Midterm Project for Deep Learning 2024-2025

Study of the Swin Transform for remote sensing images segmentation.

Group 13

Nguyen Viet Minh Duc (22BI13095)
Le Quang Minh (22BI13286)
Le Hoai Nam (22BI13321)
Nguyen Hoang Minh (22BI13291)
Nguyen Duy Nghia (22BI13331)

USTH VIETNAM FRANCE UNIVERSITY

University of Science and Technology of Hanoi

Content

Chapter 1: Introduction

a. What is Swin Transformer

- The Swin Transformer, short for Shifted Window Transformer, is a type of vision transformer designed to improve upon previous transformer architectures for image processing tasks, particularly in computer vision. Introduced in the paper "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" by Liu et al. in 2021, it has gained popularity for its efficiency and performance in various applications, including image classification, object detection, and segmentation. The Swin transformer has top-1 accuracy of 87.3 on ImageNet-1K

- When using Swin Transformers, you can upload an image and then the model will:

1. Create segmentation masks for objects that the model can identify.
2. Calculate the percentage of object present in an image

- Swin Transformers can be used for many use cases. For example:

Image Classification, Object Detection, Video Analysis,.......

b. What is Image Segmentation

- Image segmentation is the process of partitioning a digital image into multiple image segments, also known as region images or object images. Image segmentation is the technique of dividing an image into different regions based on features such as color, texture, or light intensity. In this group project, model architectures such as Swin Transform are widely used to perform segmentation task.

Chapter 2: Research Method

a) Task

The task involves predicting a set of valid masks for every given prompt by the authors (or user) to the model. The prompt can be represented as a form of points, target masks, or words.

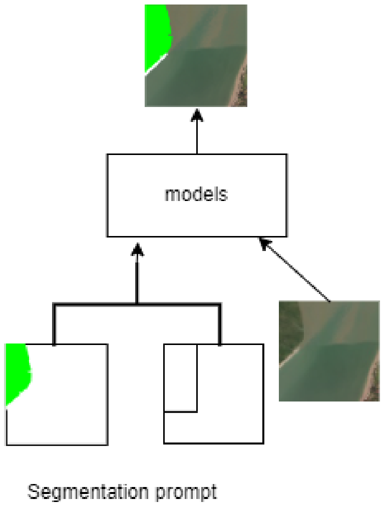


Figure 3.1: Promptable segmentation

b) Model

The input picture is resized and its pixel values are normalized as part of the preparation procedure. The image is then split up into smaller patches, each measuring 16 by 16 pixels, for a total of 196 patches for a 224 by 224 image. After that, linear projection is used to turn each of these patches into a feature vector. To calculate the links between these patches and comprehend the spatial dependencies within the picture, the model makes use of a shifting window attention method. At the end, the model produces an output that can be a segmentation map showing the different sections of the picture or a class label, such "strawberry."

