Six Step Framework For Object Detection

T.Chandrasekhar [1], K. Satya Srinija [2], K.L.V.Sai Teja[3], M.Kalyani[4], R.L.S.V.Naidu[5]
[1, 2, 3, 4, 5] Dept of ECE
[1, 2, 3, 4, 5] GIET Engineering College.

Abstract- *Object detection is a key ability required by most computer and robot vision systems. The latest research on this area has been making great progress in many directions. In the current manuscript, we give an overview of past research on object detection, outline the current main research directions, and discuss open problems and possible future directions.*

Object recognition technology has matured to a point at which exciting applications are becoming possible. Indeed, industry has created a variety of computer vision products and services from the traditional area of machine inspection to more recent applications such as video surveillance, or face recognition. In this chapter, several representatives from industry present their views on the use of computer vision in industry. Current research conducted in industry is summarized and prospects for future applications and developments in industry are discussed.

With the advancement of modern technologies areas related to robotics and computer vision, real time image processing has become a major technology under consideration. So I tried a novel approach for capturing images from the computer web cam in real time environment and process them as we are required. By using open source computer vision library (OpenCV for short), an image can be captured on the bases of its hue, saturation and colorvalue(HSV) range. The basic library functions for image handling and processing are used.Computer vision techniques become particularly important due to their fast response, high accuracy and strong adaptability. Two of the most demanding and widely studied applications relate to object detection and classification.

I. INTRODUCTION

1. OBJECT DETECTION

Object detection is a major implementation in computer vision. Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

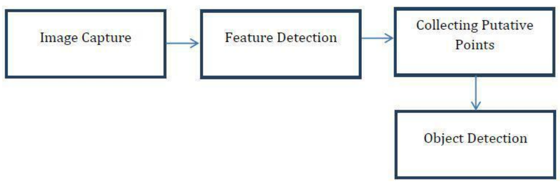
In many computer vision systems, object detection is the first task being performed as it allows to obtain further information regarding the detected object and about the scene. Once an object instance has been detected (e.g., a face), it is be possible to obtain further information, including: (i) to recognize the specific instance (e.g., toidentify the subject’s face), (ii) to track the object over an image sequence (e.g., to track the face in a video), and (iii) to extract further information about the object (e.g., to determine the subject’s gender), while it is also possible to (a) infer the presence or location of other objects in the scene (e.g., a hand may be near a face and at a similar scale) and (b) to better estimate further information about the scene (e.g., the type of scene, indoor versus outdoor, etc.), among other contextual information.

Object detection has been used in many applications, with the most popular ones being: (i) human-computer interaction (HCI), (ii) robotics (e.g., service robots), (iii) consumer electronics (e.g., smart-phones), (iv) security (e.g., recognition, tracking), (v) retrieval (e.g., search engines, photo management), and (vi) transportation (e.g., autonomous and assisted driving). Each of these applications has different requirements, including: processing time (off-line, on-line, or real-time), robustness to occlusions, invariance to rotations (e.g., in-plane rotations), and detection under pose changes. While many applications consider the detection of a single object class (e.g., faces) and from a single view (e.g., frontal faces), others require the detection of multiple object classes (humans, vehicles, etc.), or of a single class from multiple views (e.g., side and frontal view of vehicles). In general, most systems can detectonly a single object class from a restricted set of views and poses.

2. CONCEPT

Every object class has its own special features that helps in classifying the class – for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e. the center) are sought. Similarly, when looking for squares, objects that areperpendicular at corners and have equal side lengths are needed. A similar approach is used for faceidentification where eyes, nose, and lips can be found and features like skin color and distance between eyes can be found.

The research purpose of computer vision aims to simulate the manner of human eyes directly by using computer. Computer vision is such kind of research field which tries to percept andrepresent the 3D information for world objects. Its essence is to reconstruct thevisual aspects of 3D object by analyzing the 2D information extracted accordingly . Real life 3D objectsare represented by 2D images. The process of object detection analysis the input image and to determine the number, location, size, position of the objects. Object detection is the base for object tracking and object recognition, whose results directly affect the process and accuracy of object recognition. The common object detection method is: color-based approach, detecting objects based on their color values. The method is strong adaptability and robustness, however, the detection speed needs to be improved, because it requires test all possible windows by exhaustive search and has high computational complexity.The goal of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image. Typically only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur and that need to somehow be explored.



