

Report

August 6, 2024

1 Project Report: Food-101 Image Classification using Transfer Learning

1.1 1. Dataset Summary

Dataset Name: Food-101

Description: The Food-101 dataset consists of 101 food categories with 1,000 images per class. For this project, we filtered the dataset to include only the first 20 classes.

Classes: The first 20 classes are:

- Apple Pie
- Baby Back Ribs
- Baklava
- Beef Carpaccio
- Beef Tartare
- Beet Salad
- Beignets
- Bibimbap
- Bread Pudding
- Breakfast Burrito
- Bruschetta
- Caesar Salad
- Cannoli
- Caprese Salad
- Carrot Cake
- Ceviche
- Cheese Plate
- Cheesecake
- Chicken Curry
- Chicken Quesadilla

Dataset Split:

- Training set: 80% of the filtered dataset
- Validation set: 20% of the filtered dataset
- Test set: 2000 images from the filtered dataset

Transformations Applied:

- Resize to 256x256 pixels

- Center crop to 224x224 pixels
- Normalize using mean [0.485, 0.456, 0.406] and std [0.229, 0.224, 0.225]

1.2 2. Model Architecture

1.2.1 Initial Model Architecture

Base Model: ResNet-18 (pre-trained on ImageNet)

Modifications:

- Replaced the last fully connected layer to match the number of classes (20)
- All layers except the final fully connected layer were frozen during the initial training phase.

Optimizer: Adam

Loss Function: Cross-Entropy Loss

1.2.2 Post-Tuning Model Architecture

Base Model: ResNet-18 (pre-trained on ImageNet)

Modifications:

- Replaced the last fully connected layer to match the number of classes (20)
- All hidden layers were **unfrozen** during the training phase.

Optimizer: SGD with a momentum of 0.5

Loss Function: Cross-Entropy Loss

1.3 3. Training Process

Along with hyperparameter tuning, the training process involved experimenting with optimizers (Adam and SGD) and with unfreezing varying number of hidden layers in the ResNet18 model.

1.3.1 Hyperparameters

- Learning Rates: [0.01, 0.008, 0.001]
- Batch Sizes: [32, 64]
- Epochs: [3, 5, 10]

1.3.2 Training Loop

For each combination of hyperparameters, optimizer and model layer variance, the model was trained on the training dataset and evaluated on the validation dataset.

If the model achieved a validation loss less than the previous best, it was stored as the best model to be used later for further tuning and finally testing on unseen data.

Training Loop Visualization:

```
# Pseudocode for the training loop
for lr in learning_rates, batch_size in batch_sizes, epochs in epochs_list, optimizer in optimizers:
    # Train the model
```

```

train_losses, val_losses, train_accuracies, val_accuracies = train_model(...)
# Save the best model
if val_loss < best_val_loss:
    best_val_loss = val_loss
    torch.save(model.state_dict(), 'best_model.pth')

```

1.3.3 Training and Validation Losses

Training and Validation Losses Plot:

Training and Validation Accuracies Plot:

1.4 4. Evaluation Results

1.4.1 Best Model Performance

The best model was selected based on the lowest validation loss and then evaluated on the test set.

Evaluation Metrics on Test Set:

- **Accuracy:** 0.8005
- **Precision:** 0.8056
- **Recall:** 0.8005
- **F1-Score:** 0.7979

Classification Report:

	precision	recall	f1-score	support
Apple Pie	0.82	0.80	0.81	100
Baby Back Ribs	0.78	0.79	0.78	100
Baklava	0.80	0.81	0.80	100
...				
Chicken Curry	0.79	0.80	0.79	100
Chicken Quesadilla	0.81	0.80	0.81	100
Weighted Avg	0.8056	0.8005	0.7979	2000

1.4.2 Predictions Visualization

Correct Predictions:

Misclassifications:

1.5 5. Insights Gained

1. Model Performance:

- The ResNet-18 model with transfer learning achieved good performance on the Food-101 dataset with the selected 20 classes.
- Fine-tuning the entire network by unfreezing all the layers significantly improved accuracy and other metrics.

2. Hyperparameter Tuning:

- Increasing epochs for the initial model resulted in high training accuracy and low training loss, but stagnant validation accuracy and loss. This indicates that the model was overfitting and memorizing the training data.
- SGD made a significant difference in performance compared to Adam, with Adam performing poorly.

a

3. Visualization:

- Visualizing the predictions and misclassifications helped in understanding where the model struggled. Most misclassifications occurred between visually similar classes.

4. Next Steps:

- Further fine-tuning the model by exploring additional hyperparameters and using data augmentation techniques could help improve the model's performance further.
- Using a higher-end ResNet model.

1.6 6. Conclusion

The project demonstrated the effectiveness of transfer learning using a pre-trained ResNet-18 model for image classification on the Food-101 dataset. Through hyperparameter tuning and careful evaluation, a model with good accuracy and F1-score was obtained.

The insights gained from visualizing predictions provided a deeper understanding of the model's strengths and weaknesses, paving the way for further improvements.

[]: