet(Hu et al., 2018b) encodes global spatial information using global average pooling and captures channel-wise dependencies using two fully-connected layers over the previously encoded values at each channel. Input feature maps of the SE module are recalibrated with output values corresponding to each channel after applying a sigmoid activation function to produce output feature maps of the module. In this manner, the model can distinguish which channels to attend than others. GENet(Hu et al., 2018a) shows simply gathering spatial information with depth-wise strided convolution and redistributing each gathered value across all positions with nearest neighbor upsampling can significantly help a network to understand global feature context. NLNet(Wang et al., 2018) aggregates query-specific global context and adds values to each corresponding channel. GCNet(Cao et al., 2019) simplifies NLNet in a computationally efficient way using the fact that a non-local block used in the NLNet tends to produce attention map independent of query position. BAM(Park et al., 2018) efficiently enhances backbone networks by placing attention modules in bottleneck regions, which requires few increase in both parameters and computation. CBAM(Woo et al., 2018) incorporates channel and spatial attentions and employs a max descriptor as well as an average descriptor for more precise attention.

It is clear that proposed modules in aforementioned studies have brought remarkable results, most of their main focus has been on *how to capture long-range dependencies across spatial dimension*. That is, they mainly focus on contextual modeling rather than capturing inter-channel relations both of which are regarded indispensable for an attention module as depicted in Cao et al. (2019). In this work, we propose a module which strengthens model representational power by imposing non-linearities between neighboring channels in a parameter efficient manner. While this work deviates

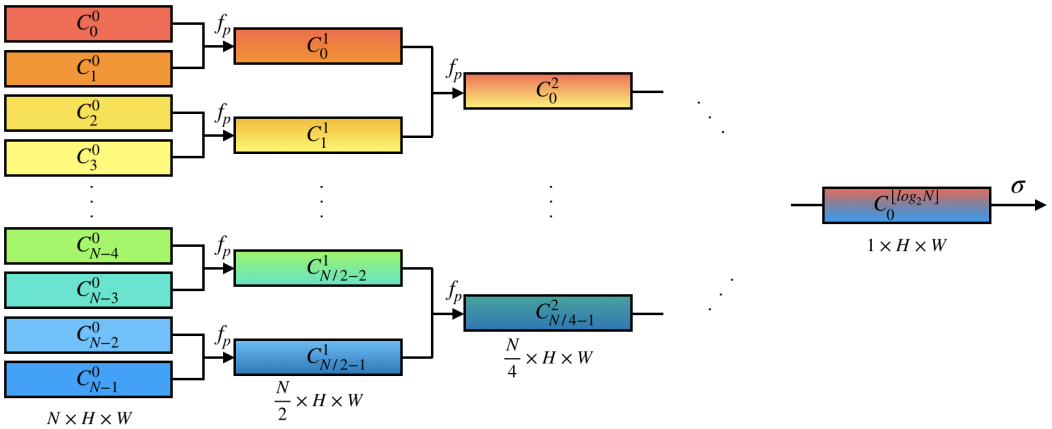


Figure 1: An instance of our proposed module with group size 2. f_p denotes a point-wise convolution followed by an activation function which combines neighboring channels. C_n^m denotes a n-th channel after applying m point-wise group convolutions to the input feature map. One channel attention map followed by a sigmoid σ is produced. A color refers to information a channel contains. The repetition of point-wise group convolution yields a tree-like structure.

from the current trend of capturing long-range dependencies within spatial dimension, we argue that taking consideration of inter-channel relations can also achieve highly competitive results **even without capturing any kind of spatial dependencies**. Our module incorporates all channels to produce a single meaningful attention map as an output whereas most previous studies restore the input channel dimension in order to attend important channels and to suppress less meaningful ones. For this, we repeatedly apply light-weight point-wise group convolution with a fixed group size to an input feature map until the number of channels becomes one. While the increased parameters and computation are almost negligible, we find this simple design remarkably boosts up the performance of various backbone networks. As we see in section 3, the module performance is highly competitive to other attention modules and enhances baseline models with few additional parameter overhead. This gives one a clue to another notion for attention deviating from the current trend of taking global context.

