IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 7, June 2022

Impact Factor: 6.252

# Object Detection using Tensor Flow API

**Sarojini V. Naik[1], Archana S. Gaikwad[2], Ajay V. Raipure[3], N. V. Hippalgaonkar[4], V. S. Kharote-Chavan[5]**

Lecturer, Department of E&TC[1,2,4,5]

Head of Department, Department of E&TC[3]

Pimpri Chinchwad Polytechnic, Pune, Maharashtra, India

**Abstract:** *This paper is based on object detection using Tensor flow. Due to object detection's close relationship with video analysis and image understanding, it has attracted much research attention in recent years. The ubiquitous and wide applications like scene understanding, video surveillance, robotics, and self-driving systems triggered vast research in the domain of computer vision in the most recent decade. Being the core of all these applications, visual recognition systems which encompasses image classification, localization and detection have achieved great research momentum. Due to significant development in neural networks especially deep learning, these visual recognition systems have attained remarkable performance. Object detection is one of these domains witnessing great success in computer vision. This paper demystifies the role of deep learning-based object detection frameworks and its representative tool, namely Convolution Neural Network (CNN). So, by studying these we decided to train a model for Mask detection and also for object detection using Single Shot Detector (SSD). After the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms, face mask being the primary one. The basic aim of the project is to detect the presence of a face mask on human faces on live streaming video as well as on images. We have used deep learning to develop our face detector model. The architecture used for the object detection purpose is Single Shot Detector (SSD) because of its good performance accuracy and high speed. Alongside this, we have used basic concepts of transfer learning in neural networks to finally output presence or absence of a face mask in an image.*

**Keywords:** Tensor Flow, Convolution Neural Network,Single Shot Detector, Object Detection, Deep Learning, etc.

## I. INTRODUCTION

Object detection's close relationship with video analysis and image understanding, it has attracted much research attention in recent years. So, our report gives a brief introduction on the deep learning and its representative tool, namely Convolutional Neural Network (CNN). Frameworks and Services of Object Detection is done with the help of Tensor Flow. The Tensor Flow Object Detection API is an open-source framework built on top of Tensor Flow that makes it easy to construct, train and deploy object detection models. There are already pre-trained models in their framework.

There are already pre-trained models in their framework. includes a collection of pre-trained models trained on various its datasets such as the –

- COCO (Common Objects in Context) dataset,
- the KITTI dataset, and
- the Open Images Dataset.

Tensor Flow bundles together Machine Learning and Deep Learning models and algorithms. It uses Python as a convenient front-end and runs it efficiently in optimized C++. Tensor Flow is at present the most popular software library. There are several real-world applications of deep learning that makes Tensor Flow popular.

Being an Open-Source library for deep learning and machine learning, Tensor Flow finds a role to play in text-based applications, image recognition, voice search, and many more.
Deep Face, Facebook's image recognition system, uses Tensor Flow for image recognition. So, the paper clarifies the role of deep learning-based object detection frameworks and its representative tool, namely CNN. So, by studying these we decided to train a model for Mask detection and also for object detection using Single Shot Detector (SSD).

### A. Methodology

Tensor Flow Object Detection API- The Tensor Flow object detection API is a software framework used for object detection tasks. Object Detection is the process of finding real-world object instances like car, bike, TV, flowers, and humans in still images or Videos. It allows for the recognition, localization, and detection of multiple objects within an image which provides us with a much better understanding of an image as a whole. It is commonly used in applications such as image retrieval, security, surveillance, and advanced driver assistance systems (ADAS).
Tensor Flow Object Detection API and other similar APIs (Keras, RCNN, YOLO) uses pre trained CNNs inside the frameworks for predictions. Most of those pretrained models are trained using, the COCO dataset, and the Open Images Dataset. These models are trained to detect some fixed type of categories of objects such as, persons, cars, dog, tooth brush and etc. If you want to detect custom objects you can retrain those models using your own datasets.

### B. Convolution

CNN stands for Convolution Neural Network. CNNs have revolutionized Pattern Recognition in last decades. As a result, CNNs are currently used for various applications such as Object Detection and Image Recognition. This process of Object Detection is carried out in four layers by CNN Models.

