**SMART SORTING: TRANSFER LEARNING FOR IDENTIFYING ROTTEN FRUITS AND VEGETABLES**

TEAM ID: LTVIP2025TMID41934

TEAM LEADER: Puvvadi Padmaja

TEAM MEMBER: N Vyshavi

TEAM MEMBER: Naru Shireesha

TEAM MEMBER: Roohi

TEAM MEMBER: Rajanala Ganga Bhavani

PROJECT OBJECTIVE:

The main objective of this AIML project is to develop an intelligent fruit and vegetable sorting system using Transfer Learning to identify and classify rotten and fresh produce. The goal is to automate the sorting process with the help of deep learning techniques, thereby improving accuracy, speed, and reducing manual labor in quality inspection.

THE SYSTEM WILL:

Apply transfer learning by using pre-trained convolutional neural networks (CNNs) such as ResNet, VGG, or MobileNet.

Fine-tune the model on a custom dataset of images containing fresh and rotten fruits and vegetables.

Accurately classify images into “Fresh” or “Rotten” categories.

Provide a scalable and efficient solution for smart agriculture, food quality control, and waste reduction.

This project demonstrates the application of AI and ML in the field of computer vision, focusing on solving real-world problems in agriculture and food processing.

TOOLS AND TECHNOLOGIES:

1.PROGRAMMING LANGUAGES:

Python: Utilized for data preprocessing, model training, evaluation, and deployment due to its simplicity and strong AI/ML library support.

2.DEEP LEARNING FRAMEWORKS:

TensorFlow / PyTorch: Core frameworks used to build and train deep learning models.

Keras: A high-level neural networks API used with TensorFlow for quick and efficient model development

3.PRE-TRAINED MODELS (FOR TRANSFER LEARNING):

AlexNet

VGG-16

GoogLeNet

ResNet50

EfficientNet

> These models are used as a base and fine-tuned with a custom dataset for classifying fresh and rotten produce.

4. DATA HANDLING&IMAGE PROCESSING:

NumPy, Pandas: For structured data manipulation and analysis.

OpenCV, PIL (Python Imaging Library): For reading, preprocessing, and augmenting image data (resizing, normalization, etc.).

5. VISUALIZATION TOOLS:

Matplotlib, Seaborn: For visualizing training metrics such as accuracy, loss, and confusion matrix.

TensorBoard: TensorFlow-based tool for visualizing the training process and model graphs.

DATASET DESCRIPTION:

The dataset used in this project is designed to support a computer vision system for **identifying fresh vs. rotten fruits and vegetables**. The data is organized to train and validate a deep learning model using **Transfer Learning**.

**1. Dataset Name:**

Rotten and Fresh Fruits and Vegetables Dataset *(can be sourced from platforms like Kaggle, GitHub, or custom-collected)*

**2. Data Type:**

* **Image Data**
* Format: .jpg, .png
* Color: RGB images
* Resolution: Varies (typically resized to 224x224 or 256x256 for model input)

**3. Number of Classes:**

* **2 main categories** per item:
  + **Fresh**
  + **Rotten**
* Multiple subcategories based on types of fruits and vegetables:
  + Examples:
    - Apple (Fresh, Rotten)
    - Banana (Fresh, Rotten)
    - Tomato (Fresh, Rotten)
    - Potato (Fresh, Rotten)
    - etc.

**4. Sample Size (Approximate):**

* Total Images: ~10,000+
* Per Class: 500–1000 images
* Balanced across categories to avoid class bias

**5. Data Annotations:**

* Each image is **labeled** based on:
  + Type (fruit/vegetable)
  + Condition (fresh or rotten)
* File/folder naming used for classification (e.g., /apple/fresh/001.jpg)

**6. Data Split:**

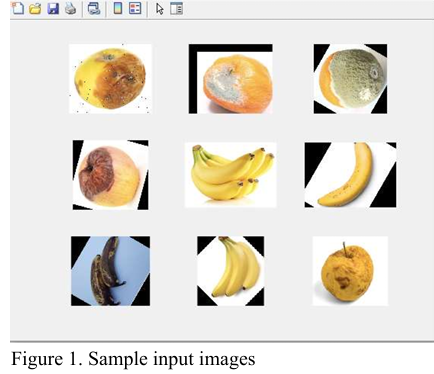
* **Training Set**: 70%
* **Validation Set**: 20%
* **Test Set**: 10%