Our contributions are two-fold:

- we propose *Tree-structured Attention Module* (TAM) which allows the network to learn inter-channel relationships using light-weight point-wise group convolutions. This tree-structure enables convolution filters in the mid and later phase of a network to have a higher variance so that it can have more presentation power.
- by proving validity of TAM with extensive experiments, we highlight the potential importance of inter-channel relations.

2 Tree-structured Attention Module

The *Tree-structured Attention Module* is a simple and light-weight module which helps the network to learn inter-channel relations by repeating a pairing function f_p. Given an input feature map X and intermediate feature map Y of the module, $f_p : X \rightarrow Y, X \in \mathbb{R}^{N \times H \times W}, Y \in \mathbb{R}^{\frac{N}{g} \times H \times W}$ where N, H, W, g means the number of input channels, height, width, and group size, respectively. Let $X = [C_0^0, C_1^0, \ldots, C_{N-1}^0]$, then the output Z of the module can be described by

$$Z = f_p^{(\lfloor log_g N \rfloor)}(W, X) = C_0^m \in \mathbb{R}^{1 \times H \times W} \tag{1}$$

where $f_p^{(m)}(W, X) = f_p(W_{m-1}, f_p^{(m-1)}(X)) = f_p(W_{m-1}, f_p(W_{m-2}, f_p(\ldots f_p(W_0, X))))$ for m repetitive operations of f_p with a fixed group size g. To ensure the module to look through all

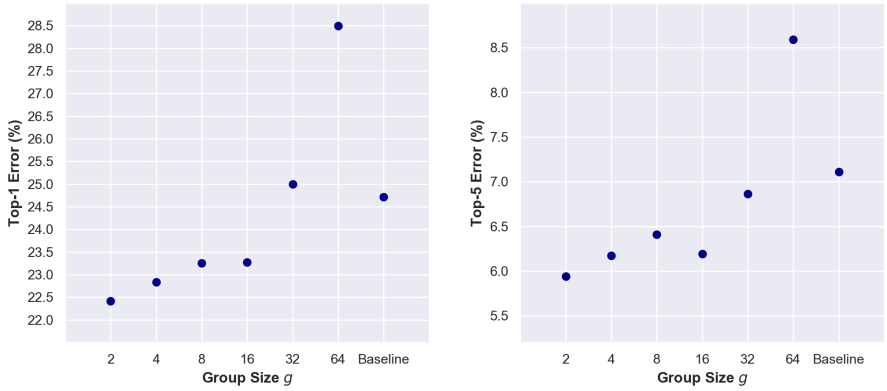


Figure 2: Top-1/5 errors (%) of our module with different group sizes on CIFAR-100 dataset. We use a ResNet50 as a backbone network (denoted as baseline). A clear trend is observed that an error decreases according to smaller group size g. For top-1 error, the case of $g = 2$ reaches 22.42%. For comparison, the original ResNet101 yields a result of 23.17% as shown in Table 1.

channels we repeat f_p until the number of remaining channels is not divisible by the factor g. Then, we apply 1x1 convolution followed by sigmoid function in order to produce one channel output. We replicate the output through channel axis to restore the original input dimension in order to recalibrate the input feature map by element-wise multiplication. Figure 1 shows an instance of our model with $g = 2$. To permit a parametric efficiency, we adopt point-wise group convolution for f_p.

Formally,

$$\begin{aligned} f_p^{(1)}(W, X) &= f_p(W_0, X) \\ &= \delta([f_{1\times1}(W_{0,0}, C_0, \ldots, C_{g-1}), f_{1\times1}(W_{0,1}, C_g, \ldots, C_{2g-1}), \ldots, \\ &\quad f_{1\times1}(W_{0,\frac{N}{g}-1}, C_{N-g-1}, \ldots, C_{N-1})]) \end{aligned}$$

where $W_i = \{W_{i,j} \in \mathbb{R}^{1\times g\times1\times1} | j = \{0, \ldots, \frac{N}{g} - 1\}\}$, $f_{1\times1}(.)$ and $\delta(.)$ refers to point-wise convolution and non-linear activation. For activation we use channel-wise PReLU function(He et al., 2015b).