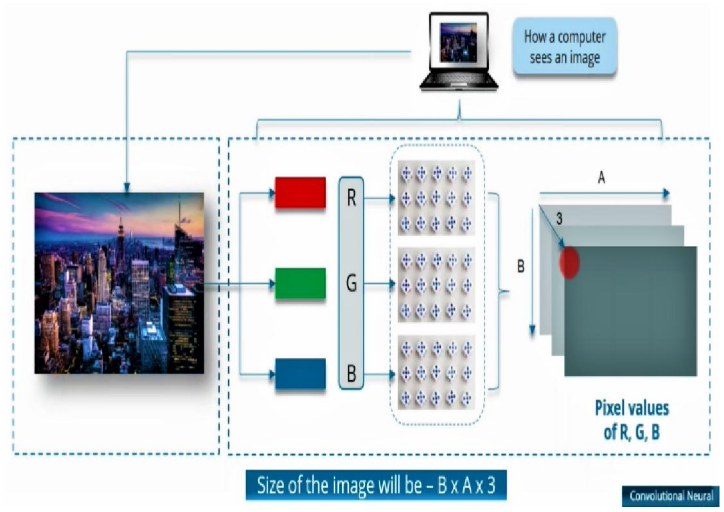

How a computer sees an image
R
G
B
A
3
B
Pixel values of R, G, B
Size of the image will be - B x A x 3
Convolutional Neural

**Figure 1:** Convolution Neural Network pattern recognition

So firstly, let us understand how exactly a computer reads an Image. In the figure 1.0 we can see the image of Dog and when we look at this image, the first thing we notice, is a Dog and a Puppy, Different Colours, and other stuffs like that, but how a computer read this image? So basically, there will be three channels- known as RGB Channel. Now each of this channel will have their own respective Pixel Values as shown in the Figure. So, when I say image size is- BxAx3 pixels it means that there are B Rows, A Columns and 3 Channels. And this is how a computer sees an image and understands through its pixel values. So, this is for coloured images for Black n White there are two channels. Now here the role of CNN is very important, it gives this pixel values for particular image or Object. And this is done by CNN Models.

IJARSCT

ISSN (Online) 2581-9429

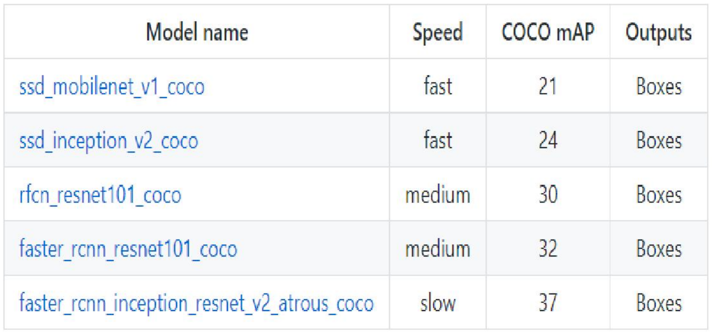
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 7, June 2022

Impact Factor: 6.252

### C. CNN models used in TensorFlow Object Detection API-

There are different types of CNN models as shown here. Basically, these CNN Models have pre-trained images i.e. It contains 100+ images trained with their pixel values. In a simple way, suppose we have to identify this dog's picture. So, CNN Model already contains a trained image of Dog with its pixel value and it compares this original pixel value with the pixel value we extracted from the image we have to identify, if it matches or comes near to the original pixel value it identifies the picture as dog.So as there are different CNN models, each model has different speed and Accuracy of object detection.



| Model name | Speed | COCO mAP | Outputs |
|---|---|---|---|
| ssd_mobilenet_v1_coco | fast | 21 | Boxes |
| ssd_inception_v2_coco | fast | 24 | Boxes |
| rfcn_resnet101_coco | medium | 30 | Boxes |
| faster_rcnn_resnet101_coco | medium | 32 | Boxes |
| faster_rcnn_inception_resnet_v2_atrous_coco | slow | 37 | Boxes |

**Figure 1.1:** Convolution Neural Network Models

### D. mAP (mean Average Precision)-

The accuracy is measured in mAP means mean average precision; it is popular metric in measuring accuracy of object detection. We cansee here that faster the speed of object detection less the accuracy and slower the speed of detection better the accuracy. Average precision computes the average precision value for recall value over 0 to 1.

