BBM 418 - Computer Vision Laboratory

Assignment 2 Report

Gökdeniz Şimşek
2210356067

Introduction

In this assignment, convolutional neural networks (CNNs) were explored from scratch and using transfer learning techniques. Experiments were conducted with different architectures, hyperparameters (learning rate, batch size, dropout), and training strategies. Models were evaluated based on training/validation accuracy, test accuracy, and confusion matrices.

Part 1 – CNNs from Scratch

1. Model Architectures

- BasicCNN: This model includes five convolutional layers with batch normalization and ReLU activation functions. Max pooling layers are used to reduce spatial dimensions progressively. After the convolutional blocks, the output is flattened and passed through a fully connected layer with a softmax output for classification.
- ResidualCNN: This model enhances the basic architecture by including residual connections that help with gradient flow in deep networks. Each residual block consists of two convolutional layers and a skip connection. When input and output dimensions do not match, a 1x1 convolution is applied in the skip path.

2. Training Settings

We experimented with different hyperparameter combinations:

- Batch sizes: 32 and 64
- Learning rates: 0.001, 0.0005, and 0.0001
- Loss function: CrossEntropyLoss
- Optimizer: Adam optimizer was chosen for its adaptive learning rate capabilities.

Data augmentation was applied only to the training set to increase the diversity of the input images and reduce overfitting. The transformations included:

- Resizing the images to 256 pixels on the shorter side
- Random cropping to 224x224 with padding to allow slight shifts
- Random horizontal flipping to simulate mirrored images
- Color jittering to slightly alter brightness and contrast, making the model robust to lighting variations

These augmentations help simulate natural variations that could appear in real-world food images, thereby improving the model's generalization ability. For validation and test datasets, only resizing to 224x224 and tensor conversion were applied to ensure consistent evaluation conditions without introducing noise.

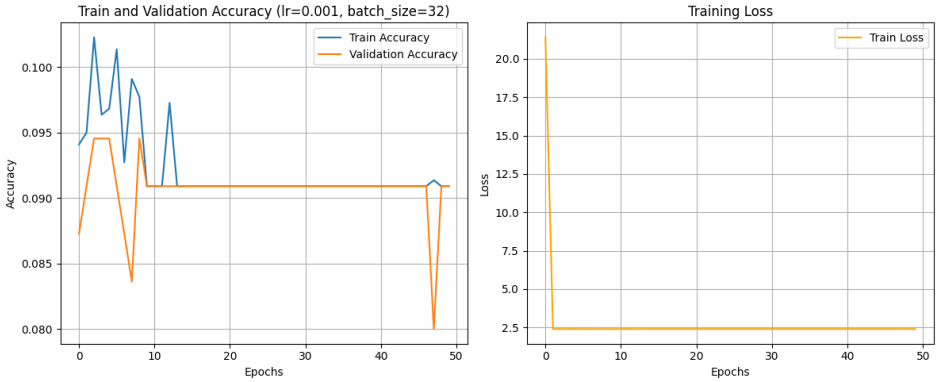
3. Dropout Experiments

Dropout layers were added to the classifier section of the models with dropout probabilities of 0.3 and 0.5. These configurations were trained and evaluated. Results showed that adding dropout led to a decrease in validation and test performance. This was likely because the data augmentation techniques (such as random cropping, horizontal flipping, and color jittering) already introduced sufficient variability to the training data, effectively reducing overfitting. When dropout was applied on top of these augmentations, the model began to underfit, failing to capture important patterns in the data. Therefore, dropout was excluded from the final model configuration.

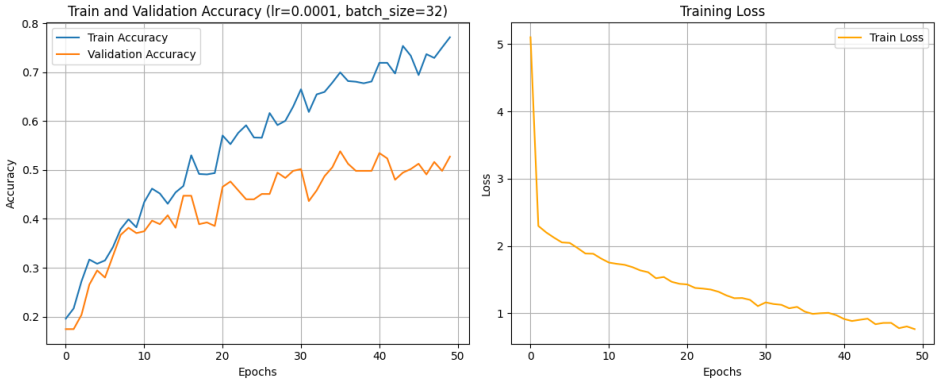
4. Accuracy and Loss Plots

For each combination of batch size and learning rate; training loss, training accuracy, and validation accuracy were plotted against epochs. These plots were used to visually identify overfitting and convergence behavior. Below are the graphs and accuracy values for each different combination:

