PEDESTRIAN DETECTION

Shriya Reddy Surusani
Clemson University

ABSTRACT

Various researches have been conducted over vision based pedestrian detection techniques for smart vehicles. In fact, it is one of the booming research topics, how a system can be developed such that a moving vehicle can detect a pedestrian in a potential hit region and warns the driver of the situation or automatically reacts to the situation by slowing down the speed. To achieve this, it falls back to the basic problem of how the pedestrian can be detected in the initial stage. Although this vision-based pedestrian detection process could be divided into three consecutive steps: pedestrian detection, pedestrian recognition and pedestrian tracking. In this paper, we deal with pedestrian detection in detail using pre-trained HOG + Linear SVM model in OpenCV and the future prospects of the research.

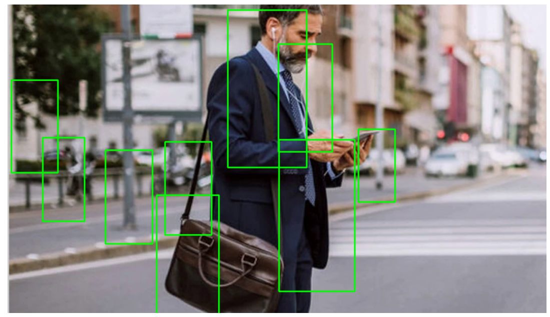
Index Terms: Pedestrian detection, Pedestrian recognition, Pedestrian tracking.

1 INTRODUCTION

Pedestrians have always been vulnerable road users. In Europe alone more than 150,000 pedestrians are injured and in United States, 12 percent of casualties in traffic accidents are pedestrians. This alarming number raises concerns in both car manufacturers and researchers. Though safety belt and other electronic equipment are in place for the protecting the drivers, there seems to be incredibly low number of safeties with respect to the pedestrians. Here, the real-time pedestrian detection is proposed as an answer for pedestrian safety by avoiding fatal accidents. Although different technologies exist for this detection. Such as, laser scanner, ultrasonic sensor and microwave radar. Our main focus is on vision-based pedestrian detection to detect wide range and obtain rich information of scenes. It proves to be a reliable method to assure driving safety and prevent the traffic accidents. We also face many challenges in this implementation since the shapes, positions, pose, motion of the pedestrians are various, including the obstructions like trees, moving cars, traffic signals etc.

We take a wide data set which includes various scenarios such as back pose pedestrians, side pose pedestrians, front facing pedestrians, pedestrians in night light etc. to check our accuracy in the results. We perform our detection on this data-set using OpenCV in python.

While Viola-Jones detectors aka HAAR cascades provided by OpenCV are extremely fast for detecting humans/faces/Objects by removing false positives. A lot of time is spent on tuning the cv2.detectMultiScale parameters. Which might not work once the image is changed. One might still end up with falsely detected humans or missing humans, due to poor parameter choice which will make the batch-processing of large datasets for human/face detection a monotonous job. Figure below shows the false-positives with this method, when tuning is not performed effectively



Though the Viola-Jones detectors were state-of-the-art and a huge motivation for the incredible advancements in object detection today. It surely is not our only choice. We can perform human/object detection using local invariant descriptors, key points, Histogram of Oriented Gradients. Exemplar models. Out of all the methods available. In this paper, pedestrian detection is performed using Histogram of Oriented Gradients + Linear SVM model as suggested by Dalal and Triggs in the seminal 2005 paper [1] where they demonstrate that the HOG image descriptor and a linear Support Vector Machine could be used to train highly accurate object classifiers.

2 SUMMARY OF THE METHOD

HOG is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. Often, local object appearance and shape can be characterized by the distribution of local intensity gradients or edge directions even without a fully accurate knowledge on gradient or edge positions. The general algorithm for training an object detector using HOG is as follows: Sample set of positives which contain the object that we want to detect and sample set of negatives which do not contain any objects that we want to detect are taken along with HOG descriptors from these samples to train a Linear Vector Machine on the positive and negative samples. Now hard-negative mining is applied. The false-positives that were found during this application is sorted by their probability and the classifier is then re-trained using the hard-negative samples. The classifier is now trained and can be applied to our test dataset. In our case, we are using pre-trained HOG that is shipped with OpenCV. We use it for scanning the image/frame and apply non-maximum suppression to remove redundant and overlapping bounding boxes.

3 IMPLEMENTATION

In Figure 1 we can see the process flow. After we process the input video, we initialize the HOG descriptor by first making a call to hog = cv2.HOGDescriptor(). Then, we call the setSVMDetector to set the Support Vector Machine to be pre-trained pedestrian detector by loading it via cv2.HOGDescriptor_getDefaultPeopleDetector().

Author's address: Shriya Surusani, Clemson University .