Unlike an ordinary point-wise convolution which increases the model parameter and computation proportional to $N_{input} \times N_{output}$, a point-wise group convolution increases them proportional to $N_{input} \times N_{output}/n_{groups}$ where n_{groups} refers to the number of groups, while allowing a model to learn channel dependencies. With a larger n_{groups}, that is a smaller g, increased parameters and computation decrease rapidly. As can be see in section 3, the increased parameters and computations are almost negligible throughout the experiments while offering huge accuracy gains over baselines.

3 EXPERIMENTS

To inspect the validity and efficiency of our module, we conduct extensive experiments on CIFAR-10/100(Krizhevsky, 2009) and SVHN(Netzer et al., 2011)[1]. We attach our module to the last part of every residual block before addition. For fair comparisons, we fix training configurations and reimplement all codes using PyTorch framework(Paszke et al., 2017).

3.1 GROUP SIZE

We first explore the model performances by varying the group size g. We experiment with ResNet50(He et al., 2016) model with group sizes $= \{2, 4, 8, 16, 32, 64\}$ on CIFAR-100 dataset

[1]We plan to upload results on ImageNet-1K dataset

as shown in Figure 2. Both top-1/5 errors go down with a smaller group size reaching the lowest point of 22.42/5.94% at $g = 2$. For a reference, the original ResNet101 produces 23.17%. Even a simple repetition of point-wise group convolution offers large gains as it permits the model to learn channel dependencies. Note that we do not apply any kind of context modeling within the module. Even though we see a favorable tendency that an error reduces with a lower g, it severely degrades the model performance with a large group size. In case of $g = 64$, it produces poorer results than the baseline model for both top-1/5. This indicates that for the module to learn channel dependencies, it is important to keep the number of channels interacting each other at a time sufficiently small. It is intuitively understandable that the model can finely process channel information with more non-linearities with a smaller group size as the number of non-linear operations increases logarithmically anti-proportional to g. For simplicity, we hereafter denote TAM as our module with $g = 2$.

Table 1: Top-1/5 errors based on various backbone networks on CIFAR-10/100 data set. For CIFAR-10, we only report top-1 error as we don't see much difference between performances. The value in parenthesis denotes the increased parameters over the baseline network.

	CIFAR-10	CIFAR-100 (%)	# PARAMS(M)	FLOPS(B)
RESNET-18	5.32	24.97/7.34	11.22	0.52
RESNET-18-SE	5.31	24.46/7.16	11.31 (+0.80%)	0.52
RESNET-18-CBAM	5.28	24.87/7.58	11.31 (+0.80%)	0.52
RESNET-18-TAM	**4.94**	**24.35/6.84**	11.22 (+0.00%)	0.52
RESNET-50	5.07	24.84/7.11	23.71	1.22
RESNET-50-SE	5.172	23.68/6.37	26.22 (+10.59%)	1.22
RESNET-50-CBAM	4.80	23.67/6.24	26.24 (+10.67%)	1.22
RESNET-50-TAM	**4.60**	**22.42/5.94**	24.01 (+1.23%)	1.22
RESNET-101	4.76	23.17/5.94	42.70	2.35
RESNET-101-SE	4.93	**21.87/5.53**	47.44 (+11.10%)	2.36
RESNET-101-CBAM	4.86	23.20/6.11	47.48 (+11.19%)	2.36
RESNET-101-TAM	**4.54**	21.90/5.75	43.34 (+1.50%)	2.35
RESNEXT-50(32X4D)	5.39	23.21/6.29	23.18	1.26
RESNEXT-50(32X4D)-SE	4.40	22.62/5.70	25.69 (+10.83%)	1.27
RESNEXT-50(32X4D)-CBAM	4.45	22.08/5.76	25.71 (+10.91%)	1.27
RESNEXT-50(32X4D)-TAM	**4.37**	**21.97/5.64**	23.48 (+1.29%)	1.26
WRN-16-8	4.94	22.81/6.18	11.01	1.44
WRN-16-8-SE	**4.92**	22.70/**6.05**	11.10 (+0.91%)	1.45
WRN-16-8-CBAM	4.97	23.07/6.15	11.10 (+0.91%)	1.45
WRN-16-8-TAM	**4.92**	**22.62**/6.11	11.05 (+0.45%)	1.45