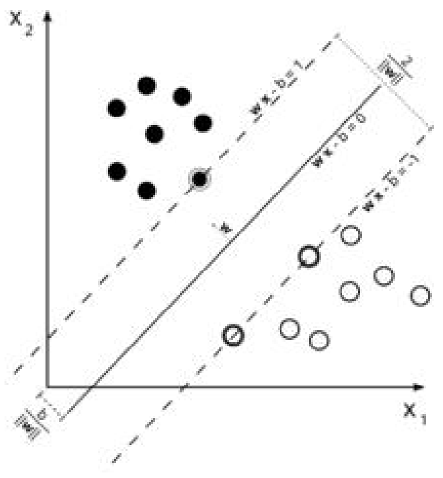
3. TECHNIQUESAND ALGORITHMS

The advantage we are having is, an image is made of pixels. So in most cases we know the location of next point, it will be connected to our current pixel. Starting with circles, take an image, convert it to gray scale, and detect edges. Move along edges, draw normal, they will intersect at center. Do this for entire circle or find connected edges and calculate Euclidean distance between center and connected points. Another algorithm is move along connected edges rotation of tangent will be uniform, because of symmetry. So whenever there is an abrupt change in rotation, you are out of circle. For squares, move along edges. First of all check if they are straight lines or not (check if pixels are having either same x or y co-ordinates). After that look for a 90 degree change in angle(if you were moving along a horizontal line then at corner y co-ordinate will stop changing and x will start changing).

3.1 SUPPORT VECTOR MACHINE

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to oneor the other of two categories, an SVM training algorithmbuildsamodel that assigns new examples to one category or the other, making it anon-probabilistic binary linear classifier.An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering and is often used in industrial applications either when data are not labeled or when only some data are labeled as a preprocessing for a classification pass.



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

4. OPEN CV

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel's research center in Nizhny Novgorod (Russia), it was later supported by Willow Garage and is now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license. To support some of the above areas,

OpenCV includes a statistical machine learning library that contains:

1.Boosting
2.Decision tree learning
3.Gradient boosting trees
4.Expectation-maximization algorithm
5.k-nearest neighbor algorithm
6.Artificial neural networks
7..Random forest
8.Support vector machine (SVM)

II. EXISTING METHODS

Viola–Jones object detection framework

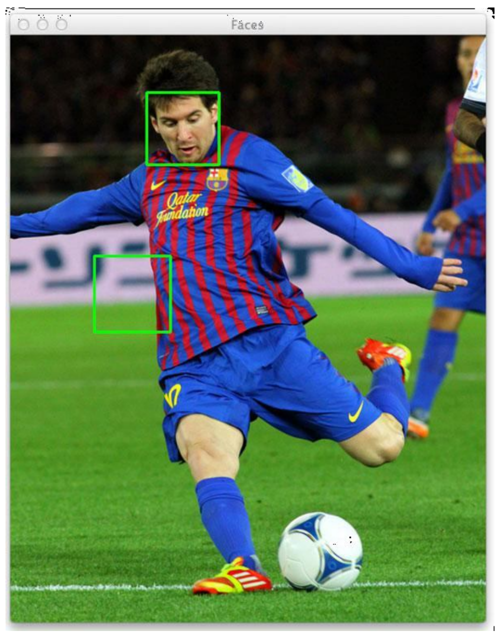
The Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection. This algorithm is implemented in OpenCV as cvHaarDetectObjects().

RELATED FACE DETECTION AND TRACKINGALGORITHM

A method similar to Viola–Jones but that can better detect and track tilted and rotated faces is the KLT algorithm. Here numerous feature points are acquired by first scanning the face. These points then may be detected and tracked even when the face is tilted or turned away from the camera,somethingViola–Jones has difficulty doing due to its dependence on rectangles.