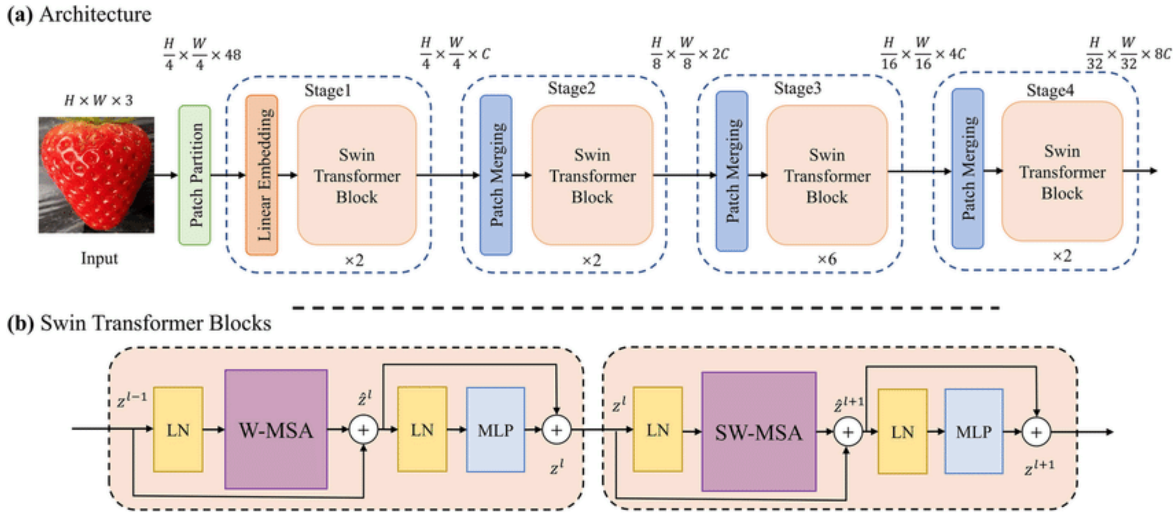


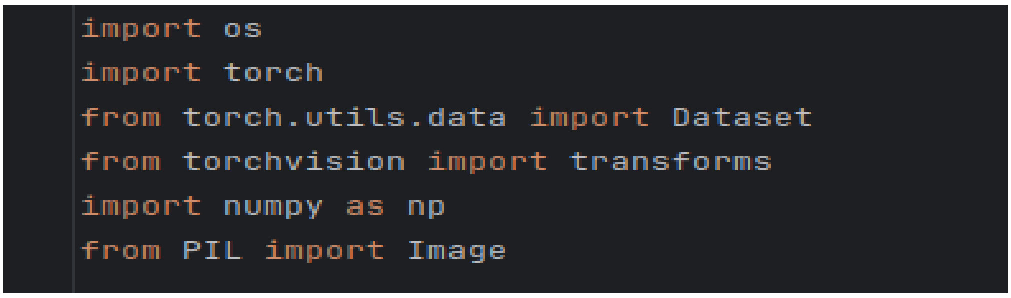
Figure 3.2: Segment Anything Model

c) Dataset

We use an open-source dataset DeepGlobe Land Cover Classification Dataset in **Kaggle** from **Land Cover Classification Dataset from DeepGlobe Challenge** comprising more than thousands images and 806 masks. The author of the dataset clearly annotates what objects correspond to the color palette, and they also give us a separate training set and test set of images.

Chapter 3: Model architecture and methods

a) Importing Necessary Libraries



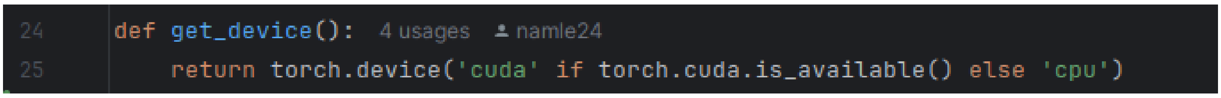
```
import os
import torch
from torch.utils.data import Dataset
from torchvision import transforms
import numpy as np
from PIL import Image
```

- torch: Core library for deep learning in PyTorch.
- timm: Provides pre-trained models, including Swin Transformer.
- PIL: Used for image loading and preprocessing.
- matplotlib: Used for plotting and visualizing results.

