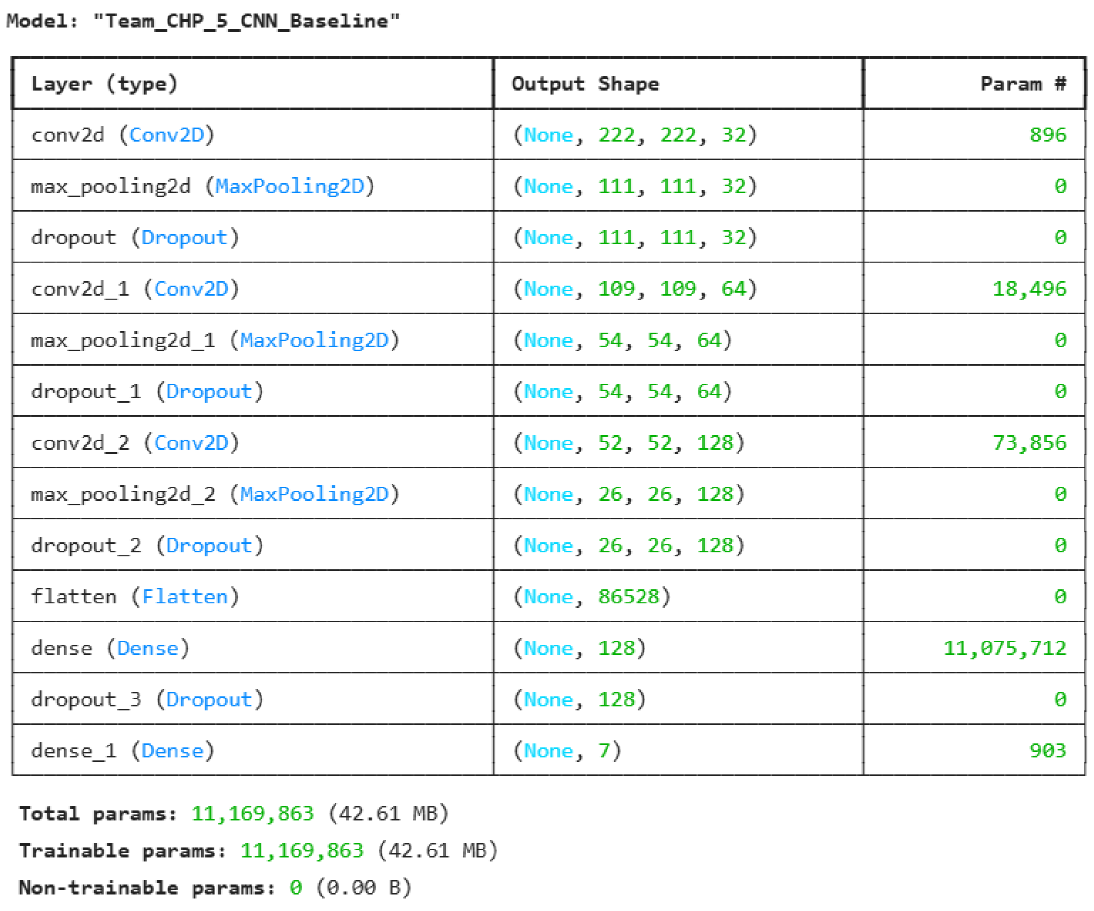
CNN Model

This part of the report focuses on how the project developed a convolutional neural network (CNN) model to classify fine-grained stages of fruit quality, including various ripeness levels for bananas and tomatoes. The model was trained using a custom image dataset divided into training, validation, and test sets. This report outlines the model design, training process, evaluation, and observations.

Model Architecture

A baseline CNN model was developed using Keras. It consists of three convolutional blocks with increasing **filter depth (32, 64, 128), ReLU** activations, and max pooling layers. Each block is followed by dropout to mitigate overfitting. The output of the convolutional layers is flattened and passed through two dense layers, the last of which uses a **softmax activation** for multi-class classification.



