CAP6411 Assignment 1: ResNet Vs. ViT

Joshua Yu

August 28, 2025

1 Introduction

This is a short write-up on my results for Assignment 1: ResNet Vs. ViT. In this assignment, we were tasked with training and evaluating a ResNet CNN model and a Vision Transformer model and comparing the results. The models were trained and evaluated on the Human Action Recognition Dataset from Kaggle. The video demo can be found at the following link: https://youtu.be/15Tm6fYYTJQ

2 ResNet Model

ResNet was the first model to be trained with transfer learning and evaluated. The specific model selected was ResNet18. In order to adapt the model to the dataset, the last fully-connected layer was replaced with a 15-output fully-connected layer to match the number of classes found in the Human Action Recognition Dataset. The total number of trainable parameters for the model was 11.18 million.

Before the model was trained, the dataset was pre-processed. For the training dataset, the images were resized to 224 x 224 and augmented with random horizontal flips and random rotations. Additionally, the images were normalized. For the validation dataset, the images were just resized to 224 x 224 and normalized.

Training parameters were used as follows. The model was trained for five epochs, with a batch size of 32. Cross entropy loss was used along with the Adam optimizer with a learning rate of 0.001.

The training log for the ResNet18 model can be found below.

```
--- Training Started for ResNet ---
Epoch [1/5], Loss: 1.7234, Val Accuracy: 46.72%
Epoch [2/5], Loss: 1.3711, Val Accuracy: 50.90%
Epoch [3/5], Loss: 1.1614, Val Accuracy: 59.68%
Epoch [4/5], Loss: 1.0407, Val Accuracy: 61.90%
Epoch [5/5], Loss: 0.9207, Val Accuracy: 58.25%
Total Training Time: 100.86 seconds
Peak GPU Memory during Training: 1223.59 MB
```

3 ViT Model

A ViT model was the second model to be trained with transfer learning and evaluated. The specific model selected was the "vit_base_patch16_224" model from the timm Python library. The last layer (the head) was replaced with a 15-output fully-connected layer to match the number of classes in the dataset, as with the ResNet18 model. The total number of trainable parameters for the model was 85.81 million.

Before the model was trained, the dataset was pre-processed the same as with training ResNet18. For the training dataset, the images were resized to 224 x 224 and augmented with random horizontal flips and random rotations. Additionally, the images were normalized. For the validation dataset, the images were resized to 224 x 224 pixels and normalized.

Training parameters were used as follows. The model was trained for five epochs, with a batch size of 32. Cross-entropy loss was used along with the AdamW optimizer with a learning rate of 0.0001.

The training log for the Vit model can be found below.

```
--- Training Started for ViT ---
Epoch [1/5], Loss: 1.5582, Val Accuracy: 68.15%
Epoch [2/5], Loss: 0.7212, Val Accuracy: 76.40%
Epoch [3/5], Loss: 0.5099, Val Accuracy: 77.62%
Epoch [4/5], Loss: 0.4167, Val Accuracy: 75.98%
Epoch [5/5], Loss: 0.3240, Val Accuracy: 77.46%
Total Training Time: 822.61 seconds
Peak GPU Memory during Training: 4897.80 MB
```

4 Results & Insights

The results of the two models are summarized in the following table, followed by confusion matrices for the two models.

Table 1: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
ResNet-18	0.58	0.65	0.58	0.58
Vision Transformer (ViT)	0.77	0.78	0.77	0.77

