UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HA NOI

INFORMATION AND COMMUNICATIONS TECHNOLOGY

INTRODUCTION TO DEEP LEARNING COURSE

SWIN TRANSFORMER FOR REMOTE SENSING IMAGE

Phạm Đức Khiêm - BA12-095

Luyện Phạm Ngọc Khánh - BA12 -093

Trần Ngọc Việt Anh - BA12-003

Ngô Huyền Anh - BA12-006

Tăng Văn Anh - BA12-007

Nguyễn Đình Hải - BA12-068

HA NOI - 2024

Table of contents

1. Introduction:

Deep learning approaches have proven to be effective in extracting useful information from large amounts of unstructured data, offering data-driven solutions for tasks like representation and decision-making. They have the potential to advance a number of practical applications and research areas, such as imaging, pattern recognition, audiovisual signal analysis, sensor technology, and imaging. New opportunities are emerging in domains like sensing, imaging, and video processing as sophisticated deep learning architectures like transformer networks, generative adversarial networks (GANs), recurrent neural networks (RNNs), deep neural networks (DNNs), and convolutional neural networks (CNNs) continue to evolve. These days, vision transformers are used for numerous applications, such as satellite imagery analysis and medical imaging, as well as for productive picture processing. The Swin Transformer is a particular field among Vision Transformers; it is essential to mention at this stage. The goal of this research is to create and evaluate the SwinTransformer model, which is used in computer vision to recognize images. The process of developing and evaluating the model involves stages like preprocessing the data, building the model, training it, testing its performance, and providing predictions for classification.

2. Define the problem:

- Using the Swin Transformer model for remote sensing image classification:
    - Input: A remote sensing image form of pixels with 3 channels (R, G, B) and labels based on datasets
    - Output: The classification model returns the probability of predicting the image for each label

