International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 04, 2020
ISSN: 1475-7192

Object Detection in a Video

[1]V. Chandra Tej, [2]T. Gayathri, [3]K. Abhilash, [4]Y. Sandeep

Abstract - *Object Detection and classification are the successful applications of image analysis and algorithm-based understanding that can be applied to an image or on a video related to computer vision in biometric research. It has enormous application value and market potential which involves deep learning and OpenCV techniques. OpenCV and python programming development is used for designing real face recognition systems. It provides access to machine-based in-depth analysis of a person's facial features. The main algorithm used for detecting the identity is the SSD algorithm. All in all, the machine successfully identifies any entity by using the combination of SSD and MobileNets. The detection module, training module and recognition module are the three major modules that are involved in the facial recognition and detection process. The main concept of our project is to train the model with some datasets with various identities and then test the model. The model will detect and recognize the identities of faces which are previously trained. This will result in image classification along with the localization. It will help to detect whether the person or the object which we a researching for, is present or not in that image or a video.*

Keywords: *Object Detection, Computer Vision, Single Shot Detector (SSD), OpenCV.*

I. INTRODUCTION

Object detection is the important topic in the area of computer vision. Object detection has got its applications in a wide range of industries, with the use cases ranging from many aspects including personal safety, productivity and many other professional work purposes. Object detection and recognition are applicable in different areas of computer vision, in conjunction with image analysis, automated vehicle registration plate recognition, privacy and security, video surveillance, character recognition, educational field, medical field, agricultural sector and many more. Object Detection is the process of categorizing an image and identifying where an object resides in a particular image. In order to obtain the bounding box i.e. (x, y)-coordinates of an image, Frame detection is used which is considered as the regression problem. That makes it easy to understand. There are many different algorithms that perform the process of detecting the objects. One of the methods of Object Detection is the Single Shot Detection [1].

Object detection (OD) system finds objects within the planet by making use of the object models which is thought a priori. This task is relatively difficult to perform for the machines as compared to Humans who perform object detection very effortlessly and instantaneously. In this Project, various techniques and approaches which are accustomed to detecting the objects in images and videos are reviewed. Basically, an object detection system is often described as the one which depicts the essential stages that are involved within the process of object detection. The general input to the object detection system can often be a picture or a scene just in the case of videos [2].

The general methodology of an object detection system comprises of two main phases namely: the learning phase and therefore the testing phase that shows the conventional working of the object detection

Received: 15 Jan 2020 | Revised: 5 Feb 2020| Accepted: 30 Mar 2020

system. The learning phase is especially meant for the classifier so that it recognizes the objects present within the image that's given as an input to the system. Learning phases are often further classified as learning through training and learning through validation. Learning through training comprises mainly the block where a correct learning scheme is defined, it might be part-based or patch-based, etc. The object template block then makes use of the learning that were done previously to represent the objects with different representations like that of the representation using the histogram, representation using random forest., etc. Whereas on the opposite hand, learning through validation blocks do not require any kind of training as they are validated beforehand itself. Therefore, after preprocessing the image, directly template matching is finished which produces the features of an object within the image. The purpose of the testing phase is to determine whether an object is present within the image that's given to the system as input and if yes then to which object class it belongs to. Here the image is looked for an object by various searching techniques just like the window technique, in line with the output of the searching mechanism, and a choice is created on the object class[3].

1.1 Problem Statement

In today's world, where data goes on increasing enormously and time becomes more and more costlier, who has the time to watch a full-length video while the data relevant to one is very less compared to video length? Consider a policeman who is investigating a case, does he have the time to watch the whole CCTV footage to find the culprit? So, we tried to ease these efforts by object detection in video using machine learning algorithms through which a lot of valuable time and effort can be saved. The major distribution of the load is contributed by two factors namely computation by testing the trained integral datasets and validation of the target images. It is mainly used in the applications of criminal detection, security surveillance system and many similar purposes. Deep learning increases the speed and accuracy of face recognition and preprocessing is done to remove noises, providing an overview study on face recognition for heterogeneous face matching.

II. SYSTEM FLOW

This section describes the flow of the paper. Major steps are explained in detail. Facts and figures related to the data has also been discussed.

2.1 Data

A data set may be a collection of information. In other words, an information set corresponds to the contents of one database table, or one statistical data matrix, where every column of the table represents a specific variable, and every row corresponds to a given member of the information set in question.