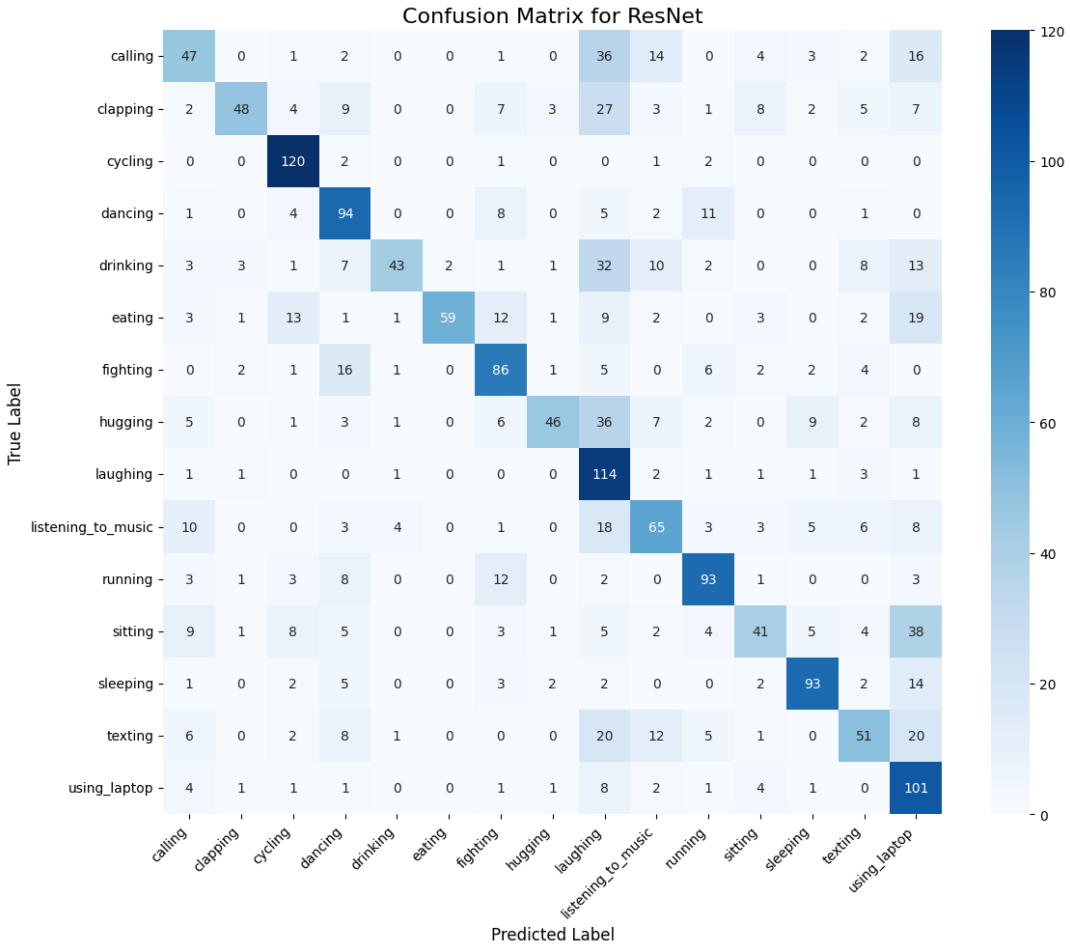

Figure 1: ResNet Confusion Matrix

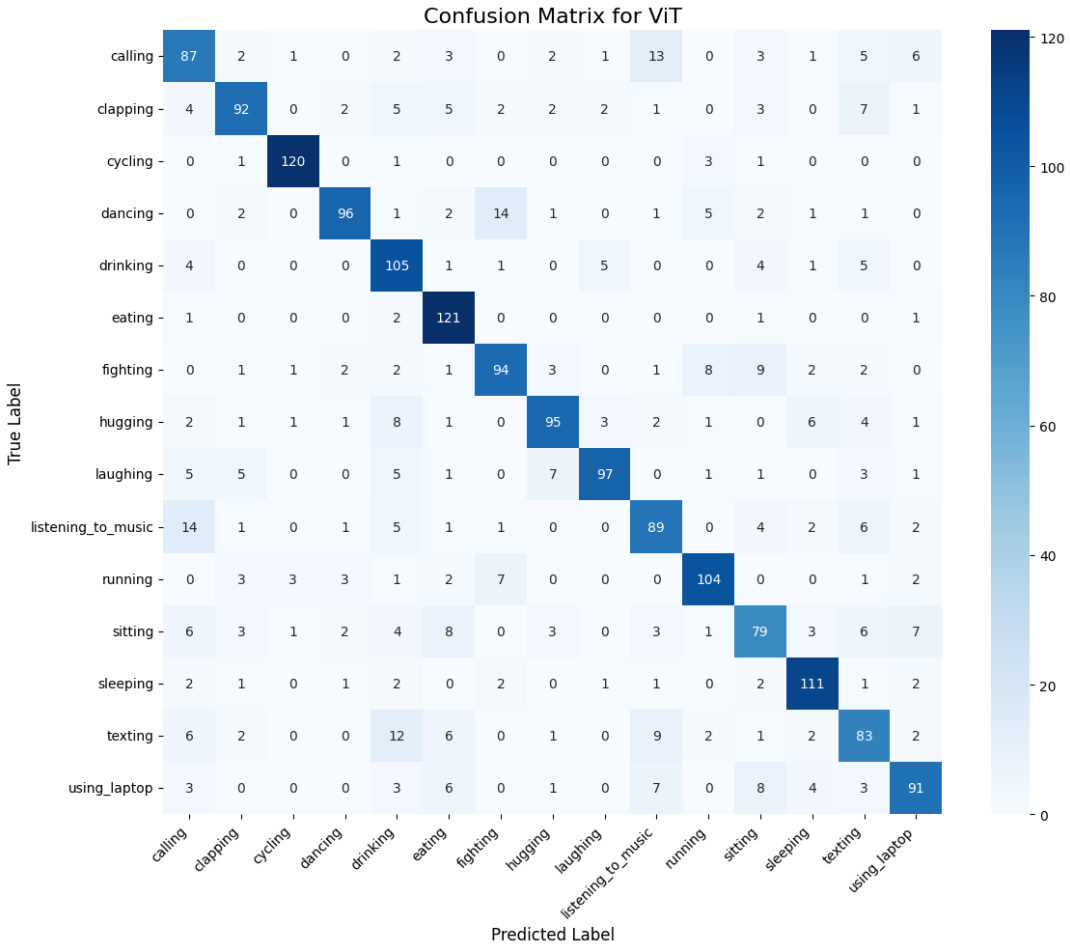

Figure 2: ViT Confusion Matrix

The Vision Transformer shows significantly better performance across the board, with an overall accuracy of 0.77 versus the accuracy of ResNet which comes in at 0.58. The Vision Transformer shows similarly elevated performance in precision, recall, and F1-score. Looking a the confusion matrices, the Vision Transformer shows less classes with a misclassification trend, e.g. clapping and drinking. What is interesting is that with both models, the classes that show high misclassification rates are not the same. For instance, with the calling class, the ResNet model consistently misclassifies inputs as laughing, listening_to_music, and using_ laptop whereas the Vision Transformer tends to misclassify calling as listeningtomusic, texting, and usinglaptop. It would be interesting to explore what the models are doing differently such that they misclassify in different ways.

Now, here are some performance metrics for the two models.

Table 2: Computational and Training Performance Comparison

Metric	ResNet-18	Vision Transformer (ViT)
Model Characteristics		
Trainable Parameters (M)	11.18	85.81
Disk Size (MB)	42.74	327.39
GFLOPs	1.82	16.87
Training Performance		
Total Training Time (s)	100.86	822.61
Peak Training GPU Memory (MB)	1223.59	4897.80
Inference Performance		
Total Inference Time (s)	2.60	9.51
Peak Inference GPU Memory (MB)	1730.78	1756.65

