ECBM E4040 Final Project - A Re-implementation of the ’Swin’ Transformer using Tensorflow 2.x

Pranav Iyenger Deevi
Department of Electrical Engineering
Columbia University
New York, USA
pid2104@columbia.edu

Sanjeev Narasimhan
Department of Computer Science
Columbia University
New York, USA
sn3007@columbia.edu

Ajay Vanamali
Department of Electrical Engineering
Columbia University
New York, USA
va2465@columbia.edu

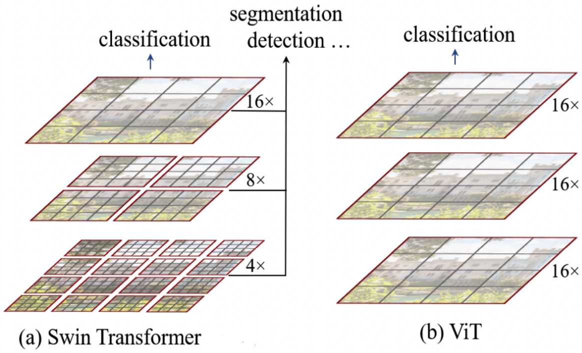
***Abstract*—This paper presents a re-implementation of the ’Swin’ Transformer in Tensorflow, a new vision Transformer that capably serves as a general-purpose backbone for computer vision. Challenges in adapting Transformer networks from language to vision arise from differences between the two domains, such as large variations in the scale of visual entities and the high resolution of pixels in images compared to words in text. To address these differences, a new hierarchical Transformer is proposed, whose representation is computed with Shifted windows. The shifted windowing scheme brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection. This hierarchical architecture has the flexibility to model at various scales and has linear computational complexity with respect to the image size. In our re-implementation, we perform image classification on the [CIFAR-10] dataset, a popular benchmark for computer vision problems, to demonstrate the capability of the Swin Transformer. The code and the models can be found here (Github).**

Keywords: Deep Learning, Transformer Networks, Swin Transformer, Image Classification, Shifted-Window Self Attention

I. Introduction

Convolutional neural networks (CNNs) have long dominated modeling in computer vision and image classification tasks. Alexnet revolutionized the field of computer vision through its ground-breaking performance on the Beginning with AlexNet [1] ImageNet image classification challenges, CNN architectures have evolved to become more powerful through larger scale [2], [3], more extensive connections [4], and more sophisticated forms of convolution [5]–[7]. the advent of ResNets re-revolutionized the entire field and opened up new avenues and greater capabilities to CNNs. However, these networks still suffer from significant drawbacks in tasks such as image segmentation and object detection.

The evolution of network architectures in natural language processing (NLP) tasks has taken a different path, where the prevalent architecture today is instead the Transformer network [8]. Designed for sequence modeling and transduction tasks, the Transformer is notable for its use of the attention mechanism to model language context. Its tremendous success in the language domain has led researchers to investigate its adaptation to computer vision, where it has recently demonstrated promising results on certain tasks, specifically image classification and joint vision-language modeling. The Swin transformer architecture is one such model which is proposed as the general-purpose backbone in computer vision problems and brings about significant improvements by overcoming on the drawbacks of ResNets and ViTs.



(a) Swin Transformer

(b) ViT

Fig. 1. (a) The Swin Transformer builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks. (b) In contrast, previous vision Transformers [9] produce feature maps of a single low resolution and have quadratic computation complexity to input image size due to computation of self-attention globally

II. Literature Review

The authors Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin and Baining Guo (hereafter referred to as ’they’/’them’/’authors’) of the paper ”Swin Transformer: Hierarchical Vision Transformer using Shifted Windows” [12] seek to expand the applicability of Transformer such that it can serve as a general-purpose backbone for computer vision, as it does for NLP and as CNNs do in vision. They observe that significant challenges in transferring its high performance in the language domain to the visual domain can be explained by differences between the two modalities. One of these differences involves scale. Unlike the word tokens that serve as the basic elements of processing in language Transformers, visual elements can vary substantially in scale, a problem that receives attention in tasks such as object detection. In existing Transformer-based models, tokens are all of a fixed scale, a property unsuitable for these vision applications. Another difference is the much higher resolution of pixels in images compared to words in passages of text. There exist many vision tasks such as semantic segmentation that require dense prediction at the pixel level, and this would be intractable for Transformer networks on high-resolution images, as the computational complexity of its self-attention is quadratic to image size.

