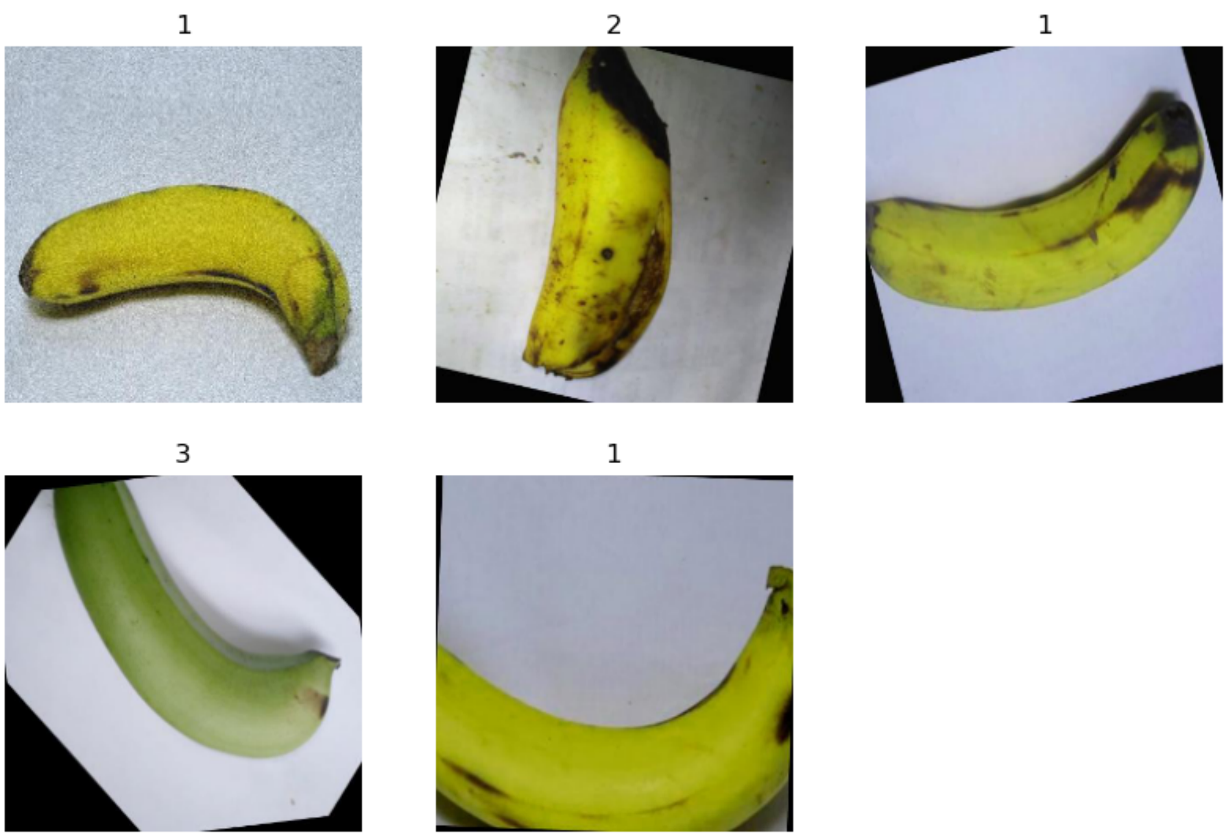
Introduction

In this project I develop a convolutional neural network to classify banana ripeness levels. The goal is to help grocery stores monitor produce conditions and decide when to restock or revisit a section. Automating ripeness detection supports better delivery scheduling and reduces manual checks. The dataset includes labeled images organized into training, validation, and test sets. Each image belongs to one of four ripeness classes and is used to train and evaluate the model.

Method

Pre-Labeled images of bananas at 4 stages of ripeness were sourced from Kaggle (Link), and resized to 224 × 224 pixels to match the networks input. Pixel values were scaled to [0,1] by dividing by 255. To improve generalization and reduce overfitting, I applied on-the-fly augmentations to training set by randomly performing a horizontal flip 50% of the time, a random rotation, and a random zoom. This was only applied to the training images while validation and test were left un-augmented.



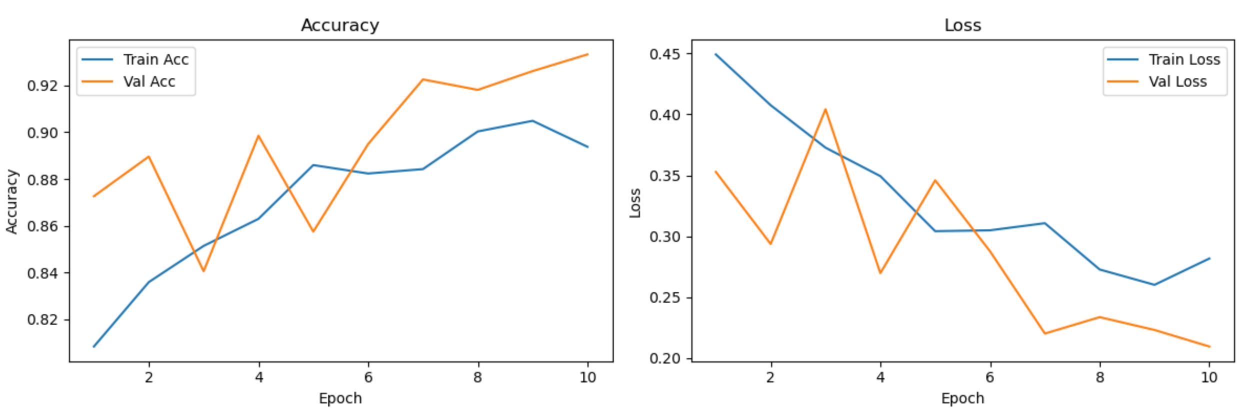
The model is a sequential CNN. It includes three convolutional layers with 32, 64, and 128 filters, followed by ReLU activations and pooling layers. A global average pooling layer reduces the output to a flat vector, followed by a dense layer with 128 units, and a final dense layer with SoftMax to output class probabilities.

