IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Cars and Pedestrians Detection and Tracking: Using Haar Cascade Classifiers

[1]Koda Indu, [2]Swathi Bhaskaran, [3]Tangi Alekhya, [4]Tangula Kaveri, [5]Varigeti Jaya Sruthi,

[1]Guide, [2]UG Student, [3]UG Student, [4]UG Student, [5]UG Student
[1-5]Department of Computer Science and Systems Engineering,
[1-5]Andhra University College of Engineering for Women, Visakhapatnam, India

Abstract: Computer vision finds an important application in traffic surveillance, management and monitoring. The aim of this paper is to review implementation of Haar Cascade classifiers in detecting cars and pedestrians from either a video input or a live stream input from surveillance cameras. The video stream is broken into frames. Each frame is taken up as an image to detect cars and pedestrians. This is done by using sliding window approach. Depending on where the window is currently positioned, each stage of the classifier marks a specific area as positive or negative. We use Computer Vision Library (OpenCV). This study also evaluates Haar Cascade method with respect to other object detection algorithms. The paper concludes with a discussion on the future scope of this work.

***Index Terms* – Object detection, OpenCV, Haar Cascade, Car and Pedestrians, Tracking, Python**

I. Introduction

Object detection is an important and interesting application of computer vision. Computer vision, image processing, and deep learning are all part of object detection, which refers to the process of identifying instances of objects in images and videos. Object detection is the action of recognizing a distinct object for an image. The primary objective of object tracking is to connect the features of the target objects with their location in subsequent video sequences. Object tracking is gaining importance in the field of computer vision due to the spread of powerful computers and the rising demand for automated surveillance systems and other applications. There are many advantages to tracking. A few of them include traffic monitoring and surveillance, security, robot vision and video communication, as well as usage in public spaces or places with massive gatherings of people like subway stations, airports, and animation.

One key application of object detection and tracking is in traffic surveillance. Detection and tracking of cars and pedestrians are crucial for managing and planning of traffic and its monitoring. A human or a driver can quickly identify and locate the objects using the segmentation's bound box, which is faster than unprocessed images. In this review, we talk about the usage of Haar Cascade classifiers which is implemented with the assistance of OpenCV, which will be written in Python. OpenCV is a computer vision algorithm library that is free and available under the BSD- license.

Haar Cascade is a method that uses machine learning to train a cascade function from many positive and negative images. After that, it is used to find objects in other pictures. It slides a window over the image to detect objects. The aspect ratio of the window remains constant, but its size changes to accommodate objects of various sizes. The algorithm finds the sum of all the image pixels in the dark area of a hair feature and the sum of all the image pixels in the light area of the same feature. The difference between them is computed.

The feature value will be close to 1 if an edge separates the bright pixels on the left from the dark pixels on the right side of the image. It means, edges are detected as the price approaches the feature value of 1. OpenCV uses a trained Haar cascade to conduct detection. It is not necessary for us to create our own positive and negative samples, train our own classifier, or stress over precisely tuning the parameter values. Instead, we only load the classifier that has already been trained to find cars or pedestrians in pictures. For each of the stops along the sliding window route, OpenCV computes five rectangular features in the background. We simply subtract the sum of pixels under the white region from the number of pixels under the black region to generate features for each of these five rectangular sections. The Haar Cascade algorithm can recognize objects in an image regardless of their size or location. This algorithm can be executed in real time and is not very difficult to implement.

1 WORKFLOW OF OBJECT TRACKING

The process includes four basic steps: Video sequence/ Real-time input, Object Detection, Object Recognition and Object tracking.

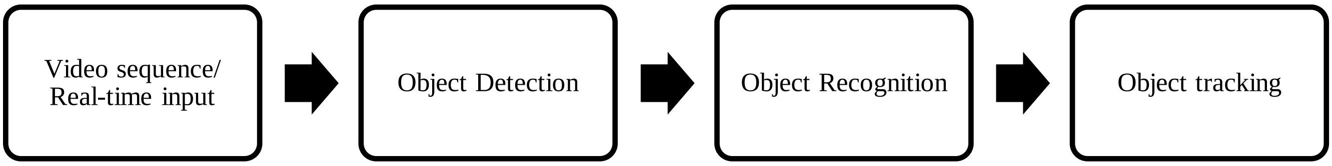

Fig.: Flow diagram of Object tracking

A computer vision approach called object detection enables a machine to identify items in an image or video. Object tracking involves continuous detection of the same object (here, car and pedestrians) in all the frames of the video sequence.