3.2 CLASSIFICATION RESULTS ON CIFAR-10/100

To prove generality of the module, we experiment with other various architectures which are ResNet18, ResNet50, ResNext50(8×64d)(Xie et al., 2016), and WideResNet(WRN)16-8(Zagoruyko & Komodakis, 2016). We follow an ordinary preprocessing, which is zero-padding 4 pixels around input pixels, normalizing each channel with mean and standard deviation from training set, allowing horizontal flip by half chance and randomly crop 32×32 pixels. We initialize weights following He et al. (2015a). We use 128 mini-batch size and SGD with 0.9 momentum and 10^{-4} weight decay. It is worth noticing that we allow the decay affects to PReLU weights as we found it works better to do so in preliminary experiments. We train models for 300 epochs and set an initial learning rate 0.1 decaying by 10 at epoch 150 and 225. Table 1 shows the results. Our model outperforms the baseline model and shows competitive results to ones with different attention modules. Unlike other modules which a parametric overhead percentage largely increases according to a depth of a backbone architecture, one from our module is negligible regardless of a depth. Interestingly, our module shows better performance over other attention modules with less than 1.5% parametric overhead on ResNet50 and ResNext50 where over 10% increase are made in other modules. This supports the effectiveness of capturing inter-channel relations.

3.3 CLASSIFICATION RESULTS ON SVHN

The Street View House Numbers(SVHN) dataset is composed of 73,257 training and 26,032 test images of 32-by-32 RGB pixels and corresponding labels. We don't do any preprocessing except for normalizing images to have range [0, 1]. We set the identical hyperparameters with CIFAR-10/100 training except that we train the model for 160 epochs and decay the learning rate at epoch [80, 120]. Table 2 shows the results. We find other modules sometimes work more poorly than baseline network. We surmise this is because of its non-negligible increase in a parameter number as it may cause overfitting for a relatively small dataset.

Table 2: Top-1 error based on various backbone networks on SVHN data set.

	TOP-1 (%)	# PARAMS(M)	FLOPS(B)
RESNET-18	5.94	11.18	0.52
RESNET-18-SE	4.51	11.26 (+0.72%)	0.52
RESNET-18-TAM	**4.29**	11.21 (+0.27%)	0.52
RESNET-34	4.18	21.28	1.08
RESNET-34-SE	4.52	21.44 (+0.75%)	1.08
RESNET-34-TAM	**4.11**	21.36 (+0.38%)	1.09
RESNET-50	4.46	23.52	1.22
RESNET-50-SE	4.71	26.04 (+10.71%)	1.22
RESNET-50-TAM	**3.94**	23.82 (+1.28%)	1.22
WRN16-8	4.28	10.96	1.44
WRN16-8-SE	**3.63**	11.05 (+0.82%)	1.45
WRN16-8-TAM	3.71	11.00 (+0.36%)	1.45

4 ANALYSIS AND DISCUSSION

4.1 TAM AS A STATIC GATE

For a spatial attention activation to be discriminable, it should have a large variance across spatial dimension. For this reason, we derive mean and variance of each activation using ResNet50 on CIFAR-100 test dataset from every attention module. Here "stage" means a stage where feature maps of a fixed spatial size are processed. Stage number starts with 1 and is increased whenever the spatial pooling is applied. Figure 3 shows the variance versus mean scatter plot. Unexpectedly, all attention maps from TAM have a zero variance which means they do not contain spatial discriminability and they are independent to input feature map. **Rather, it depends only on the location where the module is placed and controls the input signal accordingly.** Figure 4a shows how TAM manages the signal flow depending on its location. Overall, a trend is found that TAM suppresses signals flowing through earlier blocks and gradually allows signals. As abstract features which contains a larger, thus more meaningful context are delivered in the mid and later phase of a network, TAM preserves the information to some extent for the later part of the network . Also, TAM allows relatively more signals at the first block of every stage where a spatial pooling is applied, that is, at block 4, 8, and 14. It can be explained as TAM considers for a loss caused by a spatial pooling. To investigate how these activation values affect a performance, we conduct a lesion study as shown in Figure 4b. We omit each attention at a time and measure the top-1/5 accuracy. Where an attention is omitted, the flow is fully transmitted. Interestingly, when we disable the first attention within a stage rapid decrease is observed. This is counter-intuitive but explains why TAM does not fully allow the input signal but just to a certain degree.

