

Human Corona Monitoring

Use Cases:

1. Assigning a unique Id when person walks in
2. Detecting whether that person wearing a mask or not
3. Social distance monitoring

Output Video URL: [drive link](#)

In video the assigned id of person is above their head. The circle and id change to dark red and increased thickness while violating the social distance. The green box with the alert message is show around the faces with mask and red box with alert for the unmasked faces.

Approach:

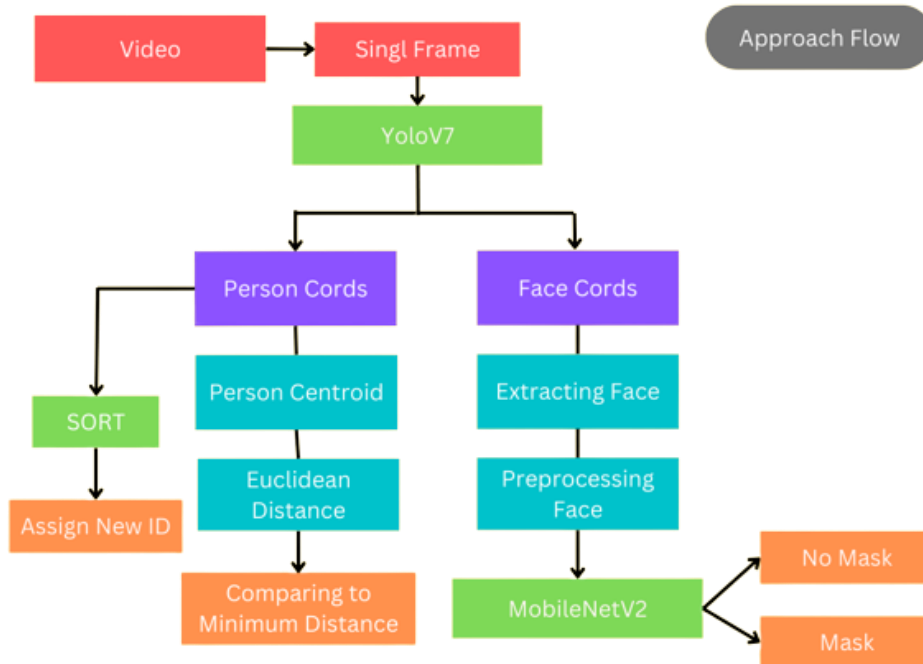
The entire approach includes the detection, tracking, classifying & mathematical calculations to achieve the given use cases is represented visually below.

The **YOLOV7** is used to detect the **person coordinates** and **face coordinates**.

The **SIMPLE ONLINE & REALTIME TRACKING** (SORT) is used for tracking the same person over different frames of the video and it is used for assigning a unique ID to each person in a video.

The person coordinates are used for monitoring the **social distance** between the people by calculating their **Euclidian distance** between the **centroid** positions of them.

The face coordinates are used to extract faces from the frame and given to **MobileNetV2** for **classifying the faces with or without mask**.



Data Collection:

For classification

URL: <https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset>

It is from the kaggle platform the dataset has 2 directory one with masked faces & another with unmasked faces.

For detection:

URL: <http://www.crowdhuman.org/download.html>

The dataset for person and face detection is downloaded from the crowdhuman which has an annotation file in the .odgt format that has the information of person position coordinates & face coordinates.

Data Preparation:

