Towards Light Weight Object Detection System

Dharma KC1,*, Venkata Ravi Kiran Dayana2, Meng-Lin Wu2, Venkateswara Rao Cherukuri2†, Hau Hwang2, and Clayton T Morrison1

1The University of Arizona
2Qualcomm Technologies, Inc.

ABSTRACT
Transformers are a popular choice for classification tasks and as backbones for object detection tasks. However, their high latency brings challenges in their adaptation to lightweight classification and object detection systems. We present an approximation of the self-attention layers used in the transformer architecture. This approximation significantly reduces the latency of the classification system while incurring minimal loss in accuracy. We also present a method that uses a transformer encoder layer for multi-resolution feature fusion for object detection. This feature fusion improves the accuracy of the state-of-the-art lightweight object detection system without significantly increasing the number of parameters. These modules can be easily integrated into existing CNN and Transformer architecture to reduce latency and increase the accuracy of the system. Finally, we provide an abstraction for the transformer architecture called Generalized Transformer (gFormer) that can guide the design of novel transformer-like architectures.

Keywords: Vision transformer, self-attention, object detection

1. INTRODUCTION
Convolutional neural networks (CNNs)1 have been widely used as backbones for object detection systems. MobileNets2 use depthwise separable convolutions to develop lightweight CNNs. MobileNetV23 further improves MobileNets using inverted residuals and linear bottlenecks. It also introduced efficient ways of applying depthwise separable convolutions to the heads of Single Shot MultiBox Detector (SSD),4 which resulted in the lightweight object detection system, SSDLite.

Recently, Vision Transformers (ViTs)5 are gaining popularity due to their ability to extract global information. However, they lack the spatial inductive biases present in CNNs. MobileViT6 presented a hybrid architecture based on CNNs and ViTs that leverages the inductive biases of CNNs and also includes global information through ViTs. MobileViT achieves impressive performance on the ImageNet-1k classification dataset.7 But, its high latency makes it less attractive for mobile devices. In this research, we significantly improve the latency of MobileViT8 while preserving its impressive performance on classification and detection.

In this work, we propose Convolution as Transformer (CAT): a module that approximates the self-attention layer in transformers. CAT has low latency and thus can be used in lightweight systems for image classification and object detection. We replace expensive transformer blocks used in MobileViT with our CAT blocks, and we show that they are competitive with the self-attention modules for image classification tasks. Moreover, CAT blocks have complexity $O(n \times d)$, unlike self-attention that has complexity $O(n^2 \times d)$, where $n$ is the sequence length, and $d$ is feature vector size.

Existing lightweight systems for object detection4,6 mainly consist of a backbone to extract features from images, followed by heads to extract features from multiple output resolutions. Predictions on object labels and localization are made directly from these multi-scale features. It is challenging to learn the relationship between these features from multiple scales, carrying different semantic information. To overcome this, we propose the module Transformer Encoder as Feature Fusion (TAFF): a single-layered transformer encoder9 which fuses...
features from multiple resolutions at different scales. We show empirically that the feature fusion performed by TAFF improves the accuracy of state-of-the-art object detection models like MobileViT\textsuperscript{8} by a large margin.

Finally, we propose Generalized TransFormer (gFormer): a general architecture that binds multiple variations of attention and transformer mechanisms under a common umbrella. From this perspective, MetaFormer,\textsuperscript{10} Transformer,\textsuperscript{9} Squeeze and Excitation Networks,\textsuperscript{11, 12} and our CAT block are all variations of gFormer. Thus it serves as a model to design new transformer-like architectures by using different functions for different blocks shown in Figure 4.

2. SYSTEM

2.1 Convolution as Transformer (CAT)

The baseline for this architecture is the MobileViT architecture\textsuperscript{8} that uses MobileNetV2 blocks along with MobileViT blocks that contain transformer layers for extracting global information. We refer to\textsuperscript{8} for the full architecture and only show the MobileViT block in Fig. 1.

![Figure 1. MobileViT block used in the MobileVit\textsuperscript{8}](image)

![Figure 2. Convolution As a Transformer (CAT) block](image)

The MobileViT architecture extracts global information with transformers. The major disadvantage of the above method is that it has high latency because of the self-attention layers used in the transformer block. We hypothesize and prove empirically that we can extract global information using a simpler function that has lower latency and a lower number of computations. Thus, we propose Convolution as Transformer (CAT)
blocks (Fig. 2) that approximate the self-attention for global feature extraction. We propose to use the following function to approximate the transformer block:

- Depthwise separable convolutional filter to extract global information:
  \[
  \text{global\_information} == \text{dep\_sep\_conv}(x)
  \]
  This is a combination of depthwise convolution with a kernel size of \((h, w)\) and a pointwise convolution. This can also be interpreted as a combination of spatial MLP and channel MLP where spatial MLP extracts information from the spatial domain and channel MLP extracts information from the channel dimension. Thus, the output vector represents approximate global information from spatial and channel dimensions.

- Broadcast global information to the same shape as intermediate feature map:
  \[
  y = \text{reshape}(\text{global\_information}, (H, W, D))
  \]

- Elementwise product between global information and intermediate feature map:
  \[
  x = \text{elementwise\_product}(x, y)
  \]
  Here, \(\odot\) represents the elementwise product between two tensors. This is similar to multi-head attention with a number of heads equal to the number of features, where the attention is calculated with respect to the single global feature vector.

2.2 Transformer Encoder as Feature Fusion (TAFF)

Fig. 3 shows the architecture of our TAFF block: We extract features from multiple-resolution intermediate feature maps. We extract \(d\) dimensional features from each anchor box from multiple resolution feature maps, we then use the Transformer encoder layer that first projects these features \([n \times d]\) feature tensor into key, query, and values tensors. It then applies multi-headed attention to these features giving each network head the ability to not only look at the feature vector at the corresponding location but also look at other feature vectors to make an informed decision. For example, the locations at the lower level can look at the features of the semantically higher layers to make a good prediction.

2.3 gFormer

MetaFormer\(^{10}\) abstracts different variations of the transformer architectures into a general framework. We further generalize multiple variations of the attention architecture into a common framework that even includes the squeeze and excite networks. Fig. 4 shows the general transformer (gFormer) architecture:

Where \(\oplus\) represents summation, and \(\odot\) represents any pairwise interaction function (e.g. elementwise product). This allows us to design new transformer-like architectures. For example,

- Transformer: If we remove residual-3, use multi-headed attention for spatial mixing, and use MLP for channel mixing with normalization layers, we recover the Transformer architecture.
• MetaFormer: If we remove residual-3 and pairwise interaction function we obtain the MetaFormer.

• Squeeze and excite networks: If we remove residual-1, residual-2, use global pooling as spatial mixing, use a linear layer with "MLP-ReLU-MLP-sigmoid" as channel mixing, repeat the output of this linear layer to original input tensor shape and take Hadamard product, we obtain squeeze and excite networks.

• MLP-Mixer: If we remove residual-3, and pairwise interaction function, use MLP for spatial mixing and MLP for channel mixing, we obtain the MLP-Mixer.

The gFormer abstraction helps us create new architectures that are optimized for different needs and still have the benefits of transformers for capturing global information. We designed our CAT block from this abstraction by using operations that can reduce the latency of the system.

3. EXPERIMENTS

3.1 Classification with CAT

For classification experiments, we use a subset of ImageNet-1k dataset which consists of 100 balanced classes from ImageNet-1k keeping the original train-validation split. We used a subset of the dataset because of the computational overhead of training on the whole ImageNet-1k. The training dataset consists of 130k images, while the validation dataset consists of 5k images. We follow and report accuracy on the validation dataset. We used PyTorch for our experiments. We trained our model up to 300 epochs with the stochastic gradient descent (SGD) algorithm with weight decay of $4e^{-5}$ and momentum of 0.5, at a batch size of 64 per GPU on 4 Nvidia RTX 2080Ti GPUs. We used the Cosine Annealing scheduler with an initial learning rate of 0.05, increasing to 0.4 within 7500 iterations, and ultimately decreasing to a minimum of $2e^{-4}$. We use an input image resolution of $256 \times 256$. We used the Swish activation function as our default activation function. We compare our results with MobileViT as a baseline method. Note that we replaced the multiple transformer layers used in MobileViT with a single layer of our CAT block. Also, our CAT block doesn’t use any feedforward layers used by the transformer layers in MobileViT.
3.2 Detection with TAFF

We use the COCO\textsuperscript{15} dataset for object detection experiments with the same setting and same hyperparameters as MobileViT.\textsuperscript{6} The input image resolution is $320 \times 320$. We evaluated the performance on the validation dataset using mAP@IOU 0.50:0.05:0.95. We first train MobileViT with SSDLite as the baseline method.

We then extract features from MobileViT and SSDLite and fuse features from multiple layers using our TAFF module. The fused features are used to make the prediction for the class and bounding box. Note that we didn’t even use positional encoding for the feature fusion that helps in keeping the small memory footprint when deployed on mobile devices. It’s interesting that the single layer of the transformer encoder was able to fuse features from multiple scales and multiple anchor boxes. We hypothesize that the projection matrices in the transformer encoder project feature from multiple scales and anchor boxes to a common domain suitable for feature fusion.

