New Model Training
(Partial Improvements based on previous report)

In this phase, we introduced a new image reading method which properly translates BGR image data from OpenCV read frames into input data for models based on RGB input images such as Tensorflow/Keras. This showed significant improvements in both the detection of the model and the classification of the model.

Figure 1: Initial image input and Detection from existing detection model (ssd mobilenet v1 300x300):

```python
while True:
    ret, frame = cap.read()
    if not ret:
        break
    start=time.time()
    image=np.array(frame)
    frame=imutils.resize(frame, width=600)
    H,W=frame.shape[:2]
    print("height: {}, width: {}".format(H,W))
    if (fc==0 or fc%4==0):
        rects=detect_boxes(image, detect_fn,thresh,H,W)
```

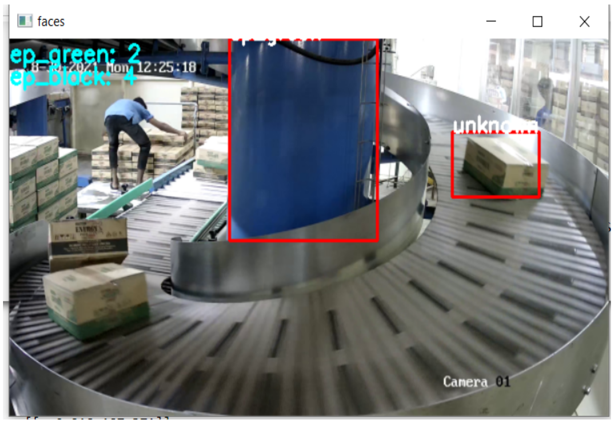
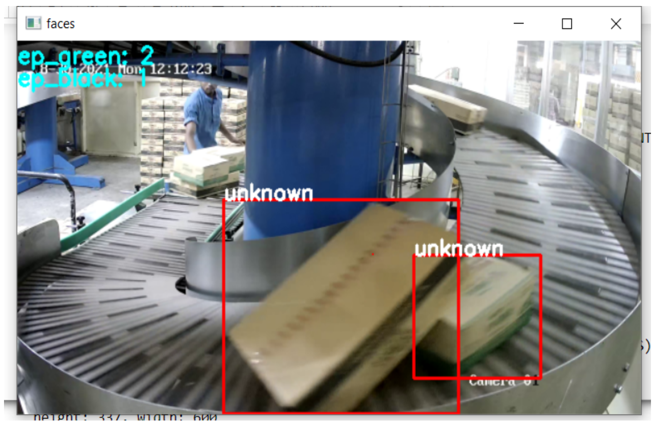


Figure 2: Image input and result from existing detection model after change in code with BGR to RGB translation:

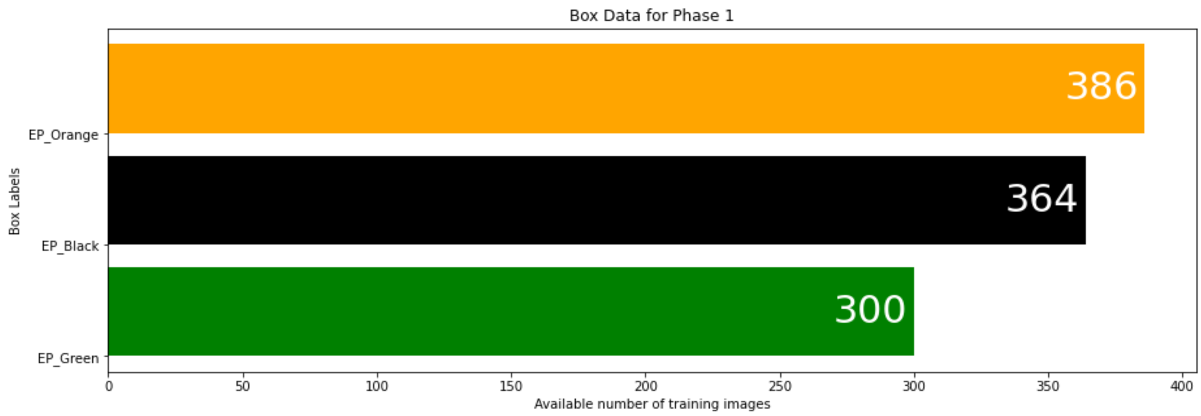
```python
while True:
    ret, frame = cap.read()
    if not ret:
        break
    start=time.time()
    img=cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
    im = Image.fromarray(img, 'RGB')
        #Resizing into dimensions you used
    im = im.resize((300,300))
    img_array = (np.array(im).astype(np.float32))
        #Expand dimensions to match the 4 D Tensor shape.

    frame=imutils.resize(frame, width=600)
    H,W=frame.shape[:2]
    print("height: {}, width: {}".format(H,W))
    if (fc==0 or fc%6==0):
        rects=detect_boxes(img_array, detect_fn,thresh,H,W)
```