### E. SSD Mobile Net:

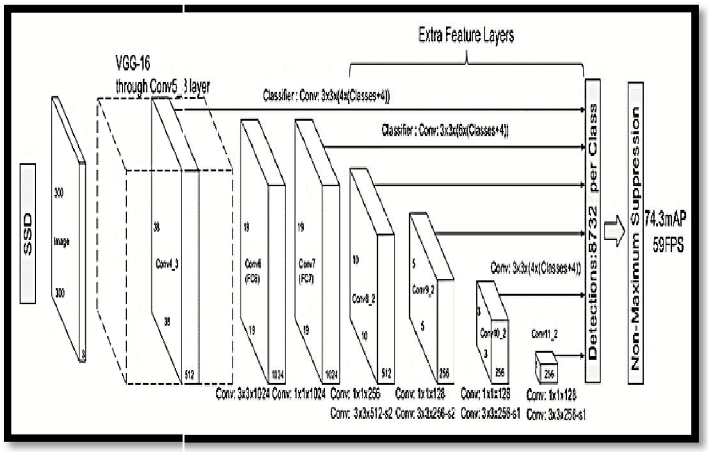


**Figure 1.2:** The architecture of SSD

This is the architecture of SSD Mobile Net in Convolutional Neural Networks (CNN), we are going to use for our object detection module as in Figure 2.0. The working of the Single Shot Detector algorithm relies on an input image with a specified bounding box against the objects. The methodology of predicting an object in an image depends upon very renowned convolution fashion.

For each pixel of a given image, a set of default bounding boxes (usually 4) with different sizes and aspect ratios are evaluated. Moreover, for all the pixels, a confidence score for all possible objects is calculated with an additional label of 'No Object'. This calculation is repeated for many different feature maps.

In order to extract feature maps, we usually use the predefined trained techniques which are used for high quality classification problems. We call this part of the model a base model. For the SSD, we have VGG-16 network as our base model. At the training time, the bounding boxes evaluated are compared with the ground truth boxes and in the back propagation, the trainable parameters are altered as per requirement. We truncate the VGG-16 model just before the classification layer and add feature layers which keep on decreasing in size. At each feature space, we use a kernel to produce outcomes which depicts corresponding scores for each pixel whether there exists any object or not and the corresponding dimensions of the resulting bounding box.

VGG-16 is a very dense network having 16 layers of convolution which are useful in extracting features to classify and detect objects. The reason for the selection is because the architecture consists of stacks of convolutions with 3x3 kernel size which thoroughly extract numerous feature information along with max-pooling and ReLU to pass the information flow in the model and adding non linearity respectively from the given image. For additional nonlinearity, it uses 1x1 convolution blocks which does not change the spatial dimension of the input. Due to the small size filters striding over the image, there are many weight parameters which end up giving an improved performance.

At the input end, we can see the VGG-16 being used as the base model. Some additional feature layers are added at the end of the base model to take care of offsets and confidence scores of different bounding boxes. At end part of the figure, we can see the layers being flattened to make predictions for different bounding boxes. At the end, non-maximum suppression is used whose purpose is to remove duplicate or quite similar bounding boxes around same objects. There may be situations where the neighbouring pixel also predicts a bounding box for an object with a bit less confidence which is finally rejected.

The problem can be solved in two parts: first detecting the presence of several faces in a given image or stream of video and then in the second part, detect the presence or absence of face mask on face. In order to detect the face, we have used the OpenCV library. The latest OpenCV includes a Deep Neural Network (DNN) module, which comes with a pre-trained face detection convolutional neural network (CNN). The new model enhances the face detection performance compared to the traditionalmodels. Whenever a new test image is given, it is first converted into BLOBS (Binary Large Object refers to a group of connected pixels in a binary image) and then sent into the pretrained model which outputs the number of detected faces. Every face detected comes out with a level of confidence which is then compared with a threshold value to filter out the irrelevant detections. After we have the faces, we need to evaluate the bounding box around it and send it to the second part of the model to check if the face has a mask or not.

The second part of the model is trained by us using a dataset consisting of images with mask and without mask. We have used Keras along with TensorFlow to train our model. First part of the training includes storing all labels of the images in a NumPy array and the corresponding images are also reshaped (224, 244, 3) for the base model. Image augmentation is a very useful technique because it increases our dataset with images with a whole new perspective. The base model that we have used here is MobileNetV2 with the given 'ImageNet' weights. ImageNet is an image database that has been trained on hundreds of thousands of images hence it helps a lot in Image classification. CNN process is carried out in 4 layers which means it extracts the pixel values of the Input image to compare with original pixel value which is already stored in Model. So, the four layers are Convolution Layer, ReLU Layer, Pooling Layer and finally Fully Connected Layer.