COCO stands for Common object in context. COCO datasets are the large-scale object detection, classification, and captioning dataset. Common Object in Context (COCO) is a large-scale image dataset created for object detection, segmentation, and person key-points detection. Its dataset contains 330,000 images, 1.5 million object instances along with greater than 200,000 labels. All the objects are classified into 80 categories. The COCO dataset is as shown in the below Fig-1.

Received: 15 Jan 2020 | Revised: 5 Feb 2020| Accepted: 30 Mar 2020

Fig-1: COCO Dataset

2.1.1 Methodology

The Object detection algorithms use features which may be extracted to acknowledge an object. This model is incredibly simple and simple to implement. Here, object detection could be a single regression problem which detects directly from bounding box coordinates and sophistication probability.

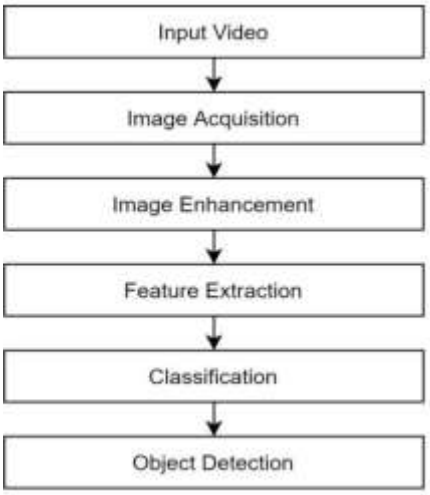

Fig-2: Flowchart of the module

The actual working strategy of our proposed work is done in two steps. They are as follows:

2.1.1.1 Pre-processing: It is the rock bottom level of abstraction. The method of pre-processing improves the image intensity by suppressing the unwanted features or enhancing them for further processing. It resizes the image size to 448*448 and normalizes the contrast and brightness effects. The image is additionally cropped and resized so feature extraction is performed easily.

Received: 15 Jan 2020 | Revised: 5 Feb 2020| Accepted: 30 Mar 2020

2.1.1.2 Feature Extraction: Its main motive is to simplify the image by considering only the important information and leaving out the additional information which isn't necessary for recognition. It uses the tactic of edge detection which may only retain the essential information. It indicates the decrease in the portion of an image as a feature vector. This approach is used especially when the dimensions of the image are very large.

2.1.2 Process Flow

Initially, the model receives any sample video from the user, and along with the video input, the model requests for the desired keyword of the object which has to be detected by it. For this model, the video is accepted only in the format of .mp4. Upon considering the input video, the model starts reading the frames of the video. Then it extracts all the features of the object which was mentioned by the user earlier. These features of the desired object are then sent to the database where the pre-trained datasets are present. They are compared with the features of the trained datasets. If the model finds the object that is matched with maximum features, it will be sent to the class box. A class box then identifies the class of the object from the dataset and finally returns the object along with its class to the user. The methodology is shown in the below Fig-3.

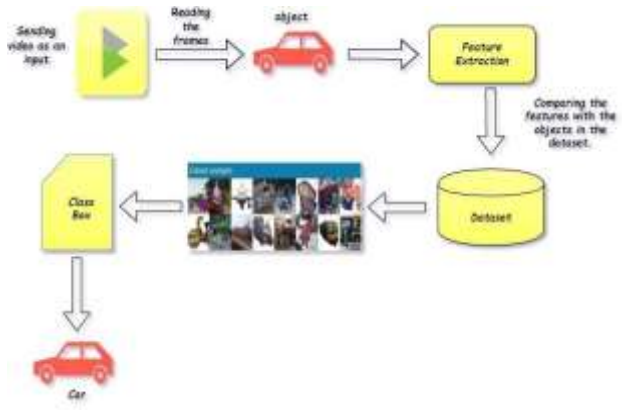

Fig-3: Process Flow

2.2 PROPOSED WORK

In the proposed model, the algorithm trains the model regarding how to accept input from the users. Along with this, the model also requests the users for a specified keyword that represents the object which has to be identified. And thereby, the model returns the output with the desired object which the user requests for.

2.2.1 System Architecture

The system architecture diagram is shown in the Fig-4.

