

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)



University of Science and Technology of Hanoi

Content

Chapter 1: Introduction.....	2
a. What is Swin Transformer.....	2
b. What is Image Segmentation.....	2
Chapter 2: Research Method.....	3
a) Task.....	3
b) Model.....	3
c) Dataset.....	4
Chapter 3: Model architecture and methods.....	4
a) Importing Necessary Libraries.....	4
b) Configuration.....	5
c) Dataset and Data Processing.....	5
d) Model training.....	6
e) Evaluation and Visualization.....	6
f) Results.....	6
Chapter 4: Discussion.....	8
a) What makes Swin Transformers an innovation ?.....	8
b) Swin Transformer's drawbacks.....	9
Chapter 5: Conclusion.....	9

Chapter 1: Introduction

a. What is Swin Transformer

- The Swin Transformer, often known as the Shifted Window Transformer, is a vision transformer developed to enhance prior transformer structures for image processing tasks, especially in computer vision. It was initially introduced in the 2021 research "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" by Liu et al., and ever since then, its performance and efficacy in many applications, including as segmentation, object detection, and picture classification, have gained it widespread acclaim. The Swin transformer has a 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.

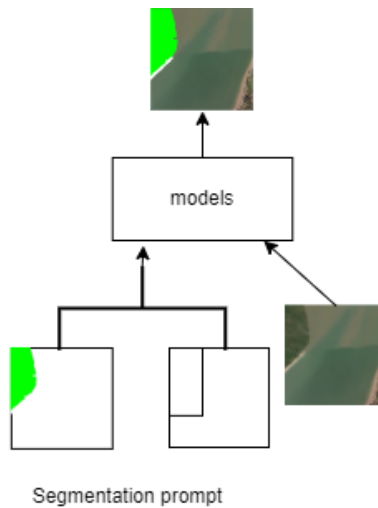


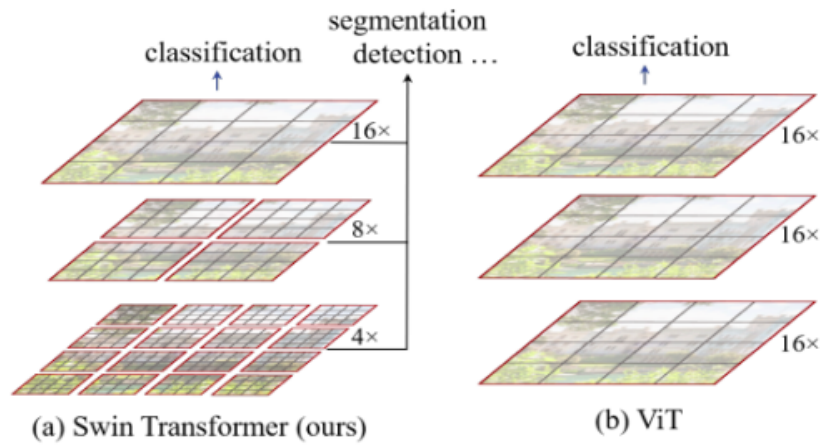
Figure 3.1: Promptable segmentation

b) Model

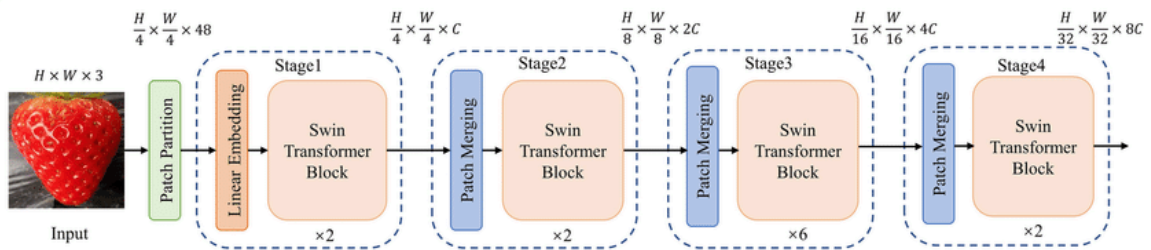
As part of the preparation process, the input image is resized and pixel values are normalized.