- **In Convolution Layer**, the image is Converted into number of pixels, then adding those pixels and dividing them into total no. of pixel is done here.
- **In ReLU Layer**, after getting those pixel values if any input pixel value is negative or zero, then the output is given as Zero.
- **Pooling Layer**, in this layer we shrink the image stack into smaller size. For Example: Four pixels image input are converted into one single pixel by giving them the value of highest pixel.
- **Fully Connected Layer,** stacking up the layer, means all of the above layer's process is done again by taking out filtered and shrinked images and putting them into single list.

Thus, Output is given as detected category of the bounding object and also their bounding rectangles parameter as (x-min, x-max) (y-min, y-max).

**F. How Object Detection is Done?**

**1.** Generates the small segments in the input image as shown in figure 3.0. So, it generates small segments in the input image as we seen in Convolution layer.

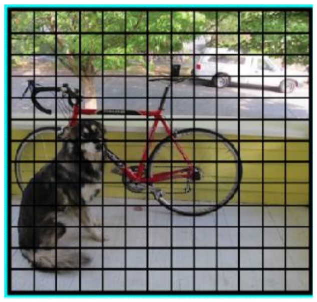


**Figure 1.3:** Small segments of a picture

**2.** Feature extraction is carried out for each segmented rectangular area to predict whether the rectangle contains a valid object as shown in the figure 3.2. Which means Extraction of Pixel values of the input image is done and Compared with the original pixel values with the trained image stored in the Model.

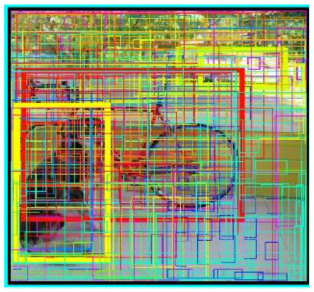


**Figure 1.4:** Extraction for each segment

**3.** Overlapping boxes are combined into a single bounding rectangle (Non-Maximum Suppression) - Thus, we get the output identified with bounding rectangle as in figure.

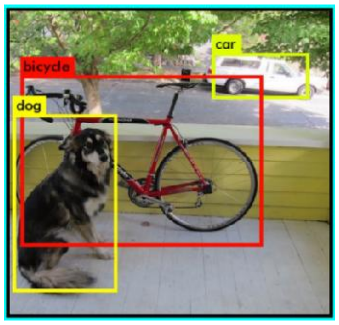


**Figure 1.4:** Object is identified

## II. CONCLUSION

As today's era is moving towards being digitalized and automated with a great speed, the youth want everything very easily and smart. Not only the youth but the people of all generation are finding it very easy to be smart effort. So, we thought of using this model.We have modelled a face mask detector using SSD architecture and transfer learning methods in neural networks. To train, validate and test the model, we used the dataset that consisted of 1916 masked faces images and 1919 unmasked faces images. These images were taken from various resources like Kaggle and RMFD datasets. The model was inferred on images and live video streams.

## REFERENCES

[1] Haddad, J., 2020. How I Built A Face Mask Detector For COVID-19 Using Pytorch Lightning, https://towardsdatascience.com/how-i-built-a-face-mask-detector-for-covid-19-usingpytorch-lightning-67eb3752fd61.

[2] Hussain Mujtaba, Jul 24, 2020 Real-Time Object detection using TensorFlow, www.mygreatlearning.com/blog/object-detection-using-tensorflow/

[3] Mustamo P. (2018)- "Object detection in sports: TensorFlow Object Detection API case study". He has developed and studied the TensorFlow Object Detection case deeply for making it applicable in sports region. The aim of his study was to explore the modern open source-based solutions for object detection in sports, in this case for detecting football players.

[4] http://jultika.oulu.fi/files/nbnfioulu-201802081173.pdfG.Eason,B Noble and I.N.Sneddon, "on certain integrals of Lipschitz Hankel type involving products of Bessel functions," Phil.Trans Roy Soc.London,Vol A247,pp.529-551,April 1995.

[5] Zhong-Qiu Zhao, Peng Zheng and Shou-Tao Xu are with the College of Computer Science and Information Engineering, Hefei University of Technology, China. Xindong Wu is with the School of Computing and Informatics, University of Louisiana at Lafayette, USA. (2017) - Here authors explained how actually object is identified and more about object detection and Deep Learning. https://ieeexplore.ieee.org/abstract/document/8627998/authors#authors.