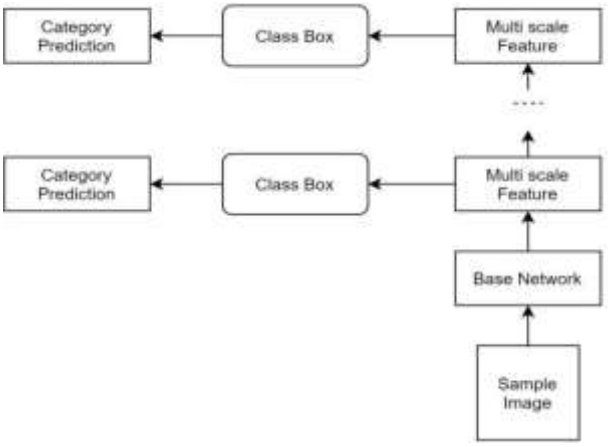

Fig-4: System Architecture

The above Fig-4 shows the planning of an SSD model. The main components of the model are a base network block and many other multi scale feature blocks connected. Here, the network block which is at the bottom is used to get the features of actual images, and generally gets the shape of a deep convolutional neural network. More anchor boxes are generated that support this feature map, and thereby allows us to detect smaller objects. This step is followed by each multi scale feature block which reduces the height and width of the feature map that is provided by the previous layer. Then the blocks use each component within the feature map in order to expand the receptive field on the input image. If the feature block is closer to the highest features, the receptive field of every element within the feature map will be larger and it suits better in order to detect larger objects. The SSD object detection comprises two parts. The primary one is to extract feature maps and also the other is to use the convolution filters to detect and classify the objects. 3.2 Description of Algorithm

Single Shot Detector is intended for object detection in real-time. Faster R-CNN uses a section proposal network to make boundary boxes and utilizes those boxes to classify objects. While it is considered the start-of-the-art in accuracy, the entire process runs at 7 frames per second. Far below what a real-time operation needs. SSD races this process by eliminating the requirement of the region proposal network. To recover the drop by accuracy, SSD applies some improvements including multi-scale features and default boxes. These improvements will make SSD to meet the accuracy of the Faster R-CNN by using low resolution images, which will further push the speed higher.

Single Shot detector discretizes the output space of bounding boxes into a collection of default boxes over the various aspect ratios and scales per feature map location. During the prediction time, the network generates scores for the presence of every object category in each default box and produces adjustments to the box to raise the article shape. Additionally, the network will combine the predictions that are made by multiple feature maps with different resolutions in order to handle the objects of different sizes.

SSD is a simple relative to methods that require object proposals because it eliminates proposal generation and subsequent pixel or feature re-sampling stages and encapsulates all computation during one network. This makes SSD easy to coach and easy to integrate into systems that need a detection component. SSD uses a tiny low convolutional filter to predict object categories and data sets in bounding box locations, using separate predictors (filters) for various ratio detections, and applying these predictors to multiple feature maps from the succeeding stages of a network so as to perform detection at different scales. With these improvements especially by using several layers for the purpose of prediction at different scales, it is possible to achieve higher accuracy rates by employing relatively lower resolution input, and thereby improving the speed of detection. These design features result in simple end-to-end training and high accuracy, even on low resolution input images, further improving the speed versus the accuracy trade-off.

The following are the sequence of steps involved within the working of SSD:

1. Get the desired frames from video and save them as JPG file.
2. Initialize the list of sophistication labels our SSD network was trained to detect.
3. Collection of bounding box colors for every class.
4. Grab the frame dimensions and convert it to a blob (Binary Large Object).
5. Pass the blob through the network and procure the detections and predictions.

2.3 Network Architecture

Initially, the input whichever the user sends as an input will be passed by the input tensor to the expansion layers. In the expansion layer, all the improvisations will be made to the input video by improving the size of the pixel and the illumination levels. Thereby the pixels get enhanced and therefore will be ready to pass on to the next layer. The succeeding layer to the expansion layer is the Depth wise layer. Before sending the data to this layer, the model will uncompress the data, filter it and then proceeds. In the depth wise layer, the model will refine the data more precisely by dividing the data layers into many number of layers in order to examine each pixel. 31 After dividing them into more layers, they will filter the data. If there is any noise in the data or any low illumination content, it will be filtered. If there is more unobservable content, the data is divided into several layers called convolution layers. After this, the data is sent to the protection layer. In this layer, the data with useful content will be saved from being added with noise and other stuff. Finally, this data is compressed back and will be sent to the outside tensor. All the pixels will be back to their original sizes and get back to default. All the formed layers will be compressed into a single layer and then shows the output. Then some extra convolution layers can be added, that will help in dealing with the bigger objects. The principle of SSD architecture can be used with any deep network base model. The network architecture is shown in the below Fig-5.