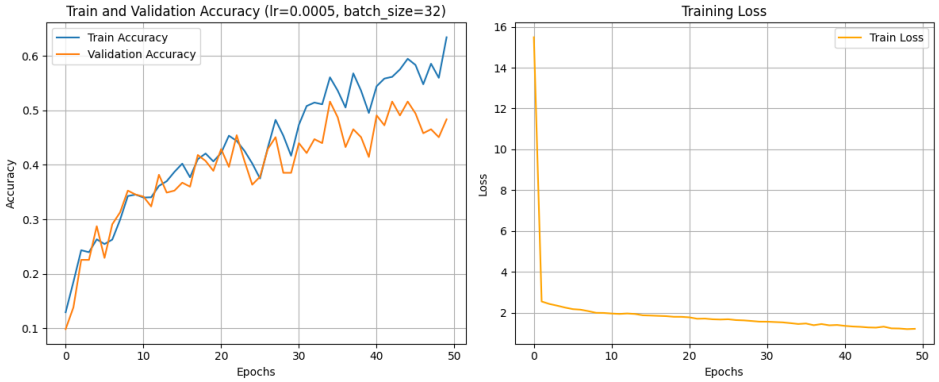
4.1 - Basic CNN:



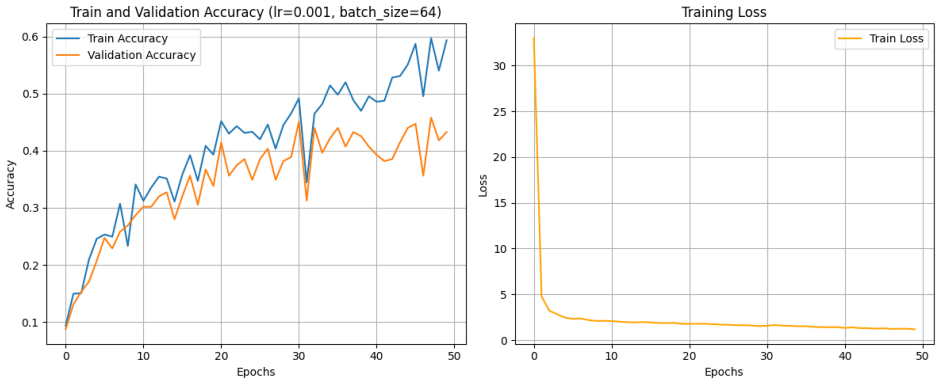
For Basic CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.001 learning rate are: Loss: 2.3980, Train Acc: 0.0909, Val Acc: 0.0909.



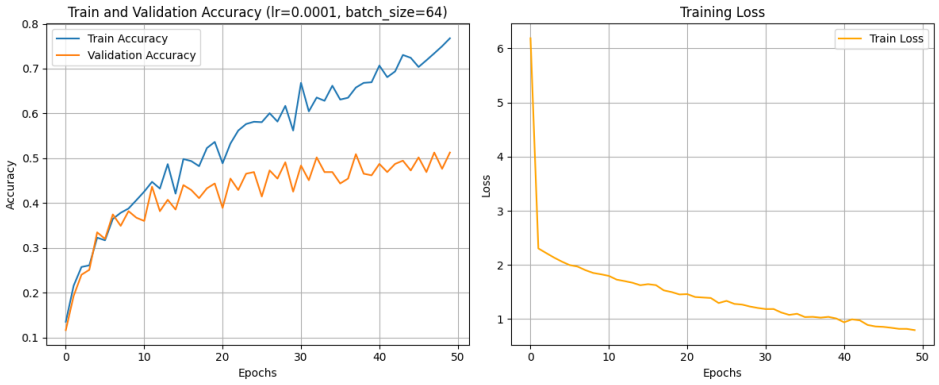
For Basic CNN, the values obtained at the end of 50 epochs with 32 batch size and 0.0001 learning rate are: Loss: 0.7653, Train Acc: 0.7714, Val Acc: 0.5273.
These are the values with the best validation accuracy among the basic CNN models.