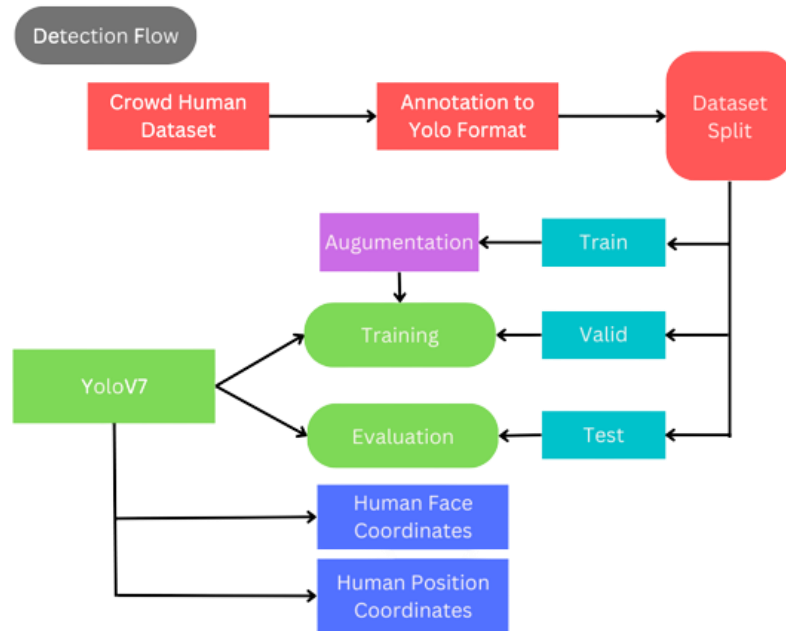
The humancrowd dataset has an annotation in odgt format first it needs to be converted to yolo format that is .txt file for each image. Next is splitting the dataset into train, validation, and test. For detection the dataset is split into the 5000 train, 3059 validation and 1311 test data. For mask classification the dataset is split into 5000 train, 400 validation and 483 test data.

In classification train dataset there are data that are randomly augmented with random process.

Training:

Person & Face Detection:

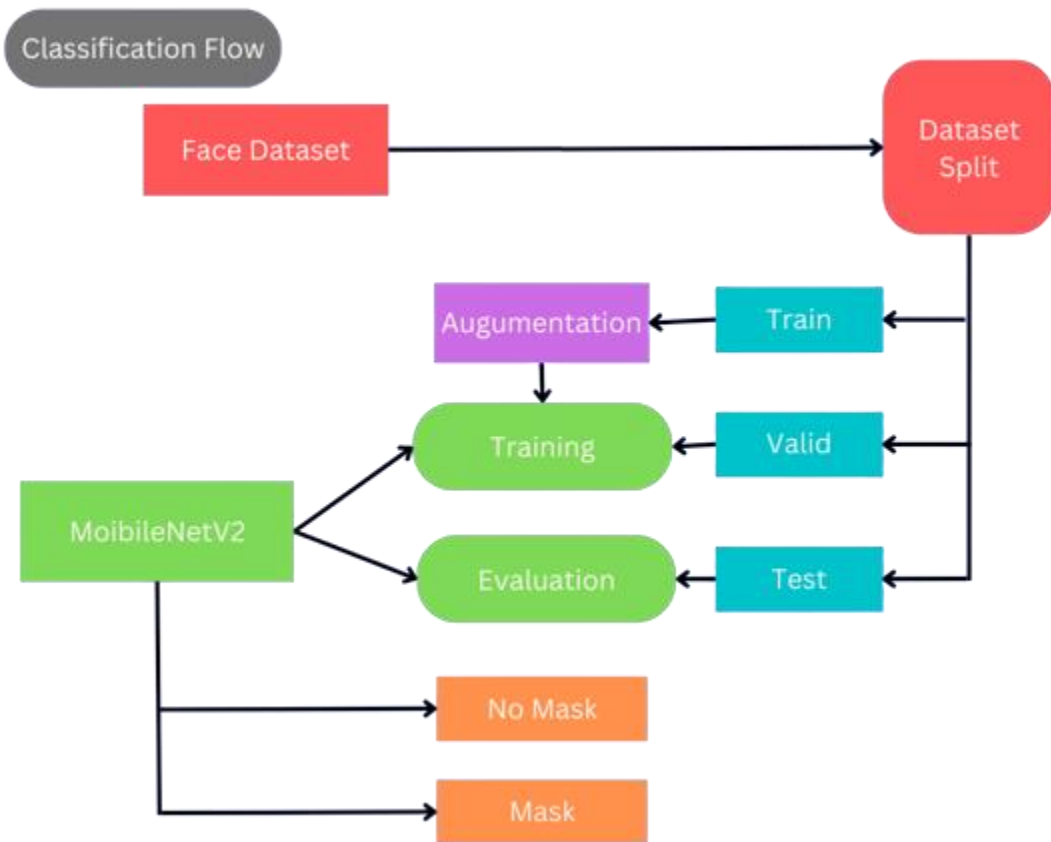
For detection, the yolo v7 architecture is used as YOLO learns generalizable representations of objects and it is faster for the real-time detection. During the training process the augmentation of images like combining multiple images into single image, shading, etc. are done randomly using the albumentation library. The detection training flow is represented visually below.