Figure 1: The process of pedestrian detection

```
#Re-sizing and making a copy
frame = imutils.resize(frame, width=min(400, frame.shape[1]))
orig = frame.copy()

#Detecting people
(rects, weights) = hog.detectMultiScale(frame, winStride=(4, 4), padding=(8, 8), scale=1.05

#Drawing original boundery boxes
for (x, y, w, h) in rects:
        cv2.rectangle(orig, (x, y), (x + w, y + h), (0, 0, 255), 2)

# applying non-maxima suppression to the bounding boxes
rects = np.array([[x, y, x + w, y + h] for (x, y, w, h) in rects])
pick = non_max_suppression(rects, probs=None, overlapThresh=0.65)

#Final boundery boxes
for (xA, yA, xB, yB) in pick:
        cv2.rectangle(frame, (xA, yA), (xB, yB), (0, 255, 0), 2)

#Write frame
out.write(frame)

#Display the frame
cv2.imshow('frame',frame)
if (cv2.waitKey(1) & 0xFF == ord('q')):
        break
```

Figure 2: Python Code

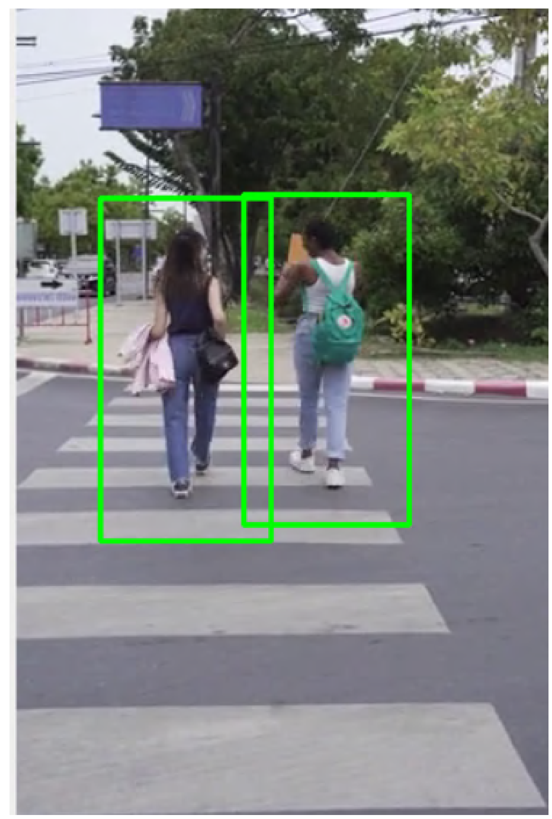
Now, to reduce detection time and increase overall detection throughput, we reduce the frame size such that the HOG features that are extracted and passed to the Linear SVM are evaluated for sure and the false-positive rate is decreased. Then, a call is made to detectMultiScale method of the HOG descriptor with scale 1.05 and sliding window of size (4,4) such that it constructs an image pyramid. The returned values of this function are rects and weights. Rects is the (x,y) – coordinates of each person in the image and weights represents the confidence value for each detection returned by SVM. Here, tuning of this scale value plays an important role. Since, not getting this right will either lead to false-positives or no detection of the pedestrians in the first place.

We now move on to drawing initial bounding boxes over our frame. Here, we will encounter a lot of overlapping bounding boxes on each person detected. This is because the same person is being detected multiple times by our program. To over come this, we apply non-maxima suppression to suppress bounding boxes that overlap, with a significant threshold value. We used the threshold value of 0.65 in our program.

Once that problem is handled, we now move on to drawing the finalized bounding boxes and display them. The implemented logic is shown in Figure 2.

4 EVALUATION

For the different inputs used, which covers the different situations the pedestrian could be in. Such as, pedestrian in low light, pedestrian in a side angle with respect to the capturing camera, pedestrian with their back facing the camera or just a blurred image of the pedestrian. We perform our detection logic on them. Here are some of the test results for the inputs used:



5 CONCLUSION

In this paper, pedestrian detection is performed using vision-based technique and has a brief overview of why HOG was used over HAAR Cascades. This could be integrated with auto-mobiles to increase pedestrian safety. Different scenario-based inputs were taken to check if the detection is accurate. One take away from what has been discussed would be, how tuning of the scale is so important, both when HOG is used or HAAR Cascades. It holds the key in getting about the right detection over the pedestrian. Scaling values in our program were obtained by trial and error. There is a lot of scope for development in pedestrian detection apart from this conventional vision-based method. One can use sensors such as thermopile/infrared where it measures the objects' presence within their respective field-of view. It uses the detection theory of, the object of interest being at a higher temperature and then directly obtains the 2D thermal image. The rate of detection using this method is proven to be more than the conventional vision-based method, according to [5]

REFERENCES

[1]: http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

[2]: https://www.researchgate.net/publication/4253959_A_Review _on_Vision-Based_Pedestrian_Detection_for_Intelligent_Vehicles

[3]: https://www.researchgate.net/publication/337595492 _Research_on_Pedestrian_Detection_Algorithm_Based_on_Image

[4]: https://www.pyimagesearch.com/2014/11/10/hitogram -oriented-gradients-object-detection/

[5]: https://ieeexplore.ieee.org/document/1505072

[6]: https://paperswithcode.com/paper/pedestrian-synthesis-gan-generating

[7]: https://paperswithcode.com/paper/illuminating-pedestrians-via-simultaneous

[8]: https://iopscience.iop.org/article/10.1088/1742-6596/1345/6/062023/meta