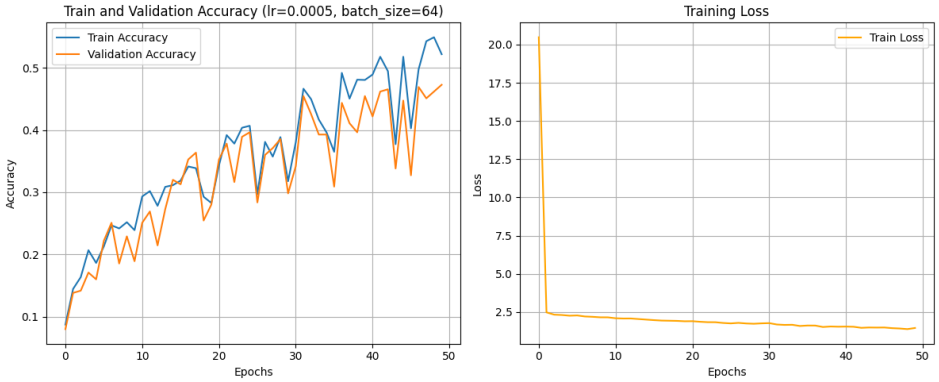
For Basic CNN, the values obtained at the end of 50 epochs with 32 batch size and 0.0005 learning rate are: Loss: 1.2092, Train Acc: 0.6345, Val Acc: 0.4836.



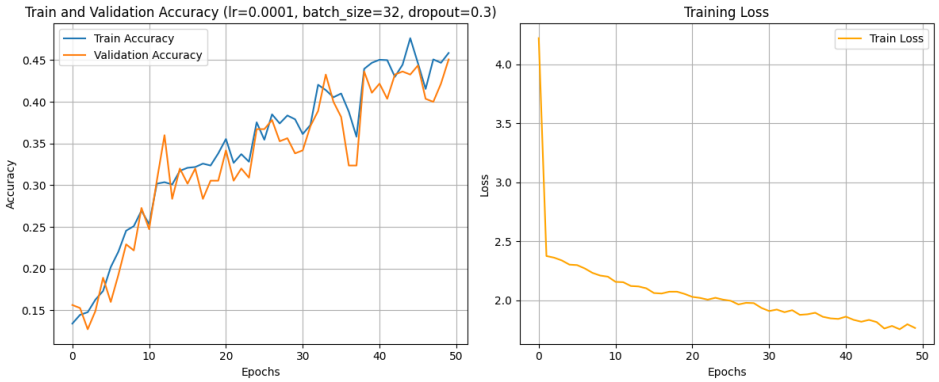
For Basic CNN, the values obtained at the end of 50 epochs with 64 batch size and 0.001 learning rate are: Loss: 1.1841, Train Acc: 0.5936, Val Acc: 0.4327.



For Basic CNN, the values obtained at the end of 50 epochs with 64 batch size and 0.0001 learning rate are: Loss: 0.7961, Train Acc: 0.7677, Val Acc: 0.5127.

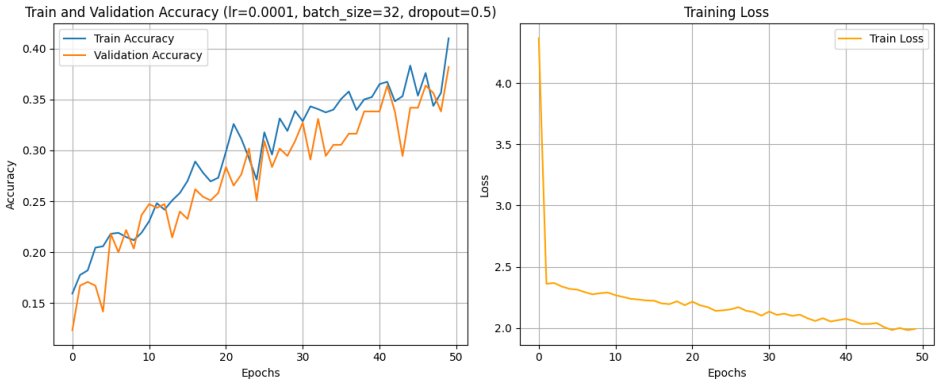


For Basic CNN, the values obtained at the end of 50 epochs with 64 batch size and 0.0005 learning rate are: Loss: 1.4580, Train Acc: 0.5218, Val Acc: 0.4727.



For Basic CNN, the values obtained at the end of 50 epochs with 32 batch size, 0.0001 learning rate and 0.3 dropout value are:

Loss: 1.7671, Train Acc: 0.4586, Val Acc: 0.4509.