2 WORKFLOW OF OBJECT DETECTION

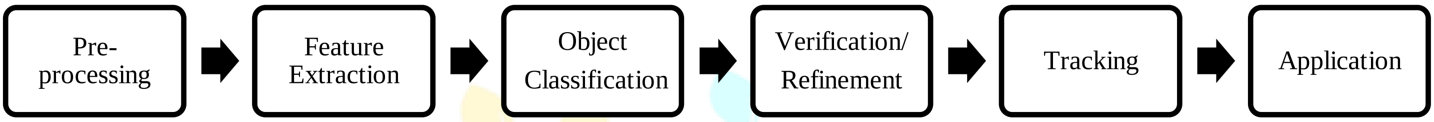

Fig.: Flow diagram of Object detection

2.1 Pre-processing

Pre-processing involves several tasks that are necessary at the beginning of the procedure, including, but not limited to, exposure time, gain adjustments, and camera calibration.

2.2 Feature Extraction

We can categorize the various features as: Generic qualities: Qualities including color, texture, and shape. When they are according to the degree of abstraction, further divided into the following groups:

2.2.1 At the pixel level:

Features that are decided at each pixel include color and position.

2.2.2 Local features:

These are those characteristics that are calculated across the outcomes of band on the image, image segmentation, or edge detection.

2.2.3 Global features:

These are the features that are estimated throughout a complete image or just a regular section of an image.

2.2.4 Domain-specific features:

Domain-specific features only include application-dependent features, such fingerprints, human faces, and conceptual features. Nothing more than a synthesis of low-level traits for a certain domain, these features.

2.3 Object Classification

It combines motion-based and appearance-based detection methods to give a contextual combination which can then detect an object from the image as a pedestrian or a car.

2.4 Verification/Refinement

Most cutting-edge systems employ a tracking module to monitor identifiable cars and pedes over time. This stage is crucial because it helps prevent false positives over time, forecasts the positions of cars and pedestrians for the segmentation algorithm. At a higher level, it infers useful insight into the behavior of pedestrians and cars.

2.5 Tracking

Though it is a challenging issue in image processing, object tracking that can use video sequences is feasible. More other problems seem to be caused by the occlusion of the object to scene, object to object, complex object motion, real-time processing requirements, and the improper or deformed shape of the object.

2.6 Application

Any intelligent video surveillance system must perform the crucial and important duty of pedestrian detection since it supplies the vital data for the semantic analysis of the video material. As a result of the ability to enhance safety systems, it has an obvious extension to automobile applications. This approach is helpful for preventing accidents and traffic issues by applying clever techniques.

3 METHODOLOGY AND IMPLEMENTATION

The section outlines the basic methodologies used in this review.

3.1 Computer Vision

The study of computer vision makes it possible for machines to mimic the human visual system. It's a branch of artificial intelligence that gathers data from digital photos or videos and then manipulates it to create attributes. Image acquisition, screening, analysis, identification, and information extraction are all part of the process. Computers can grasp any visual content and respond to it appropriately thanks to this extensive processing.

To collect multi-dimensional data, computer vision projects convert digital visual input into clear descriptions. To assist in the decision-making process, this data is subsequently translated into computer-readable language. Teaching machines to gather data from pixels is the fundamental goal of this area of artificial intelligence.

3.2 Object detection

A computer vision approach called object detection enables a machine to identify items in an image or video. A human or driver may quickly identify and locate the objects using this bounded box from the segmentation, which is faster than using raw photos. In order to apply this intelligence into a computer, object detection's objective is to detect objects. There are various methods for performing object detection. Convolutional neural networks CNNs are often used in popular deep learning techniques like YOLO and SSD that automatically learn to detect objects inside frames.

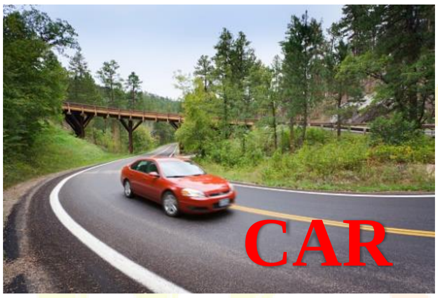

Fig: Image recognition

Fig: Object Detection

3.3 OpenCV

OpenCV is a computer vision algorithm library that is free and available under the BSD- license. Haar Cascade classifier is implemented with the assistance of OpenCV, which will be written in Python. The OpenCV library maintains a repository of pre-trained Haar cascades.

3.4 Haar Cascade Classifiers

In their 2001 paper titled "Rapid Object Detection using a Boosted Cascade of Simple Features," Paul Viola and Michael Jones proposed an efficient method for object detection called the Haar Cascade classifier. A cascade function is trained from a large number of positive and negative images in this machine learning strategy. The next step is to take features out of it. Haar features are used for this. They resemble convolutional kernel in every way. By dividing the total number of pixels in the white rectangle by the total number of pixels in the black rectangle, a single value is created for each feature. Then, based on the training, it is used to find objects in the other images. Negative images are those without the target objects, whereas positive images contain the object in question.

Positive Negative

Fig: Examples of Positive and Negative images

There is individual .xml files with a lot of features; each xml file is related to a very specific kind of use case. These files are used to train the model to detect the object from images. The speed of Haar cascades is one of their main advantages; it's difficult to match it. In the system's detection phase, a window of the target size is primarily moved over the input image, and the calculation of a Haar-like feature is done for each subpart of the image. Then, this difference is contrasted with a learnt threshold that distinguishes between objects and non-objects. Background Subtraction Using MOG2 is performed with OpenCV library functions as a performance-enhancing pre-processing step.

The XML files for the various objects (target objects) that need to be detected are then created, with one file for each bus, car, two-wheeler, and pedestrian. A special tool known as the Haar Training tool is used to carry out the steps. The various files and folders in this tool are used to create XML files. Several scaling factor values are evaluated, and the best one for each car and pedestrian detection was selected; as a result, rectangles are painted onto the identified objects. Examples of the results after using different xml files to detect items like a car or pedestrian.

The steps involved in creating the XML files is shown below:

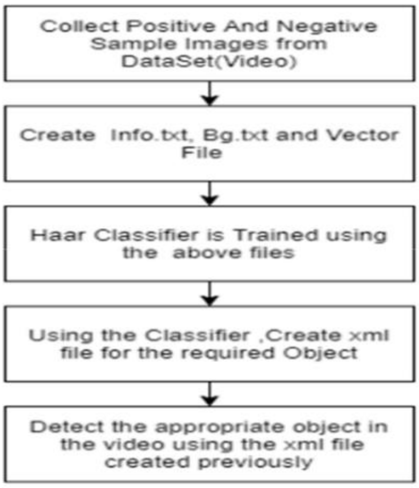

Fig: Flow Diagram of the various processes involved

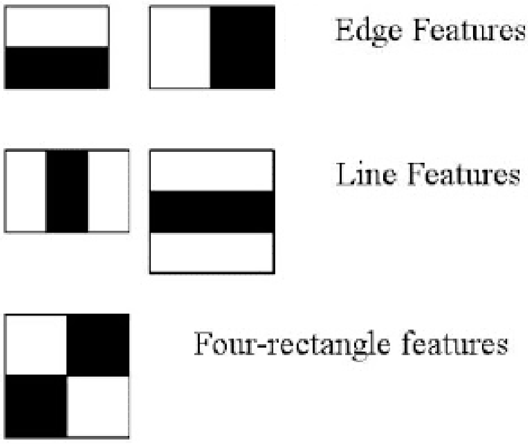

Fig.: The 5 different types of Haar-like features

II. RESULTS AND DISCUSSION

Haar cascades are fast and can work well in real-time. They're very fast at computing Haar-like features due to the use of integral images (also called summed area tables). They are also very efficient for feature selection through the use of the AdaBoost algorithm. It is simple to implement and less computing power required.

One major limitation would be that Haar cascades are prone to false-positives and it is less accurate than deep learning-based techniques.Manual tuning of the haar cascade detection parameter can solve the problem of false-positive detection. Despite the limitation of the haar cascade algorithm still widely used when we can tolerate some false-positive detection.

Video file is given to the system as an input where the system will detect the cars and Pedestrian on road.