b) Configuration

- Device Configuration

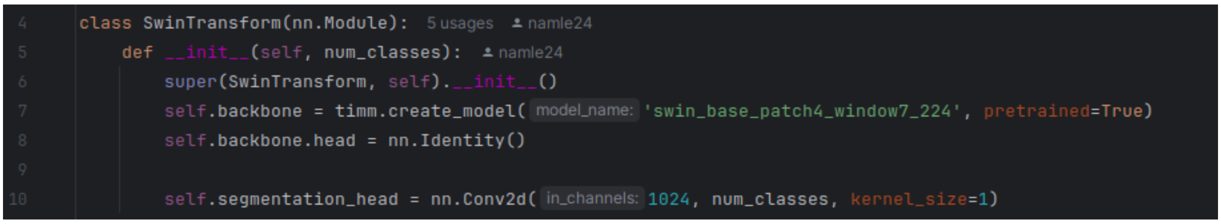
To ensure our model utilizes GPU (if available), we check for the availability of cuda. This allows faster computations, which is crucial for training deep learning model.



```
def get_device():  4 usages  namle24
    return torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

- Model Configuration

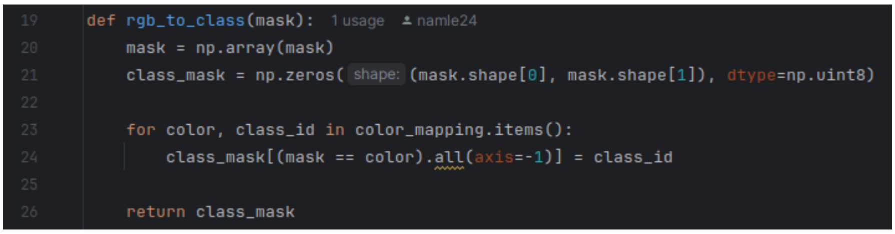
The Swin Transformer is loaded from timm. This specific model version is swin_base_patch4_window7_224, which is pre-trained on ImageNet. After loading, we configure the model to predict 7 classes corresponding to different land types (forest, urban,...).



```
class SwinTransform(nn.Module):  5 usages  namle24
    def __init__(self, num_classes):  namle24
        super(SwinTransform, self).__init__()
        self.backbone = timm.create_model( model_name: 'swin_base_patch4_window7_224', pretrained=True)
        self.backbone.head = nn.Identity()

        self.segmentation_head = nn.Conv2d( in_channels: 1024, num_classes, kernel_size=1)
```

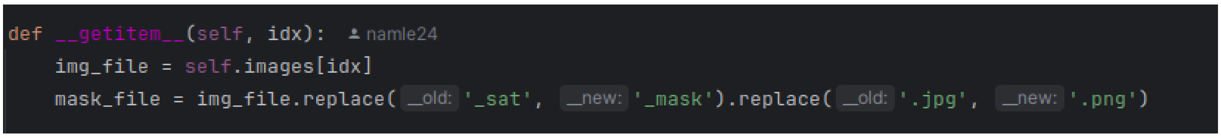
c) Dataset and Data Processing

Using the rgb_to_class function, each colored pixel in the mask is converted to its corresponding label value.



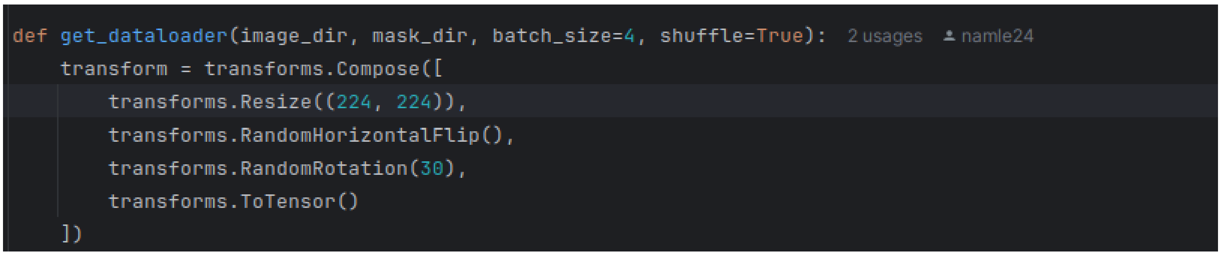
```
def rgb_to_class(mask):  1 usage  namle24
    mask = np.array(mask)
    class_mask = np.zeros( shape: (mask.shape[0], mask.shape[1]), dtype=np.uint8)

    for color, class_id in color_mapping.items():
        class_mask[(mask == color).all(axis=-1)] = class_id

    return class_mask
```

Each satellite image will have an accompanying mask file, and through the __getitem__ method, the image and mask data pair will be returned as a tensor for use in the model.