4. RESULTS

4.1 Classification with CAT

Table 1 shows the accuracy, number of parameters (in Million), number of floating point operations (FLOPS), and latency (measured for a single image on Nvidia RTX 2080 Ti by averaging over 1000 runs). The results show that our CAT block closely approximates the self-attention module of the MobileViT architecture with a slight decrease in accuracy but with less than half the number of parameters, FLOPs, and latency.

| Model     | Accuracy | Parameters | FLOPS | Latency | Complexity |
|-----------|----------|------------|-------|---------|------------|
| MobileViT | 84.84    | 5.7 M      | 4 G   | 11.17 ms| $O(n^2d)$  |
| CAT       | 83.84    | 2.4 M      | 1.27 G| 4.69 ms | $O(nd)$    |

Table 1 also shows the advantage of our method compared to other methods in terms of the run-time complexity of the self-attention module. We can see that our method has linear complexity $O(n)$ with respect to sequence length, while MobileViT has $O(n^2)$ complexity.

4.2 Detection with TAFF

Table 2 shows the effectiveness of our TAFF module in object detection tasks on the COCO validation dataset, which is a challenging object detection dataset. We demonstrate that adding this module on top of the SSD architecture with the MobileViT backbone significantly increases the accuracy of object detection by about 2.5mAP on the COCO validation dataset without increasing the number of parameters by a large amount.

| Method     | mAP  | Parameters |
|------------|------|------------|
| MobileNetv1| 22.2 | 5.1 M      |
| MobileNetv2| 22.1 | 4.3 M      |
| MobileNetv3| 22.0 | 4.9 M      |
| MobileViT  | 27.7 | 5.7 M      |
| TAFF(ours) | 30.1 | 6.1 M      |

5. CONCLUSION

In this paper, we have proposed the CAT block that can decrease the latency and FLOPs of transformer-based backbones such as MobileViT, while increasing the inference speed using simple approximation functions composed of depthwise-separable convolution and Hadamard product. Then, we proposed the TAFF module that improves the accuracy of existing anchor-based object detection systems by fusing features from multiple scales. Finally, we have presented a general framework called gFormer that helps us design new architectures.
There are multiple ways to combine these ideas with existing systems to develop further lightweight, accurate, and fast object detection systems. One such approach would be fusing these ideas to anchor-free object detection systems. For example, our CAT block and TAFF module can be integrated with DETR. We will explore this approach in future research.

REFERENCES

[1] LeCun, Y., Bengio, Y., et al., “Convolutional networks for images, speech, and time series,” The handbook of brain theory and neural networks 3361(10), 1995 (1995).
[2] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., “Mobilnet: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861 (2017).
[3] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., “Mobilenetv2: Inverted residuals and linear bottlenecks,” in [Proceedings of the IEEE conference on computer vision and pattern recognition], 4510–4520 (2018).
[4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., “Ssd: Single shot multibox detector,” in [European conference on computer vision], 21–37, Springer (2016).
[5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N., “An image is worth 16x16 words: Transformers for image recognition at scale,” in [International Conference on Learning Representations], (2021).
[6] Mehta, S. and Rastegari, M., “Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer,” in [International Conference on Learning Representations], (2022).
[7] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., “Imagenet large scale visual recognition challenge,” International journal of computer vision 115(3), 211–252 (2015).
[8] Mehta, S. and Rastegari, M., “Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer,” arXiv preprint arXiv:2110.02178 (2021).
[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I., “Attention is all you need,” Advances in neural information processing systems 30 (2017).
[10] Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., and Yan, S., “Metaformer is actually what you need for vision,” in [Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition], 10819–10829 (2022).
[11] Hu, J., Shen, L., and Sun, G., “Squeeze-and-excitation networks,” in [Proceedings of the IEEE conference on computer vision and pattern recognition], 7132–7141 (2018).
[12] Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al., “Mlp-mixer: An all-mlp architecture for vision,” Advances in Neural Information Processing Systems 34, 24261–24272 (2021).
[13] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., “Pytorch: An imperative style, high-performance deep learning library,” Advances in neural information processing systems 32 (2019).
[14] Ramachandran, P., Zoph, B., and Le, Q. V., “Searching for activation functions,” arXiv preprint arXiv:1710.05941 (2017).
[15] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., “Microsoft coco: Common objects in context,” in [European conference on computer vision], 740–755, Springer (2014).
[16] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S., “End-to-end object detection with transformers,” in [European conference on computer vision], 213–229, Springer (2020).