Project Report - ECE 285

Abhishikta Panja
ECE
A69027311

Geetika Agrawal
ECE
A59026563

Abstract

Our project implements the Swin Transformer for image classification, utilizing a shifted windowing mechanism and a hierarchical architecture. The shifted window scheme enhances performance by restricting self-attention to non-overlapping local windows while enabling cross-window connections. This project highlights the Swin Transformer's potential as a foundational general-purpose vision backbone, showcasing its robustness and efficiency in handling visual tasks.

1 Introduction

Traditionally, Convolutional neural networks (CNNs) have achieved revolutionary breakthroughs in computer vision, mostly dominating applications like object detection and image categorization. However, managing visual features with significant scale variations and processing high-resolution images efficiently are challenges for CNNs. Transformer architectures have shown remarkable potential using attention mechanisms to model long-range dependencies effectively.

The motivation behind this project stems from the need to address specific challenges in adapting transformers for vision tasks. Unlike fixed-scale word tokens in NLP, visual elements can vary widely in scale, and high-resolution images require dense pixel-level predictions, making quadratic computational complexity of traditional Transformers impractical. To overcome these issues, Swin Transformer, constructs hierarchical feature maps by merging image patches and achieves linear computational complexity by confining self-attention to local windows. A key innovation is the use of shifted windows between layers, which enhances connectivity and modeling power without high computation costs. This approach allows the Swin Transformer to handle various vision tasks efficiently and effectively.

Our results shows the Swin Transformer's[8] achieve an accuracy of 78.10% for the SWIN-T variant and 79.5% with a deeper and more complex SWIN-B variant. We experimented with various methods in positional encoding, multi-head self-attention (MSA) and swin transformer architecture.

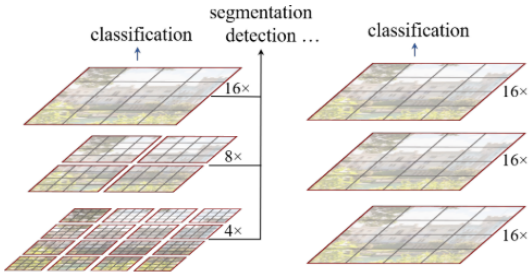

Figure 1: (a) Swin Transformer[8]; (b) ViT

2 Related Work

Convolutional Neural Networks (CNNs) have been the most used tool for computer vision, with significant milestones marked by state-of-the-art architectures like AlexNet [2], VGG [3], ResNet [4], and DenseNet [5]. These networks have advanced the field through deeper layers, improved connectivity through skip connections, and sophisticated convolution techniques. However, the Vision Transformer (ViT) [6] introduced a new paradigm by applying Transformer architectures directly to image patches, demonstrating a compelling speed-accuracy tradeoff but requiring large-scale datasets for optimal performance. DeiT [7] refined ViT's approach, enabling effective performance on smaller datasets like ImageNet-1K through innovative training strategies.

While ViT focuses on image classification, its quadratic complexity with respect to image size and low-resolution feature maps limit its applicability for dense vision tasks. To address these limitations, the Swin Transformer [8] introduces a hierarchical structure with shifted windowing as in Figure 1, achieving linear computation complexity relative to image size. This design results in high-resolution feature extraction and efficient modeling across vision tasks, including object detection and semantic segmentation. DeiT [7] introduces training strategies that make ViT effective with smaller datasets like ImageNet-1K. Despite encouraging results in image classification, ViT's architecture is not suited for dense vision tasks or high-resolution inputs due to low-resolution feature maps and quadratic computation complexity with image size.

Swin Transformer [8] addresses these limitations by constructing hierarchical feature maps and achieving linear computational complexity through cross-window self-attention. This allows Swin Transformer to excel in various vision tasks, including image classification, object detection, and semantic segmentation, achieving state-of-the-art accuracy and efficiency. Some works have also attempted to build multi-resolution feature maps on Transformers, but Swin's linear complexity and cross-window operation make it more effective in modeling high correlation in visual entities.

3 Method

3.1 Model Architecture

The Swin Transformer is a hierarchical vision transformer with several key innovations, including a hierarchical architecture, a shifted window mechanism, cyclic shifts, masking post-cyclic shift, and specialized Swin blocks. Below is a detailed explanation of these components and their benefits.

