# ML Final Project

Student ID: 110550014 Student Name: 吳權祐

## Run Inference

Please modify the path of test data, the path to the pretrained model (should be the folder I provide) and the output path (if necessary).

```
model_path = './110550014_model.pt'          # path to the pretrained model
data_path = './test'                         # path to the test data
output_csv = './110550014_prediction.csv'    # path to output csv file
```

Here is an example directory

/110550014_final

- ㄴ110550014_inference.py
- ㄴ110550014_model.pt
- ㄴ/test
  - ㄴimage.jpg…
- ㄴ110550014_prediction.csv (after running inference file)

## Environment Details

- Python version: 3.10.13
- Framework: Pytorch

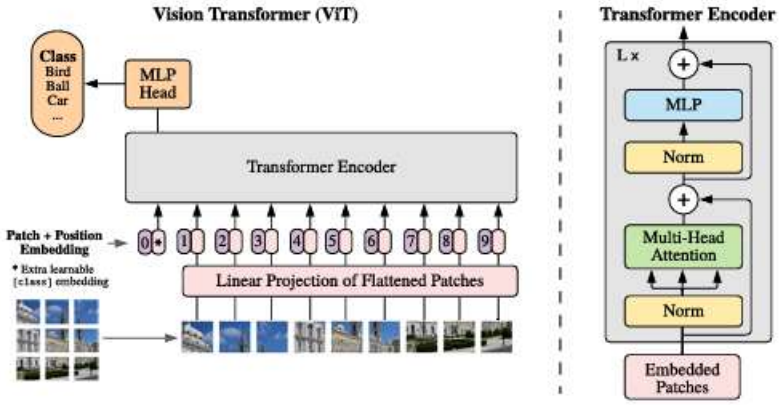
```
import os
import sys
from transformers import ViTImageProcessor, ViTForImageClassification
from transformers import TrainingArguments, Trainer
from datasets import load_dataset, Features, ClassLabel, Image
import torch
```

- Hardware:
  - GPU: NVIDIA GeForce GTX 1080Ti 11G with CUDA 12.3
  - CPU: Intel i7-12700

## Implement Detail

- Model architecture: Vision Transformer
  - num of hidden layers: 12
  - num of attention heads: 12
  - patch size: 16

  I use a single Vision transformer (ViT) as the main architecture in my model. ViT divides input image into fixed-size patches and get a input sequence for the encoder, and the encoder outputs a sequence of vector representations for each patch, and a classification token. Then this output will be send to the MLP Head to get a prediction class.




- Hyperparameters
  - Epoch: 5
  - Batch size: 10
  - Learning rate: 2e-5
  - Weight decay: 1e-4
- Training strategy

  First, it is necessary to split the train dataset into validation data and train data to make the model easily be evaluated during training. I have tried to separate 15% or 5% of the original data into the validation data, and the final ratio I choose is 5% since the I found that the model the model trained under this setting performs better.

  I have tried to finetune several pretrained models in Transformers library, and the final pretrained model I choose is a model from Google pretrained on ImageNet at resolution 224 x 224 (model name: google/vit-base-patch16-224).

  Then I trained a lot of models with different epochs and batches, finally I choose this set of hyperparameter.

