1 Introduction

This report outlines the efforts made to classify a food image dataset into 27 categories by using a nueral network model. With a dataset comprising 6,281 training images, the objective was to build a robust classification model. This report details the utilization of generic ML techniques and the domain-specific insights gained while handling the dataset.

2 Domain-Specific Insights and Understanding

Exploratory Data Analysis (EDA) was a fundamental step in understanding the nuances of the image dataset, by addressing class imbalances and visualizing sample images for each category.

2.1 Class Imbalance Assessment

An initial exploration was counting the number of images for each category. In Figure 1(a), the result revealed varying frequencies across the 27 food categories. Several classes such as 4, 5, 9, 10, 11, 12, 15, and 21 showed much less frequencies than other classes, potentially impacting model performance during training and evaluation. Strategies such as weighted loss function were considered to address this imbalance and thus prevent biases and ensure equal representation during training.

2.2 Sample Images Visualization

For more comprehensive understanding, sample images from each class were visualized. This step offered insights into the diversity and complexity of images within each food category and provided valuable context for subsequent preprocessing steps such as data augmentation.

3 Utilization of ML Techniques

In the process of developing an effective food image classification model, several machine learning techniques were employed to enhance model performance and robustness. Testing with variety of models, the primary challenge was the divergence between train accuracy and validation accuracy. Thus the main point was to enhance the model's ability to generalize and reduce overfitting.

3.1 Shallow Model Architecture

A shallow version of the ResNet architecture was used, consisting of nine layers in total. This architecture comprised two layers of convolution, a residual block, two additional convolutional layers, another residual block, and a fully connected layer. This model will be referred to as ResNet9 from here. ResNet9 was chosen among several models, including Simple CNN, ResNet18, ResNet34, and ResNet50, after a comparative analysis. The Simple CNN was insufficiently deep to learn all the features of the training data, while ResNet34 and ResNet50, being more complex, led to longer training time and overfitting. ResNet18 and ResNet9 yielded similar results. However, due to its superior efficiency, ResNet9 was chosen as the final model.

3.2 Train-Validation Splitting

The dataset was partitioned into training and validation sets, enabling the model to train on one portion and validate its performance on another. After experimenting various splits from 8:2 to 9.5:0.5, the 9:1 split was chosen. This split yielded favorable accuracy results while maintaining consistency in validation accuracies and minimizing disparity between validation and test set accuracies due to the smaller validation set size.

3.3 Image Augmentation and Normalization

After analyzing food images across various categories as mentioned in Section 2.2, it became apparent that these images were captured under diverse lighting conditions and perspectives. This diversity extended to the test set, indicating variations in real-world situations. Additionally, the dataset's relatively small size compared to larger standard datasets like CIFAR-10, posed a challenge.

Thus, various augmentation techniques such as geometric transformations, random augmentations, and CutMix were employed. Five distinct datasets were created by duplicating the training dataset, each subjected to diverse augmentation methods: AutoAugment, TrivialAugmentWide, AugMix, a custom composition of RandomPerspective and RandomResizedCrop, and a custom random transformation involving simple augmentations like rotation, flipping, Gaussian blur, and noise. This approach aimed to enhance the model's ability to generalize with limited data. Furthermore, normalization was applied across all datasets (train, test, and validation sets) for each channel to ensure consistency and stability in the dataset. Figure 1(b) displays the outcomes of each augmentation method, with the original normalized image in the initial frame.

After experimenting with several augmentation methods, the random augmentations mentioned above exhibited better regularization effects compared to others, although subtly. However, the application of CutMix to those rigorous augmentations occasionally led to instances of underfitting. Consequently, the implementation of the two-phase training method described below in Section 3.7 was introduced.

3.4 Class Weights in Loss Function

To address class imbalances within the dataset, adjustments were made using class weights within the loss function, employing the cross-entropy loss. This strategy aimed not only the imbalance in classes with smaller frequencies but also for instances where certain classes exhibited lower accuracy when trying out variety of models.

3.5 Optimization Strategies

Stochastic Gradient Descent (SGD) with momentum, coupled with high weight decay(0.01 in the first training phase and 0.05 in the second) for regularization purposes, was used during the training process. This strategy aimed to optimize the model's weights while preventing overfitting. The experiment revealed that higher weight decay slowed the increase in training accuracy, reducing disparity between validation and training accuracy which indicates less overfitting.

3.6 Learning Rate Scheduling

The learning rate was adjusted using the cosine annealing warm restarts scheduler, helping the fine-tuning of the learning rate across extended epochs. This scheduler gradually reduced the learning rate, which improved performance during prolonged epochs. Additionally, its periodic sharp increase of learning rate prevented the model from getting trapped in local minima, contributing to its enhanced performance, according to various experiments.

