Slide 1 – Title Slide

Hello everyone, and thank you for being here. My name is Zahra Dastfal, and this project was completed by our team: Umesh Chander Reddy, Vinod Kumar Reddy, and myself. Our instructor is Dr. I-Hua Tsai. Today, we present our Machine Learning project titled 'Food-101 Image Classification using CNN and EfficientNetB0.' This project focuses on building deep learning models to classify 101 different categories of food images.

Slide 2 – Agenda

In this presentation, we will cover the following: Introduction, Problem Statement, Dataset description, Methodology, Baseline CNN results, EfficientNet initial results, Fine-tuning results, Confusion Matrix, Model comparison, Conclusion, and Future Work. This structure helps guide our analysis step-by-step.

Slide 3 – Introduction

Food classification is an important computer vision task with many real-world applications. These applications include dietary tracking, calorie estimation, restaurant menu automation, and healthcare-related food logging systems. However, food images are often challenging to classify because food items vary widely depending on lighting, presentation, angle, and plating styles. Deep learning is especially effective for handling such variability.

Slide 4 – Problem Statement

The Food-101 dataset contains 101 different food categories, many of which look visually similar. Images vary significantly in brightness, presentation, and background. The problem we aim to solve is: How can we classify 101 food categories accurately using deep learning models? Our goal is to compare a simple CNN model with EfficientNetB0, both in initial and fine-tuned forms.

Slide 5 – Dataset

We use the Food-101 dataset, which contains 101,000 images in total. Each of the 101 categories contains 1,000 images. The dataset is split into 75,750 training images and 25,250 testing images. It is a highly challenging dataset because many food types look similar to each other. This makes it an excellent benchmark for testing deep learning models.

Slide 6 – Methodology

Our methodology includes five major steps. First, we load the dataset and perform preprocessing, such as resizing and normalizing images. Second, we train a baseline CNN model from scratch. Third, we apply transfer learning using EfficientNetB0. Fourth, we fine-tune the model by unfreezing the top layers. Finally, we evaluate all models using accuracy, loss curves, confusion matrix, and F1-scores to compare performance.

Slide 7 – Baseline CNN

The baseline CNN consists of convolution layers, max-pooling layers, and dense layers at the end. It is a simple model used to provide a reference point. Since the model starts from scratch with no pretrained knowledge, we expect its performance to be limited.

Slide 8 – Baseline Accuracy Curve

Here we see the baseline CNN accuracy curve. The model shows very low accuracy, with little improvement across epochs. This indicates the model is underfitting and struggling to learn meaningful food features. Because the dataset is large and complex, the model lacks the capacity to capture detailed patterns.

Slide 9 – Baseline Loss Curve

The baseline loss curve shows high loss values with inconsistent reduction. Validation loss does not improve significantly, reinforcing that the CNN model is not adequate for Food-101 classification.

Slide 10 – Baseline Analysis

In summary, the baseline CNN fails to learn complex representations required for 101 food categories. It underfits heavily, and its low accuracy motivates the use of transfer learning models like EfficientNetB0.

Slide 11 – EfficientNet Initial Training

EfficientNetB0 is a state-of-the-art convolutional neural network that uses pretrained ImageNet weights. In initial training, we freeze the base layers so only the classification head trains. This approach quickly improves accuracy compared to the baseline CNN because the model already knows useful image features.

Slide 12 – Initial Accuracy Curve

The accuracy curve for initial training shows rapid improvement. Validation accuracy is stable, indicating good generalization. This demonstrates the benefits of transfer learning, even without fine-tuning the deeper layers.

Slide 13 – Initial Loss Curve

Validation loss decreases smoothly during initial training. The model converges quickly because the pretrained layers already capture rich visual features.

Slide 14 – Initial Training Analysis

In this stage, EfficientNetB0 significantly outperforms the baseline CNN. The model generalizes well, and its performance shows that pretrained networks offer major advantages on complex datasets.

Slide 15 – Fine-Tuning Overview

In fine-tuning, we unfreeze the top layers of EfficientNetB0 and train with a very small learning rate. This allows the model to adapt its higher-level features specifically to food images. Fine-tuning increases accuracy even further.

Slide 16 – Fine-Tuning Accuracy Curve

The fine-tuning accuracy curve shows clear improvement over initial training. The model continues to learn useful representations for food classification.

Slide 17 – Fine-Tuning Loss Curve

Fine-tuning reduces validation loss further, showing improved convergence and better fit to the dataset.

Slide 18 – Fine-Tuning Analysis

Fine-Tuned EfficientNetB0 is our best-performing model. It is more stable, accurate, and robust than both the baseline and initial EfficientNet models. Fine-tuning demonstrates the benefit of adjusting pretrained layers to domain-specific data.

Slide 19 – Confusion Matrix

The confusion matrix shows many correct predictions along the diagonal. Although some misclassifications occur, this is expected for a dataset with 101 classes. Overall, the model performs well with consistent prediction patterns.

Slide 20 – Model Comparison

Our comparison shows the following: the baseline CNN performs poorly, the initial EfficientNet achieves strong accuracy, and the fine-tuned EfficientNet gives the best results with around 71% accuracy. Transfer learning and fine-tuning clearly offer major advantages.

Slide 21 – Conclusion

We conclude that EfficientNetB0 significantly outperforms a baseline CNN for food classification. Fine-tuning improves performance even more. Our final model achieves strong accuracy on a large, complex dataset.

Slide 22 – Future Work

Future improvements may include training for more epochs, applying data augmentation, experimenting with larger EfficientNet versions, and performing hyperparameter tuning. Another improvement would be deploying the model as a mobile or web application.