The Swin Transformer splits an input RGB image into non-overlapping patches using a patch splitting module. Each patch is treated as a token, with its feature set as the concatenation of raw pixel RGB values. For our implementation, we use a patch size of 4×4, resulting in a feature dimension of $4 \times 4 \times 3 = 48$. A linear embedding layer projects this raw-valued feature to a dimension C.

Swin Transformer blocks, each with modified self-attention computation, are applied to these patch tokens. In "Stage 1", the number of tokens is maintained as $\frac{H}{4} \times \frac{W}{4}$. As the network deepens, patch merging layers reduce the number of tokens. Each merging layer concatenates features of 2×2 neighboring patches and applies a linear layer on the $4C$-dimensional concatenated features, halving the resolution and doubling the dimension. This process is repeated, producing hierarchical feature maps at resolutions of $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, and $\frac{H}{32} \times \frac{W}{32}$ in subsequent stages.

3.1.1 Swin Transformer Block (STB)

The Swin Transformer has two successive blocks. The second block replaces the standard multi-head self-attention module with a shifted window-based MSA, followed by a 2-layer MLP with GELU nonlinearity. Each module is preceded by LayerNorm (LN) and followed by a residual connection, as shown in Figure 2(b).

3.1.2 Stage Modules

The model consists of 4 stages. Each stage includes a Patch Merging layer, which downsamples the input by concatenating neighboring patches and applying a linear transformation to reduce the number of tokens. Each stage further consists of multiple Swin Transformer blocks, where the number of tokens is progressively reduced, and the feature dimensions are increased.

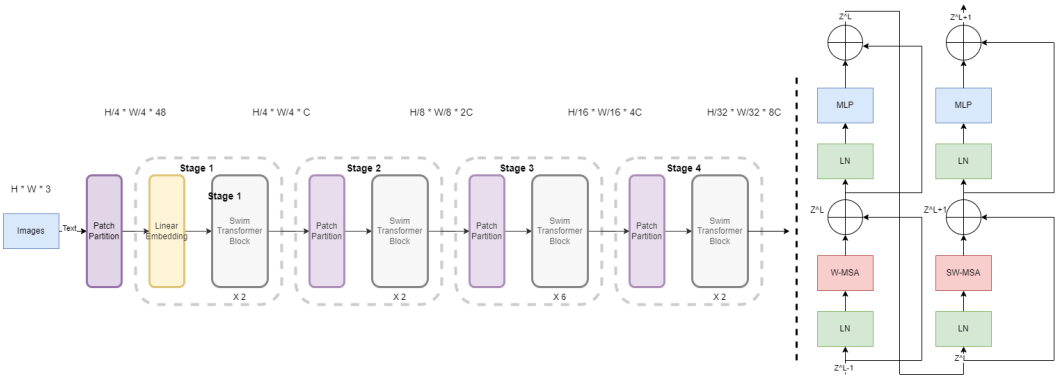

Figure 2: (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks. W-MSA and SW-MSA: Multi-head self-attention modules with regular and shifted windowing settings

3.1.3 Shifted Window Mechanism

To introduce connections across different windows, the window partitioning is shifted between consecutive layers. Patches are the basic units (tokens) derived from the input image. Each patch corresponds to a token. Windows are are groups of patches within which self-attention is computed. A window contains multiple patches (tokens), but it is not considered a single token. For example, an input image of size 224×224 can be divided into 4×4 patches, resulting in (224/4)×(224/4) = 56×56 patches. Each patch is a token. During the self-attention computation, a window size of 7×7, each window will contain 7×7 = 49 patches (tokens).

In Figure 3, the first the layer l is divided into non-overlapping local windows. Each window contains multiple patches. Each window has 4×4 patches where self-attention is computed within these 4×4 windows independently. This significantly reduces the computational complexity compared to computing self-attention globally, from quadratic to linear, across the entire image. In the next layer l+1, the window partitioning is shifted. The windows are displaced by a certain number of patches (here, half the window size) in both horizontal and vertical directions, which result in an overlap with the previous windows. The shifted windows ensure that patches from different original windows now share a window, facilitating connections and interactions between them.