Then, the image is divided into 196 small patches of 16 x 16 pixels to create an image of 224 x 224 pixels. Each of these patches is converted into a feature vector using linear projection.

To determine the relationship between these patches and understand the spatial dependence in the image, the model applies a shifting window attention technique



(a) Architecture



(b) Swin Transformer Blocks

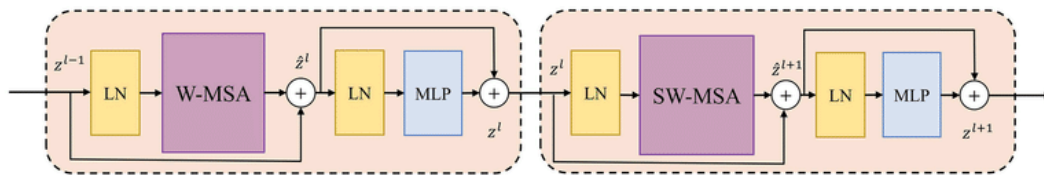


Figure 3.2: Segment Anything Model

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.

```
24 def get_device(): 4 usages 1 name24
25     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,...).

```
4 class SwinTransform(nn.Module): 5 usages 1 name24
5     def __init__(self, num_classes): 1 name24
6         super(SwinTransform, self).__init__()
7         self.backbone = timm.create_model(model_name='swin_base_patch4_window7_224', pretrained=True)
8         self.backbone.head = nn.Identity()
