CSE 244 Final Project Report

-Nistha Kumar

Experimental Setup:

I started with training ResNet50 with very little data augmentation. Later shifted to EfficientNetB3 and was able to achieve an accuracy around 53%.

Then I started experimenting some models from timm library and this boosted my accuracy. I tried various models of **ViT** and **DEiT3**, for eg: '**vit_tiny_patch16_224**', '**deit3_huge_patch14_224**'.
But I achieved the best accuracy by using **'swin_large_patch4_window7_224'** which came out to **73.7** on the test dataset on Kaggle.

Some Data Augmentation Techniques used:
RandomResizedCrop, RandomHorizontalFlip, RandomVerticalFlip, ColorJitter and RandomGrayscale.

Also, I tried to increase the train data by adding data from The Food 101 dataset for all the classes present in our original dataset.

A bit about my code:

I used Kaggle(GPU P100) to train my model. Since I have done the whole code on Kaggle, there are some cells in the ipynb file which would run only on Kaggle and not on Colab or anywhere else.
A few to mention:

```
!pip install gdown
!gdown --id 1-3_5KxKYPqsDDYLdmjQoaoHuPFN0uF9c
```

# The above code is used to pull the dataset from the google colab to Kaggle which is in the zip format

```
! unzip data.zip
```

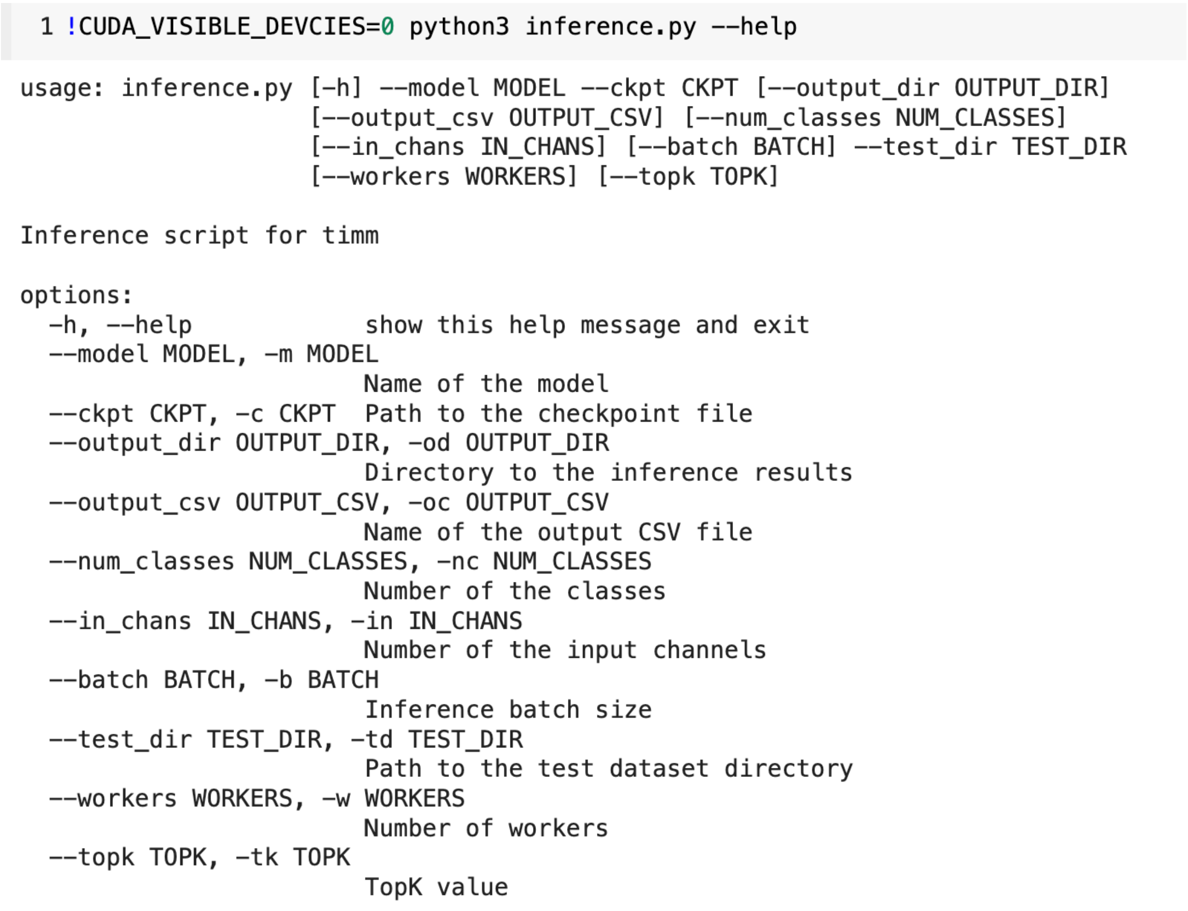
# Unzips the dataset and stores it under /kaggle/working/data