```
def __getitem__(self, idx):  namle24
    img_file = self.images[idx]
    mask_file = img_file.replace( __old: '_sat', __new: '_mask').replace( __old: '.jpg', __new: '.png')
```

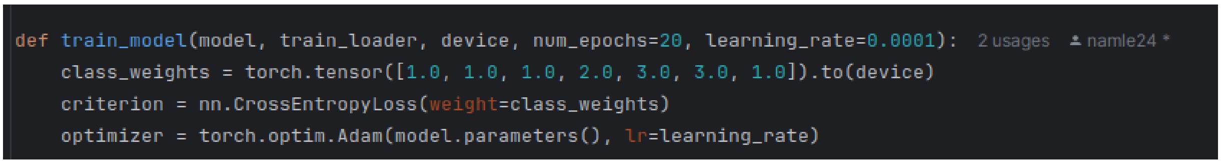
Create a dataloader for easy access to batch data. Create image transforms including resizing, flipping, rotating to optimize training.



```
def get_dataloader(image_dir, mask_dir, batch_size=4, shuffle=True):  2 usages  namle24
    transform = transforms.Compose([
        transforms.Resize((224, 224)),
        transforms.RandomHorizontalFlip(),
        transforms.RandomRotation(30),
        transforms.ToTensor()
    ])
```

d) Model training

The model is trained using CrossEntropyLoss with class weights and the Adam optimizer. The training loop iterates over the dataset for a specified number of epochs, calculating loss and accuracy



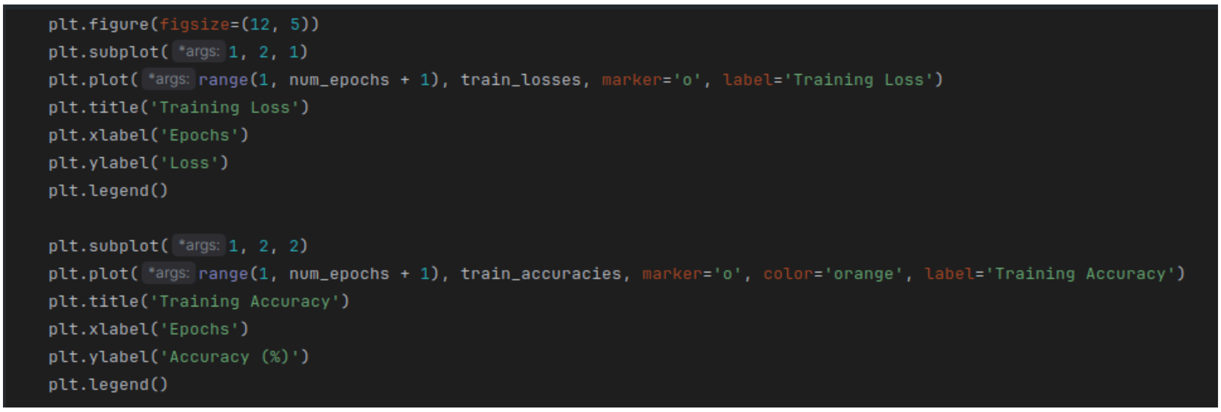
```
def train_model(model, train_loader, device, num_epochs=20, learning_rate=0.0001):  2 usages  namle24 *
    class_weights = torch.tensor([1.0, 1.0, 1.0, 2.0, 3.0, 3.0, 1.0]).to(device)
    criterion = nn.CrossEntropyLoss(weight=class_weights)
    optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