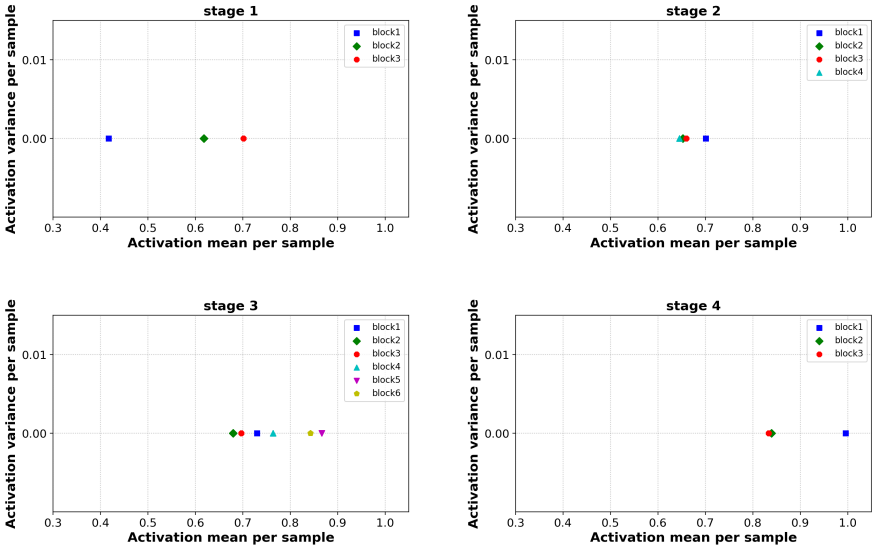
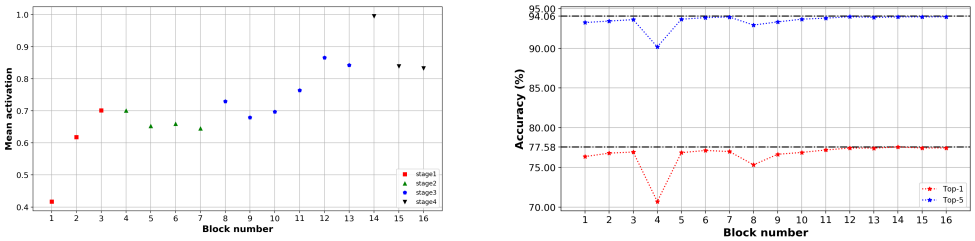


Figure 3: Scatter plot of variance versus mean of activation per data sample from CIFAR-100 10,000 test images. Note that this is a scatter plot for 10,000 samples. Each dot is composed of 10,000 activations as **all variances have a zero value and share the same mean activation**.

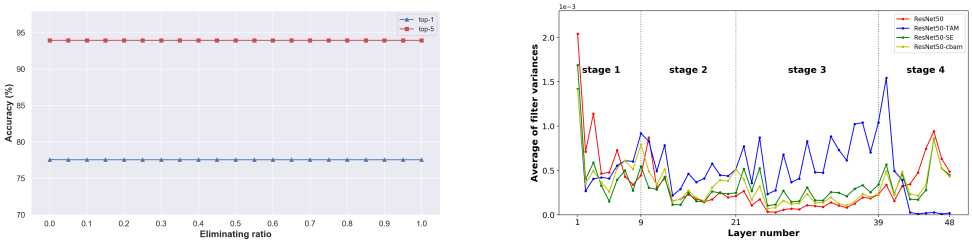


(a) (b)

Figure 4: (a) depicts how TAM controls signals depending on its location. (b) Lesion study on attention at every block. Dashed line denotes the accuracy without omitting any attention.