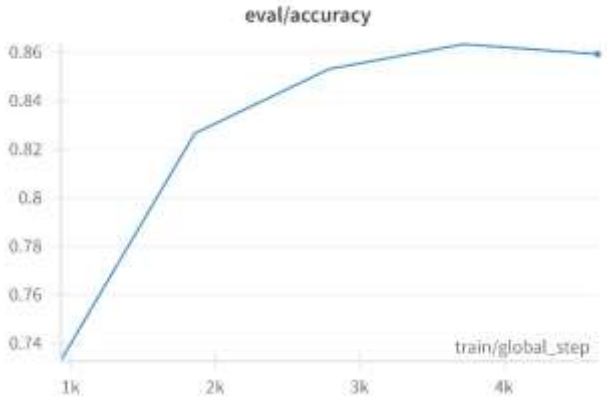
## Experimental Result

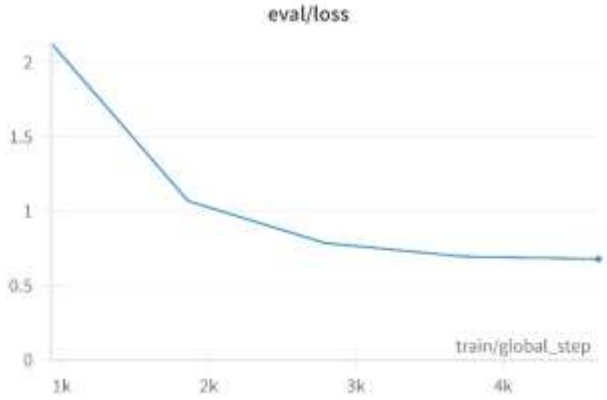
- Evaluation metrics

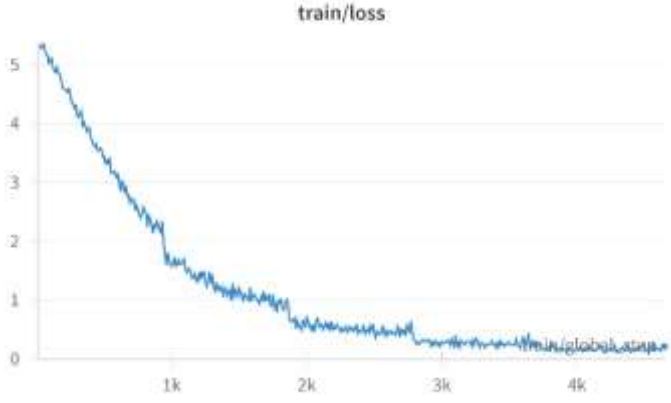
  I use evaluation accuracy and training loss as evaluation metrics

- Learning curve

  I use wandb to visualize my training result



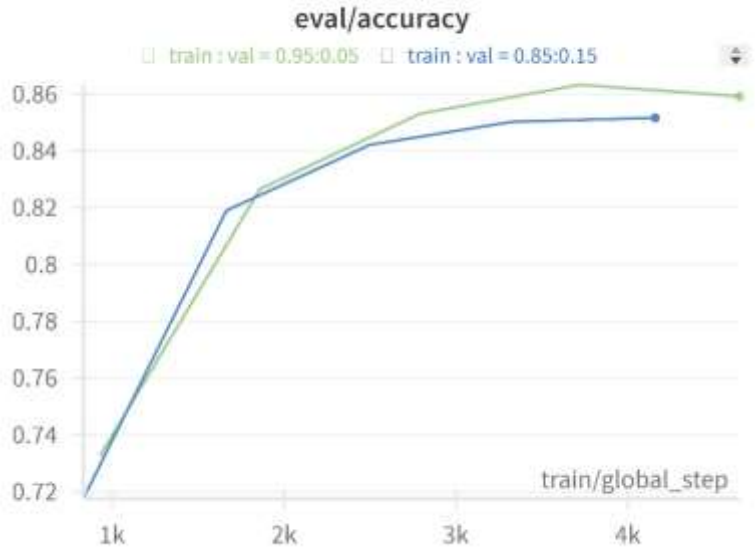



◦ Ablation study

I have done some experiments to study the effects of the ratio between training and validation data.

The following graph shows that the evaluation accuracy of the model trained on more training data is higher.

And finally, the performance of this model is 0.7% higher than another model in the competition.




## Bonus

◦ Discussion

I have tried to finetune several models from the pretrained ResNet50 model, but the performance of these models are not that good. So I started to train some model based on transformer and also I started to learn some knowledge of vision transformer via reding some paper or article online including first paper about ViT *( An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale ).*

But, after I read these papers or articles, I found out that I couldn't understand why these models could get that huge success. Compare to other architectures like CNN or ResNet, I think ViT has less interpretability. So I think I should be more careful to use such ViT based models in the future.