9
10        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.

```

19 def rgb_to_class(mask): 1 usage 1 name24
20     mask = np.array(mask)
21     class_mask = np.zeros(shape=(mask.shape[0], mask.shape[1]), dtype=np.uint8)
22
23     for color, class_id in color_mapping.items():
24         class_mask[(mask == color).all(axis=-1)] = class_id
25
26     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): 1 name24
    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 1 name24
    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 1 name24 *
    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

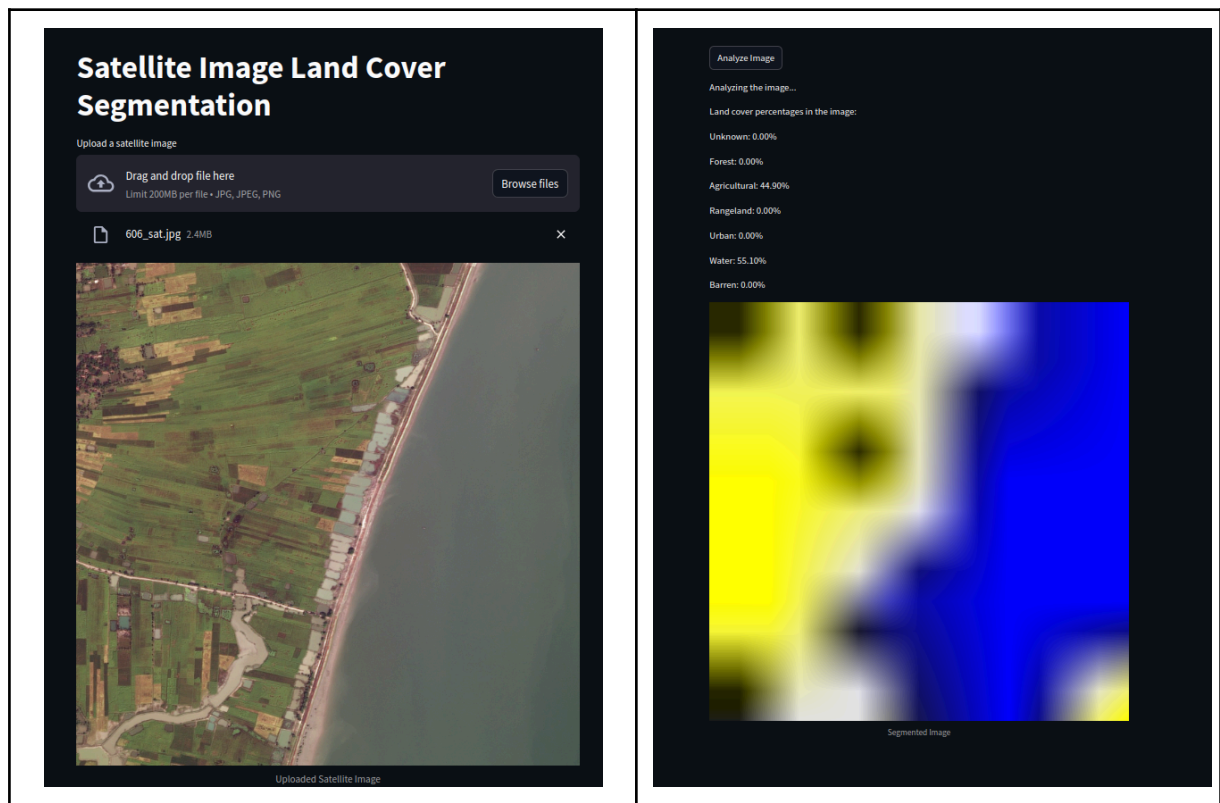
Finally, we will plot a comparison between the original RGB image and the image with segmentation annotations using `plt.show()` command.

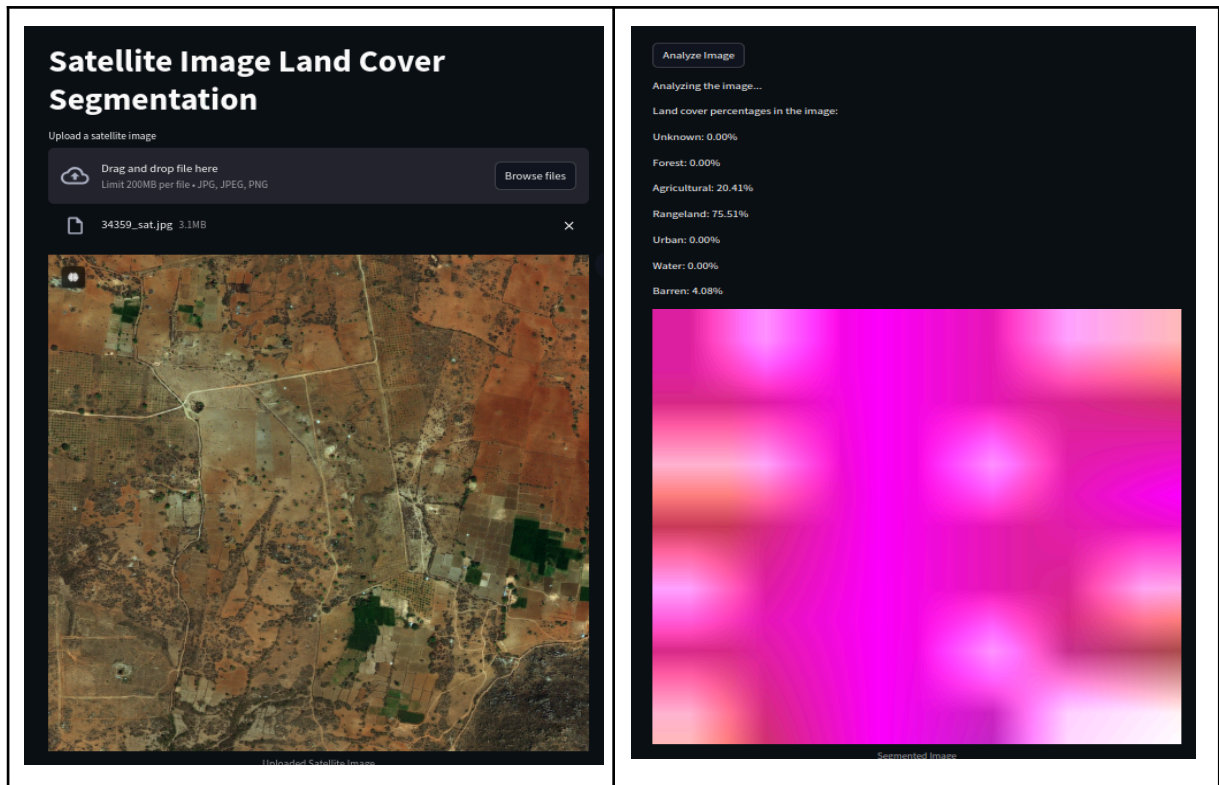
f) Results

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()
```





