Visual Object Detection and Tracking using YOLO and SORT

Akansha Bathija
M.Tech Student, Dept of Computer Engineering
K J Somaiya College of Engineering
Mumbai, Maharashtra, India

Prof. Grishma Sharma
Assistant Professor, Dept of Computer Engineering
K J Somaiya College of Engineering
Mumbai, Maharashtra, India

***Abstract*— Over the past two decades, computer vision has received a great deal of coverage. Visual object tracking is one of the most important areas of computer vision. Tracking objects is the process of tracking over time a moving object (or several objects). The purpose of visual object tracking in consecutive video frames is to detect or connect target objects. In this paper, we present analysis of tracking-by-detection approach which include detection by YOLO and tracking by SORT algorithm. This paper has information about custom image dataset being trained for 6 specific classes using YOLO and this model is being used in videos for tracking by SORT algorithm. Recognizing a vehicle or pedestrian in an ongoing video is helpful for traffic analysis. The goal of this paper is for analysis and knowledge of the domain.**

Keywords:- Tracking-by-detection, You Only Look Once (YOLO), Simple Online and Realtime Tracking (SORT), visual tracking.

I. INTRODUCTION

Video tracking is aimed at associating target objects in consecutive video frames. When the objects move quickly relative to the frame rate, the association can be particularly difficult. Another situation that increases the problem's complexity is when the object being tracked changes orientation over time.

An algorithm analyses sequential video frames and outputs target motion between frames to execute video tracking. Various algorithms exist, each with strengths and weaknesses. When selecting which algorithm to use, it is essential to consider the planned use. A visual tracking scheme has two main parts: target representation and location, as well as filtering and association of information.

Object Tracking [4] is a computer vision job that consists of extracting an object's movement from a series of pictures estimating its trajectory. Object detection and tracking; Detection - A detection algorithm asks the question: is something there? Tracking - A tracking algorithm wants to know where something is headed.

Visual tracking is a difficult job in computer vision owing to target deformations, variations in illumination, changes in scale, rapid and abrupt movement, partial occlusions, movement blur, deformation of objects and background clutters. Recent progress in object detection techniques has resulted in a number of tracking-by-detection approaches being developed [4]. The scope of this project is to detect and track objects (Vehicles and Pedestrians) in a video by using tracking-by-detection approach.

The rest of this paper is organized as follows: Sec II presents Literature Survey, Sec III presents Methodology followed, Sec IV presents carried out Experimental Results, Sec V present Concluding remarks.

II. LITERATURE SURVEY

Neural networks are a group of algorithms designed to recognize patterns, modeled loosely after the human brain. They view sensory data by means of a form of raw input device perception, marking or clustering. Detect faces, identify people in pictures, recognize facial expressions. Identify objects in pictures or videos. The identification of similarities is clustering and grouping. Deep learning may create associations between, say, pixels in a picture and a person's name with classification. Neural Networks is currently one of the most common algorithms for machine learning. The fact that neural networks outperform other algorithms in accuracy and speed has been clearly proved over time. With different variants such as CNN (Convolutional neural network) [7], RNN (Recurrent Neural Networks), Deep Learning, etc. Deep-learning networks are differentiated by their depth from the more common single-hidden-layer neural networks; that is, the number of node layers that information can move through in a pattern recognition multi-stage system.

Most researchers use techniques of deep learning to extract qualified deep characteristics. In many challenging tasks, which historically rely on hand-crafted features such as location, monitoring, identification, human crowd detection, self-stabilization, obstacle and crash avoidance, perception of forest or mountain trails, and object tracking, they have exhibited magnificent results. With the rise of autonomous vehicles, smart video surveillance, facial recognition and numerous people-counting applications, the demand for quick and accurate object detection systems is increasing. Such systems require not only the identification and classification of each object in an image, but also the location of each object by drawing around the correct bounding box. This makes identification of objects a much harder task than their conventional counterpart in computer vision, the recognition of images [3][4].

Object Detection [1][4] is modeled as a classification problem where at all possible locations we take gaps of fixed sizes from the input object to feed these patches into an image classifier. Every window is fed to the classifier which determines the object's class in the window. Therefore, we know the category and location of the image objects [6].