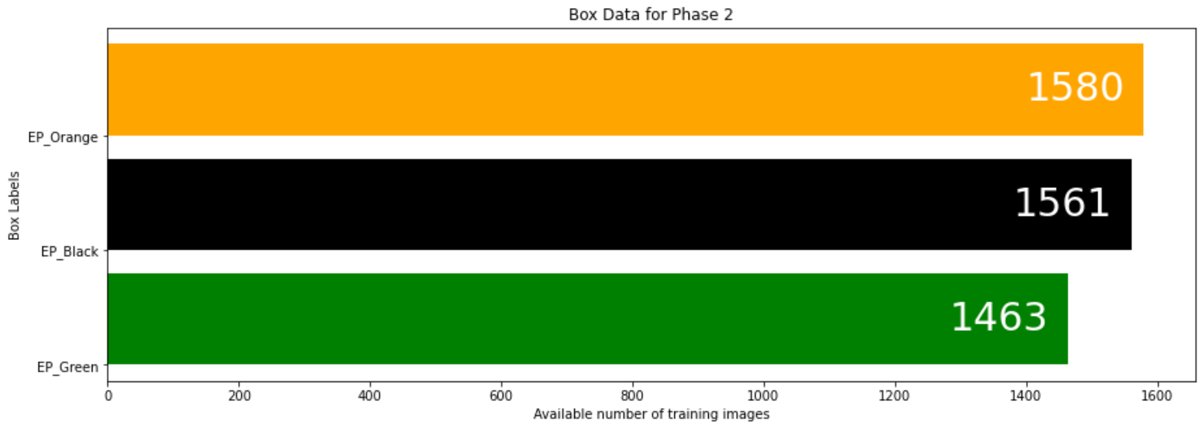
Siamese Backbone Training Changes

In the first trial we used the resnet model that we previously used. The training data of 300 per class was removed with training data of 1500 per class.