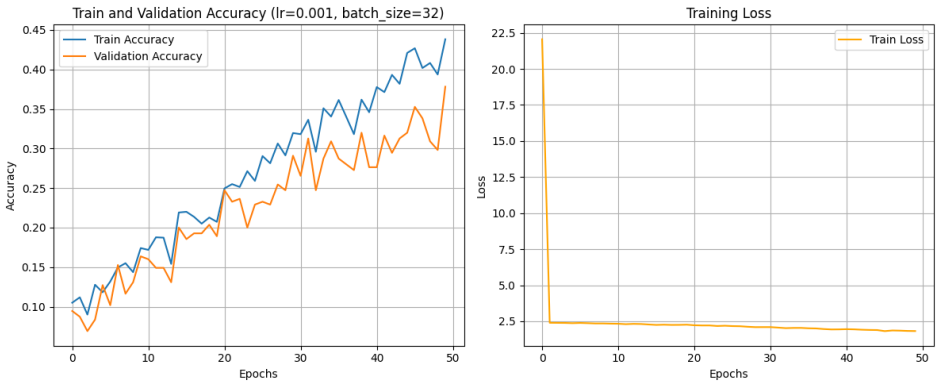


For Basic CNN, the values obtained at the end of 50 epochs with 32 batch size, 0.0001 learning rate and 0.5 dropout value are:

Loss: 1.9928, Train Acc: 0.4100, Val Acc: 0.3818.

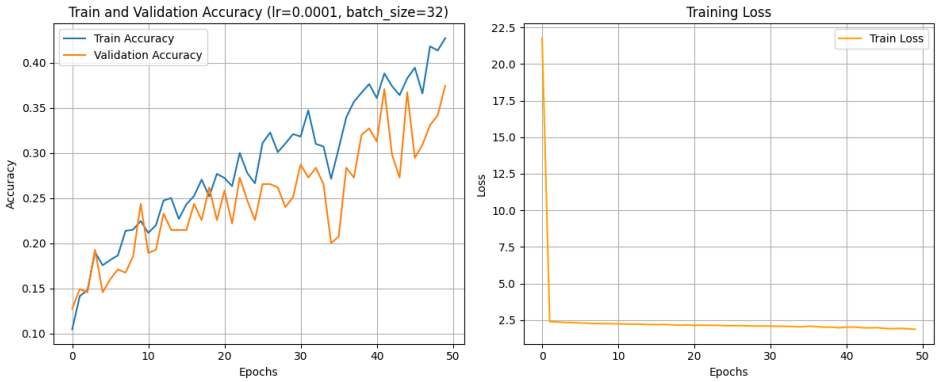
As a result, it was determined that the model with no dropout, a batch size of 32 and a learning rate of 0.0001 had the best validation accuracy for the Basic CNN model. The test accuracy obtained as a result of the test conducted with this model was 0.4655.

4.2 - Residual CNN:

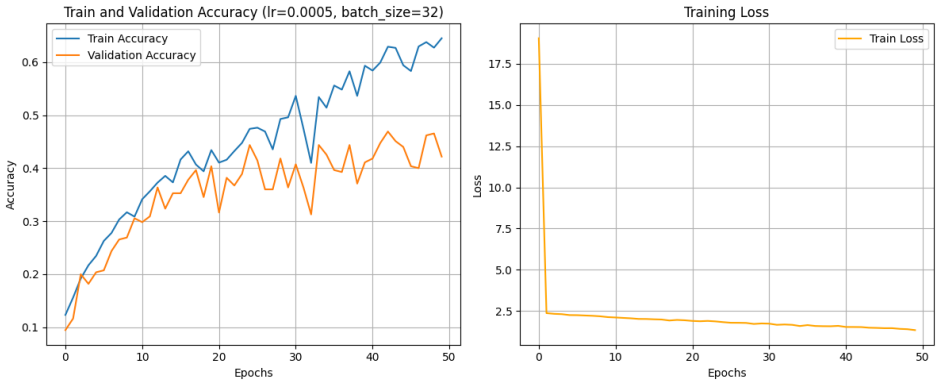


For Residual CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.001 learning rate are:

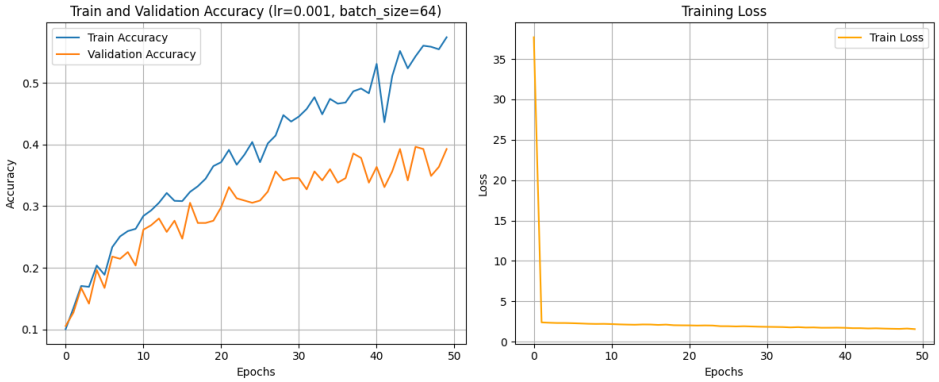
Loss: 1.8247, Train Acc: 0.4382, Val Acc: 0.3782.



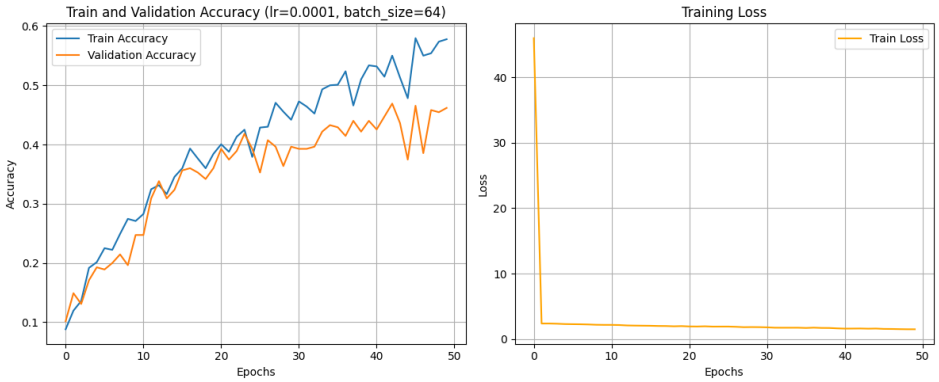
For Residual CNN, the values obtained at the end of 50 epochs with 32 batch size and 0.0001 learning rate are: Loss: 1.8787, Train Acc: 0.4273, Val Acc: 0.3745.



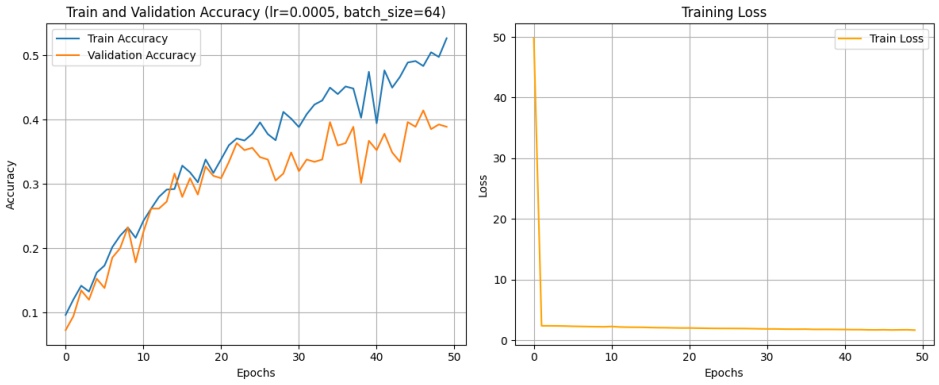
For Residual CNN, the values obtained at the end of 50 epochs with 32 batch size and 0.0005 learning rate are: Loss: 1.3467, Train Acc: 0.6450, Val Acc: 0.4218.



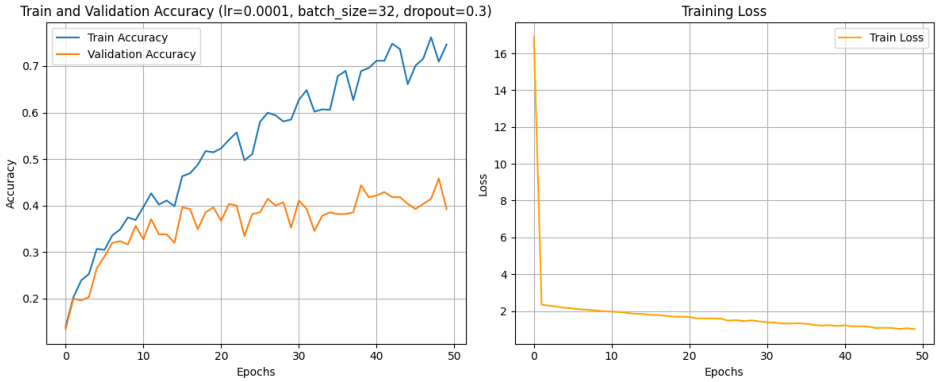
For Residual CNN, the values obtained at the end of 50 epochs with 64 batch size and 0.001 learning rate are: Loss: 1.5561, Train Acc: 0.5741, Val Acc: 0.3927.