To overcome these issues, the authors propose a general-purpose Transformer backbone, called Swin Transformer, which constructs hierarchical feature maps and has linear computational complexity to image size. As illustrated in Figure 1(a), the Swin Transformer constructs a hierarchical representation by starting from small-sized patches (outlined in gray) and gradually merging neighboring patches in deeper Transformer layers. With these hierarchical feature maps, the Swin Transformer model can conveniently leverage advanced techniques for dense prediction such as feature pyramid networks (FPN) [10] or U-Net [11]. The linear computational complexity is achieved by computing self-attention locally within non-overlapping windows that partition an image (thus the name windowed-MSA or shifted window-MSA). The number of patches in each window is fixed, and thus the complexity becomes linear to image size. These merits make Swin Transformer suitable as a general-purpose backbone for various vision tasks, in contrast to previous Transformer based architectures [9] which produce feature maps of a single resolution and have quadratic complexity.

The main element of the Swin Transformer is its shift of the window partition between consecutive self-attention layers, as illustrated in Figure 2. The shifted windows bridge the windows of the preceding layer, providing connections among them that significantly enhance modeling power. The attention is computed between the local neighborhood of patches which acts similar to a ’convolution’.

Initially, the image pixels are grouped into smaller sub-elements known as patches. These patches reduce the computation required when computing attention and makes the use of transformers feasible for vision related tasks. To improve the learning capabilities of the transformer model, these patches are passed through positional embedding layer that can efficiently encode them in higher dimensional space (128). The embeddings are then passed through consecutive windowed-self attention blocks (W-MSA/SW-MSA) and the final output is then passed through a patch merging layer that reshapes the output and then linearly projects the final dimension from C’ to C’/2.

The proposed Swin Transformer architecture achieves strong performance on the problems of image classification, object detection, and semantic segmentation. The authors perform a comparison with other state-of-the-art models and demonstrate that the shifted-window technique is particularly useful for tasks requiring finer analysis such as pixel-wise classification.

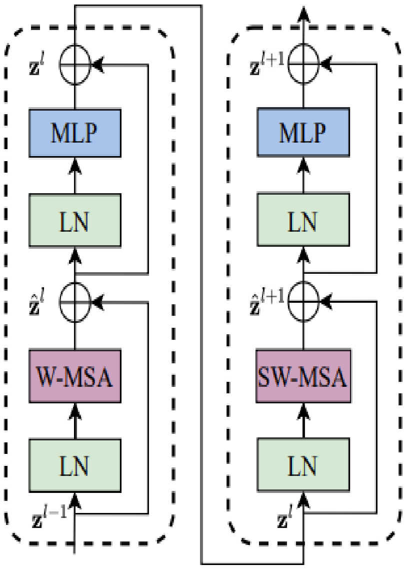


Fig. 2. (a) The first block is a normal windowed attention module (b) The second block is the shifted windowed attention module.

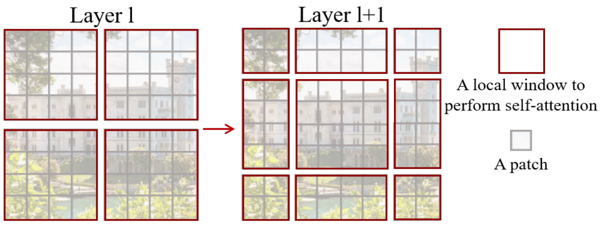


Fig. 3. (a) The Swin Transformer builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). (b) In contrast, ViT produces feature maps of a single low resolution and has quadratic computation complexity.

III. Dataset

For our implementation and experiments, we use the **CIFAR-10** image classification Dataset, a popular computer vision set used for bench-marking models. The CIFAR-10 dataset consists of 32 × 32 RGB images, with 60K training samples and 10K test samples among 10 object classes (Figure

Fig. 4. Sample Images from the CIFAR-10 dataset

3). The classification task proves to be particularly challenging due to the low image resolution. In our experiments, we use the test set as our validation set during training to observe model performance. All images are augmented by performing random flips and re-scaled to (0-1) values before being fed into the model. We avoid other augmentations that may drastically change the image feature space due to the low resolution.

IV. Architecture

We implement a modified version of the original Swin Transformer architecture for our experiments (Figure 4). The original Swin transformer comes in three variants (tiny, base, large) with Swin_T having the least parameters (28M) and consists of 4 stages (2,2,6,2 Swin-blocks respectively). In contrast, our architecture consists of 3 stages (2,4,4 blocks) with a total of 17M parameters, with our choice as a result of lower image sizes in our dataset compared to the original ImageNet experiments. We also use a patch size of 2×2 and a window size of 4 for the same reasons.