Model: "Team_CHP_5_CNN_Baseline"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
dropout (Dropout)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
dropout_1 (Dropout)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
dropout_2 (Dropout)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 128)	11,075,712
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 7)	903

```
Total params: 11,169,863 (42.61 MB)
Trainable params: 11,169,863 (42.61 MB)
Non-trainable params: 0 (0.00 B)
```

Figure 1: *Baseline CNN architecture used in the experiment. It includes three convolutional layers followed by pooling, dropout, and fully connected layers.*

Training and Callbacks

The model was compiled with the **Adam optimizer** and sparse categorical crossentropy loss. Early stopping and model checkpointing were employed to prevent overfitting and retain the best model. The data was normalized and resized to **224x224 pixels.**

Training Analysis

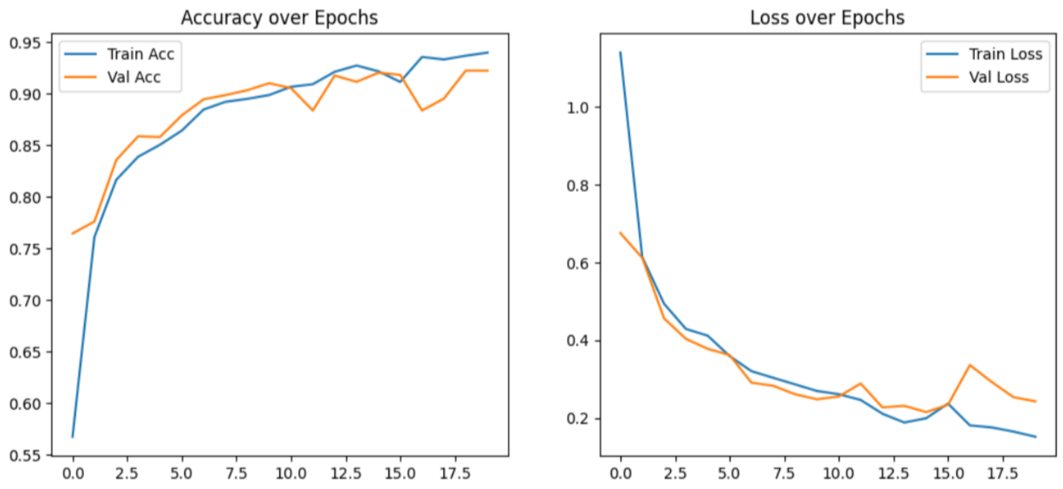
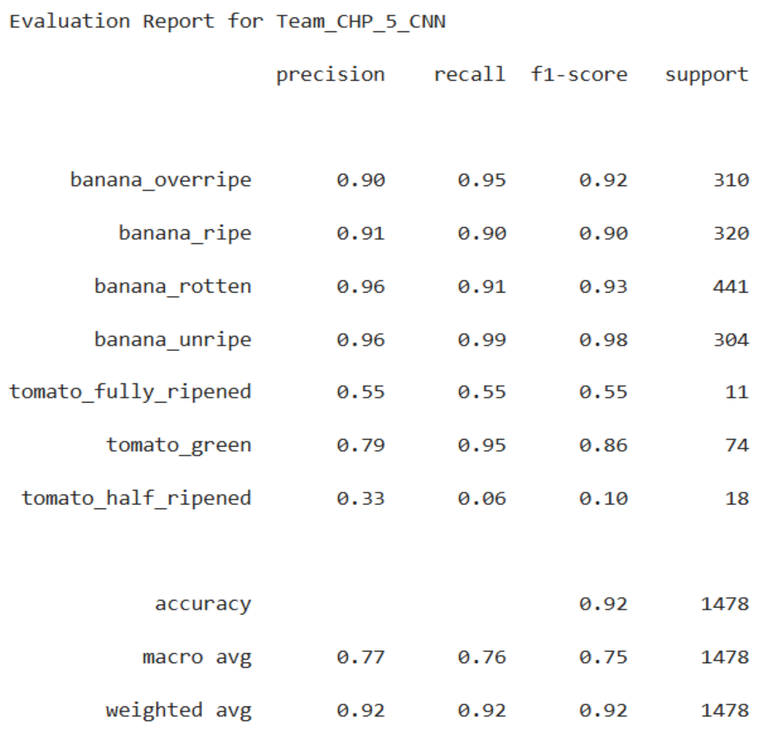


Figure 2: *Training and validation accuracy/loss curves showing convergence over **30 epochs**. The model reached a peak **validation accuracy of ~92.2%** around **epoch 20**, indicating good generalization with minimal overfitting.*