Plot 1: Barplot of Available Data in training phase 1

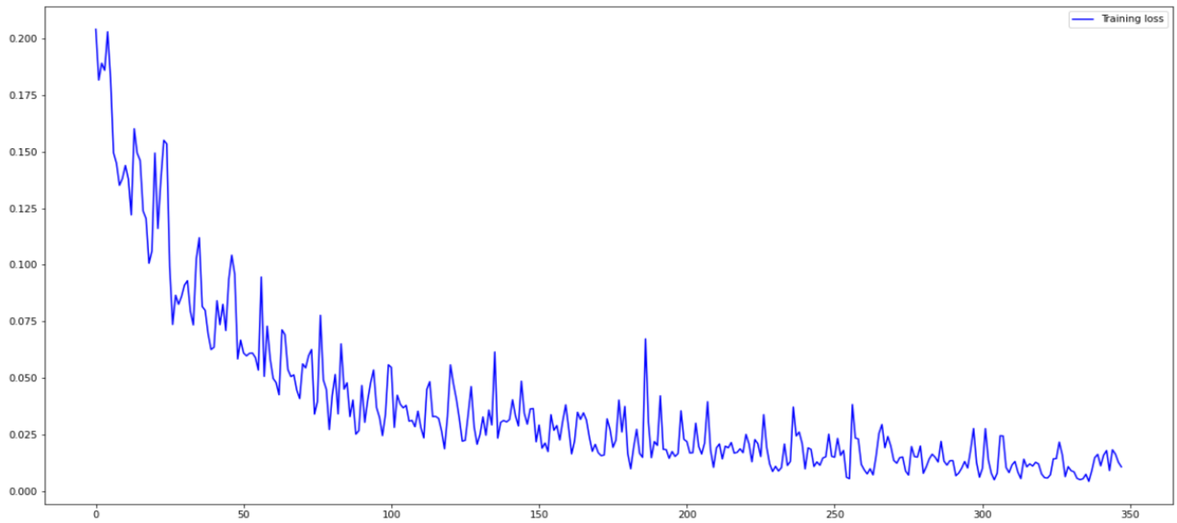


Plot 2: Barplot of Available data in training phase 2



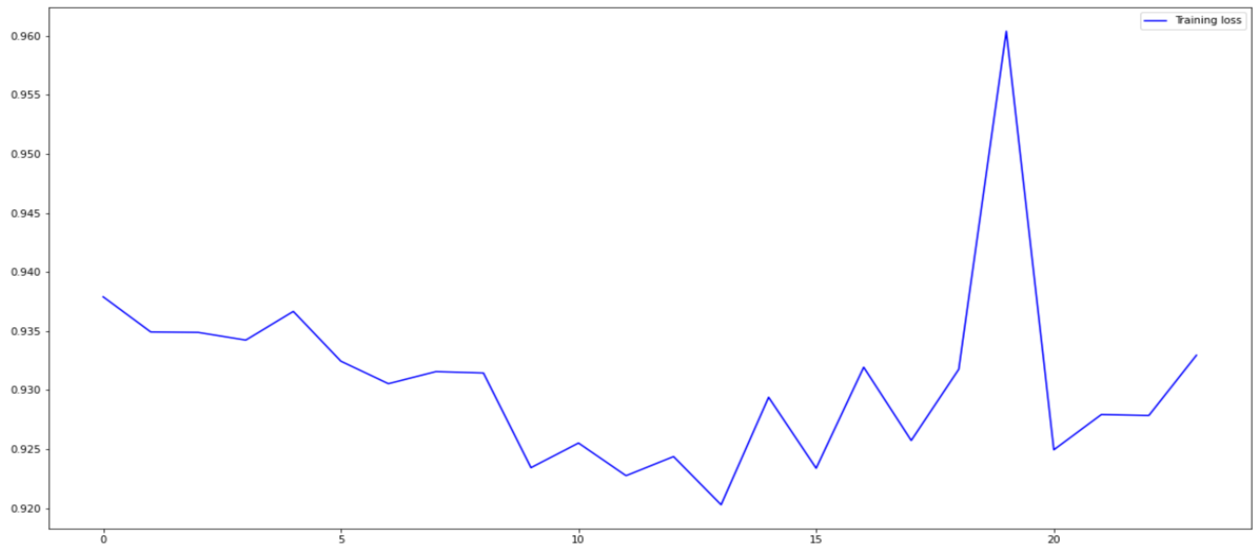
This showed us a significant improvement of the model from a loss of value of 0.01 to a value of 0.004 compared to our previous model. The training graph is given below.

Plot 3: Training Results of New Data on existing Siamese Architecture.



As second experiment we tried to implement a model with less amounts of layers so we used the model architecture of Siamese paired neural networks but with an extra dense layer of 128. The results showed deterioration in training loss. The training loss didn’t go below the 0.9 threshold in the first 25 epochs thus it was terminated. The training graph is given below:

Plot 4: Training results on model based on FaceNet paper



After many further trials we have come to the conclusion that it is not possible to reach better training loss for now with our existing intellectual resources and given project time with a lower number of training layers in architecture. Further research shall be done upon this. We also could not implement the grayscale image inputs for training as the backbone resnet from keras api only supports 3 channelled input, but grayscale images have one channelled input, thus the backbone of the embedding model would need to be built from the ground up. Since we already found a way to translate the BGR to RGB channels, we did not implement it in this stage, but it can be done to further optimize the model.

Findings in further SNN training and testing:

It turns out that the SNN is very good at spotting specific differences it is trained with. According to my empirical evidence, I would say that if say you train SNN boxes of different categories, in which, some are of different colors, details and orientations whereas as some have differences in either only color or orientation. Now, for training your data, you choose three classes at random and most of your training data for positives and negatives have mainly contrast in color and orientation. Then once its trained, the model would be able to identify boxes differently if they were of different colors, that is, their embedding vectors would have high euclidean distance. Whereas in the case of boxes of different categories with similar color and orientation, the pair of embedding vectors would have a comparatively lower euclidean distance.