Traffic Surveillance using activity detection by YOLO Algorithm and PSO

By

Dharaneesh (19BPS1031) & Deepsi Kumari (19BCE1064)

Abstract...

The project mainly focuses on the detecting a particular object, basically a car suspected to be dangerous, by using advanced algorithms to do the task with flexible accuracy. We use the YOLO algorithm to detect the objects and then based on the information collected we try to match the objects with the image of the suspected using PSO, which allows us to decide the level of accuracy to work with, thus makes it flexible for various purposes , this can even be used to detect a certain species of animals, a person suspected to have a chronic disease or a criminal record thus helps us for searching anything in the frame with providing the flexibility to choose the accuracy level.

OBJECT DETECTION

USING YOLO ALGORITHM

YOLO(You Only Look Once)

YOLO is a state-of-the-art, real-time object detection algorithm that is fast and accurate.

A neural network predicts bounding boxes and class probabilities directly from full images in one evaluation.

YOLO make each cell predict more bounding boxes and also put the idea of multi-scale into it in order to process small objects.

How it works?

YOLO divides the input image into an SxS grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those boxes.

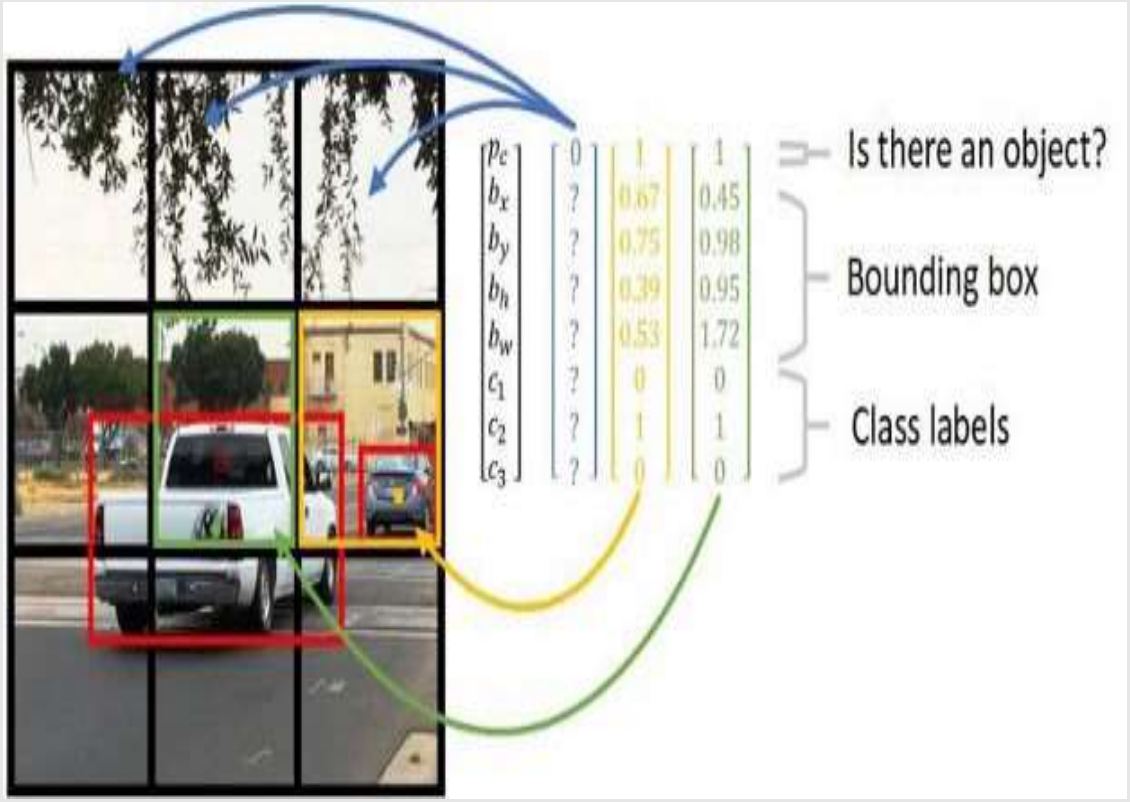
Each bounding box can be described using four descriptors: center of a bounding box (**bx,by**),width (**bw**),height (**bh**),value **c** is corresponding to a class of an object (such as: car, traffic lights, etc.).