3. Methodology:

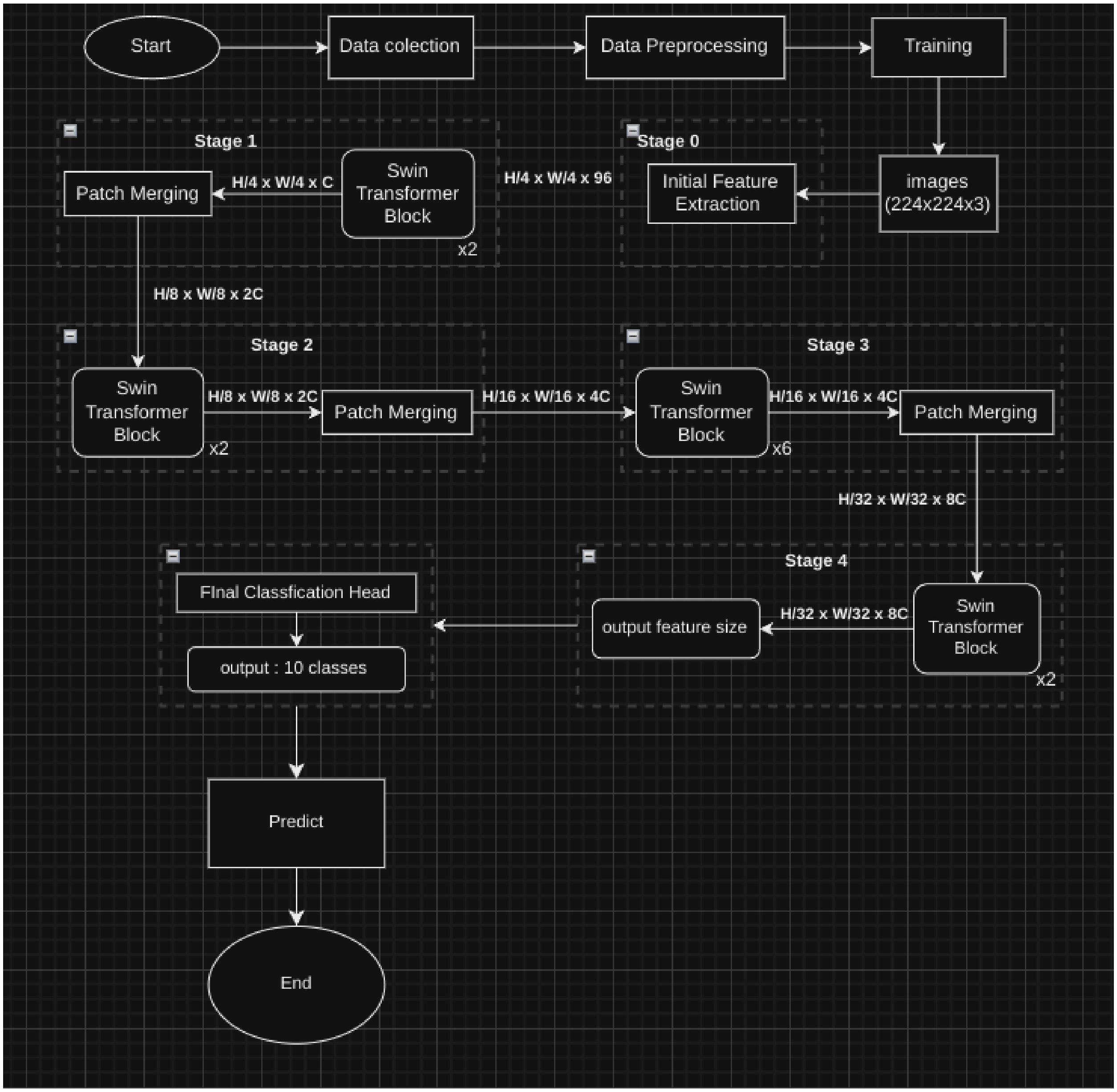


Figure 3.1. The diagram of the processing flow

3.1 Setup:

Using Pytorch framework. Load the model swin_t from torchvision

3.2 Data collection:

Sentinel 2 satellite pictures are used to classify land use and land cover using the EuroSAT dataset. With a total of categorized and geotagged photos, it spans LULC classes and features spectral bands. EuroSAT: A Novel Dataset and Deep

Learning Benchmark for Land Use and Land Cover Classification and Introducing EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification are the research articles that are linked to this dataset. The A JPG-formatted RGB version of the dataset displaying the optical R, G, and B frequency bands is available for download in the EuroSAT RGB zip file, which includes 10 labels (Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial, Pasture, Permanent Crop, Residential, River, Sea Lake) and around 27000 images.

3.3 Data preprocessing:

Data preprocessing plays a crucial role in the process as it helps transform raw data into a format. As it assists in converting unformatted data into a format, data preparation is an essential step in the process. Before training our model, there were a few missing values in the input dataset that needed to be changed to the default format of the Swin Transformer. We used preprocessing techniques to prepare the dataset for modeling:

- Resize: resize image to 224x224 pixel
- To Tensor: Pytorch requires the input data for the model to be in tensor format. Tensor can be used on GPU to speed up the computation.
- Normalize: It helps the model learn faster and improves performance.
- Activity: by formula (tensor - mean)/std = normalize

Channel	Mean value	Std
R	0.485	0.229
G	0.456	0.224
B	0.406	0.255

3.4 Training Model:

Swin Transformer model:

- Stage 0: The single image we take while gathering data has three dimensions: height (H), width (W), and channel (3). as well as the image's color. The number of channels is increased from 3 channels (RGB) to C channels, and the spatial dimension is decreased to 1/4 of the original height

(H) and width (W). While increasing the number of channels aids in the model's ability to retain more significant information from the original image, decreasing the spatial dimension lowers the computational effort. The model merely processes the first raw data in this stage, setting the foundation for subsequent processing layers.

