DSAA PROJECT REPORT

- Group 20

Problem Statement:

Damaged or compromised products in a manufacturing industry are at present to be manually picked by humans. This is a very time consuming and laborious work. And humans at many instances may oversee this process. We plan to aid humans here or replace the necessity of human interference.

To identify apples and oranges in a conveyer belt with multiple fruits with good accuracy so that we can separate oranges from apples.

Ground Truth:

To identify all the apples and oranges in an image with no false positives i.e no oranges should be identified as apples.

Confusion Matrix:

	NEGATIVES	POSITIVES
True	**19**	**59**
False	**18**	**11**

Conclusion -

Total test examples = 107

True Positives = 59
True Negatives = 19
False Positives = 11
False Negatives = 18

Avg. Accuracy = 72.8%

Validation:

Fscore = (2 * TP) / (2 * TP + FP + FN)
= 0.80272
Where,
TP = True Positives
FP = False Positives
FN = False Negatives

Method of Implementation:

This project implements SSD - Single Shot Multibox Detector. This is heavily influenced by ssd.pytorch implementation using Python, OpenCV, Pandas, and PyTorch. An already trained model on a subset of VOC 2012 dataset with average precision (0.6755) was re-trained on Google Open Images dataset for Apples and Oranges.

A python script was used to download and divide the dataset into train, test and validation classes. The overall dataset contained more than 11,000 annotated apples and more than 13, 000 annotated oranges across 1100+ images.

The final retrained model with an average loss of 2.99 was used. The code for validation threw a lot of errors like a lot of functions were depreciated, and due to dependency issues. Therefore, the confusion matrix was manually built by looking at 107 validation images split using the download script.

The video capturing capabilities of OpenCV and FFmpeg building capabilities of imageio are used to give output in real time.

One major problem faced by us is that of scaling. The detector is unable to identify individual entities from a group as the Multi-box concept makes a box around all the areas of interest. Also, we tried to figure out where does it exactly lag in misidentifying the classes. A closer look at memorizations in deep neural networks paper and some youtube videos on similar concepts revealed the kind of randomness and the actual meaning of learning in deep learning to us in terms of just minimising the loss.
We are still trying to wrap our heads around the Vector Calculus used so that we may suggest some changes to be made in the model, if not implement it as such.

We believe that this problem will be solved when we implement this using YOLO implementation.

INDIVIDUAL CONTRIBUTION:

TANAY RATHORE - SSD Implementation of the problem

VISMITH ADAPPA - YOLO Implementation of the problem and documentation

ANIRUDH AMBATI - YOLO implementation

JASWANTH BALABOMMU - Obtaining dataset and GUI