The training reached a plateau after **epoch 20**, achieving training accuracy of **94.2%** and **validation accuracy of 92.2%.** The loss curves indicate that the model is generalizing well, with no significant overfitting observed.



```
Evaluation Report for Team_CHP_5_CNN

                      precision    recall  f1-score   support


     banana_overripe       0.90      0.95      0.92       310

         banana_ripe       0.91      0.90      0.90       320

       banana_rotten       0.96      0.91      0.93       441

       banana_unripe       0.96      0.99      0.98       304

tomato_fully_ripened       0.55      0.55      0.55        11

        tomato_green       0.79      0.95      0.86        74

  tomato_half_ripened      0.33      0.06      0.10        18


            accuracy                           0.92      1478

           macro avg       0.77      0.76      0.75      1478

        weighted avg       0.92      0.92      0.92      1478
```

Figure 3: *Baseline CNN Model Evaluation Report*

Evaluation and Confusion Matrix

Final evaluation on the validation set shows strong performance for banana categories with **F1-scores >90%** across most banana classes. However, the tomato classes, especially 'tomato_half_ripened', suffer from low recall and F1-scores, likely due to data imbalance.

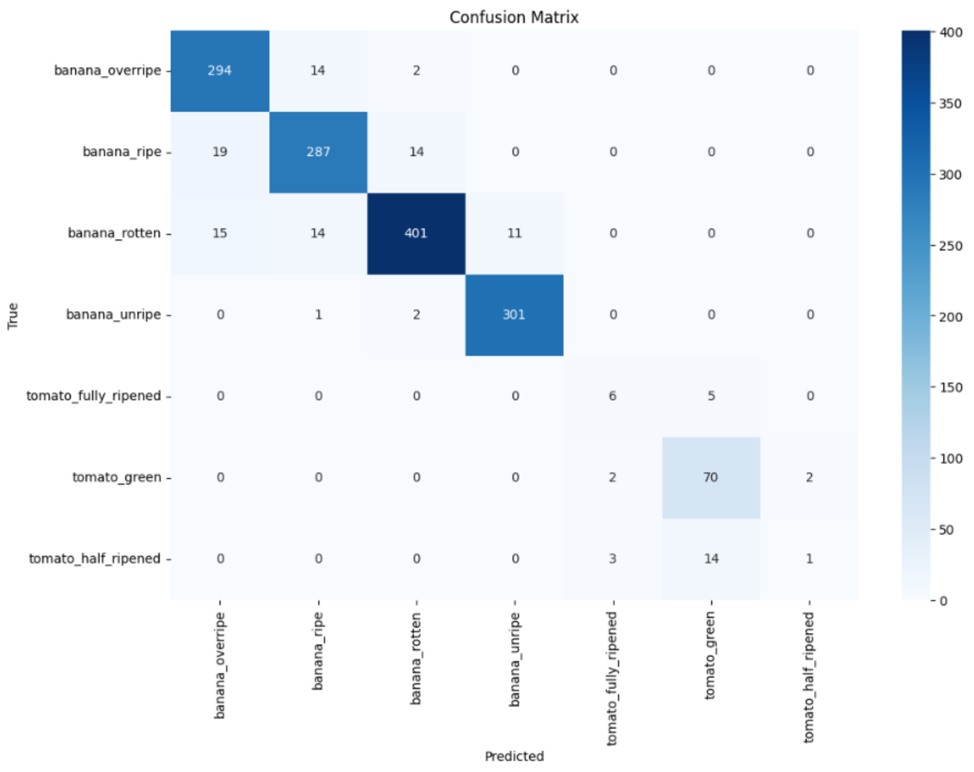


Figure 4: *Confusion matrix on validation data. Most banana classes are well classified, but tomato classes, especially 'tomato_half_ripened', show poor recall and precision.*

Test Set Performance

The model achieved an overall **test accuracy of 92%.** Predictions were exported to CSV for submission. This result reflects consistent performance from training to unseen data, affirming the robustness of the baseline CNN for fruit quality classification.

Conclusion

This baseline CNN achieved commendable performance on a fine-grained fruit quality classification task. It performs reliably on banana categories due to sufficient representation in the dataset. It is robust to overfit owing to dropout layers and early stopping. However, it showed underperformance on underrepresented tomato classes. Overall macro metrics are affected due to class imbalance.