Files Submitted:

- cse244-project.ipynb
- inference.py
- best_model_params.pth.tar(https://drive.google.com/file/d/14ArnrDIp3prTwc5QEt8tjyRcwiHTHGnD/view?usp=drive_link)
- output.csv

Steps to reproduce the accuracy:

The weights for the best model are saved in **best_model_params.pth.tar.** And there is a script **inference.py** which can be run to achieve the final accuracy.



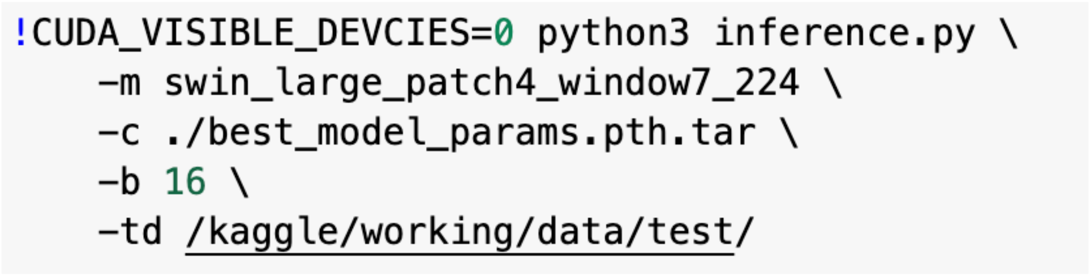
```
!CUDA_VISIBLE_DEVCIES=0 python3 inference.py --help
```

```
usage: inference.py [-h] --model MODEL --ckpt CKPT [--output_dir OUTPUT_DIR]
                    [--output_csv OUTPUT_CSV] [--num_classes NUM_CLASSES]
                    [--in_chans IN_CHANS] [--batch BATCH] --test_dir TEST_DIR
                    [--workers WORKERS] [--topk TOPK]

Inference script for timm

options:
  -h, --help            show this help message and exit
  --model MODEL, -m MODEL
                        Name of the model
  --ckpt CKPT, -c CKPT  Path to the checkpoint file
  --output_dir OUTPUT_DIR, -od OUTPUT_DIR
                        Directory to the inference results
  --output_csv OUTPUT_CSV, -oc OUTPUT_CSV
                        Name of the output CSV file
  --num_classes NUM_CLASSES, -nc NUM_CLASSES
                        Number of the classes
  --in_chans IN_CHANS, -in IN_CHANS
                        Number of the input channels
  --batch BATCH, -b BATCH
                        Inference batch size
  --test_dir TEST_DIR, -td TEST_DIR
                        Path to the test dataset directory
  --workers WORKERS, -w WORKERS
                        Number of workers
  --topk TOPK, -tk TOPK
                        TopK value
```

This is the way I ran to test the model. Here '**/kaggle/working/data/test/**' is the test directory on Kaggle where all the test files are present.



```
!CUDA_VISIBLE_DEVCIES=0 python3 inference.py \
    -m swin_large_patch4_window7_224 \
    -c ./best_model_params.pth.tar \
    -b 16 \
    -td /kaggle/working/data/test/
```

Note: I have noticed that the same model when run with different epochs sometimes end up giving 72.7 or 70.7 accuracy as well. But the best so far I got was 73.7 on Kaggle.