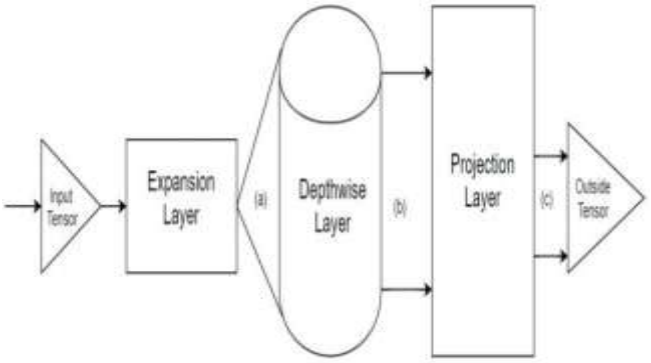

Fig-5: Network Architecture

The sub-components that are mentioned in the diagram are as follows:

(a) Uncompressing/unzipping the data

(b) Filter the data

(c) Compress the data

III. RESULTS AND OBSERVATIONS

In general, no machine is hundred percent accurate enough for application. Here, several instances are taken to estimate the accuracy of the machine in identifying the objects in the video. TensorFlow in python is used to make the work easier by converting required data into serialized format. It is observed that the machine yields result at a faster rate.

Upon running the model, the model requests for the execution of the algorithm followed by the video path and the keyword for detecting the object. The output terminal after running the model in shown in the below Fig-6.

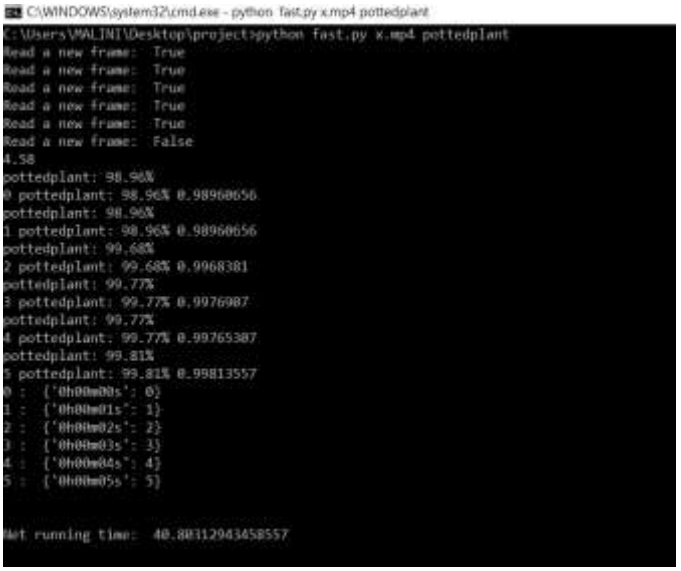

Fig-6: Observation from the terminal for Potted Plant Since, the desired keyword that has been mentioned by the user is the potted plant, the model first reads the frames of the video. And then the model classifies the frames in which the plant encounters. All the time frames during which the plant appears in the video will be recorded and the snapshots of the respective time frames will be taken and saved in the destined location with the names of the detected sequential frame numbers. The rate of accuracy in detecting the object as well as the net running time for performing the entire action are also mentioned in the output terminal. Fig-7 shows the snapshots of the detected potted plant that appears in different frames over the entire video.

Fig-7: Creation of frames for Potted Plant Similarly, the machine is fed with another video

where the user sends person as the keyword which means that the user wants the machine to identify the person who appears in the video along with the time frame. The output terminal is shown in the Fig-8.