4.2 Eliminating non-linearities between channels

To further investigate the working mechanism of TAM, we eliminate non-linearities within the module. We first average weights, i.e. negative slopes, of PReLU per layer within the attention module. We sort them out from one closer to 1 to one far from 1. We gradually replace the weights with 1 which eliminates non-linearities of the layers. Figure 5a shows the result. Surprisingly, the performance remains almost constant over eliminating ratio. Unlike conventional thoughts that non-linearities are an indispensable part of a neural network, TAM, once training is finished, does not depend on activation functions. To answer how TAM offers benefits to a network, we plot average variances of each convolutional filter weight within residual blocks from the trained ResNet50, ResNet50-SE, ResNet50-CBAM and ResNet50-TAM. Figure 5b shows this. Compared to the baseline and ones with other attention modules, the average variances have higher values except for the last part of the network. We believe this is because it is more important to pass selected meaningful features to the classifier. This indicates that TAM gives more choices of various filters in the early and mid phase and focuses on a few important filter weights in the last stage by allowing learning inter-channel relationships.



(a) (b)

Figure 5: (a) depicts how TAM controls signals depending on its location. (b) Lesion study on attention at every block. Dashed line denotes the accuracy without omitting any attention.

4.3 Convergence speed

As earlier noted in Hu et al. (2018a;b), TAM also shows faster convergence speed during training. Figure 6 presents this. We attribute this to its light-weight property of TAM. As TAM does not require many additional weights which may require more time to adjust while helping the network to understand inter-channel relations.

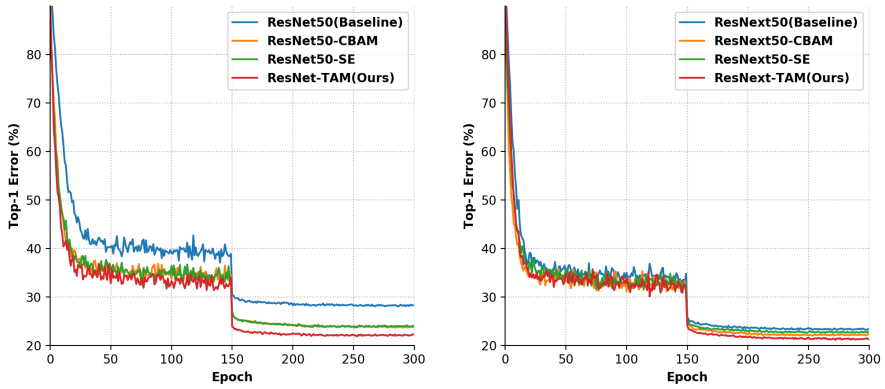


Figure 6: Top-1 validation errors versus epoch on ResNet-50 and ResNext-50.

In summary,

- TAM behaves like a static gate which only depends on its location and controls signal flow.
- TAM allows the model to have various filters in the early and mid phase of the network and narrows the choices the choices down to important ones in the last stage.
- TAM helps a network to converge fast.

5 Conclusion

In this paper, we propose *Tree-structure Attention module* which enables a network to learn inter-channel relationships which deviates from the current trend of capturing long-range dependencies in attention literature. TAM adopts light-weight point-wise group convolutions to allow communication between neighboring channels. Once trained, TAM acts as a static gate controlling signal at a certain location which does not depend on input feature but on the location where it is placed. Moreover, TAM permits higher variances in filter weights in the early and mid phase and helps the filters to focus on important ones at the last phase before classifier. On top of that, TAM produces

favorable performance gains with only a few additional parameters to a backbone network. These advantages of TAM shed a light on a new way to attend features.

REFERENCES

Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *CoRR*, abs/1904.11492, 2019. URL http://arxiv.org/abs/1904.11492.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015a. URL http://arxiv.org/abs/1502.01852.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL http://arxiv.org/abs/1704.04861.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 9401–9411. Curran Associates, Inc., 2018a. URL http://papers.nips.cc/paper/8151-gather-excite-exploiting-feature-context-in-convolutional-neural-networks.pdf.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018b.

Yang Hu, Guihua Wen, Mingnan Luo, Dan Dai, and Jiajiong Ma. Competitive inner-imaging squeeze and excitation for residual network. *CoRR*, abs/1807.08920, 2018c. URL http://arxiv.org/abs/1807.08920.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. BAM: bottleneck attention module. *CoRR*, abs/1807.06514, 2018. URL http://arxiv.org/abs/1807.06514.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016. URL http://arxiv.org/abs/1611.05431.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL http://arxiv.org/abs/1605.07146.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7354–7363, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/zhang19d.html.