# INTRODUCTION

An Object Detection model has been built to detect whether a person is wearing mask or not and show the exact location of the object in the image.

# DATASET

The dataset being used is a collection of around 1,510 images consisting of both masked and unmasked persons. Besides images, annotation files are also present for images which contain the labels of either class i.e., 0 for mask or 1 for no mask, along with coordinates of bounding box.

# DATA PREPROCESSING

1. Removed multi-faced images: The annotation files for each image were scanned and those images were removed which contained more than one person face (We are mainly focusing on single-object detection in this use case).

2. Formatting images compatible for feeding into model: Each image was being converted to grayscale and then resized to 244 x 244 pixels (The dimension is same as the shape of the input for the CNN model which has constructed later).

3. Conversion into tensors: Each image, label and coordinates were converted into tensors using TensorFlow library.

4. Splitting the data: The data was then split into train (70%), validation (25%) and test (5%) sets.

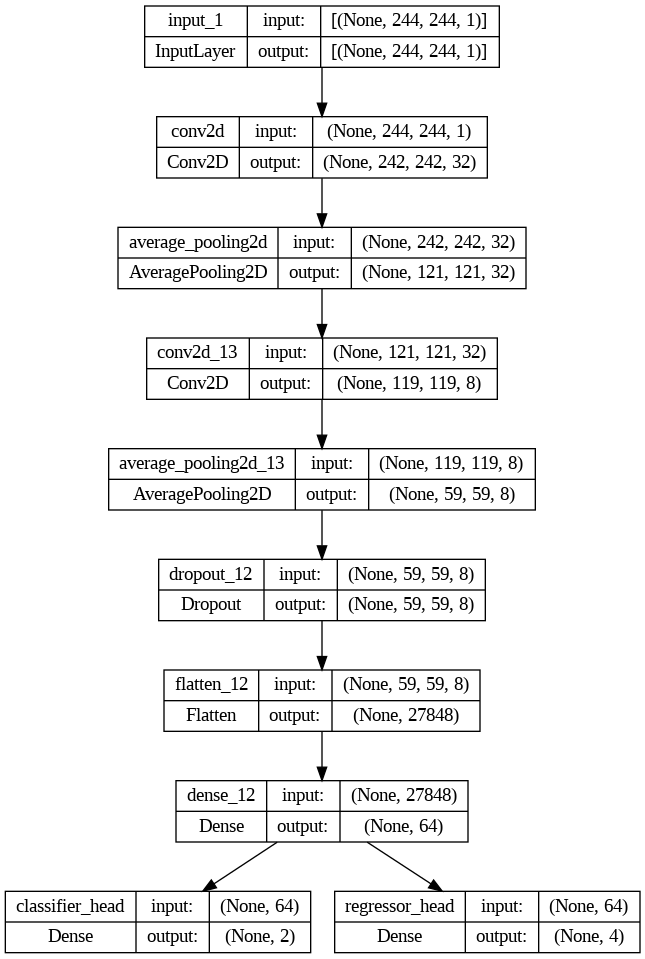
# CNN ARCHITECTURE

Built a custom CNN architecture for this use case. It has 9 layers including Input layer, hidden layer and output layer.

Input layer takes image input as shape of 244 x 244 x 1.

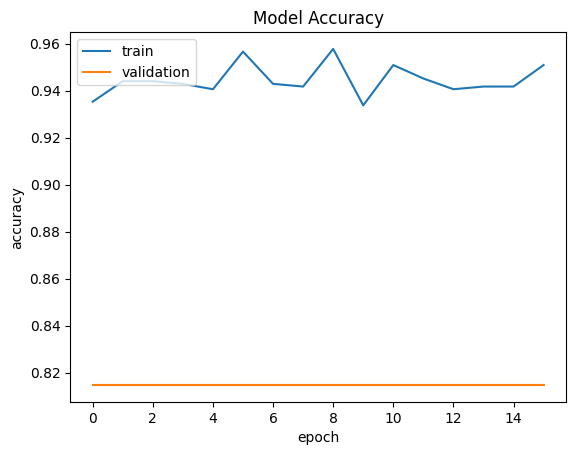
Output layer contains two heads viz. Classifier head and Regressor head. Output shape of classifier head is 1x2, which contains probability of both the classes. Output shape of regressor head is 1x4, which contains coordinates of bounding box.