PROBLEMS IN USING HAAR CLASSIFIERS

- Problem 1: Haar cascades are extremely slow to train, taking days to work on even small datasets.
- Problem 2: Haar cascades tend to have an alarmingly high false-positive rate (i.e. an object, such as a face, is detected in a location where the object does not exist).
- Problem 3: What's worse than falsely detecting an object in an image? Not detecting an object that actually does exist due to sub-optimal parameter choices.
- Problem 4: Speaking of parameters: it can be especially challenging to tune, tweak, and dial in the optimal detection parameters; furthermore, the optimal parameters can vary on an image-to-image basis!



In order to detect faces/humans/objects/whatever in OpenCV (and remove the false positives), you'll spend a lot of time tuning the cv2.detectMultiScale parameters. And again, there is no guarantee that the exact same parameters will work from image-to-image. This makes batch-processing large datasets for face detection (or any other type of object detection) a tedious task, since you'll be very concerned with either falsely detecting faces or missing faces entirely, simply due to poor parameter choices on a per image basis.

There is also the problem that the Viola-Jones detectors are nearing 15 years old. If this detector were a nice bottle of Cabernet Sauvignon, I might be pretty stoked right now. But the field has advanced substantially since then. Back in 2001, the Viola-Jones detectors were state-of-the-art, and they were certainly a huge motivating force behind the incredible new advances we have in object detection today.

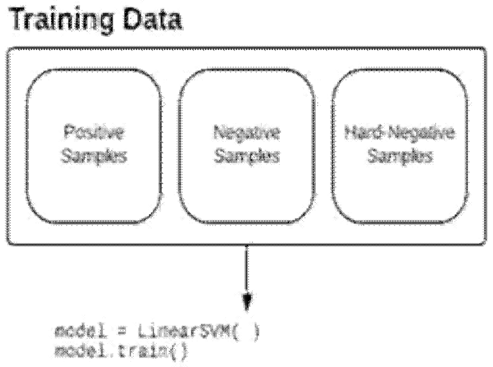
III. PROPOSED METHOD

This framework will consist of many steps and components, including

- Step 1: Experiment preparation
- Step 2: Feature extraction
- Step3: Detector training
- Step 4: Non-maxima suppression
- Step 5: Hard negative mining
- Step6: Detection retraining

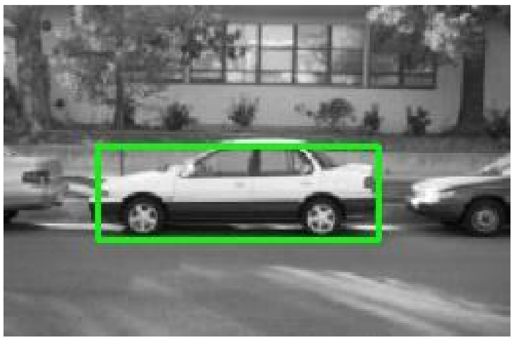
IV. OBJECTIVE

Understand the concept of an object detection framework. Explore our training data, allowing us to make critical downstream decisions. •Explore our "negative images dataset".

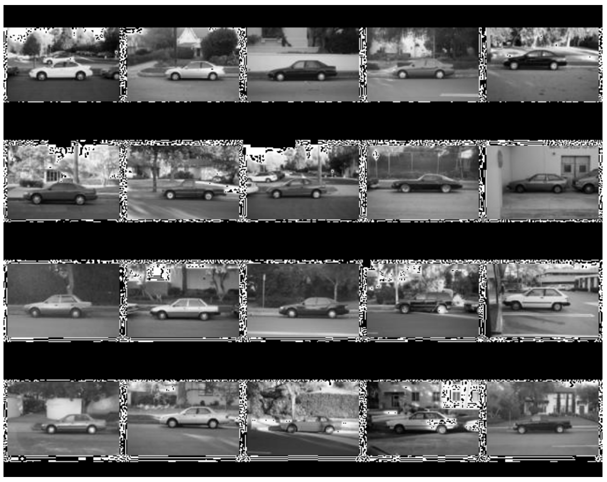


FRAMEWORK GOALS

The primary goal of this module is to build an object detection framework that can be used to easily and rapidly build custom object detectors. When building each object detector (e.x., chair, car, airplane), the code should have to change onlyminimally, and ideally, not at all.



Once trained, our object detector will be able to detect the presence (or lack thereof) of a car in an image, followed by drawing a bounding box surrounding it.Again, this framework is not specific to side views of cars ,it can be used to create any custom object detector of our choice.The car side choice is simply an example.