e) Evaluation and Visualization

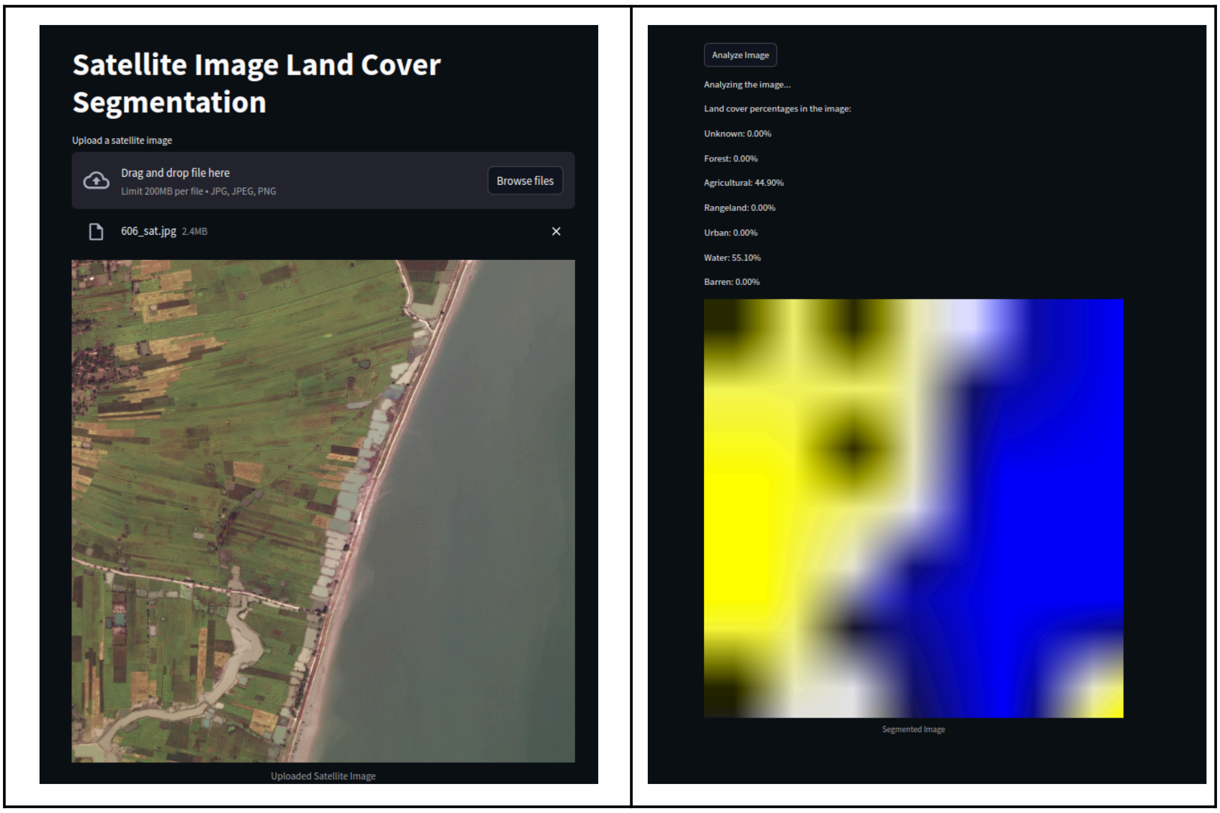
Finally, we will plot a comparison between the original RGB image and the image with segmentation annotations using plt.show() command.

f) Results

Track training loss and accuracy over epochs and plot them to ensure proper convergence.



```
plt.figure(figsize=(12, 5))
plt.subplot( *args: 1, 2, 1)
plt.plot( *args: range(1, num_epochs + 1), train_losses, marker='o', label='Training Loss')
plt.title('Training Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.subplot( *args: 1, 2, 2)
plt.plot( *args: range(1, num_epochs + 1), train_accuracies, marker='o', color='orange', label='Training Accuracy')
plt.title('Training Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy (%)')
plt.legend()
```



Satellite Image Land Cover Segmentation

Upload a satellite image

Uploaded Satellite Image

Analyzing the image...

