

# Analysis of Object detection Method using Open CV – Python

<sup>1</sup>Nandan Verma, <sup>2</sup>Mohit Jauhari, <sup>3</sup>Mrs. Rashmi Mishra

<sup>1,2</sup>B.tech 4th Year, <sup>3</sup>Assistant Professor, ABES Engg. College, Ghaziabad, U.P, India.

**ABSTRACT** - Object detection is a very well-known computer technology connected with computer vision, image processing, data collection and character recognition that focuses on detection of any object or its instances of a certain class (such as humans, flowers, animals, number plates, vehicles etc.) presented in the form of digital images and videos. There are various applications on object detection that have been well researched including face detection & recognition, character recognition and prediction, and number plate detection. Object detection and recognition are used in very vast cases and scenarios including retrieval, surveillance, detection of overspeeding of vehicles and a lot more cases . In this research, various basic concepts used in object recognition and detection while making use of OpenCV library of python 3.8, increasing and improvising the efficiency & accuracy of object recognition and detection are presented.

**Keywords** - Object Detection, IOU, OpenCV, Python, Matlab.

## I. INTRODUCTION

Object detection [9] and recognition has become crucial in many fields of the society due to increasing concern about security and this technology provides a fast, economic and easy method of achieving it. This is not a new technique but there is still scope of improvement in order to achieve the targeted objective more efficiently and accurately. The main aim of studying and researching computer vision is to simulate the behavior & manner of human vision directly by using a computer and after that to develop the system which reduces efforts and also to find the most optimistic method to increase efficiency. In the research field of computer vision we try to perceive & represent the 3D information for all the present objects. The main purpose of analyzing the 3D and extracting information is to reconstruct a virtual aspect of it in 2D. Real life objects are 3D objects that are represented by 2D for reducing complexity. The purpose of object detection analysis is to accurately quantify and locate the number, size, position of the targeted object in the input image. Object detection is the basic concept used in tracking & recognition of objects, which affects the efficiency and accuracy of object recognition. A common object detection method known as color-based approach which detects the object based on their color values [4]. This method is used because of the method's strong adaptability & robustness. However, the speed detection needs more improvement, because it requires rigorous testing which can cover most of the windows by exhaustive search and have high computational complexity. Objects or images or videos which have a very big background or have too many colours or have a large number of other objects are very complex in nature and

challenging for image processing . The goal of this project is to identify objects placed over such a background using various techniques. The detection of the objects can be implemented in automation and robotics for plucking/choosing of the objects like apples, bananas from the corresponding tree using the image processing and computer vision techniques and it will be easier, faster and convenient to pluck/choose the apples and bananas rather than the plucking performed by any human. Intersection-over Union (IOU) is the standard performance measure which is commonly used. In a given object the similarities between the predicted region and the ground-truth region for an object present in found with the IOU performance measures and can be defined as the size of the intersection divided by the union of the different regions. For example, if any particular algorithm predicts each and every pixel of an image to be its background, the IOU performance measure can effectively penalize for it, as the intersection between the predicted and ground-truth regions would be zero because predicted and ground truth doesn't have anything similar between them, Hence it will give performance measure as zero. In python3.8 libraries such as openCv and NumPy are used to help in object detection .

## II. OBJECTIVE

The main objective of object detection is reconstructing Real life objects in 3D but to reduce the processing and for increasing the speed of object detection a visual aspect in 2D is created. The 2D image is scaled to a smaller image and converted into black and white image so that computation of image becomes easy. After that image is

converted into matrix and in each block of matrix it contains some info about the character of the image.

### 2.1 Concept:

The detection and recognition of objects in videos and images are the sub-domain of computer vision detection. various terms related to object detection:-

Edge matching:- count the number of edges overlapping. find edges of objects.

Divide and Conquer search:- All positions are to be considered as a set. The lower bound is determined at the most preferable position in the cell. The cell is cropped if the bound is too large. The process stops when a cell becomes small enough.

Gradient matching:- Comparing image gradient to determine change in intensity, texture etc.

## III. PROBLEMS IN EXISTING SYSTEM

### 3.1 Object Classification In Moving Object Detection:-

The classification can be done under various classes such as trees, animals, humans, objects etc. Tracking objects and analyzing their features is a key concept of object classification.

Shape based:- BLOB (known as Binary Large Objects which consists of connected pixels divided in groups) is the parameter for classification at each and every frame. Histograms are used to store these results.

Motion Based:- For classification of motion of the human body the non-rigid articulated human motion is considered as periodic motion (generally) Based on this useful clue, human motion can be distinguished from other objects motion.