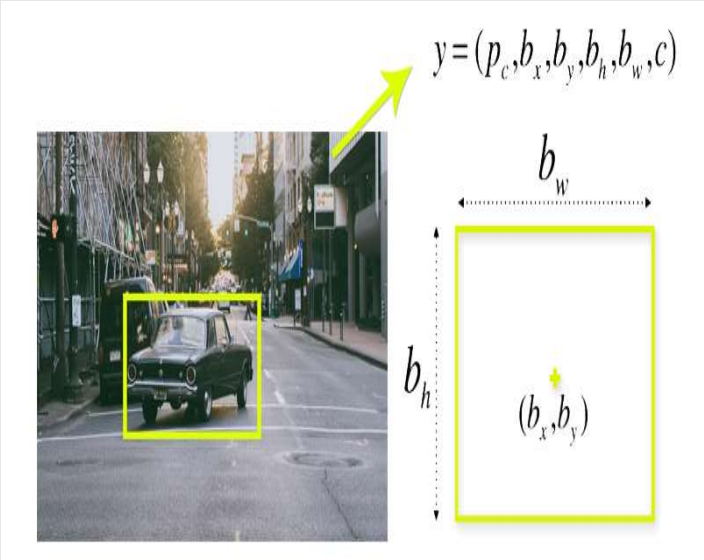
In addition, we have to predict the Pc value, which is the confidence value:

->Confidence Value= Pr(Object) * IOU ,where IOU is intersection Over Union value.

IOU= area of intersection
area of union.

IOU value >=0.5 denotes a well defined and an appropriate bounding box to detect an object.





As we are splitting our image into cells, using S*S grid (typically ,a 19×19 grid). Each cell is responsible for predicting 5 bounding boxes. Therefore, we arrive at a large number of 1805 bounding boxes for one image.

Most of these cells and bounding boxes will not contain an object. Therefore, we predict the value Pc, which serves to remove boxes with low object probability (Pc<=0.5) and bounding boxes with the highest shared area in a process called **non-max suppression** and this process is called filtering. At last, bounding box with high IOU and high Probability(object) denoting a high confidence value is used to detect the car.

Particle swarm Optimization (PSO)

PSO is a computational method which tries to find the optimum solution of a problem by iteratively trying to improve the candidate solution with regard to the given measure of quality.

In traditional method to find the minimum among them...

610	354	524	21	44	43	12	45	13	14
234	223	425	6756	6556	425	234	234	342	1234
3523	367	2452	897	8907	5687	6875	678	2356	876
345	3675	678	5567	5678	575	567	68	567	567
3354	897	6789	6789	6789	7867	6789	456	2345	5687
9879	234	436	4567	3456	3475	436	345	2343	234
2324	344	345	235	342	2134	2345	3242	234	235
134	5464	2346	2344	236	2345	235	2366	2245	1436

Using PSO...

610	354	524	21	44	43	12	45	13	14
234	223	425	6756	6556	425	234	234	342	1234
3523	367	2452	897	8907	5687	6875	678	2356	876
345	3675	678	5567	5678	575	567	68	567	567
3354	897	6789	6789	6789	7867	6789	456	2345	5687
9879	234	436	4567	3456	3475	436	345	2343	234
2324	344	345	235	342	2134	2345	3242	234	235
134	5464	2346	2344	236	2345	235	2366	2245	1436

Mathematical model of PSO

$$\overrightarrow{X_i^{t+1}} = \overrightarrow{X_i^t} + \overrightarrow{V_i^{t+1}}$$

$$\overrightarrow{V_i^{t+1}} = w\overrightarrow{V_i^t} + c_1 r_1 \left(\overrightarrow{P_i^t} - \overrightarrow{X_i^t}\right) + c_2 r_2 \left(\overrightarrow{G^t} - \overrightarrow{X_i^t}\right)$$

Inertia

Cognitive component

Social component

Our first goal is to detect the objects in the frame....

So we use motion for doing this as in our case objects definitely move... the idea is...

