

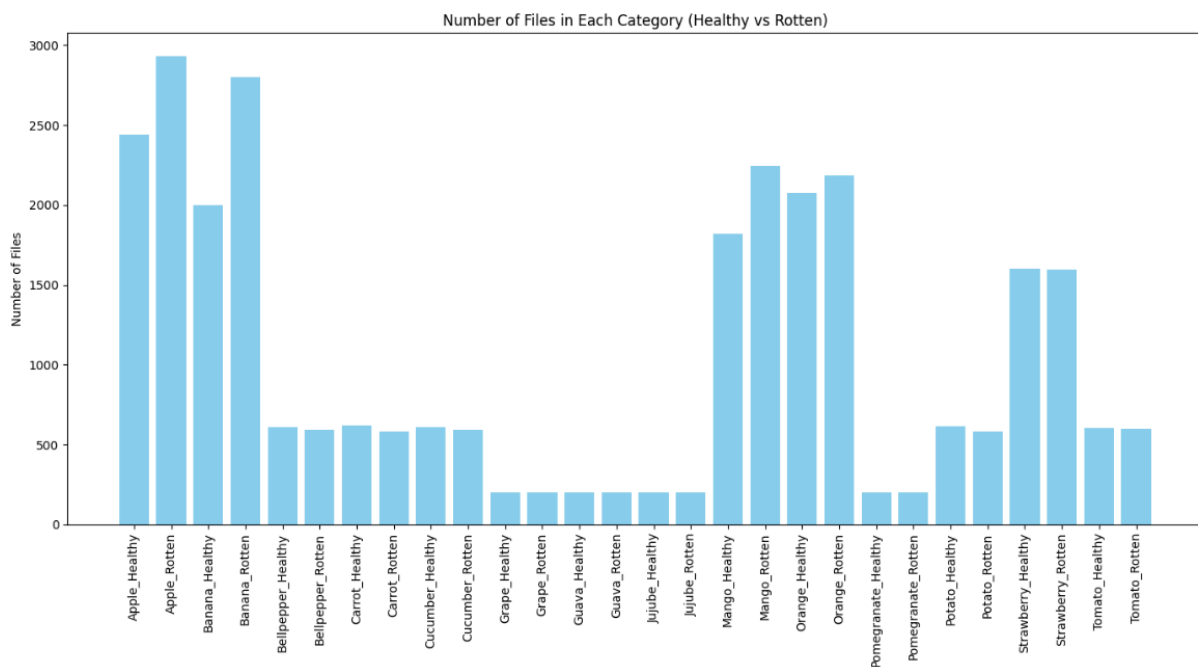
Fruits and Vegetables Disease Detection

Introduction:

This project is based on predicting whether a given fruit or vegetable is rotten or fresh. This is one of the major problems for both producers and consumers so that early detection of disease avoids spreading it in the neighboring plants.

Dataset:

The dataset for this project was chosen from Kaggle. There are 14 distinct fruits and vegetables and each item has two variants, 'Fresh' and 'Rotten' in total containing around 29,400 colored image files which are classified into 28 different classes. The dataset is slightly imbalanced. Please see the below bar graph for analyzing the data:



Preprocessing:

All images are not of the same size, so they were resized to (256,256) size. Converted all images to Pytorch tensors and normalized using standard normalization method.

ML Models and Methodology:

The model used for this project is CNN. Because it is better than other models in classification problems. The dataset is divided into 72,8,20% for train, validate and test data respectively.

I used Adam for gradient descent optimization with learning rate = 0.001 as this is the optimal algorithm in this aspect. Used cross entropy loss for loss function. I used a function for calculating accuracy:

```
def calculate_accuracy(loader, model):
    correct = 0
    total = 0
    model.eval()
    with torch.no_grad():
        for inp, out in loader:
            inp, out = inp.to(device), out.to(device)
            scores = model(inp)
            _, predictions = scores.max(1)
            correct += (predictions == out).sum().item()
            total += predictions.size(0)
    model.train()
    return correct/total
```

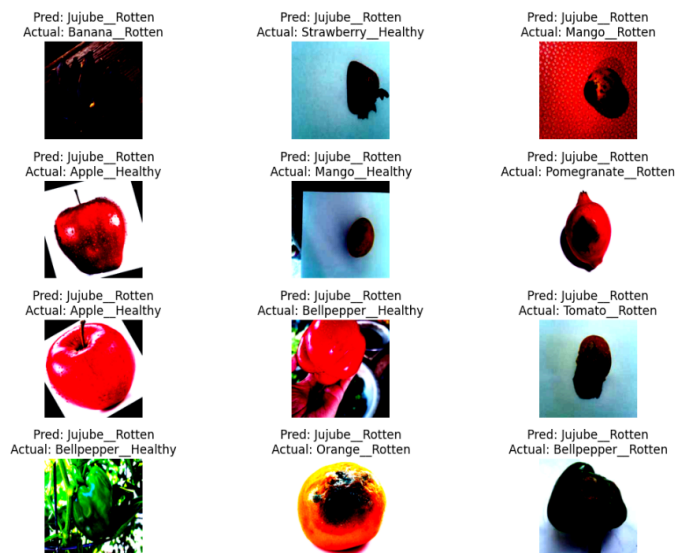
Results:

I got accuracy values for train, test and validation as 68.28%, 68.27% and 68.25% respectively. But when I used only training and testing data (the split then was 80% and 20%), I got nearly 97% training and 84% testing accuracy. When I searched about it online, i came to know that early stopping when loss is very less is not preferable as the model might not be trained much. (But i preferred early stopping because the dataset is large and its taking time for epoch running). Here is the screenshot of model's accuracy:

Training Accuracy is 0.6828528072837633
Validation Accuracy is 0.6825938566552902
Testing Accuracy is 0.682710360129715

Limitations:

I came to know that Adam optimizer is not efficient in large datasets compared to small datasets. So, my model is not trained much and it's always predicting the same class when tested through a sample dataset. This is the screenshot of sample prediction made by my model.



Improvements and Future plan:

As the data is imbalanced, the model is trained more with high frequency classes and not predicting nicely. To improve this, I'll use class weighting and balance the view for all the classes. Instead of Adam, I'll use Stochastic gradient descent algorithm for my future work. I will run the model through epochs without stopping in the middle although loss is too less.

I will implement class wise prediction and calculate accuracy. Also, I had an idea of implementing two models, one for predicting the name of fruit or vegetable and other for predicting its rotten or fresh and then combining the accuracy. Also, this time I didn't go much deeper into CNN, I'll understand it and implement it again with a new boost. Also, I'll try using other models suitable for this project and save this project for my future references.

Conclusion:

The model is not trained well, so its only predicting the same class on sample test data. Please see the kaggle [code](#) for this project.

References:

CNN model: [code](#)

Sample predictions: [code](#)

I am always open for suggestions, please suggest me through mail:

Potla Naga Sai Prajith if you have any.