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_6 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_10 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_7 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_11 (Conv2D)	(None, 52, 52, 128)	73,856
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 128)	0
dense_4 (Dense)	(None, 128)	16,512
dense_5 (Dense)	(None, 4)	516

The design allows the model to detect color and texture changes across ripeness levels without too much complexity. The model uses the Adam optimizer and sparse categorical cross-entropy loss. Accuracy is tracked during training which used batches size 32 for up to 10 epochs, with early stopping based on validation accuracy.

Outcome

The training process showed consistent improvement in accuracy and reduction in loss. Validation accuracy reached about .93, and validation loss was around 0.20.

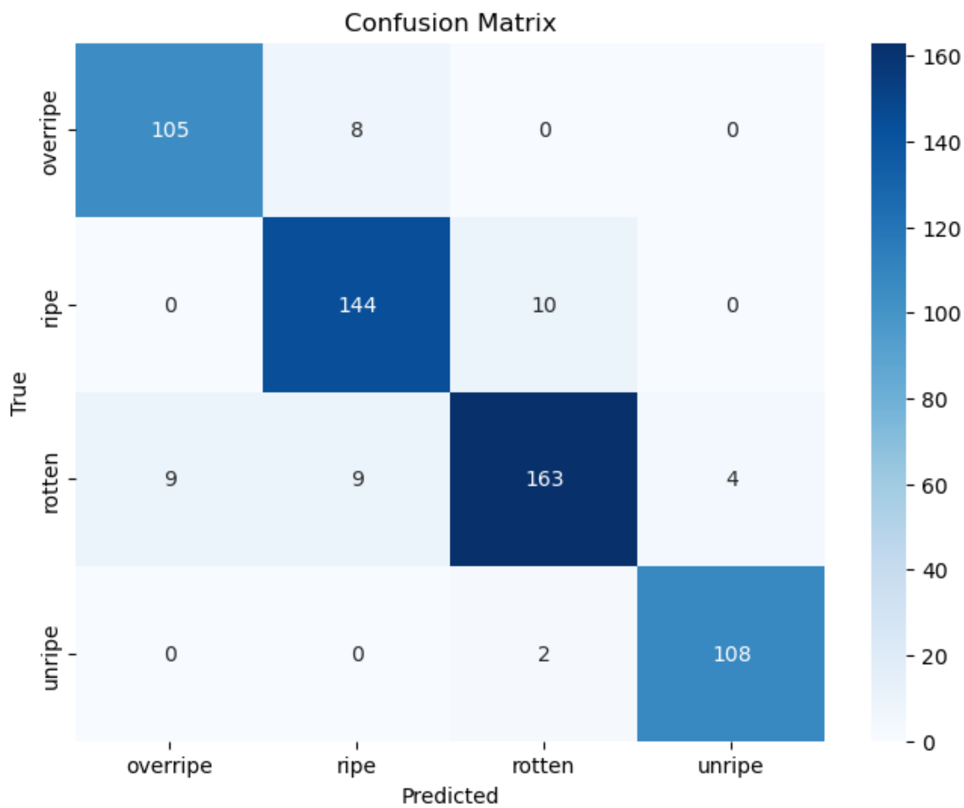


With more compute power, we would train for many more epochs, but because this was trained on a macbook air, we stuck with batch size 32 for 10 epochs. That said, we reached satisfactory performance after about 40 minutes of training.

On the test set, the model has a macro-avg F1score of .92 with precision and recall hovering right around 93%. This performance was relatively consistent across the four classes with ripe having the lowest precision, and rotten having the lowest recall.

	precision	recall	f1-score	support
overripe	0.9211	0.9292	0.9251	113
ripe	0.8944	0.9351	0.9143	154
rotten	0.9314	0.8811	0.9056	185
unripe	0.9643	0.9818	0.9730	110
accuracy			0.9253	562
macro avg	0.9278	0.9318	0.9295	562
weighted avg	0.9256	0.9253	0.9251	562

We can dive deeper by looking at the confusion matrix for each class of banana ripeness.



Here we see that not only is ripeness classification accurate, but when bananas are misclassified, most of the time it is from one level of ripeness to the adjacent level. This further increases our trust in the model.

Conclusion

This model successfully classifies banana ripeness and does so correctly > 90% of the time. With more computing power, and exploration this performance could be augmented, but is satisfactory for the time being. This would be a great step for grocery stores to get ahead of inventory management and continue to improve their delivery planning.