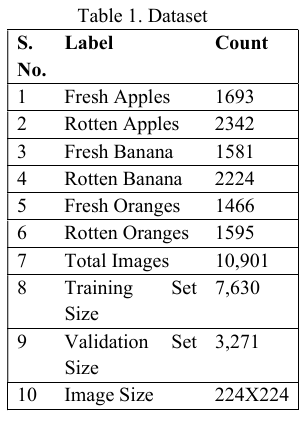
**7. Preprocessing Steps:**

* Image Resizing (e.g., 224x224)
* Normalization (pixel values scaled to 0-1 or -1 to 1)
* Augmentation (for robustness):
  + Rotation, Flipping, Zooming, Brightness

**8. Data Source**

* Open-source datasets (e.g., Kaggle: “Fruits and Vegetables Image Recognition Dataset”.





ARCHITECTURE:



The block diagram represents the steps in detecting the quality of fruits. Initially, the images in the dataset are preprocessed to resize images to 224X24X3 pixels. We are experimenting with four CNN Models AlexNet, GoogleNet, VGG19 and ResNet50. We select one model every time and replace all the final layers. The pre-trained network’s fully connected layer and classification layer are set up for 1000 classes. We swap out these two layers for new ones that are appropriate to the new dataset in order to retrain a previously trained network to categorise new images. However, we had to categorize it into six classes, thus we created a six-unit output softmax layer. The softmax layer establishes the probabilities for each class to which an input image might belong. The trained model can distinguish between fresh and rotten fruit.

PERFORMANCE EVALUATION:

The models are assessed using metrics like precision, accuracy, recall, and F1 score. The accuracy metric is utilized to assess the categorization performance of the model. It indicates the percentage of correctly categorized samples among all samples examined and calculated as follows:

Accuracy=

where

•TP denotes true positives, those who belonged to the class and were appropriately classified in the class;

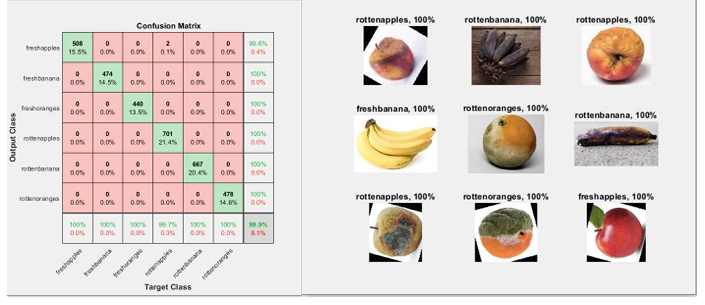
•TN for true negatives, those who did not belong to the class but were appropriately classified in another class;

•FP for false positives, those who did not belong to the class but were incorrectly assigned to the class; and

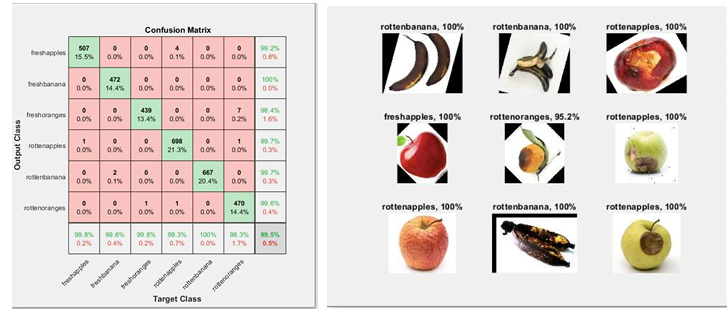
finally, FN for false negatives, those who belonged to the class but were incorrectly classified in another class.

The number of predictions made by a model for each class, as well as the classes to which those predictions belong, are summarized in a confusion matrix.