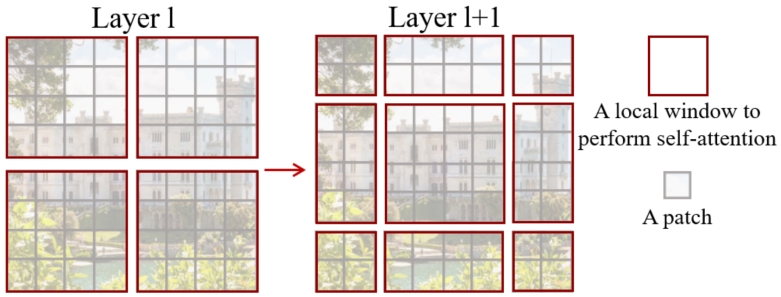

Figure 3: Shifted window approach

Suppose each window contains $M \times M$ patches. The computational complexities for global and window-based MSA on an image of $h \times w$ patches are:

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C, \tag{1}$$

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC, \tag{2}$$

3.1.4 Cyclic Shift and Masking

Due to the shifted window approach, certain windows at the boundaries become smaller than the window size. The cyclic shift mechanism and subsequent masking is used to handle this challenge.

1. **Cyclic Shift:** A cyclic shift operation rolls the feature map by a certain displacement, ensuring that the window boundaries shift as intended. As seen in Figure 4, shifting the windows by a specified displacement (half the window size) in both horizontal and vertical directions. This shift results in a new arrangement where patches from different original windows (labeled A, B, and C) are now within the same window. After the self-attention computation, a reverse cyclic shift restores the original arrangement of patches, now enriched with information from neighboring windows. This process allows the model to capture long-range dependencies efficiently without significantly increasing computational complexity.
2. **Masking:** The cyclic shift can cause patches that are not spatially adjacent in the original image to be grouped together, potentially leading to invalid or unintended interactions during the self-attention computation. Masking ensures that the self-attention mechanism only considers valid interactions within each shifted window. Without masking, the attention mechanism could incorrectly compute attention scores between patches that are not meant to interact.

   Masks are created to indicate which positions in the attention matrix should be ignored. These masks assign a very large negative value (negative infinity) to the positions that should not be considered during self-attention computation.

   - textbfUpper/Lower Mask: This mask prevents attention from considering interactions across the top and bottom boundaries of the shifted windows. For example, patches that are shifted to the top of the window should not interact with patches that were originally at the bottom.
   - textbfLeft/Right Mask: Similarly, this mask prevents interactions across the left and right boundaries of the shifted windows.

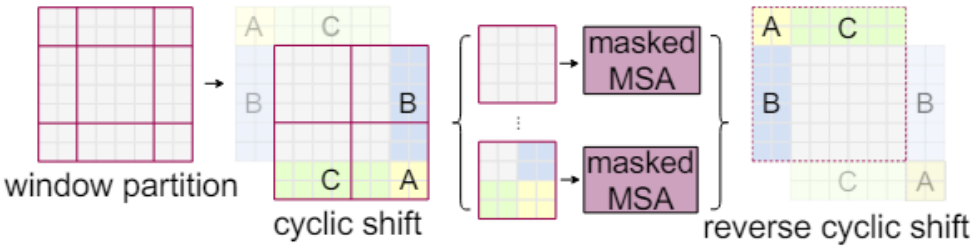

Figure 4: Visualisation of cyclic shift for multi-head self-attention

3.2 Training and Testing Algorithms

We have utilised two variants of Swin Transformer architectures Swin-T (Tiny) and Swin-B (Base). Swin-Tiny is designed for efficiency and consists of 4 stages with [2, 2, 6, 2] layers in each stage, embedding dimensions of [96, 192, 384, 768], and uses [3, 6, 12, 24] attention heads per stage. This configuration makes Swin-T suitable for environments with limited computational resources.

On the other hand, Swin-Base is aimed at using a deeper network. It also comprises 4 stages but with [2, 2, 18, 2] layers, embedding dimensions of [128, 256, 512, 1024], and [4, 8, 16, 32] attention heads per stage. Swin-B is ideal for more resource-intensive tasks requiring higher computational power, offering enhanced performance for large-scale image classification, object detection, and segmentation tasks.

