Model Training Report

Jakkula Adishesh Balaji
AI24BTECH11016

May 12, 2025

1 Summary Report

1.1 Final Accuracy and Class-wise Metrics

The model was trained across multiple configurations, and the following metrics were obtained:

- **Best Validation Accuracy:** 0.8238 from the final ResNet-18 model with parameters listed
- **Final Test Accuracy:** 0.85023% (on the test dataset after training the best model).

For the finetuning experiments, the following metrics were achieved:

- **Experiment 1:**
  - **Learning Rate:** 10^{-4}, **Batch Size:** 64, **Hidden Units:** 256, **Dropout:** 0.2
  - **Train Accuracy:** 88.00%, **Validation Accuracy:** 81.00%
  - **Precision:** 0.7896, **Recall:** 0.7761, **F1-score:** 0.7630
- **Experiment 2:**
  - **Learning Rate:** 10^{-4}, **Batch Size:** 64, **Hidden Units:** 1024, **Dropout:** 0.5

- **Train Accuracy:** 86.66%, **Validation Accuracy:** 83.25%
  - **Precision:** 0.8355, **Recall:** 0.8169, **F1-score:** 0.8112

- **Experiment 3:**
  - **Learning Rate:** 10^{-3}, **Batch Size:** 128, **Hidden Units:** 256, **Dropout:** 0.5
  - **Train Accuracy:** 57.09%, **Validation Accuracy:** 64.40%
  - **Precision:** 0.5734, **Recall:** 0.5598, **F1-score:** 0.5358

- **Final Model :**
  - **Architecture:** ResNet18 with pretrained weights, layers frozen up to `layer4`
  - **Classifier:** Custom classifier with one hidden layer of 512 neurons and 315 output classes
  - **Loss Function:** CrossEntropyLoss
  - **Optimizer:** AdamW with learning rate = `5e-4`, weight decay = `1e-4`
  - **Scheduler:** CosineAnnealingLR with `T_max = 15`
  - **Training:** 15 epochs, early stopping enabled
  - **Precision:** Mixed precision using `GradScaler`
  - **Data Augmentation:** Strong augmentations including occlusion, geometric and color transforms
  - **Train Accuracy:** 81.72%, **Validation Accuracy:** 83.01%
  - **Train Loss:** 0.0817, **Validation Loss:** 0.8130

- **EfficientNet-B0 Model:**
  - **Final Accuracy:** This model was worse than the simple overfitting models since it took a longer time to run and achieved similar validation accuracies, but on the flipside, the validation and train accuracies were close

1.2 Challenges Faced During Training and How They Were Overcome

Several challenges were encountered during the training process:

- **Overfitting:** The model initially showed signs of overfitting, especially when training on the smaller datasets. To counter this, early stopping, stronger data augmentation, and regularization techniques such as L2 regularization were employed.
- **Suboptimal Learning Rates:** During some experiments, the learning rate was either too high or too low, causing the model to either converge too slowly or not converge at all. The learning rate scheduler and adjustments to batch sizes helped address this issue.
- **Training Time:** The model took a considerable amount of time to train with multiple experiments due to the complexity of the models and the dataset size. This was mitigated by using a more powerful GPU and optimizing the batch size.

1.3 Reflections

What Worked Well: Freezing early layers of ResNet18 allowed faster convergence without significant loss in performance, thanks to the pretrained weights capturing low-level features. Using AdamW with weight decay and CosineAnnealingLR improved training stability and generalization, this reflects in the final test accuracy. I used the simpler models which had similar validation accuracies as the final model but got an accuracy of 52 %. Mixed precision training also helped reduce memory usage and speed up computation.

What Didn't Work Well: I couldn't reach a validation accuracy of greater than 90 % on the validation dataset. I couldn't show a clear difference between simpler models and the more advanced final model based on the train and validation accuracies alone. I couldnt experiment with deeper models properly, the showed worser performances than simpler models. I think ResNet50 could reach higher accuracies like 90 % after similar training and higher epochs

Impact of Augmentation and Architecture Choices: Stronger data augmentations, especially random occlusion and affine transformations, significantly improved model robustness by forcing it to learn more general features. The use of ResNet18 with frozen early layers struck a good balance between performance and training time. The combination of augmentation and architecture was key to achieving over 83% validation accuracy.

What We'd Improve with More Time or Data: With additional time and data, we would explore deeper architectures (e.g., ResNet50), perform hyperparameter tuning (e.g., learning rate schedules, dropout), and experiment with advanced techniques like self-supervised pretraining or test-time augmentation. Further dataset balancing and hard example mining could also enhance performance.