CNNs consist of neurons with learning weights and biases such as neural networks. Each neuron receives multiple

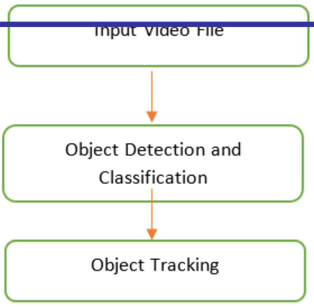

Fig -1: Basic flow diagram for multiple object tracking

inputs, takes over a weighted sum, passes it through an activation function, and provides an output response.

These are often used to recognize patterns like edges (vertical / horizontal), shapes, colours, and textures in object detection. Example of a CNN architecture: [INPUT — CONV — RELU — POOL — FC]. There are quite a few algorithms for object detection which have been developed over the years.

Inspired by the ground-breaking image classification results obtained by CNN [7] and the success of selective search in regional proposal for hand-crafted apps, Girshick et al. were among the first to explore CNN for generic object detection and developed Region-based Convolutional Neural Networks(R-CNN) [4], which combines AlexNet [8] with regional proposal system selective search.

Since R-CNN's proposal[4], many improved models have been proposed, including Fast R-CNN, which jointly optimizes classification and bounding box regression tasks, Faster R-CNN[7], which allows an additional subnetwork to produce local proposals, and YOLO[1], which performs object detection by means of a fixed grid regression. All bring different degrees of improvements in detection efficiency over the primary R-CNN and make object recognition more feasible in real-time and accuracy [6].

YOLO is one of the fastest algorithms out there to detect objects. Although it is no longer the most accurate algorithm for object detection, when you need real-time detection without losing too much precision, it is a very good choice. YOLO uses a single convolutional network to predict several bounding boxes and category probabilities for these boxes at the same time [1].

For computer vision, object tracking is an important field. Object trackers can be categorized into TBD (Tracking by Detection) and DFT (Detection-Free Tracking) and online and offline trackers can also be separated by whether potential frames are used [4]. This includes the method of tracking an object through a series of frames that could be a human, a ball or a vehicle. This begins with identifying all possible detections in a frame in object tracking and assigning them an ID. For the following images, the current object ID is attempted to be carried forward. If the object moves away from the image, the ID will be removed. If a new object appears, a fresh ID will begin. This is a challenging task because objects may look similar, forcing the template to change IDs, an object may become occluded as when an item or entity is concealed behind something, or some objects may disappear and reappear later.

III. METHODOLOGY

The followed method is for visual object tracking in videos which consists of Object detection and tracking using YOLO [1] and SORT [2]. The whole implementation is done in python.

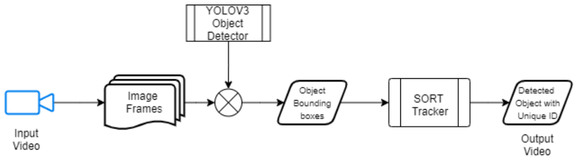

Fig -2: Flowchart representation for

Visual Object Detection and Tracking

Custom dataset [6] consisting 800 images having 6 classes: Person, Car, Truck, Bus, Bicycle and Motorbike was used for training YOLOv3 which was already pre-trained for MS COCO [7] dataset consisting of 80 classes. Model was trained for 320 epochs using Google Colab [14]. All the 800 Images were annotated manually using LabelImg tool [12]. Dataset was trained with help of PyTorch library [5].

Image were labeled in the YOLO format. Total of 200 images were used for validation. All the images have a specified .txt associated to them after annotation were done in the format of YOLO. Images can be labelled in PascalVOC version as well. Below is the snapshot of Google Colab using which the custom dataset was trained

.

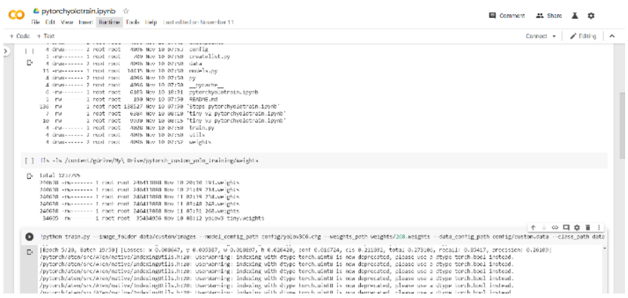

Fig -3: Snapshot of Google Colab

Once labelling is completed, images and annotations/ labels are placed in a directory and all this information is either passed as parameters or code inside the main file and with the help of PyTorch library, YOLOv3 is trained for our custom dataset with the number of epoch decided depending on the size of dataset and trying to achieve maximum accuracy. A weights file is the final output after training which will be used for object detection in our model.