Received: 15 Jan 2020 | Revised: 5 Feb 2020| Accepted: 30 Mar 2020

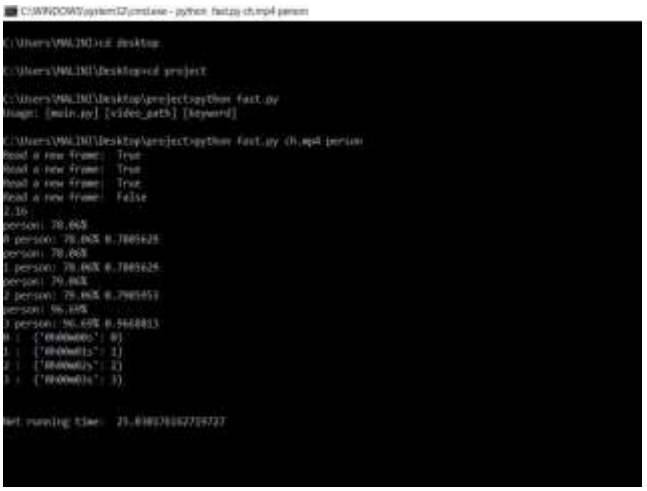

Fig-8: Observation from the terminal for Person Here, the person appears in the above-mentioned

time fraction. And these snapshots will be saved as per the above-mentioned frame numbers in the destined address. These frames are shown in the Fig-9.

Fig-9: Creation of frames for Person

3.1 Performance Analysis

Since the proposed method in this paper yields results with a great speed and accuracy, it is necessary to evaluate the performance of the model. In order to calculate the performance, Precision and Recall are calculated.

Precision is the total number of true positives divided by the sum of true positive and false positive.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall is calculated as the total number of true positives that is divided by the sum of true positives and false negatives.

$$\text{Recall} = \frac{TP}{TP+FN}$$

Received: 15 Jan 2020 | Revised: 5 Feb 2020| Accepted: 30 Mar 2020

Table-1: Comparison of SSD testing between an image and a video

Input	Image	Video
Accuracy	High	Moderate
Time	0.17 ~ 0.23 (second / per frame)	0.1830 ~ 0.237 (second / per frame)
Speed	Fast	Fast

The above Table-1 illustrates the performance testing of SSD over different inputs. The average precision for all the object categories is reported in the following Table-2. The highest accuracy for the COCO dataset was found to be 0.924637. The current state-of-the-art best mAP value is reported to be 0.839.

Table-2: Average precision of class objects

Class	**Average Precision**
Motor Bike	0.7241
Bottle	0.8127
Bird	0.7925
Cat	0.8554
Chair	0.9091
Aeroplane	0.7245
Person	0.8978
TV Monitor	0.7154
Sofa	0.7598
Potted plant	0.9458
Car	0.9245
Bus	0.8154

IV. CONCLUSION

Single shot detector algorithm along with the Tensor flow is implemented in this project. When considering time as the pivot constraint, SSD is the best applicable algorithm. The model which is developed using the SSD implements reinforcement learning. Even though the model shows errors in the beginning of

its learning, the machine learns over and over with the training data. Hence based on the results, it is observed that single shot detector algorithm is fast but not an accurate one because it aims to predict only a few classes but not all of them. The performance of this algorithm is efficient when applied to static video visuals and helps in reducing the time taken to process the video. Certain video stabilization algorithms can be applied to the dynamic camera visuals for obtaining optimal results. The idea of this model can be taken to a higher level by designing and developing sophisticated visually embedded systems. The main objective of this idea is to assist the human operators to detect and analyze unusual events in the videos and thereby responding rapidly to implement the counter measures to avoid the collateral damage.

REFERENCES

1. https://honingds.com/blog/ssd-single-shot-object- detection-mobilenet-opencv/
2. A real time object detection algorithm for a video, Shengyu Lua, Beizhan Wanga, Hongji Wanga, Lihao Chenb, Ma Linjiana, Xiaoyan Zhangc, 19 july 2019
3. A review and an approach for object detection in an image, Kartik Umesh Sharma and Nileshsingh V. Thakur, january,2017
4. [https://www.researchgate.net/publication/312037041_A_review_and_an_approach_for_object_detectio n_i n_images]
5. A Review of Detection and Tracking of Object from Image and Video Sequences, Mukesh Tiwari, Dr. RakeshSinghai, [https://ripublication.com/ijcir17/ijcirv13n5_07.pdf]
6. Object Detection: Current and Future Directionns, Rodrigo Verschaeand Javier Ruiz-del-Solar, December.