For training, we used the Adam optimizer, employing a StepLR learning rate scheduler for 50 epochs. The Swin-T and Swin-B models were trained on the CIFAR-10 dataset, focusing on image classification tasks. Various configurations listed in Proposed techniques, such as the number of heads, layers, and positional encoding techniques, were experimented with to optimize performance.

For testing, we used the CIFAR-10 test dataset to evaluate the performance of the models under the various experimental configurations.

3.3 Proposed Techniques

All experiments were done on both Swin-T and Swin-B. The baseline version of both uses dot product similarity between query and key matrices, absolute positional encoding, two successive swin transformer blocks in every stage, with the first block with window multi-head self-attention (W-MSA) and the second block with shifted multi-head self-attention (SW-MSA) using cyclic shift.

Each of the experiments below are tested individually with the baseline version mentioned above.

1. Implementing Relative Positional Encoding instead of Absolute Positional Encoding.
    - **Relative Positional Encoding:** Calculated by finding the relative distance between tokens and encoding it into a matrix, thus, enhancing spatial relationship understanding in transformers.
    - **Absolute Positional Encoding:** Calculated by assigning unique encoding vectors to tokens based on their absolute positions, thus, aiding sequence comprehension in transformers.
2. Utilizing Cosine Similarity between query and key matrices instead of scaled dot product similarity.
    - **Cosine Similarity:** Similarity between vectors by computing the cosine of the angle between them which is commonly used in applications for measuring similarity.
    - **Scaled Dot Product Similarity:** Calculated by scaling the dot product of two vectors by square root of their dimensionality, generally used in transformer self-attention mechanisms for capturing similarities between sequences.
3. Exploring different configurations of multi-head self-attention in the Swin Transformer blocks:
    (a) Both blocks using W-MSA.
    (b) The first block using W-MSA and the next block using SW-MSA.
    (c) Both blocks using SW-MSA.
4. Applying SW-MSA with Zero Padding or Cyclic-Shift.

4 Experiments

4.1 Datasets

We conducted our experiments on CIFAR-10 dataset[9], which consists of 60,000 images, each with a resolution of 32x32 pixels, categorized into 10 distinct classes. We used augmentation techniques like random horizontal flips and random cropping with padding. The images were then normalized to standardize the dataset and improve model performance. This data is formatted as JPEG images, categorized into distinct classes, with annotations provided for each image. These preprocessing steps ensure consistency and enhance the model's ability to capture and classify intricate visual details accurately, providing a robust benchmark for evaluating performance in image classification tasks.

4.2 Results

We evaluated the performance of Swin Transformer models, Swin-T and Swin-B, on the CIFAR dataset. The detailed architecture and accuracy results are presented in Table 1:

Highest Model Performance: Both SWIN-T(78.10%) and SWIN-B(79.5%) have highest performance using the shifted window mechanism with cyclic shift, relative positional encoding and with a scaled dot-product similarity. The SWIN-B model consistently outperforms SWIN-T under all experiments.

4.3 Ablation Study

We performed an ablation study to understand the impact of different components and configurations on the performance of our Swin Transformer models. The study includes the following evaluations and analysis:

In all cases, SWIN-B performs better than SWIN-T.

1. **Positional Encoding:**

Configuration	Swin-T Accuracy	Swin-B Accuracy
Absolute Positional Encoding	76.72%	78.65%
Relative Positional Encoding	**78.10%**	**79.5%**
STB-1: W-MSA, STB-2: W-MSA	75.20%	77.40%
STB-1: W-MSA, STB-2: SW-MSA	76.72%	78.65%
STB-1: SW-MSA, STB-2: SW-MSA	**77.50%**	**79.20%**
Scaled Dot-Product Similarity	76.72%	76.72%
Cosine Similarity	54.32%	55.71%
SW-MSA (with zero padding)	75.8%	77.30%
SW-MSA (cyclic-shift)	76.72%	78.65%

Table 1: Accuracy for SWIN-T and SWIN-B

- **Absolute Positional Encoding:** Results in accuracy 76.72% for Swin-T and 78.65% for Swin-B
- **Relative Positional Encoding:** Improved accuracy over absolute positional encoding to 78.10% for Swin-T and 79.5% for Swin-B.
- **Rationale:** Relative positional encoding dynamically adjusts to the positions of patches relative to one another, which allows the model to better understand the context and spatial configurations within the image, irrespective of their absolute positions. Relative positional encoding provides a more generalized approach and it is invariant to global shifts and changes in the image.