Land cover percentages in the image:

Unknown: 0.00%

Forest: 0.00%

Agricultural: 44.90%

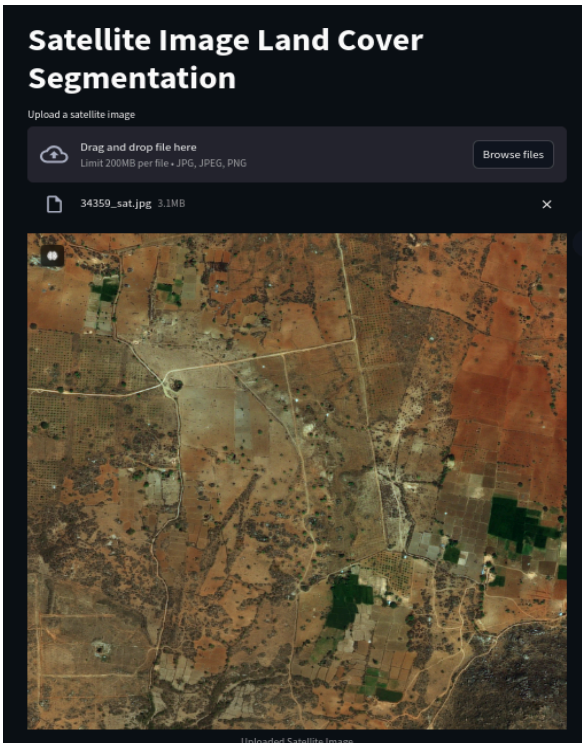
Rangeland: 0.00%

Urban: 0.00%

Water: 55.10%

Barren: 0.00%

Segmented Image

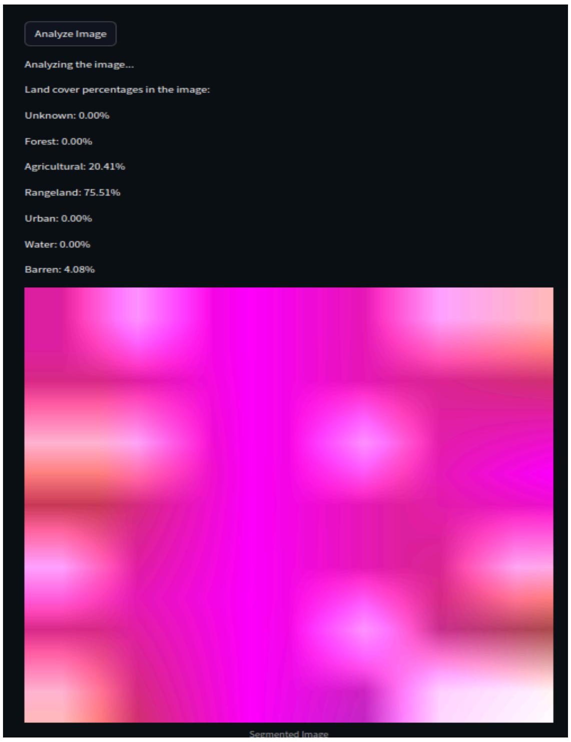


Satellite Image Land Cover Segmentation

Upload a satellite image

34359_sat.jpg 3.1MB

Uploaded Satellite Image



Analyzing the image...

Land cover percentages in the image:

Unknown: 0.00%

Forest: 0.00%

Agricultural: 20.41%

Rangeland: 75.51%

Urban: 0.00%

Water: 0.00%

Barren: 4.08%

Segmented Image

Chapter 4: Discussion

a) What makes Swin Transformers an innovation ?