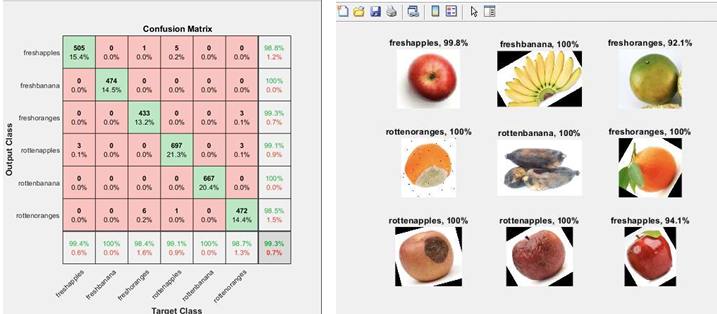
EXPERIMENTAL RESULTS:Top of Form



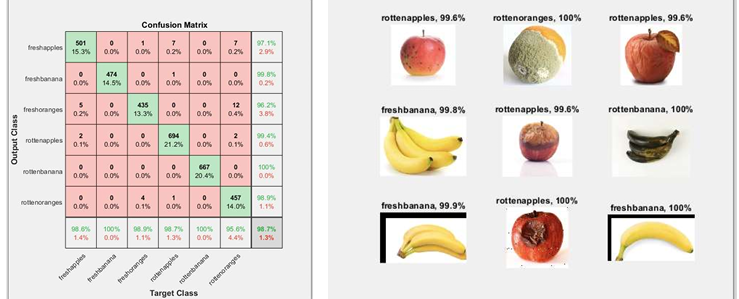
Figure(a). Confusion Matrix of VGG19 (b) Sample Output ImagesBottom of Form



Figure(b). Confusion Matrix of AlexNet (b) Sample Output Images

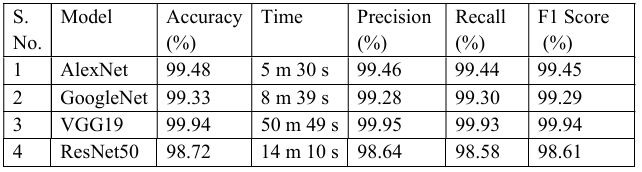


Figure(c) Confusion Matrix of GoogleNet (b) Sample Output Images



Figure(d). Confusion Matrix of ResNet50 (b) Sample Output Images

RESULT OBTAINED:



DISCUSSION:

We applied the transfer learning method to solve the issue of determining fruit quality. We have evaluated the different CNN models like AlexNet, VGG19, GoogleNet and ResNet50 with batchsize 64, learning rate 0.0001, 6 epochs and SGDM optimizer on the dataset to sort the fruits based on their quality. Fresh/rotten apples, fresh/rotten bananas, and fresh/rotten oranges were included in a dataset of six fruit groups to evaluate these pre-trained models. Each pre trained model goes through the fine-tuning procedure to prepare it for training on the gathered dataset. The processing time and validation accuracy of these models are assessed. All the models have good validation accuracy. The Vgg19 model has the best validation accuracy at 99.94%, however, the processing time is excessively long at roughly 50m 49 s. Both AlexNet and VGG19 work admirably, however, AlexNet outperforms VGG19 in terms of processing speed. As shown in Table above, AlexNet has 99.48% validation accuracy and 5 m 30 s processing time. Again, the results above show us that, the best accuracy is obtained by VGG19 and the least is by ResNet50. According to the other metrics precision, recall and F1 score, VGG19 performs the best. We observed that AlexNet is the best architecture for detecting the quality of fruits.

CONCLUSION:

We have evaluated the accuracy and processing time of four different CNN architectures, AlexNet, GoogleNet, VGG19, and ResNet50 and found the most effective method for distinguishing fresh and rotten fruits. The results indicate that the AlexNet model performs the best for sorting fruits and achieves state of art accuracy of 99.48% for a dataset with 64 batchsize, 0.0001 L.R. and 6 epochs with SGDM optimizer and the processing time was 5 m 30 s. Thus, AlexNet performs best in both time and accuracy. Furthermore, we evaluated the models with other metrics like precision, recall, and F1 score. We concluded that the AlexNet architectures can be useful to the producers to improve their sorting process to detect fresh and rotten fruits. As a future aspect, we can evaluate different CNN models to obtain the best accuracy and the least processing time.