Chapter 4: Discussion

a) What makes Swin Transformers an innovation ?

Various leveled highlight learning: Permits learning highlights at numerous levels, from neighborhood to worldwide, comparable to CNN, making the show viable in portioning both little and expansive objects.

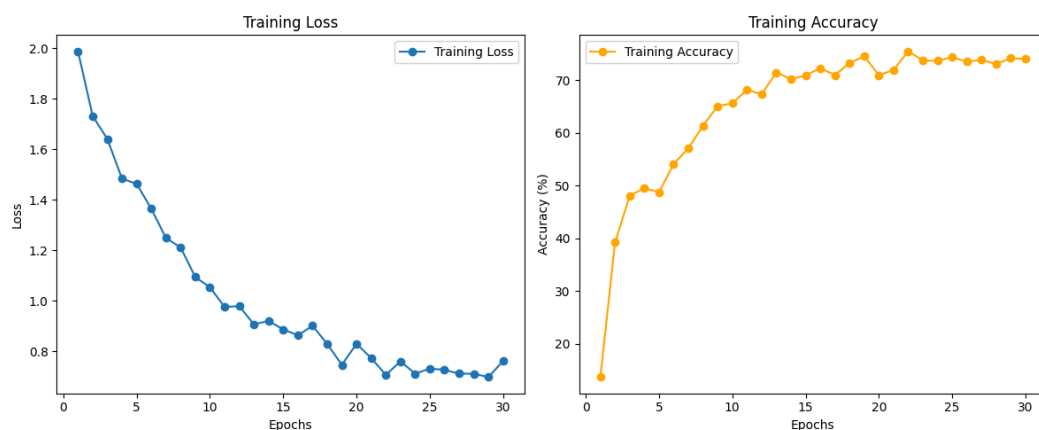
Adjusted window consideration: Diminishes computational complexity by computing consideration in nearby windows and exchanging them over layers, capturing long-range connections between picture locales.

Productive for high-resolution pictures: The window-based consideration instrument altogether diminishes computational costs, permitting Swin Transformer to prepare high-resolution pictures whereas protecting fine subtle elements .

Reasonable for complex division assignments: This show exceeds expectations in division assignments due to its capacity to capture fine subtle elements and long-range conditions.

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

After the training process, Swin Transformer achieved result: average loss: 0.9, average accuracy 80%. Swin transformer is one of the finest neural organize topologies for profound learning, especially picture preparing. Since it can prepare pictures of diverse sizes, learn highlights from basic to complex, and center on imperative districts with a

solid consideration component, this demonstrate is exceptional. It also has a hierarchical structure of Swin Transformers show great promise for a variety of future applications in object recognition, image classification, and segmentation because to their remarkable performance in several benchmarks and computational optimization prospects. Considering everything, this model is a major breakthrough 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=IwY2xjawFnoydleHRuA2FlbQlXMAABHSjfTKKU9msYBpWmyb99L_yUpBGlidule9ouyu9Xz9he0MYGfMtHsCfKA_aem_8qBNG1CdCvyNctNi30GXOw
3. <https://github.com/microsoft/Swin-Transformer>
4. https://www.youtube.com/playlist?list=PL9iXGo3xD8jokWaLB8ZHUKjiv5Y_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>