3.7 Training Strategy

A two-stage training strategy was employed. Initially, the model underwent 80 epochs of training without CutMix augmentation, utilizing all five augmented datasets aforementioned in Section 3.3 alongside the original training dataset. Upon reaching a saturation point in validation loss and accuracy after 80 epochs, the second phase ensued. In this phase, training continued for an additional 40 to 80 epochs, exclusively employing CutMix on lightly augmented images—specifically, datasets subjected to geometric and simple augmentations, along with the original dataset—until another point of saturation in validation loss and accuracy was reached.

3.8 Model Ensemble

After applying the techniques mentioned earlier, the validation accuracies reached up to 79%. However, a significant disparity of up to 9% were shown between the validation accuracy and the test accuracy (as calculated by Kaggle) each time. This variance might be attributed to the small size of the validation set. Consequently, an ensemble approach employing majority voting was introduced

Three different networks, employing the same model but utilizing distinct validation sets and training policies, produced validation accuracies of 76.825%, 78.856%, and 78.378%—the top three validation accuracies observed. By merging predictions through majority voting, the ensemble model was created. Initially, the test accuracy of each individual network was lower than its respective validation accuracy. However,

after ensembling, the test accuracy nearly aligned with the validation accuracy.

Prior to ensemble integration, the best test accuracy stood at 77.037%. Following the ensemble method application, the aggregated predictions yielded a notable improvement, elevating the test accuracy to 78.148%.

These techniques collectively contributed to enhancing the model's ability to classify food images accurately and efficiently.

4 Results, Analysis, and Conclusion

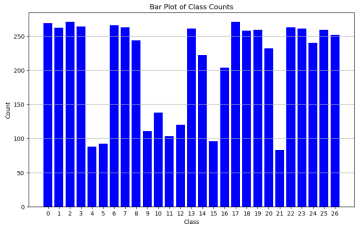
The training progression of the initial phase can be observed in Figure 1(d), while the complete training evolution is illustrated in Figure 1(e). Notably, there's a marked spike in the training loss after the implementation of CutMix. Interestingly, despite this surge in loss, a modest rise in validation accuracy can be observed. Furthermore, in comparison to an earlier occasion with less weight decay and no CutMix in Figure 1(f), the difference between the training and validation loss is smaller. It's worth noting that the inclusion of a learning rate scheduler has helped mitigate this discrepancy; without it, the difference might have been more pronounced.

Examining the class-wise accuracy depicted in Figure 1(c), a consistent smoothness in accuracy is evident across most classes. However, the accuracy for the least frequent class (Class 4) remains notably lower than that of other classes. Efforts to address this discrepancy by assigning more weight to this class didn't yield a significantly improved result.

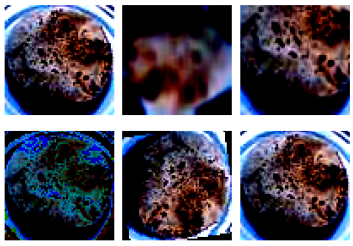
The primary focus of model enhancement revolved around augmenting its generalization capabilities. Augmentation strategies and regularization techniques were primarily employed to mitigate overfitting and enhance the model's adaptability to diverse real-world scenarios.

The marginal improvement in test accuracy, below the 80% mark, appears to be the consequence of constrained dataset size. The evident limitation in achieving higher accuracy levels emphasizes the pivotal role of larger datasets in enhancing model performance and robustness. Although, there is room for improvement. Refining the model architecture to accommodate class imbalances more effectively or even fine-tuning the training process further could be ways to substantially elevate the model's accuracy and performance.

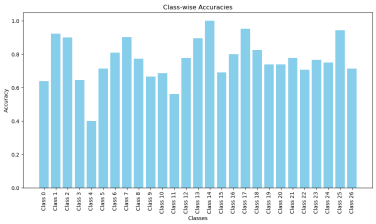
5 Appendices



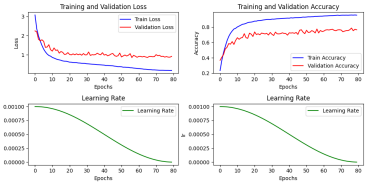
(a) Bar Plot of Class Counts.



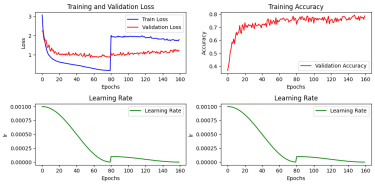
(b) Examples of Augmented Images.



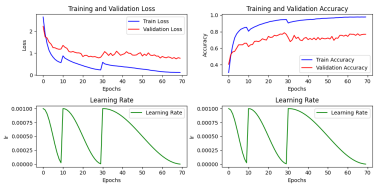
(c) Class-wise Accuracy



(d) Training Progress of Phase 1.



(e) Whole Training Progress.



(f) Training Progress with Less Regularization.

Figure 1: Figures.