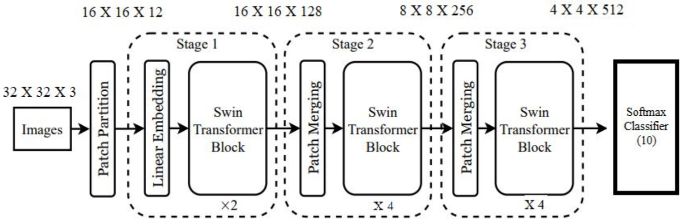


Fig. 5. Our Swin Architecture with blocks 2, 4, 4

Architecture Comparison:

Feature	Our Swin Model	Original Swin_T
Stages	3 (2,4,4)	4 (2,2,6,2)
Input	32 x 32	224 x 224
Embedding	128	96
Window Size	4	7
Patch Size	2	4
#Params	17M	28M

V. Experiments

We train the model for the classification task and measure the performance on the validation set. Training is done with a batch size of 64 for 35 epochs. We use a learning rate of 0.001 and weight decay of 0.00001 with the AdamW optimizer as proposed by the authors. One key difference is that we implement the OneCycle learning rate schedule as opposed to the Linear Warmup + Cosine Decay in the original implementation by the authors. Our choice of the learning rate schedule is motivated by encouraging results of the OneCycle policy, showing much faster convergence for training as opposed to previous scheduling policies for state-of-the-art models. We measure the top-1 (acc@1) and top-3 (acc@3) accuracy on the test set.

For comparison, we trained several other architectures (see table) with the same hyperparameters and learning rate schedule. We observed that the Swin transformer outperformed the other architectures on the classification problem, achieving 72% accuracy on the dataset, with an inference time of 82ms.

CIFAR-10	acc@1	acc@3	No. of params	Training Time	Inference Time
Swin	72.35%	93.23%	17M	3h 49m	82.97ms
ViT	56.3%	85.29%	10M	1h 10m	67.28ms
ResNet 50	61.67%	86.32%	23M	27m	56.13ms
Efficient-NetB4	61.12%	88.19%	17M	44m	67.80ms

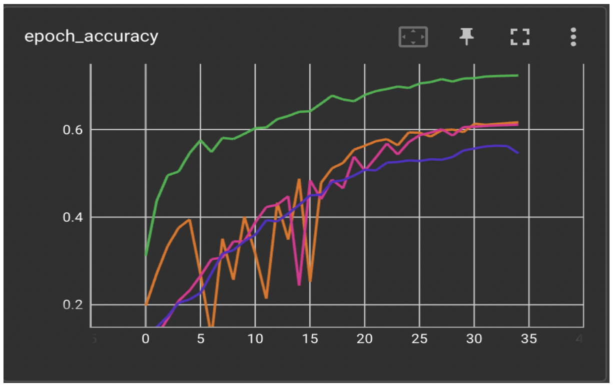


Fig. 6. Tensorboard Accuracy Curves on the CIFAR-10 dataset. Green: Our Swin Transformer model; Orange: ResNet50; Pink: EfficientNetB4; Purple: ViT

VI. Challenges and Improvements

The main issue with the classification on the CIFAR-10 dataset is that the models begin to overfit the data. While experimenting with various model complexities, we observed that reducing the complexity only caused the models to saturate with lower accuracy. We additionally experimented with L2 regularization (0.01, 0.001, 0.005) and found that the models were unable to converge, indicating that other regularization techniques such as dropout/complex image augmentation might be useful in increasing training accuracy. As a future goal, we could possibly use the newer versions of the Swin Transformer (Swin v2) which show even better performance on the problems of classification with an updated architecture that helps reduce training instability.

VII. Contributions

A. Code

Done by **Iyenger Deevi, Pranav (pid2104)**:

- Architecture/Approach (Equal)
- Pytorch to TensorFlow porting
- Dataset Handling

- Commenting and organizing code.
- Debugging and code refactoring

Done by **Narasimhan, Sanjeev (sn3007)**:

- Architecture/Approach (Equal)
- Model Definition
- Training of Swin
- Training of Other Architectures
- Learning rate scheduler
- Hyper-Parameter Tuning

Done by **Vanamali, Ajay (va2465)**:

- Architecture/Approach (Equal)
- Layer Definitions
- Layer Function Definitions
- Refactoring of Github code/directory organization

B. Report:

Written by **Iyenger Deevi, Pranav (pid2104)**:

- Introduction
- Architecture (Equal)
- Dataset/Training
- Challenges and Improvements

Written by **Narasimhan, Sanjeev (sn3007)**:

- Literature Review
- Architecture (Equal)
- Experiments
- References

Written by **Vanamali, Ajay (va2465)**:

- Abstract
- Architecture (Equal)
- Challenges and Improvements

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural net- works. In Advances in neural information processing systems, pages 1097–1105, 2012.

[2] 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.

[3] Sergey Zagoruykoand Nikos Komodakis. Wide residual networks. In BMVC, 2016.

[4] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional net- works. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017.

[5] Saining Xie, Ross Girshick, Piotr Dollar̀, 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.

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proceedings of the IEEE International Confer- ence on Computer Vision, pages 764–773, 2017.

[7] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9308–9316, 2019.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko- reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, pages 5998–6008, 2017.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations,2021.

[10] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U- net: Convolutional networks for biomedical image segmen- tation. In International Conference on Medical image com- puting and computer-assisted intervention, pages 234–241. Springer, 2015.

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv:2103.14030