Colour based:- Color-histogram based technique is implemented for detecting the fast moving automobiles in real world scenarios. Though color is not a standalone precise method for detecting and tracking of objects, but it takes minimal computation costs.

Texture Based:- It provides better accuracy, by using overlapping local contrast normalization but may require more time, which can be improved using some fast techniques and high computational costs.

## IV. LITERATURE SURVEY

### 4.1 VARIOUS OBJECT DETECTION ALGORITHMS IMPLEMENTED IN PYTHON

#### 4.1.1 Haar Like Features

#### 4.2 SSD(Single Shot MultiBOX Detector):

It is a method which is dependent on machine learning algorithms for object detection where we train a classifier from a lot of images. After that classifier is used in detecting targeted objects in an image. Firstly for training purposes algorithm needs to be trained with faces (positive images) and images without faces (negative images) to train a classifier to distinguish between these images and then extract features from the classifier. All features are not applied at once we divide these features into sets for different stages of the classifier and apply one by one. Discard the window in case it fails in the first stage. If it passes the stage then the process goes on. The window which clears through all the stages will be our desired region

#### 4.1.2 Circular Hough Transformation:-

This transformation was initially meant to find arbitrary shapes from an image. Later it was updated to find circular objects in low-contrast noisy images and referred as Circular Hough Transformation. CHT relies on three parameters, which require too much time for processing and memory and it increases the complexity to extract information from the image. For simplicity, CHT programs are provided with a constant value of radius or provided with a range of radius prior to running the application.

#### 4.1.3 Template Matching:-

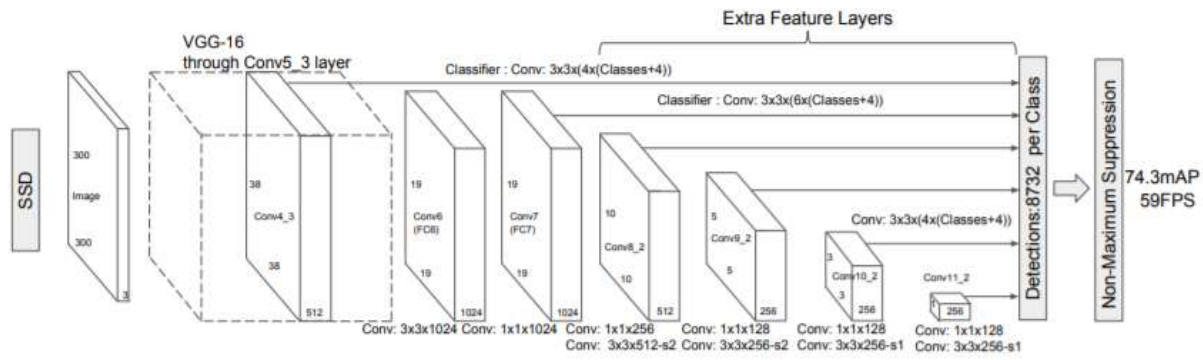
Template matching is a high-level machine vision technique to detect objects having similarity of patterns . This technique compares the base image with the template image or patch. If the image has strong features, the feature based approach may be used otherwise template-based approach is used.

#### 4.1.4 BLOB Matching:-

Binary Large Objects(BLOB) consists of a set of connected pixels. BLOB is a region in the image in which all points can be considered to be similar to each other. There are two methods of detection of differential method and local extreme method.

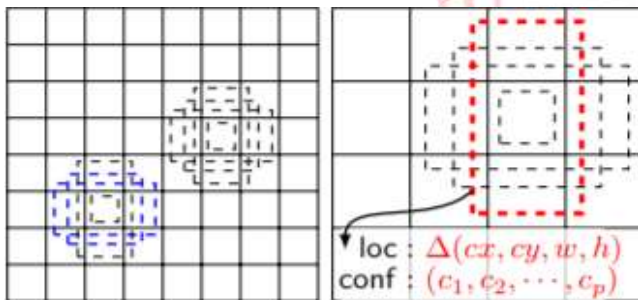
#### 4.1.5 Local Binary Pattern:-

The LBP feature vector is created in the following manner. It divides the window into cells. For each pixel in the cell, compare it with its 8 adjacent pixels. If the value of the pixel is more than its neighbor, write "0", else "1". This gives an 8 digit binary number which is converted to decimal for convenience. Aggregate histogram of all the cells is calculated and compared. Normalize the histogram. Merge histogram of all cells. This gives the feature vector of the entire window. This vector now can be used to classify the images.