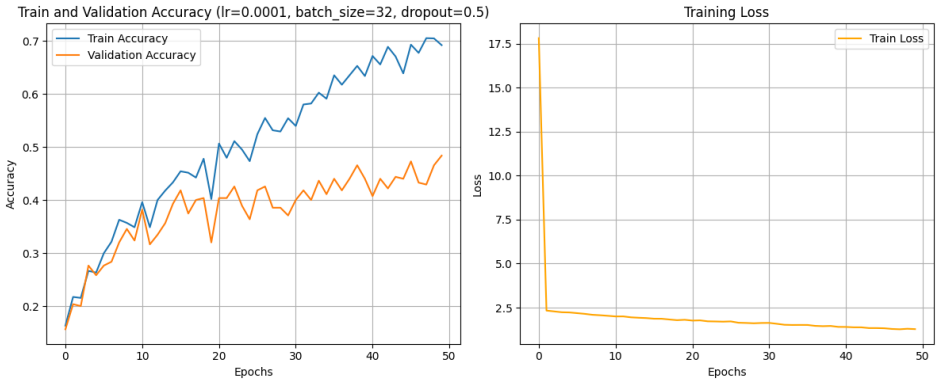
For Residual CNN, the values obtained at the end of 50 epochs with 64 batch size and 0.0001 learning rate are: Loss: 1.5118, Train Acc: 0.5777, Val Acc: 0.4618.



For Residual CNN, the values obtained at the end of 50 epochs with 64 batch size and 0.0005 learning rate are: Loss: 1.6873, Train Acc: 0.5268, Val Acc: 0.3891.



For Residual CNN, the values obtained at the end of 50 epochs with 32 batch size, 0.0001 learning rate and 0.3 dropout value are:
Loss: 1.0143, Train Acc: 0.7464, Val Acc: 0.3927.



For Residual CNN, the values obtained at the end of 50 epochs with 32 batch size, 0.0001 learning rate and 0.5 dropout value are:

Loss: 1.2818, Train Acc: 0.6918, Val Acc: 0.4836.

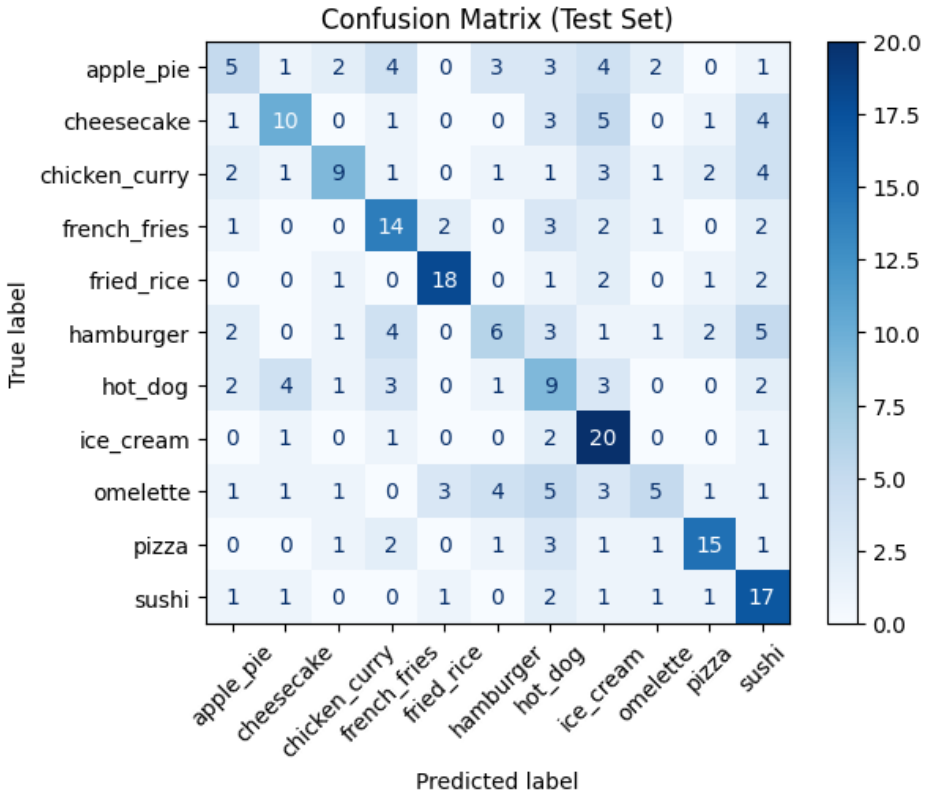
These are the values with the best validation accuracy among the basic CNN models.