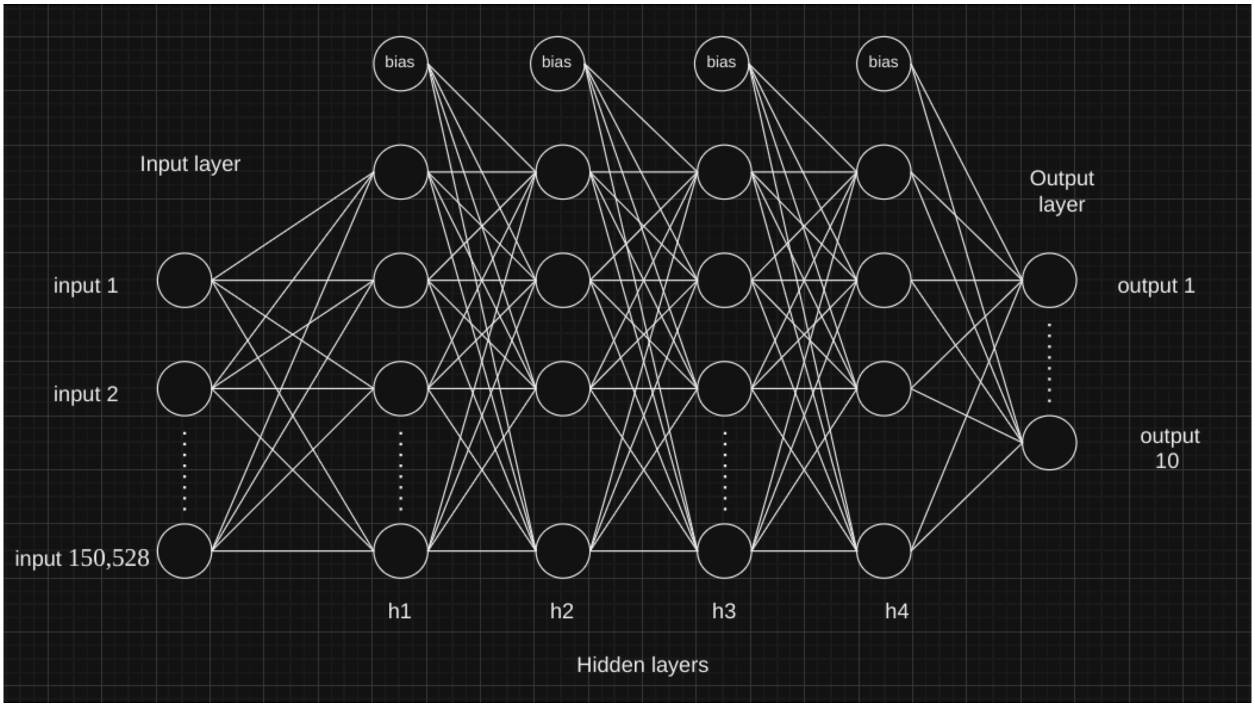
- Stage 1: In order to maintain the matrix size (h/4 x w/4 x C), with C being 96, it is crucial to process information both within and between patches. This must be done without altering the overall dimensions. However, when reducing the spatial size (for instance, from (H/4 x W/4) to (H/8 x W/8), one observes that both the height and the width are halved. Consequently, the number of feature channels is doubled. Perform the patch merging, which combines groups of 2x2 patches into a single patch; as a result, (the width and height are cut in half) and (the number of features per token is doubled: H/8, W/8, 2C). This reduction makes it possible to process the data more efficiently, but it also means that the doubled feature count must be carefully considered.
- Stage 2: Similar to the stage 1
- Stage 3: The model can access and process features as in stages 1 and 2 but more fully when it has more layers added (6 layers) to it when the Swin Transformer Block is reduced or made smaller (H/16 x W/16 x 4C). Since the model contains a significant quantity of data gathered from earlier stages, increasing the number of channels is imperative. The more layers in the model, the more sophisticated information it can interpret and amplify, as it will be able to capture. Careful calibration is required, though, as one must take into account the possibility of declining returns. The advantages are obvious, but implementation needs to be done carefully.
- Stage 4: The same as stage 1 and stage 2

The experiment will display the findings as follows after each stage:

	Input	Output	Layer Numbers
Stage 0	(H xW x3)	(H/4 x W/4 x C)	-
Stage 1	(H/4 x W/4 x C)	(H/8 x W/8 x 2C)	2
Stage 2	(H/8 x W/8 x 2C)	(H/16 x W/16 x 4C)	2
Stage 3	(H/16 x W/16 x 4C)	(H/32 x W/32 x 8C)	6
Stage 4	(H/32 x W/32 x 8C)	(H/32 x W/32 x 8C)	2

Note: As the default format of Swin Transformer : C= 96 and layer numbers ={2, 2, 6, 2}

