

## CSE 144 Final Report

**Group**Neil Grover : [negrover@ucsc.edu](mailto:negrover@ucsc.edu)Russell Elliott: [rdelliot@ucsc.edu](mailto:rdelliot@ucsc.edu)**Model**

Vision Transform 16-bit with custom Fully Connected layer as shown below

```

Sequential(
  (0): CustomResNetHead(
    (fc): Sequential(
      (0): BatchNorm1d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (1): Dropout(p=0.4, inplace=False)
      (2): ReLU()
      (3): Linear(in_features=2048, out_features=100, bias=True)
      (4): Softmax(dim=1)
    )
  )
)

```

**Libraries Used**

```

os
torch.utils.data → Dataset, DataLoader
torchvision.transforms → transforms
PIL → Image
torch
torch.nn
torch.optim
torchvision → models
torch.multiprocessing
pandas
tqdm → tqdm

```

**How to Run**

1. Navigate to Canvas to submission for this report and open the Jupyter notebook
2. Select run all cells. This should:
  - a. Load and augment training images into train, validation split
  - b. Run the CNN training loop using transfer learning on Resnet50
  - c. Print loss and accuracy curves for train and validation
  - d. Generate the test labels to submit to Kaggle.
3. The output `predictions.csv` can be submitted to kaggle for competition results.

## **Model Weights**

These can be accessed as a .pth file through:

1. Canvas as a part of the report submission
2. Generated by the jupyter notebooks along with the test predictions.

The csv and pth files can be accessed at this Google Drive:

<https://drive.google.com/drive/u/0/folders/10oZ2-4IH5XbTAqlzsBQN1lkgmy7rVit>

Note that the train and test folders are accessed through there rather than locally.