2. **Similarity in self-attention methods:**
   - **Scaled Dot-Product Similarity:** Swin-T achieved 76.72% accuracy and Swin-B 78.65%.
   - **Cosine Similarity:** Swin-T and Swin-B obtained 54.32% and 55.71% accuracy respectively, indicating that cosine similarity is less effective for capturing complex visual features in this context.
   - **Rationale:** The expected result would be for cosine similarity to perform better, however, the lower performance of cosine similarity on the CIFAR-10 dataset, compared to dot-product similarity, can be due to several factors. The small 32x32 pixel size of CIFAR-10 images may limit the effectiveness of cosine similarity, which excels with high-dimensional data. Also, cosine similarity normalizes the vectors, potentially losing crucial information that dot-product similarity retains. Dot-product similarity may simply be better suited for the specific feature distribution and resolution of the CIFAR-10 dataset, explaining its superior performance in this context.
3. **Combinations of W-MSA and SW-MSA:**
   - **STB-1: W-MSA, STB-2: W-MSA:** Swin-T achieved 75.20% accuracy and Swin-B achieved 77.40% accuracy. Using window-based multi-head self-attention in both blocks without any shift leads to slightly lower performance.
   - **STB-1: W-MSA, STB-2: SW-MSA:** Swin-T achieved 76.72% accuracy and Swin-B achieved 78.65% accuracy. Introducing shifted windows in the second block helps in improving its performance.
   - **STB-1: SW-MSA, STB-2: SW-MSA:** Swin-T achieved 77.50% accuracy and Swin-B achieved 79.20% accuracy. Using shifted windows in both blocks gives the highest accuracy.
   - **Rationale:** On using shifted windows, the model can bridge the gaps between non-overlapping windows, allowing patches that were previously isolated in different windows to interact. This helps in capturing long-range dependencies within the image.
4. **Padding and Cyclic Shift:**
   - **Shifted Window with Padding:** Swin-T achieved 75.8% accuracy and Swin-B achieved 77.30% accuracy.
   - **Shifted Window with Cyclic Shift:** Swin-T achieved 76.72% accuracy and Swin-B achieved 78.65% accuracy.
   - **Rationale:** With cyclic shift, we retain more information from the image because it uses the original pixel values by wrapping around the image. However, zero padding adds zeros to the overflowing windows, which provides no meaningful information about the image.

4.4 Conclusion

Our experiments with the Swin Transformer models, Swin-T and Swin-B, demonstrate their effectiveness in handling image classification tasks on the CIFAR-10 dataset. The hierarchical architecture and shifted window mechanism significantly enhance the model's performance, enabling it to manage high-resolution images and varied visual scales effectively. The combination of **shifted window multi-head self-attention (SW-MSA) with cyclic shift** and **relative positional encoding** yielded the best performance, highlighting the importance of cross-window interactions for capturing long-range dependencies.

5 Implementation

Our code includes a swin_transformer_model.py which contains the architecture for the swin_transformer model and main.py contains data loading and pre-processing, model training and evaluation.

Code has been written from scratch in both files. Zero-padding shift, cosine similarity and other experimentation is written from scratch, except, base class for cyclic shift multi-head self-attention [10].

References

[1] Zhao, H., Jia, J., Koltun, V. (Year). Exploring Self-attention for Image Recognition.

[2] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.

[3] Simonyan, K., Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.

[4] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

[5] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700-4708.

[6] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*.

[7] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H. (2021). Training data-efficient image transformers & distillation through attention. *Proceedings of the International Conference on Machine Learning (ICML)*, 10347-10358.

[8] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012-10022.

[9] CIFAR-10. (Year). Learning Multiple Layers of Features from Tiny Images. Toronto: Alex Krizhevsky, Vinod Nair, Geoffrey Hinton.

[10] Microsoft. Swin Transformer. Available at: `https://github.com/microsoft/Swin-Transformer`.