As a result, it was determined that the model with 0.5 dropout, a batch size of 64 and a learning rate of 0.0001 had the best validation accuracy for the Residual CNN model. The test accuracy obtained as a result of the test conducted with this model was 0.3818.

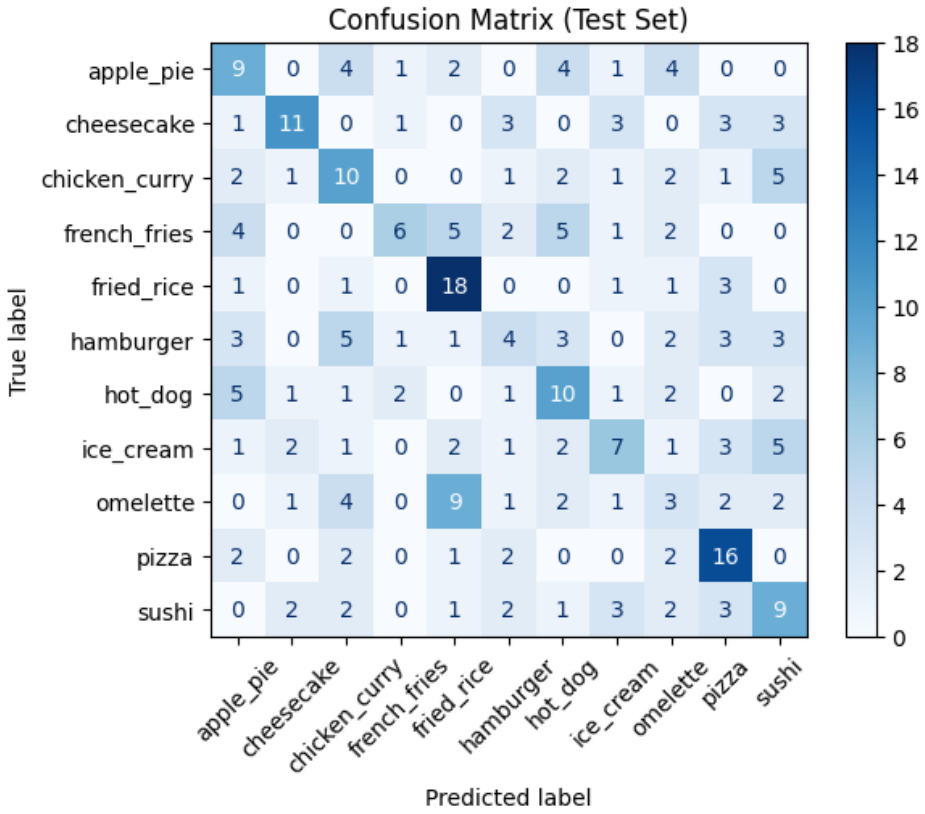
5. Confusion Matrix

A confusion matrix was generated using the test dataset for the best model. This matrix helped identify which food classes were commonly misclassified. Most errors occurred between visually similar classes such as fried rice and fried noodles.

5.1 - Basic CNN Confusion Matrix:



5.2 - Residual CNN Confusion Matrix:



Part 2 – Transfer Learning

1. Pretrained Model: MobileNetV2

MobileNetV2 was selected for transfer learning due to its lightweight architecture and high performance. The model was loaded with pretrained ImageNet weights. The final classification layer was replaced with a new linear layer that outputs 11 logits for the food categories.

Initially, all layers in the feature extractor were frozen. Only the classifier layer was trained to adapt the pretrained features to our task.

2. Fine-Tuning Strategy

After training only the classifier for 50 epochs, fine-tuning was performed by unfreezing the last two convolutional blocks (features[17] and features[18]). A new optimizer with a reduced learning rate (0.00001) was used to avoid large weight updates. However, due to the small size of the training dataset, fine-tuning caused overfitting. Validation and test accuracy decreased, indicating that the pretrained weights were already well-optimized for generic features and fine-tuning disrupted them.