Face Classification:

The MobileNetV2 model is used for classification of faces. The architecture delivers high accuracy results while keeping the parameters and mathematical operations as low as possible as it is designed particularly for mobile devices. It is also very effective feature extractor. The **Adam** with the **learning rate of 3e-4** and **loss function of categorical entropy** is used for training.

The classification training flow is represented visually below.

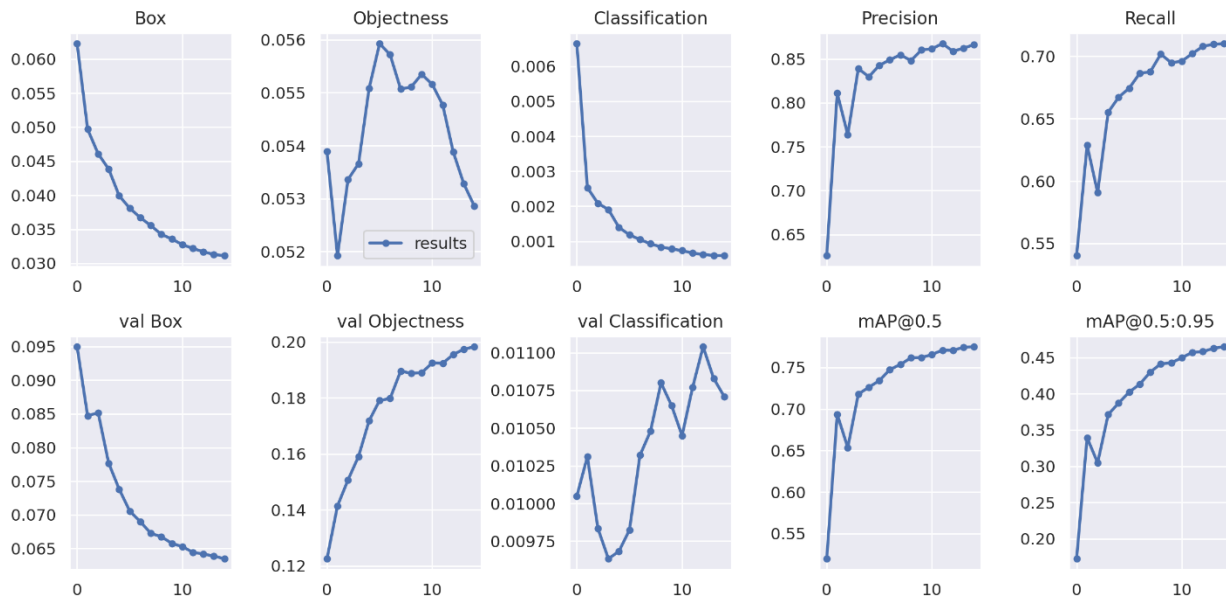


Model Evaluation:

For Detection:

The training IOU is 77.5 % and testing IOU is 46.5%

Class	Images	Labels	P	R	mAP@.5	
all	3007	142286	0.866	0.71	0.775	0.465
body	3007	71143	0.864	0.752	0.821	0.502
head	3007	71143	0.868	0.668	0.729	0.428



For Classification:

The training accuracy is 100% and test data accuracy is 99.9%. The evaluation based on the test dataset are

	precision	recall	f1-score	support
WithMask	0.9979	1.0000	0.9990	483
WithoutMask	1.0000	0.9980	0.9990	509
accuracy			0.9990	992
macro avg	0.9990	0.9990	0.9990	992
weighted avg	0.9990	0.9990	0.9990	992

Social Distancing:

To monitor the social distance between persons we need to map pixels to measurable units that are based on the calibration of the camera. In here the **centroid** of the person is calculated based on the **bounding box** of person from yolo. After the **distance between the centroids** of people are calculated using the **Euclidian distance**. If the calculated pixel distance (in pixels) is less than the minimum distance (in pixels) then social distance is violated and alert is shown in top left corner.

Assigning Unique ID:

The unique id to the person is assigned by using SORT. When the first frame is sent to SORT it assigns the id to all coordinates. The working of the sort is represented visually below. Based on the position of the person in last frame the kalman filter predicts the position if that specific person in the next frame. Then the person position from the current frame along with the previous predicted position by kalman filter is given to Hungarian algorithm it checks for IOU if the iou is greater than the threshold it sets the same id to the person if not it sets a new id.