An Input video is passed through the system and then at first total number of frames are extracted and forwarded to object detector which is YOLO in this case. Being an object detector YOLO generated bounding boxes with class ID and confidence for each bounding box [1]. As we are following tracking-by-detection approach, these detections are then forwarded to our tracker which is SORT [2].

SORT is an online tracker which works on the principle of tracking by detection. This method uses a strong detector to detect objects and the Hungarian algorithm and Kalman filter are used to track objects. SORT tracks each detection by assigning unique id to each bounding box, as soon an object is lost due to occlusion, wrong identification, etc. tracker assigns a new ID and start tracking the newfound object.

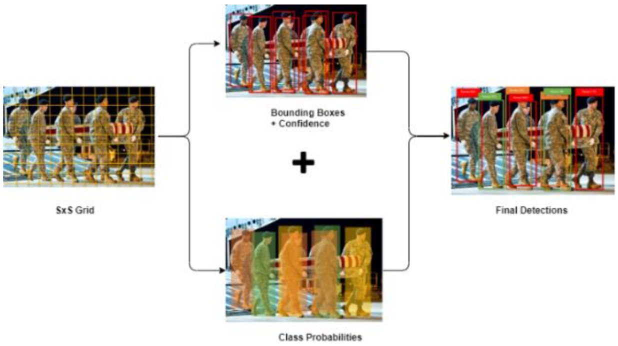

Fig -4: YOLOv3 on custom database

IV. EXPERIMENTAL RESULTS

On several videos, the proposed system is tested. The experiment is divided into two sections, the identification and tracking of objects. The project design is python-based and evaluated on five different video sequences and runs with strong FPS.

Once training is completed the weights files in used for object detection in videos. Input video file is broken down into total number of frames and passes each image to our trained object detector and once detection is done bounding box information is passed onto SORT tracking algorithm and object tracking performed.

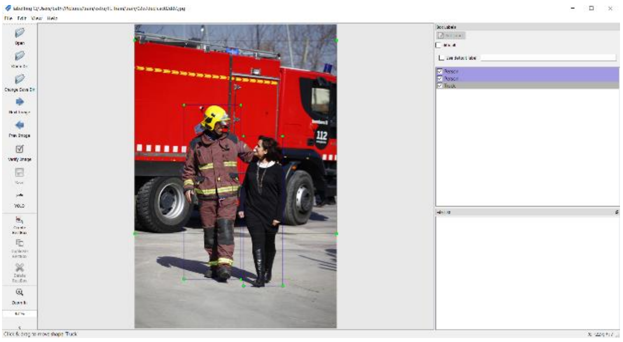

Fig -5: Labelling images using LabelImg tool

We tested our object detector for few images to check how well it was trained and the following precision and recall graph was obtained.

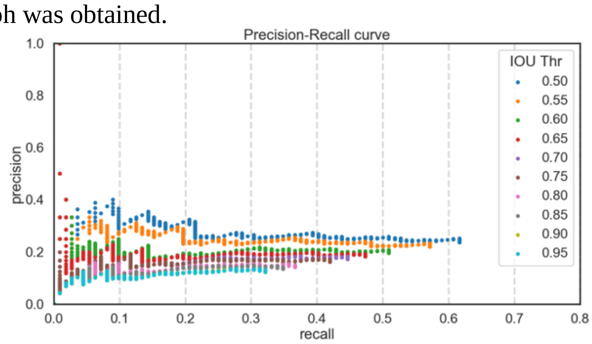

Fig -6: Precision Recall graph for custom dataset

The quantitative analysis is performed these parameters True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

- TP: Where the model correctly predicts a positive object class
- FP: Where the model incorrectly predicts a positive object class
- FN: Where model incorrectly predicts a negative object class
- TN: Where the model correctly predicts a negative object class

Here, TP = a, TN = b, FP = c, FN = d.