3. Accuracy Results

For MobileNetV2 CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.0001 learning rate are:

Loss: 0.0937, Train Acc: 0.9945, Val Acc: 0.8836. Test Accuracy for before fine-tuning: 0.7855

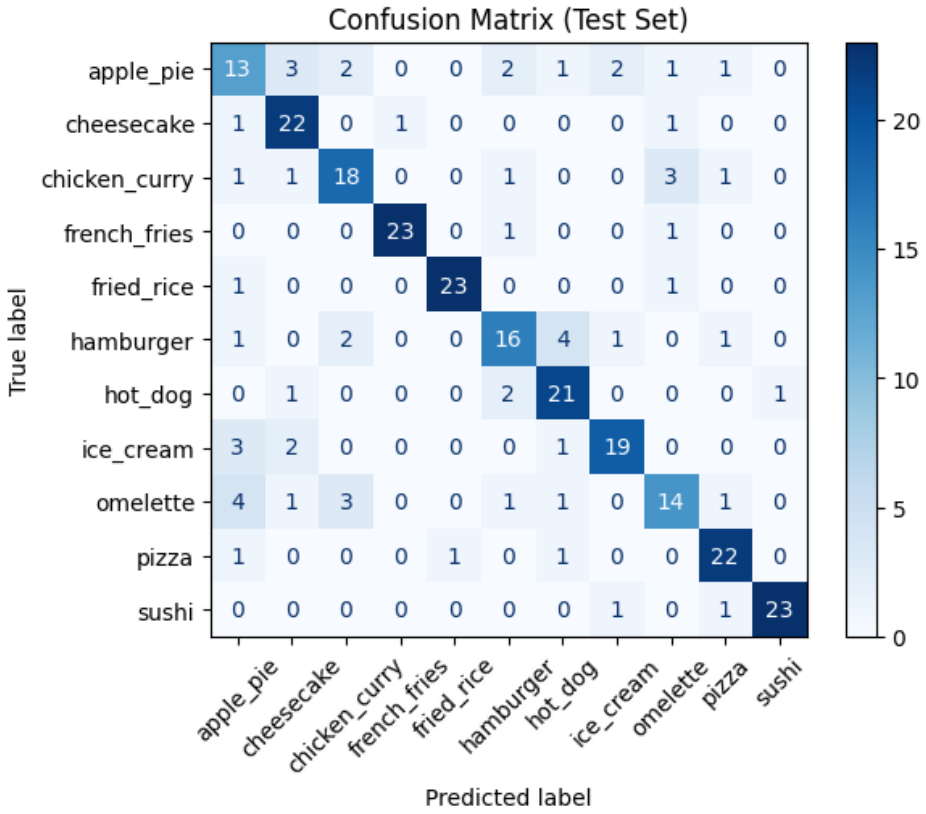
For MobileNetV2 CNN Model after fine-tuning, the values obtained at the end of 10 epochs with 32 batch size and 0.00001 learning rate are:

Loss: 0.0612, Train Acc: 0.9964, Val Acc: 0.8909. Test Accuracy after fine-tuning: 0.7782

Although the validation accuracy value is higher after fine-tuning; test accuracy decreased, indicating that the pretrained weights were already well-optimized for generic features and fine-tuning disrupted them.

4. Confusion Matrix

A confusion matrix for the classifier-only MobileNetV2 model was plotted. Like the scratch models, errors occurred in categories with high visual similarity. However, fewer extreme misclassifications were observed compared to scratch models. Predictions were largely correct.



Comparison: Part 1 vs Part 2

In this section, we compare the best-performing model from the scratch-designed CNNs with the MobileNetV2 transfer learning model. The best accuracy from Part 1 was obtained using BasicCNN with batch size of 32, learning rate of 0.0001, and without dropout. This configuration achieved a test accuracy of 46.55%, outperforming deeper networks like ResidualCNN under the same conditions. In contrast, MobileNetV2 with only the classifier layer trained (before fine-tuning) yielded a significantly better performance with a test accuracy of 78.55%.

This result clearly demonstrates the power of transfer learning, especially when using pretrained models on small datasets. While BasicCNN had fewer parameters and required more epochs and tuning to reach moderate accuracy, MobileNetV2 achieved superior performance with less effort and computation time.

Conclusion

This assignment provided valuable insights into CNN design, hyperparameter optimization, and transfer learning. The experiments demonstrated that:

- Dropout can harm performance when the dataset is small.
- Transfer learning can outperform scratch models in terms of efficiency.
- Fine-tuning pretrained models requires careful regularization and lower learning rates to prevent overfitting.

Selecting the best model depends on the trade-off between computational cost and accuracy. The best performing model developed from scratch for this dataset was BasicCNN without dropout, with batchsize of 32 and learning rate of 0.0001.