SSD Discretize the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. The network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. SSD is easy to train compared to other algorithms. SSD is one of it's kind algorithms. only one algorithm(YOLO algorithm) can reduce computational cost but it's not better than SSD.

#### 4.2.1 Predict Object Position with SSD:-



Two parameters are required to train SSD i.e. a base image and ground truth with boxes. Evaluation is done on small groups of boxes having default boxes with different aspect ratio and scale and in a convolutional fashion . The major contribution to efficiency of SSD is of default boxes on different layers of output and different scales. To measure the advantage gained, we progressively remove layers and compare results. For a fair comparison, every time we remove a layer, we adjust the default box tiling to keep the total number of boxes similar to the original (8732). This is done by stacking more scales of boxes on remaining layers and adjusting scales of boxes if needed. We do not exhaustively optimize the tiling for each setting.

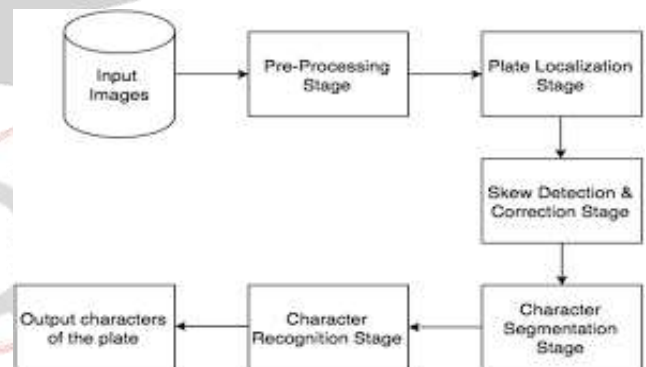
#### 4.2.2 Scaling Problem using SSD:-

Firstly we need to understand what is a scale problem. In other algorithms if an object size is too large then it will probably not be able to detect that image of the object as an object. but in SSD the image goes to many layers so that the objects can be easily downsized so that each and every algorithm can easily identify it.

#### 4.2.3 Libraries helpful in implementing SSD:-

Torch:- It contains dynamic graphs by which we can very efficiently compute the Gradients of composition functions( because in a neural network we have several layers and each layer have some function of previous layers and previous layers have function of previous previous layers ) which will be helpful in determining weights of function.

### V. FLOW CHART



**Input images :-** Input images can be frames from a captured real time video, image, or a video as an input. Basically in input images we find our targeted object.

**Pre-Processing Stage :-** In this stage a 2D aspect is created from a real life 3D image so that complexity can be reduced and processing can be made faster.

Firstly grayscale conversion is done on colour of 24 bit by using the following formula. This value is in 8 bit format.

$$\text{grayscale}(x, y) = 0.59 * \text{Red}(x, y) + 0.30 * \text{Green}(x, y) + 0.11 * \text{Blue}(x, y)$$

Secondly Median filtering is done, generally this filter is non-linear. This value removes noises from the image.

Lastly Contrast enhancement is done and the technique used in it is known as equalization technique.

**Plate Localization stage :-** In this stage we extract our targeted object from the processed input image. There are many methods for it like edge matching, pattern matching etc which are already discussed above.

**Skew detection and correction stage :-** Skew means tilted pixels or blurry pixels. In this step these types of pixels are



identified and corrected so that these pixels don't create problems in recognition of targeted objects. Some of the techniques used in this stage are Hough transform, projection profile etc.

| Detector             | 10    | 31    | 50    | 65    | 78      | 95    | 167   |
|----------------------|-------|-------|-------|-------|---------|-------|-------|
| Viola-Jones          | 76.1% | 88.4% | 91.4% | 92.0% | 92.1%   | 92.9% | 93.9% |
| Viola-Jones (voting) | 81.1% | 89.7% | 92.1% | 93.1% | 93.1%   | 93.2% | 93.7% |
| Rowley-Baluja-Kanade | 83.2% | 86.0% | -     | -     | -       | 89.2% | 90.1% |
| Schneiderman-Kanade  | -     | -     | -     | 94.4% | -       | -     | -     |
| Roth-Yang-Ahija      | -     | -     | -     | -     | (94.8%) | -     | -     |

**Character segmentation stage :-** In this method characters are detected by using edge matching (generally) technique.

**Character recognition stage :-** In this stage all the detected characters are processed and recognised using character recognition stage.

**Output character of plate :-** In this stage all the characters are recognised and sent to the server for further processing.

## VI. PROPOSED METHOD

**6.1 Basetransform( imported from data library ) class:-** Do the required transformation that will convert a simple input image into a neural network compatible network. As we know neural networks have a certain set of format for images and base transform classes are used to transform input images into that neural network format image.