With the models selected, the Vision Transformer is way bigger with its 85.51 million parameters versus the ResNet model's 11.18 million parameters. With that, it is expected that the model has great classification performance. However, it can be seen that this performance comes at great computational cost. The model takes up way more space on disk and takes much more memory to train. Additionally, it is slower to train and at inference time. The Vision Transformer is almost four times slower than the ResNet model!

It may be worthwhile to explore Vision Transformers of different sizes, as this one's performance is not high enough to justify the computational cost compared to ResNet. The author did a little experimentation with a small ViT model (vit_tiny_patch16_224) with around 5.5 million parameters and discovered that the model offered increased performance compared to the ResNet model (0.72 test accuracy) while running at almost the exact same speed.

5 Challenges

Several challenges were encountered while attempting to train and evaluate the models, mostly pertaining to the software environment and computational restraints. Initial training was attempted on a local machine. Upon initial training of the ViT, it was discovered that the training was so slow that nothing would show up in the progress bar. This prompted the use of Google Colab Pro, which alleviated some of these issues. However, using Colab had its own challenges. Translating the Colab environment into a local environment that could reproduce this code was very challenging. Python environment and package management proved to be the most challenging part of this assignment.

6 Conclusion

With this experimentation, it has been demonstrated that with the specific models tested, the ViT is much more performant than the ResNet CNN. However, this comes at great computational cost. A more optimal solution would be a highly-optimized, compact ViT with fewer parameters. More experimentation and research remains to be done.