CSPE74 - Image Processing

Project Report

Topic : Grading Fruit Quality using Deep Learning.

Team Number : 43

Team Members :

- **Siva Sundar – 106120026**
- **Joshua George – 106120046**
- **Rahul Shanker - 106120094**

Index

Problem Statement

Design and implement a robust and efficient deep learning system for automating the grading process of fruits based on quality attributes such as size, color, and defects.

The system should be capable of accurately classifying and grading individual fruits in real-time, providing a reliable and objective assessment that can replace or complement manual grading methods in the agricultural industry.

Design

- Our implementation consists of 2 main parts –

  - The CNN model which helps users to identify the quality of the input images of fruits.
  - The web application that allows users to upload images of different fruits and check their quality.

- The network architecture of the CNN model that we have used for fruit quality detection is given below –

  - **Input Layer :**
    - The input layer expects images of size (64, 64, 3), representing 64x64 pixels with three color channels (RGB).
  - **Convolutional Layers :**
    - Two sets of convolutional layers are employed to capture spatial features in the input images.
    - The first convolutional layer utilizes 32 filters of size (3, 3) with a Rectified Linear Unit (ReLU) activation function to introduce non-linearity.

- The second convolutional layer also has 32 filters and uses ReLU activation.
- Max pooling layers with a pool size of (2, 2) follow each convolutional layer, reducing spatial dimensions and providing translation invariance.

- **Flattening Layer :**
  - The flatten layer is inserted to transform the output of the convolutional layers into a 1D vector. This step is essential before transitioning to fully connected layers.
- **Fully Connected Layers :**
  - Dense layers with varying numbers of units and ReLU activation functions are stacked.
  - The choice of 32, 64, 128, and 256 units in these layers introduces non-linearities and enables the network to learn complex relationships within the data.
  - The final dense layer contains 6 units with a softmax activation function, representing the six classes. This layer outputs probabilities for each class, and the class with the highest probability is predicted.
- **Compilation :**
  - The model is compiled using the Adam optimizer, which adapts the learning rate

during training, and categorical cross-entropy loss, suitable for multiclass classification tasks.
    - Accuracy is chosen as the evaluation metric to assess the performance of the model during training and validation.
- **Data Augmentation :**
    - To enhance the model's ability to generalize, data augmentation techniques are applied to the training set. These include rescaling, shearing, zooming, and horizontal flipping.
- **Training :**
    - The model is trained using the **fit_generator** method, which utilizes the image data generators for both training and testing sets.
    - Training occurs over 20 epochs, and the model learns to classify fruit images into the specified categories by adjusting its weights based on the optimization process.

The web application is written in Python using the Flask framework. It simply asks the user to enter an input image of a fruit which is fed to the pre-trained CNN model and displays the prediction of fruit quality.

Tools Used

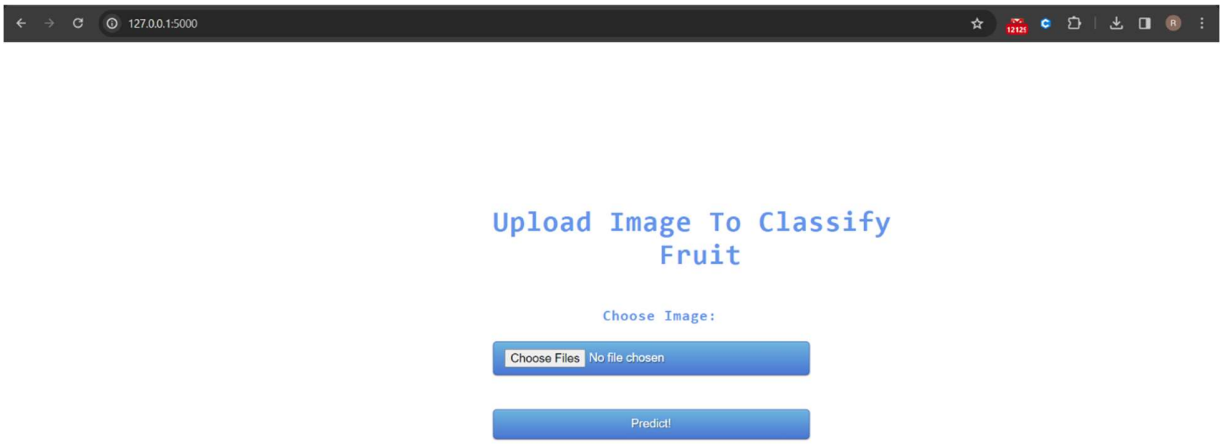
- Python was used to code both the CNN model and the web application.
    - In the CNN model, TensorFlow framework serves as the backend and Keras library was used to define and build the neural network.
    - In the web application, Flask was used to setup the web server.

Code Link

The relevant code for this project can be found in the following repository - https://github.com/rahuls1307/Fruit-Grading-using-DL/tree/main

Dataset used is from Kaggle: https://www.kaggle.com/datasets/sriramr/fruits-fresh-and-rotten-for-classification/code

Sample Testcases



Uploaded Image:

Fresh Banana

Predict Another image

Uploaded Image:

Fresh Apple

Predict Another image

Demo Video

The video can be accessed through the link:
https://github.com/rahuls1307/Fruit-Grading-using-DL/blob/main/Final%20rec.mp4

Novelty of Solution

The solution proposed by us includes augmentation techniques for images to enhance learning of the model. The techniques include shearing, rescaling, zoom and flipping.

1. **Shearing:**
    - Shearing involves shifting the pixels of an image in a fixed direction, distorting its shape and introducing variations that help improve model generalization.
2. **Rescaling:**
    - Rescaling is the process of adjusting the intensity values of pixel colors to a specific range, often between 0 and 1, to standardize the input data and facilitate model training.
3. **Zoom:**
    - Zooming in data augmentation involves randomly enlarging parts of an image, simulating the effect of capturing the subject from different distances and enhancing the model's ability to recognize patterns at various scales.
4. **Flipping:**
    - Flipping involves horizontally or vertically mirroring an image, providing the model with additional perspectives and helping it generalize better to diverse orientations of objects in the dataset.

Contribution

All three members worked equally on training CNN model, building Flask framework and coding the frontend.