Fresh and Rotten Fruits Classification using Convolutional Neural Networks (CNN)

Aman Varshney, Abhay Jagtap, Siddhesh Bhosale and Tejas Gundale
IIIT Vadodara,
Gandhinagar, Gujarat

***Abstract*— This project aimed to develop a deep learning model for classifying fresh and rotten fruits using a dataset of 12,000 augmented images of bananas, apples, and oranges. The data was collected from Kaggle and preprocessed using the ImageDataGenerator from TensorFlow. The model architecture consisted of convolutional and dense layers, followed by a softmax activation for multi-class classification. The model was trained using the Adam optimizer and evaluated using accuracy as the performance metric. The results showed that the trained model achieved a high accuracy of 97.58% on the validation data. However, further analysis is needed to fully understand the strengths and limitations of the model, such as its generalization to other methods or adaptability to individual preferences. The conclusion highlights the potential for future research and development in this field, including experimentation with different model architectures, hyperparameter tuning, and additional data augmentation techniques. Overall, this project serves as a foundation for fruit quality classification using deep learning techniques and provides insights for further improvements in model performance.**

I. INTRODUCTION

A. Background

The identification and sorting of fresh and rotten fruits play a critical role in quality control in the agriculture and food industry. Ensuring that only healthy fruits are sold and consumed can reduce waste and improve customer satisfaction. In this project, we propose a CNN-based approach for fresh and rotten fruit classification to accurately identify and sort fruits based on their freshness status.

B. Objective

The main objective of this project is to develop a Convolutional Neural Network (CNN) model that can accurately classify images of fresh and rotten fruits. The model aims to achieve high accuracy in identifying the freshness status of fruits, which can be used for quality control purposes in the agriculture and food industry.

C. Scope

This project focuses on the classification of fresh and rotten fruits using CNNs. It uses a dataset of labeled fruit images for model training and evaluation. However, it does not cover other aspects of fruit quality, such as taste or nutritional content. The performance of the model may be affected by factors such as lighting conditions, image quality, and fruit variations. The project does not include real-time implementation or extensive testing in different environments.

D. Methodology

The project followed a standard machine learning workflow, including data collection, data preprocessing, model architecture design, model training, model evaluation, and model deployment. We collected a diverse dataset of labeled images of fresh and rotten fruits and preprocessed the data by resizing images, normalizing pixel values, and augmenting the dataset. We designed a CNN model architecture with multiple convolutional and pooling layers, followed by fully connected layers for classification. The model was trained using the preprocessed dataset and evaluated on a separate testing set to assess its performance. Finally, the trained model was deployed for inference in a real-world environment.

II. PROJECT DATA SET

The dataset for this project was obtained from Kaggle website and consisted of 2000 images of fresh and rotten bananas, apples, and oranges. The dataset was augmented to increase the diversity and quantity of the data. Augmentation techniques, such as rotation by a certain angle and addition of sandpaper noise, were applied to the original images, resulting in a total of approximately 12000 images.

The purpose of data augmentation was to enhance the model's ability to generalize to different variations of the fruits, such as different orientations and surface conditions. The augmented dataset was used for training and evaluating the Convolutional Neural Network (CNN) model in this project.

III. DATA PREPROCESSING

In the data preprocessing step, we used the ImageDataGenerator class from the Keras library in TensorFlow to apply image augmentation and perform data scaling. The 'rescale' parameter was set to 1./255, which normalized the pixel values of the images to the range of [0,1]. This normalization step helps in mitigating the effect of varying pixel intensity values across different images. The 'validation_split' parameter was set to 0.2, which split the training data into a training subset and a validation subset with a 80:20 ratio, respectively. This allows us to have a separate validation set to assess the model's performance during training.

The 'target_size' parameter was set to (20,20), which resized the input images to a fixed size of 20x20 pixels. The 'classes' parameter was set to the list of labels obtained from the 'os.listdir' function, which represents the six classes of fresh and rotten fruits. The 'batch_size' parameter was set to 25, which determines the number of images processed in each iteration during training. These preprocessing steps were applied to both the training and validation data, ensuring that the input images are properly scaled and augmented for training the CNN model.

IV. MODELLING THE DATA

The model architecture consisted of multiple layers, starting with a Conv2D layer with 32 filters and a kernel size of (3,3), followed by a ReLU activation function. This layer served as the input layer and was designed to extract low-level features from the input images of size 20x20 pixels with 3 channels representing RGB colors. A MaxPooling2D layer with a pool size of (2,2) was added after the first Conv2D layer to downsample the extracted features and reduce the spatial dimensions of the feature maps.

The model was then further extended with another Conv2D layer with 64 filters and a kernel size of (3,3), followed by another MaxPooling2D layer with the same pool size. This additional layer helped in capturing higher-level features from the input images. The flattened output from the last MaxPooling2D layer was passed through two fully connected Dense layers with 128 units each and ReLU activation function, which served as the intermediate feature representation layers.