**6.2 Build\_ssd( imported from SSD package ):-** It's used as a constructor of SSD neural networks. This will include all the internal architecture of algorithms.

**6.3 Imageio :-** Used to capture frames from the videos. It helps in capturing frames from high frame videos. It detects each and every frame of videos which will certainly help in detecting an object more accurately and more fastely.

## VII. CONCLUSION AND FUTURE WORK

Multiple scales and locations are checked of an image for the final detector. The scaling is not done by scaling of the image, it is done by the detector. It is a precise method because scale and cost are needed not to be changed. Factor of 1.25 is used as it gives highest accuracy.

We have an approach for object detection which maximises performances and minimises computation time and achieves high detection accuracy. The approach was used to construct an object detection system which is approximately 1.5x faster than any previous approach.

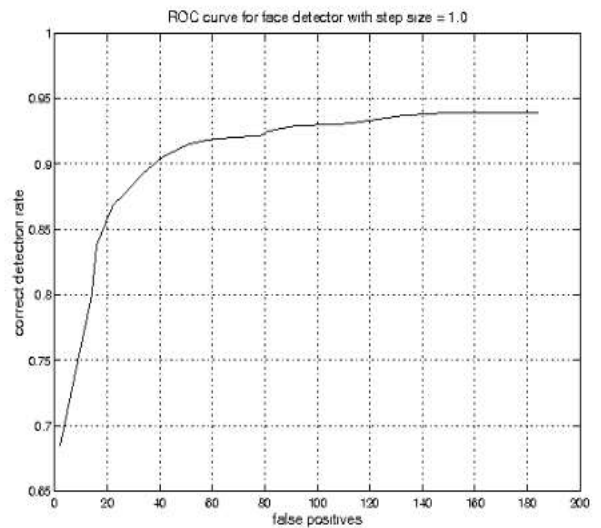
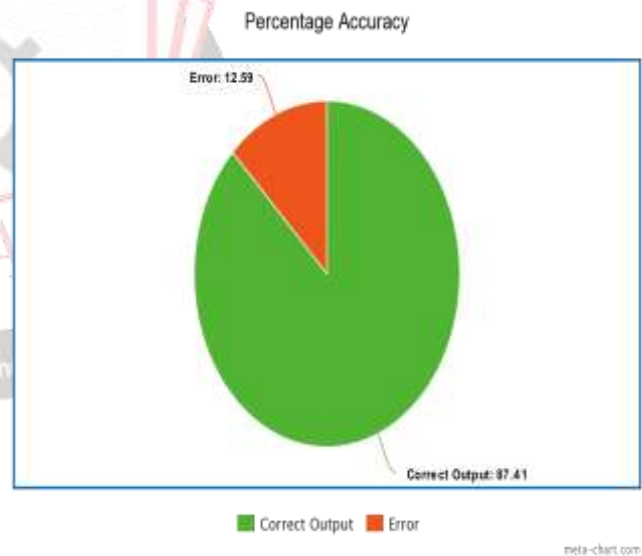


Figure 6: ROC curve for our face detector on the MIT+CMU test set. The detector was run using a step size of 1.0 and starting scale of 1.0 (75,081,800 sub-windows scanned).

The below mentioned meta chart is created after testing the program with 100 images and it clearly shows increased accuracy of the program.



## REFERENCES

- [1] (2017, January 17). Object Detection [Online]. Available: [http://en.m.wikipedia.org/wiki/Object\\_detection](http://en.m.wikipedia.org/wiki/Object_detection)
- [2] Opencv.org, "About OpenCV", 2017 [Online]. Available: <http://www.opencv.org/about>
- [3] Numpy.org, 2017, [Online]. Available: <http://www.numpy.org>
- [4] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J., "Hypercolumns for object segmentation and fine-grained localization", CVPR., 2015.

[5] Wei Liu 1, Dragomir Anguelov 2 , Dumitru Erhan 3 , Christian Szegedy 3 , Scott Reed 4 , Cheng-Yang Fu 1 , Alexander C. Berg 1, “SSD: Single Shot MultiBox Detector” .

[6] Paul Viola and Michael Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features”, IEEE, Conf. on computer vision and pattern recognition, December 2001.

[7] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS. (2010)

[8] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. In: ICLR. (2015).

[9] Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: ECCV 2012. (2012).

[10] Uijlings, J.R., van de Sande, K.E., Gevers, T., Smueulders, A.W.: Selective search for object recognition IJCV (2013).