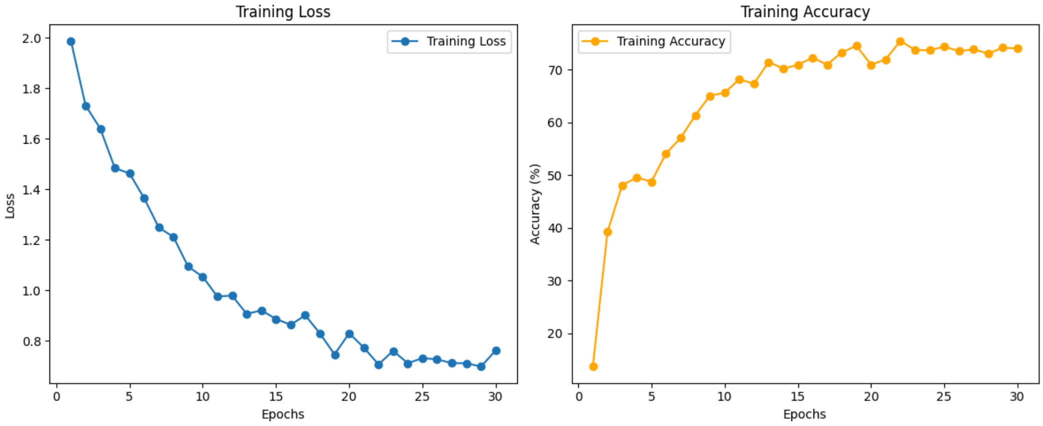
- **Hierarchical Feature Learning:** Allows learning features at multiple levels, from local to global, similar to CNN, making the model effective in segmenting both small and large objects.
- **Shifted Window Attention:** Reduces computational complexity by computing attention within local windows and shifting them across layers, capturing long-range relationships between image regions.
- **Efficient for High-Resolution Images:** The window-based attention mechanism significantly reduces computational costs, enabling Swin Transformer to handle

high-resolution images while preserving fine details.

- **Suitable for Complex Segmentation Tasks:** The model excels in segmentation tasks due to its ability to capture fine details and long-range dependencies.

b) Swin Transformer's drawbacks

1. **System resource :** We performed the test on 2 machines, one is a laptop 8GB Ram, Intel i5 gen 9th, one is a Dell 12gb, both train on cpu. The first machine took 7 hours to finish the test and during the run time consumed 2.5gb-4 GB memory for first train, but after 5th train we spent over 10 -30 minutes for each test.
2. **Model parameters :** The first epoch's results are always the lowest and the completion time is the longest, the model requires large resources, the complex structure makes it more difficult to deploy and optimize, it is difficult to segment rare classes in the dataset, often leading to low accuracy for these classes.



Chapter 5: Conclusion

Swin Transformers is a powerful neural network architecture in the field of deep learning, particularly in image processing. This model stands out for its ability to handle images of various sizes, its hierarchical structure that learns features from basic to complex, and its efficient attention mechanism to focus on important regions. With high performance in numerous benchmarks and the potential for computational optimization, Swin Transformers promise many potential applications in object recognition, image classification, and segmentation. Overall, this model represents a significant advancement in deep learning technology.

Bibliography

1. https://www.facebook.com/DataScienceWorld.Kan/posts/swin-transformer-m%C3%B4-h%C3%ACnh-ph%C3%A2n-c%E1%BA%A5p-transformer-k%E1%BA%BFt-h%E1%BB%A3p-d%E1%BB%8Bch-chuy%E1%BB%83n-windowswin-tra/4894441707257793/

2. https://openaccess.thecvf.com/content/ICCV2021/papers/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper.pdf?fbclid=IwY2xjawFnoydleHRuA2FlbQIxMAABHSjfTKKU9msYBpWmyb99L__yUpBGIidule9ouyu9Xz9he0MYGfMtHsCfKA_aem_8qBNG1CdCvyNctNi30GXOw

3. https://github.com/microsoft/Swin-Transformer

4. https://www.youtube.com/playlist?list=PL9iXGo3xD8jokWaLB8ZHUkjjv5Y_vPQnZ

5. https://huggingface.co/microsoft/swin-base-patch4-window7-224

6. ▶ Swin Transformer - Paper Explained

7. https://viblo.asia/p/tim-hieu-ve-swin-transformers-5OXLAAoaLGr