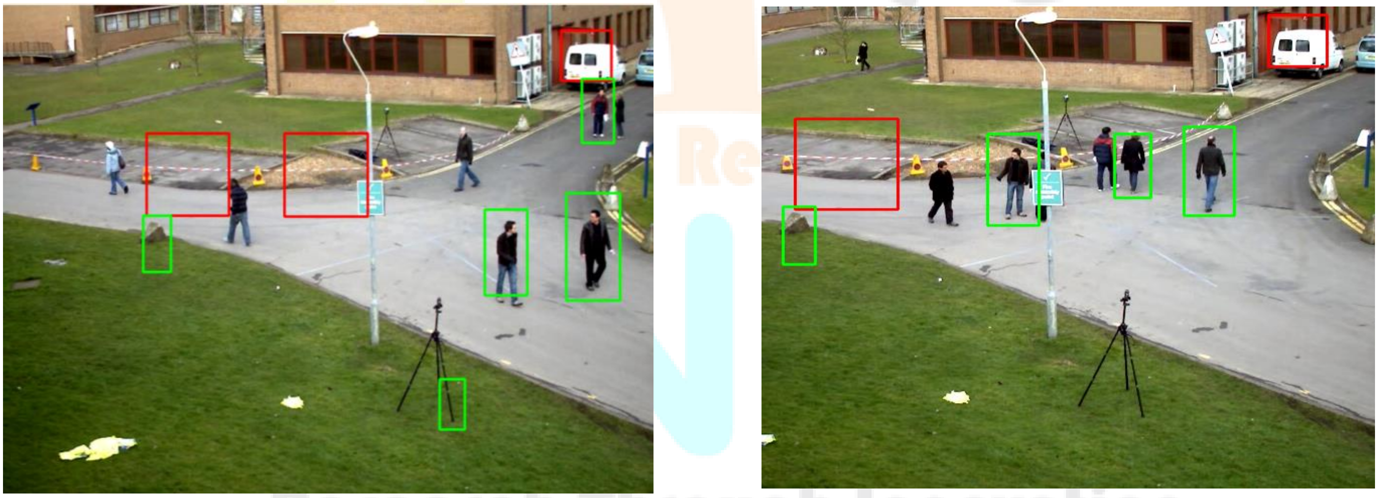

Fig: Detection and tracking of pedestrians (in green) and cars (in red)

In below output image, we can see cars are detected with red color rectangles.

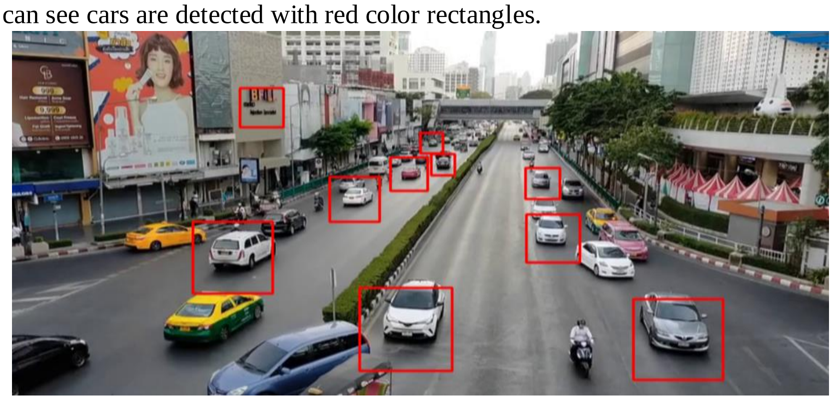

Fig: Detection and tracking of cars from input video stream

The same tracking technology might be used to build and evaluate intricate video sequence simulations. In the typical scenario, an object of the same color or a larger obstruction with a longer occlusion period will obscure moving items. If there are more items, the tracking algorithm will be more effective and useful. Weight parameters could be used to adjust the intensity of each individual pixel. In any image, the possibility that the foreground also has comparable pixel coordinates is decreased if the intensity value is assigned as foreground based on the current frame, allowing the BG weightage for the said pixel to be reduced to the lowest value feasible. The prior pixel value can be discarded with the least likelihood as opposed to the evolved scene by introducing a weightage that is smaller than the starting value.

III. References

[1] Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, I-511 — I-518. doi: 10.1109/CVPR.2001.990517.

[2] OpenCV Cascade Classifier from https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html.

[3] Aggarwal, A., Biswas, S., Singh, S., Sural, S. & Majumdar, A.K., 2006. Object Tracking Using Background Subtraction and Motion Estimation in MPEG Videos, in 7th Asian Conference on Computer Vision. SpringerVerlag Berlin Heidelberg, pp. 121–130. doi:10.1007/11612704_13.

[4] Aldhaheri,A.R.&Edirisinghe,E.A.,2014.DetectionandClassificationofaMovingObjectinaVideoStream,in:Proc.oftheIntl.Conf. onAdvancesinComputing andInformation Technology.InstituteofResearch Engineersand Doctors.