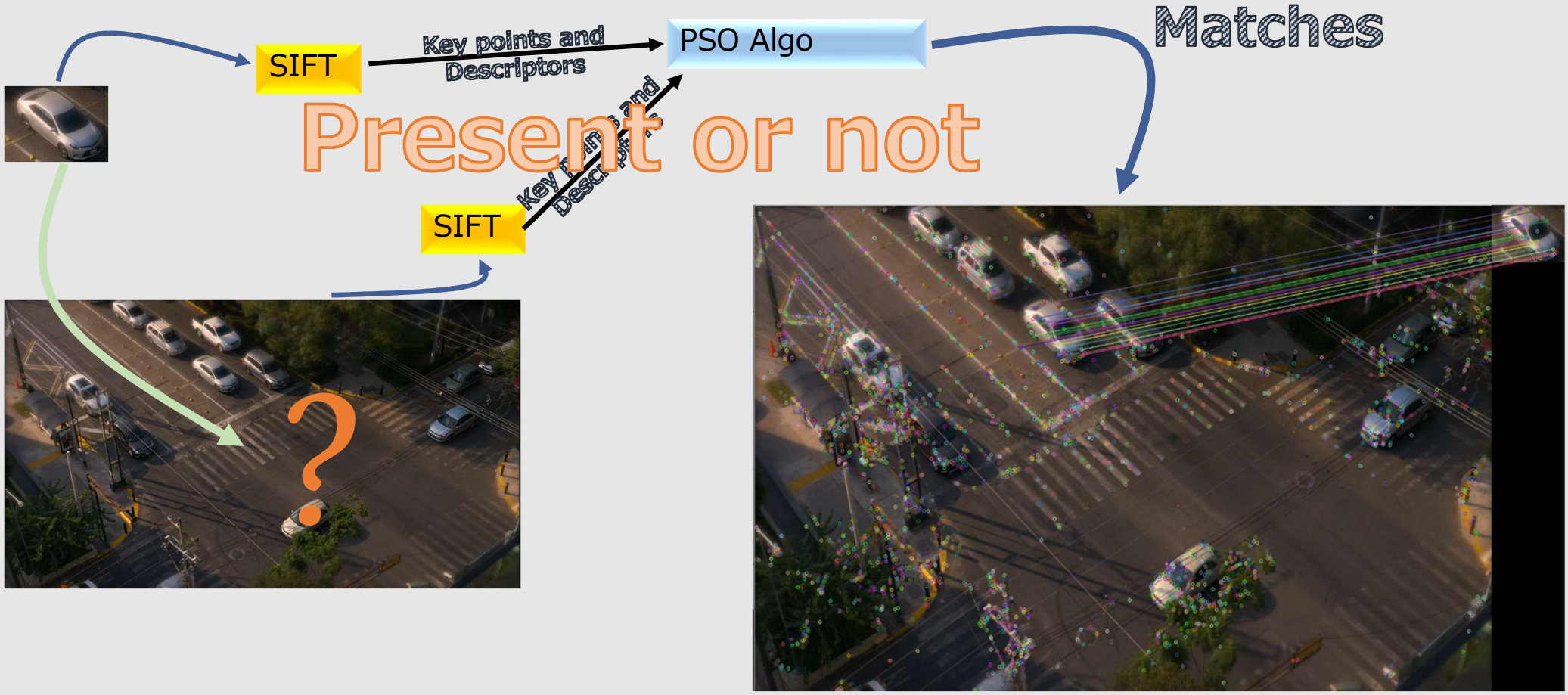
-

=

Object Detection

The idea behind detecting the objects is, we take a reference frame in which no objects(vehicles and people) are present then we find the subtraction result of this frame with other frames in the succession so that we find the objects and then do required processing on them

Our Second goal is to find the required object in the frame....