3.5 Neural Network Architecture:



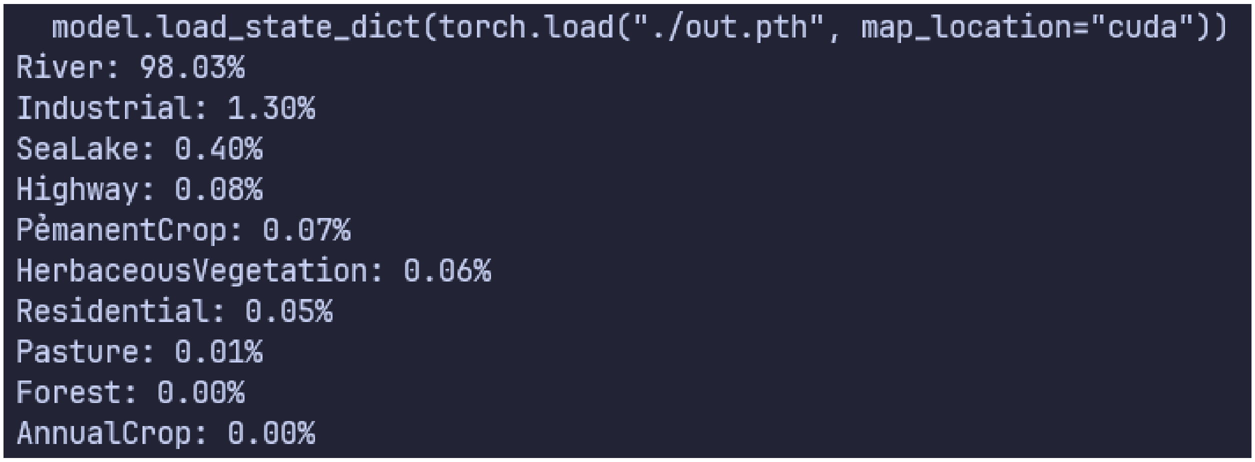
3.5.1 Figure of Architecture of Neural Network

- Input nodes: 150,528 (corresponding to the flattened image size of 224x224x3).
- Hidden Layers:
  - Layer 1: 301,056 nodes.
  - Layer 2: 150,528 nodes.
  - Layer 3: 75,264 nodes.
  - Layer 4: 37,632 nodes.
- Output nodes: 10 (for classification)

3.6 Criterion and optimizer:

- Criterion: **CrossEntropyLoss** measures the average number of bits required to identify an event from one probability distribution, p, using the optimal code for another probability distribution, q. In other words, cross-entropy measures the difference between the discovered probability distribution of a classification model and the predicted values. The cross-entropy loss function is used to find the optimal solution by adjusting the weights of a machine learning model during training. The objective is to minimize the error between the actual and predicted outcomes. A lower cross-entropy value indicates better performance.
- Optimizer: SGD (stochastic gradient descent) is a variation of Gradient Descent. It is an algorithm to find minimum value of J

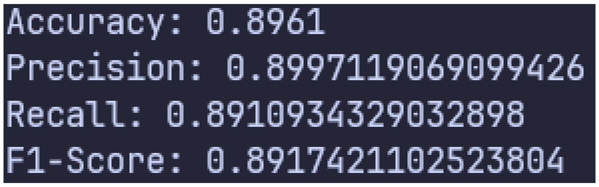
3. Result of experiment:



```
  model.load_state_dict(torch.load("./out.pth", map_location="cuda"))
River: 98.03%
Industrial: 1.30%
SeaLake: 0.40%
Highway: 0.08%
PẻmanentCrop: 0.07%
HerbaceousVegetation: 0.06%
Residential: 0.05%
Pasture: 0.01%
Forest: 0.00%
AnnualCrop: 0.00%
```

4.1 Figure of the predict

After the training process, the experiment starts to get the prediction. The prediction is the probability of which is the classification



```
Accuracy: 0.8961
Precision: 0.8997119069099426
Recall: 0.8910934329032898
F1-Score: 0.8917421102523804
```

4.2Figure of evaluate mode

5. Reference:

- Pytorch Module Library
- Pytorch Models and Pre-Trained weights Library
- Zero Sat Web
- Papers With Code
- Swin Transformer Papers: Hierarchical Vision Transformer using Shifted Windows