Accuracy = (a + b)/(Total)
Precision = (a) / (a + c)
Recall = (a) / (a + d)

TABLE I. QUANTITATIVE ANALYSIS OF THE PROPOSED SYSTEM

Video#	Total Frames	Accuracy	Precision	Recall
1	812	0.794	0.843	0.934
2	930	0.851	0.958	0.93
3	1160	0.75	0.9	0.9
4	835	0.781	0.833	0.892
5	590	0.444	0.5	0.889

While performing visual object detection and tracking task, video is broken down into frames and each frame as well as a video output is saved with detection and tracking information obtained for each input video after using YOLO and SORT for object detection and tracking respectively.

Below are output screens of videos tested, which provide output as bounding boxes with class name and confidence scores as well as unique ID which is assigned to them by SORT.

Fig -7: Qualitative analysis of system for traffic video

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

Fig -8: Qualitative analysis of system for pedestrian video

V. CONCLUSIONS

In this paper, visual object tracking is done on videos by training detector for custom dataset consisting of 800 images for specific 6 classes. The moving object detection is done using YOLO detector and SORT tracker for tracking the objects in consecutive frames. Accuracy and precision can be worked upon by training the system for more epochs and fine tuning while training the detector. Performance of SORT tracker totally depends upon the detectors performance as it is a tracker which follows tracking by detection approach.

For Future work, the system can be trained for more classes (more types of objects) as it can be used for different domains of videos and different objects can be detected and tracked. Our system is limited to pedestrian and vehicles, this can be expanded to multiple objects or can be reduced for a specific object with different number of dataset images. Different types of object detectors (For eg: YOLOv1, YOLOv2, YOLOv3, R-CNN, SSD, etc) and object trackers (For eg: Deep SORT, Centroid, IOU tracker, CNN + LSTM, etc) can be implemented and tried for proposed object detection and tracking and different set of results will be obtained which can be studied for analysis.

VI. REFERENCES

[1] D J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.
[2] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing, Phoenix, AZ, 2016, pp. 3464-3468.
[3] S. Moon, J. Lee, D. Nam, H. Kim and W. Kim, "A comparative study on multi-object tracking methods for sports events", 2017 19th International Conference on Advanced Communication Technology (ICACT), 2017.
[4] A. Mekonnen and F. Lerasle, "Comparative Evaluations of Selected Tracking-by-Detection Approaches," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 4, pp. 996-1010, 2019.
[5] S. Nayak, "Training YOLOv3: Deep Learning based Custom Object Detector | Learn OpenCV," Learnopencv.com, 2019. [Online]. Available: https://www.learnopencv.com/training-yolov3-deep-learning-based-custom-object-detector/.
[6] S. Shinde, A. Kothari and V. Gupta, "YOLO based Human Action Recognition and Localization," Procedia Computer Science, vol. 133, pp. 831-838, 2018.
[7] L. Liu et al., "Deep Learning for Generic Object Detection: A Survey," International Journal of Computer Vision, 2019.
[8] Zhao, Z.Q., Zheng, P., Xu, S.t., Wu, X., “Object Detection with Deep Learning: A Review,” arXiv:1807.05511v1 [cs.CV] 15 Jul 2018
[9] "Object Tracking in Deep Learning - MissingLink.ai," MissingLink.ai, 2019. [Online]. Available: https://missinglink.ai/guides/computer-vision/object-tracking-deep-learning/.
[10] P. SHARMA, "A Step-by-Step Introduction to the Basic Object Detection Algorithms (Part 1)," Analytics Vidhya, 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/.
[11] D. Parthasarathy, "A Brief History of CNNs in Image Segmentation: From R-CNN to Mask R-CNN," Medium, 2019. [Online]. Available: https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4.
[12] "LabelImg," Tzutalin.github.io, 2019. [Online]. Available: https://tzutalin.github.io/labelImg/.
[13] R. Khandelwal, "Computer Vision—A journey from CNN to Mask R-CNN and YOLO," Medium, 2019. [Online]. Available: https://medium.com/datadriveninvestor/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-1d141eba6e04.
[14] "Google Colaboratory," Colab.research.google.com, 2019. [Online]. Available: https://colab.research.google.com/.