Name: Julius Chang , Ethan Cheng
E-mail: jchang90@ucsc.edu , echeng15@ucsc.edu

Contributions:
Work was split evenly, we each did about half.
Julius: data augmentations, data preprocessing, training/validation, optimizing hyperparameters, model and design choices
Ethan: loading Kaggle data, data preprocessing, training/validation, optimizing hyperparameters, model and design choices

Code:
Google Colab:
https://drive.google.com/file/d/1mtwNQxc0nJFRT27bSqos1skImQAWGzvv/view?usp=sharing

Model:

- Architecture: Vision Transformer (ViT) with a configuration of vit_l_16
  - Pre-trained on ImageNet-21k dataset with SWAG weights
  - 24 laters
  - 16x16 input patch size
  - 307M Parameters
- Modifications:
  - Custom classification head replacing the original final layer, composed of a linear layer reducing features to 512, following by ReLU activation, a dropout layer (p=0.4), and final linear layer outputting 100 classes
- Training set-up:
  - Optimizer: AdamW with a learning rate of 0.001
  - Learning rate scheduler: StepLR with step size of 10 and gamma of 0.1
  - Loss function: cross entropy loss
  - Total epochs: 17

Environment:
torch, torch.nn, torch.optim, torch.optim.lr_scheduler, numpy, pandas, torchvision.datasets, torchvision.transforms, matplotlib.pyplot, os, PIL, torch.utils.data.random_split, torch.utils.data.DataLoader

Usage:

1. Setup Kaggle
   a. upload kaggle.json file and place in it the appropriate directory
2. Download and unzip data:
   a. use kaggle API command to download the dataset
   b. Unzip the downloaded data
3. Set the Seed
   a. We used a seed of 42 for the uploaded final model weights
4. Data preparation:

    a. apply data transformations for both training (with augmentation) and validation (without augmentation)
    b. split the training dataset into training and validation set using a 80:20 split
5. Model training:
    a. Load the pre-trained ViT model and replace the classification head
    b. freeze the backbone parameters and fine-tune the classification head using the AdamW optimizer and StepLR scheduler
    c. train the model for 17 epochs
6. Evaluation:
    a. Validate the model on the validation set
    b. run inference on the test set to generate predictions
7. Submission:
    a. map the predicted labels to the corresponding class names and the save the results in a submission.csv file

Weight:
https://drive.google.com/file/d/12FMpiljU3B9L-fZsfMCmeQc-MuGyLk9_/view?usp=sharing

Accuracy: 0.81939 on Kaggle