Finally, a Dense output layer with 6 units and softmax activation function was added to classify the input images into 6 classes representing fresh and rotten fruits. The model architecture was designed to capture both local and global features from the input images and make predictions based on the learned features, enabling it to effectively classify fresh and rotten fruits in the subsequent training and evaluation stages.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 18, 18, 32)        896

 max_pooling2d (MaxPooling2D  (None, 9, 9, 32)         0
 )

 conv2d_1 (Conv2D)           (None, 7, 7, 64)          18496

 max_pooling2d_1 (MaxPooling  (None, 3, 3, 64)         0
 2D)

 flatten (Flatten)           (None, 576)               0

 dense_2 (Dense)             (None, 128)               73856

 dense_3 (Dense)             (None, 128)               16512

 dense_4 (Dense)             (None, 6)                 774

=================================================================
Total params: 110,534
Trainable params: 110,534
Non-trainable params: 0
_________________________________________________________________
```

V. TRAINING THE MODEL

The CNN model was compiled using the Adam optimizer, which is a popular stochastic gradient descent (SGD) optimization algorithm known for its efficiency in training deep neural networks. The categorical cross-entropy loss function was used as the objective function to measure the dissimilarity between the predicted class probabilities and the true class labels, which is suitable for multi-class classification tasks.

Additionally, the accuracy metric was used to evaluate the performance of the model during training. The model was then trained using the training data generated from the ImageDataGenerator, and specifies the number of steps per epoch as the length of the training data. The model was trained for 20 epochs, with the validation data being used to evaluate the model's performance after each epoch. The validation steps were also set to the length of the validation data to ensure that the entire validation dataset was used for evaluation at each epoch.

The training process involved updating the model's weights iteratively based on the computed gradients, with the goal of minimizing the loss function and improving the model's accuracy over time. The training history, including the loss and accuracy metrics at each epoch, was stored for further analysis and visualization of the model's performance.

VI. MODEL EVALUATION

In the Model Evaluation part, the trained model was assessed to determine its performance on the validation data. The accuracy of the model provides an indication of how well the model is performing in terms of correctly classifying fresh and rotten fruits.

The strengths of this evaluation method include the ability to quantitatively assess the performance of the model using objective metrics such as accuracy. It provides a straightforward way to measure the accuracy of the model's predictions on the validation data, which can be useful for evaluating the effectiveness of the trained model in accurately classifying fresh and rotten fruits.

However, there are some shortcomings to this evaluation method. One limitation is that it only provides a general measure of accuracy without providing insights into the specific errors made by the model. It may not capture nuances in the model's performance, such as its ability to correctly classify certain types of fruits or its sensitivity to variations in lighting conditions, image quality, or other factors.

The generalizability of this method to other methods and datasets may depend on various factors such as the quality and diversity of the training data, the complexity of the model architecture, and the specific task being performed. It may require further evaluation on different datasets and with different methods to assess its performance in different contexts.

The controllability of the model may be limited, as it may not be able to adapt to individual preferences or specific requirements without further customization. Developing the model further could involve fine-tuning the hyperparameters, increasing the size and diversity of the training data, exploring different model architectures, or incorporating other techniques such as transfer learning or ensemble methods to improve the performance of the model.

The overall accuracy of the model was found to be nearly 97.58%, indicating a high level of accuracy in classifying fresh and rotten fruits. This high accuracy demonstrates the effectiveness of the CNN model in accurately classifying fruits based on their freshness status.

Overall, while the evaluation method used in this project provides a quantitative measure of accuracy, it has limitations and further experimentation and analysis may be required to fully understand the strengths and weaknesses of the trained model and its potential for adaptation to individual preferences or other applications.

VII. CONCLUSION

In conclusion, this project aimed to develop a deep learning model for classifying fresh and rotten fruits using a dataset of 12,000 augmented images of bananas, apples, and oranges. The data was collected from Kaggle and preprocessed using the ImageDataGenerator from TensorFlow. The model architecture consisted of convolutional and dense layers, followed by a softmax activation for multi-class classification. The model was trained using the Adam optimizer and evaluated using accuracy as the performance metric.

The results of the model evaluation showed that the trained model achieved a high accuracy of 97.58% on the validation data. However, it is important to note that the model's performance may have limitations and further analysis is needed to fully understand its strengths and weaknesses, such as its ability to generalize to other methods or adapt to individual preferences.

In conclusion, this project successfully developed a deep learning model for fresh and rotten fruit classification with promising accuracy. Further improvements and fine-tuning may be possible through experimentation with different model architectures, hyperparameter tuning, and additional data augmentation techniques. This project provides a foundation for future research and development in the field of fruit quality classification using deep learning techniques.

Link to Project's Colab Notebook

VIII. REFERNCES

Introduction to Convolutional Neural Network (CNN) using Tensorflow By Govinda Dumane

Fresh and Rotten Fruits Dataset