Classifier head has categorical cross-entropy as loss function and Regressor head has MSE as loss function.

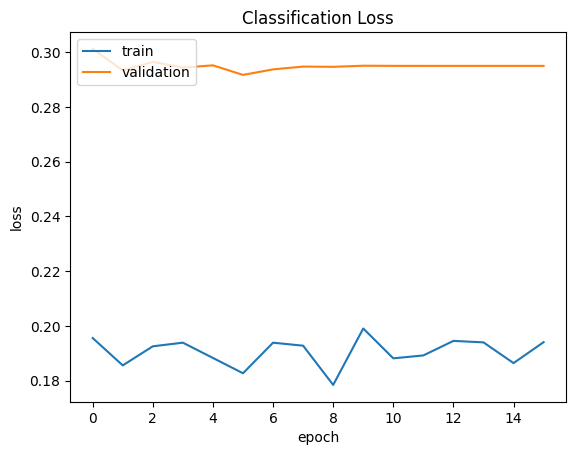


# PERFORMANCE OF MODEL ON TRAINING AND VALIDATION SET

Model Accuracy



Model Loss



Bounding Box Loss

A graph with blue lines and orange lines

Description automatically generated

# HYPERPARAMETER TUNING

The CNN object detector was being tuned using Keras tuner. The following hyperparameters were being tweaked:

1. Number of hidden layers

2. Number of filters/kernels

3. Activation function for each layer

4. Optimizer being used

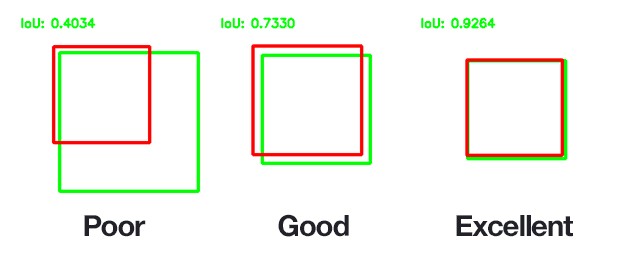
# EVALUATION METRIC

1. IoU (Intersection over Union): IoU score tells us how well the predicted bounding box overlaps the actual bounding box. The idea behind IoU is pretty simple; compare the intersection and union area between the predicted and actual boxes by dividing the intersection by the union, as shown in the following image.

A diagram of a company

Description automatically generated

IoU value ranges from 0 to 1.



If IoU is greater than 0.5 and predicted object is true class, then that is True Positive (TP).

If IoU is less than or equal to 0.5, then that is False Positive (FP).

If there is no detection or IoU is greater than 0.5 but prediction is wrong, then that is False Negative (FN).

1. Precision: Precision is equal to TP/(TP + FP). In other words, of all bounding box predictions, what fraction was located correctly.
2. Recall: Recall is equal to TP/(TP + FN). In other words, of all target bounding boxes, what fraction did we correctly detect.
3. mAP (Mean Average Precision): The mAP is computed by first calculating the Average Precision (AP) for each class, and then taking the mean of the AP values across all classes. To compute the AP for a single class, the precision-recall curve for that class is first created. The area under this curve (AUC) represents the AP for that class. The mAP value ranges from 0 to 1, where 0 indicates poor performance and 1 indicates perfect performance.

# PERFORMANCE OF THE MODEL ON TEST DATA

The test dataset contains 65 images. Tested model performance on all these 65 images and achieved mAP equals to 0.42.



# SCOPE OF IMPROVEMENT

Taking in consideration, limited data was available for training and the model was trained from scratch without using any pre-trained models like Yolo or transfer learning techniques, a mAP equals to 0.41 has been achieved. This metric can be improved in the following ways:

1. The first option to improve the bounding box and classification performance is to get access to more image data.
2. Trying more hyperparameter combinations for.eg. number of layers, number of filters, activation function etc., can reduce the classification and regression errors.
3. Instead of custom CNN architecture, transfer Learning can be used to improve the accuracy of model.