Question: Face Mask Detection in a crowd.

ZIP FILE CONTENTS

- Modified Dataset
- Trained Model
- Main.py
- Test Images.

DATASET:

Link: https://www.kaggle.com/andrewmvd/face-mask-detection

I have used this dataset for training. It consists of 853 images. A split of 80-20 was done and the training images consists of 683 images while the validation set consists of 170.

The XML files which consist of labels were split in the same ratio (80% training ,20% validation). To make things easier for training I parsed the respective XML files into one CSV file. Hence, the training set and validation set each consists of one csv file.

The dataset I split and prepared for training is present in the zipped folder for further inspection.

MODEL:

For making the model I used the DETECTO library. It makes use of PyTorch and it's built on **Faster R-CNN ResNet-50 FPN.**

The model was trained on colab with GPU runtime enabled.

Libraries used:

- Detecto
- Pandas
- Torch, Torchvision
- Matplotlib
- PIL

```
import pandas as pd
from detecto.core import DataLoader, Model
from detecto import core, utils, visualize
import matplotlib.pyplot as plt
from PIL import Image
from torchvision import transforms
from detecto.utils import normalize_transform
import torch
import torchvision.transforms
from detecto.utils import read_image
```

Transforms applied to the data to increase variety so that the model can learn more features.

```
custom_transforms = transforms.Compose([
    transforms.ToPILImage(),
    transforms.RandomHorizontalFlip(0.5),
    transforms.ColorJitter(saturation=0.2),
    transforms.ToTensor(),
    normalize_transform(),
])
```

The dataset fed into the models:

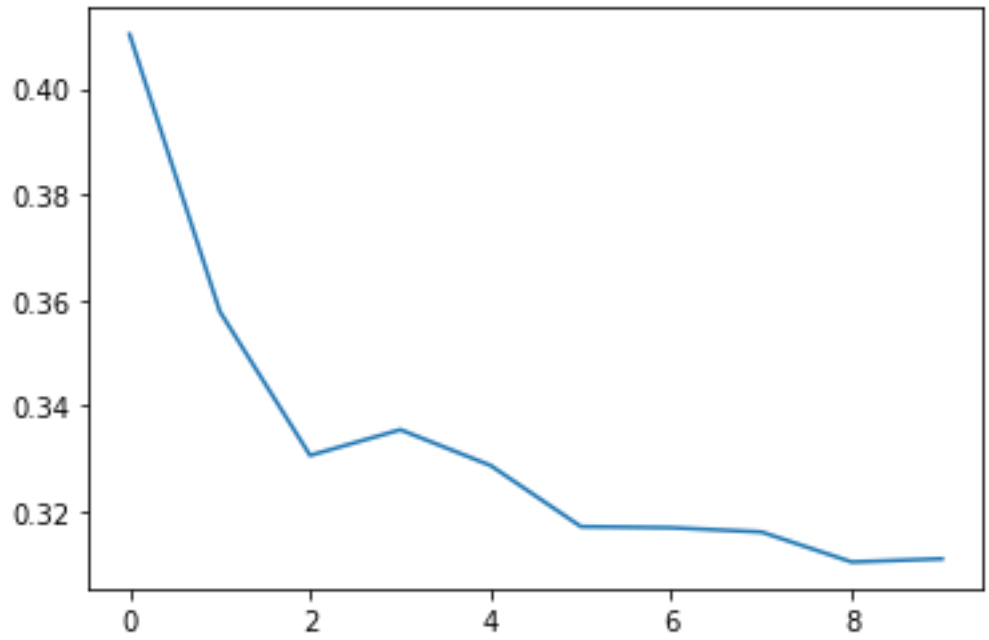
```
training_data = core.Dataset(training_label_path, '/content/facemask_detection/facemask_detection/training_images'
                             , transform = custom_transforms)

validation_data = core.Dataset(validation_label_path,'/content/facemask_detection/facemask_detection/validation_images',
                               transform = custom_transforms)
```

TRAINING:

```
loader = DataLoader(training_data, batch_size=4, shuffle=True)

model = Model(['with_mask', 'mask_weared_incorrect', 'without_mask'])

losses = model.fit(loader, validation_data , epochs=10, learning_rate=0.001,
                   gamma = 0.2, lr_step_size=5,
                    )

plt.plot(losses)
plt.show()
```

DataLoader and Model are a part of the Detecto library.