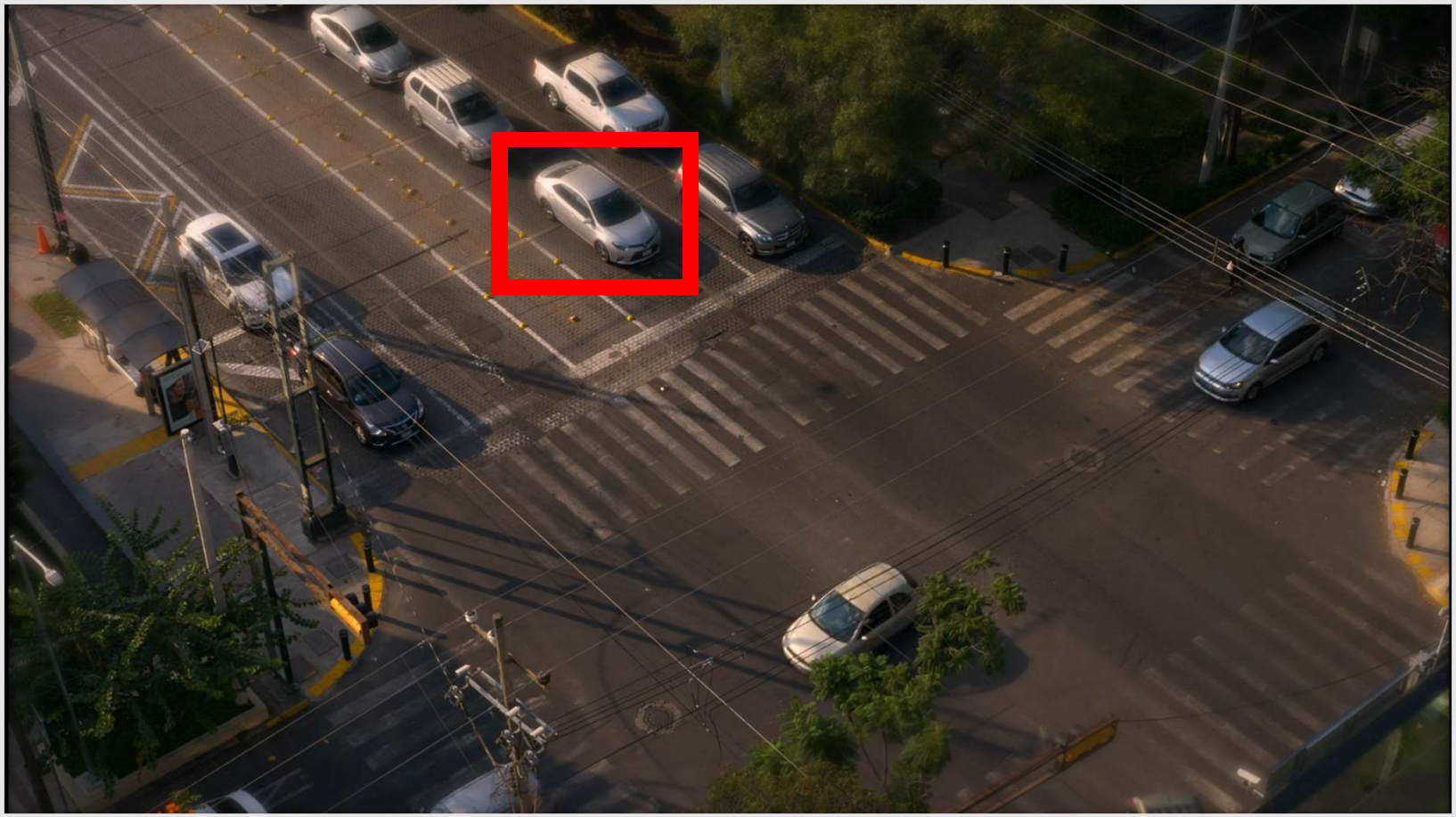


Finding similarities

Once we have a target object to detect in the frame ,we use the sift object in OpenCV along with flann's algorithm to generate the image descriptors and keypoints, for both the reference image and the object found in the footage

Once we generated the descriptors we use knn algorithm to find the similarities between the descriptors of the images and then find the no of matches they have, based on the no of matches we can set a threshold value so that we detect the desired object in the footage

Finally....



Showing the Output

Once we found the object in the footage we should draw a rectangle around the object we found using the software which lets us know that it found the object given to it as a reference, which makes the task much easier than looking into hundreds of footages and finding the object.

https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11069/110690S/An-improved-ORB-feature-point-image-matching-method-based-on/10.1117/12.2524178.short?SSO=1

Thank you