The unique labels we are trying to predict are the “with_mask” , “mask_weared_incorrect” , “without_mask” hence they are passed onto the Model so that it learns the features to predict them.



Epoch v/s Loss

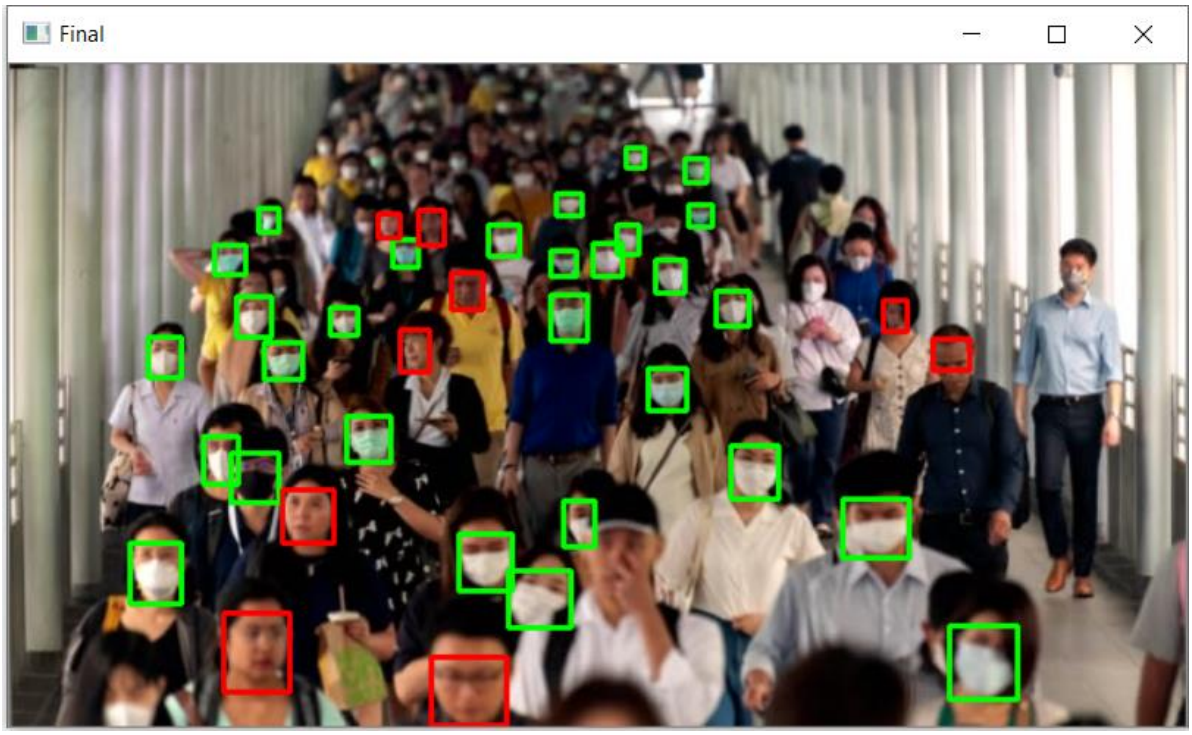
Once the training is done the result was plot. The graph shows the change in loss per every epoch.

The trained model is stored as ‘detector.pth’ in the zipped file.

RESULTS

Test Image:

Result Image:



Green bounding box – People wearing a mask

Blue – People wearing the mask incorrectly. (None in this test image)

Red bounding box - People without a mask

There are some people who have been skipped because the confidence is less than 80%. A higher quality(input) image will give better results and a greater number of detections.

The bounding boxes are drawn for people only with a score of 80% and above.

(refer main.py from the zipped folder)