RE-TRAINING AND RUNNING THE CLASSIFIER

Now that we have our extra negative examples, we need to re-train our classifier using all three sets of training data:

1. Original Positives
2. Original Negatives
3. Hard-Negatives

After re-training our Linear SVM on this data, the detector should report less false-positives while still retaining the true-positive detections.

OBJECTIVES

•Use hard-negative mining examples mined from our previous lesson to re-train our object detector.

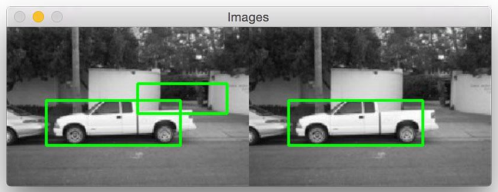
•Test our newly re-trained object detector and see if our false-positive detection rate has been reduced.

Now that we have applied hard-negative mining, the next step is to re-train our classifier. To accomplish this, we'll use the same train_model.py from our initial training phase lesson.To re-train our object detector using the hard-negative examples, just issue the following command.$python train_model.py --confconf/cars.json --hard-negatives 1.

Again, notice how the --hard-negatives switch is used to indicate that we want to include the hard-negative examples in the training process.

To test out our newly re-trained model, execute test_model.py , only this time, our newly re-trained object detector will be automatically used.

```
$ python test_model.py --confconf/cars.json \
--image
datasets/caltech101/101_ObjectCategories/car_side/image_00
16.jpg
```



On the left, we have our original detections (after non-maxima suppression [NMS]) with no hard-negative mining. Then, on the right, we have the re-trained object detector results (after NMS) after applying hard-negative mining. As you can see, the false-positive detection has been removed.

Applying hard-negative mining has lead to a reduction in false-positive detections, with no loss in true-positive detections.By using our hard-negative examples as additional training data, we were able to reduce the false-positive detection rate of our classifier while still retaining the true detections.Applying hard-negative mining nearly always increases detection accuracy, so it's something that should be performed (or at the very least tested) for our own custom object detectors.

V. APPLICATIONS

1. FACE DETECTION

Popular applications include face detection and people counting. When we upload a photo Facebook instantly detects faces.This is a simple application of object detection that we see in our daily life.

2. PEOPLE COUNTING

Object detection can be also used for people counting, it is used for analysing store performance or crowd statistics during festivals. These tend to be more difficult as people move out of the frame quickly (also because people are non rigid objects).

3. VEHICLE DETECTION

Similarly when the object is a vehicle such as a bicycle or car, object detection with tracking can prove effective in estimating the speed of the object. The type of ship entering a port can be determined by object detection(depending on shape, size etc). This system for detecting ships are currently in development in some European countries.

4.MANUFACTURING INDUSTRY

Object detection is also used in industrial processes to identify products. Say you want your machine to only detect circular objects. Hough circle detection transform can be used for detection.

5. ONLINE IMAGES

Apart from these object detection can be used for classifying images found online. Obscene images are usually filtered out using object detection.

6. SECURITY

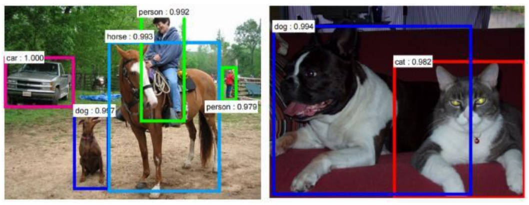
In the future we might be able to use object detection to identify anomalies in a scene such as bombs or explosives (by making use of a quadcopter).

VI. FUTURE ADVANCEMENTS

OBJECT DETECTION USING DEEP LEARNING

An image classification problem is predicting the label of an image among the predefined labels. It assumes that there is single object of interest in the image and it covers a significant portion of image. Detection is about not only finding the class of object but also localizing the extent of an object in theimage. The object can be lying anywhere in the image and can be of any size(scale) as can be seen in the figure 1. So object classification is no more helpful when:

1.Multiple objects in image
2.Objects Are Small
3.Exact location and size of object in image is desired



VII. CONCLUSION

Object detection is a key ability for most computer and robot vision system. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated in assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning. Itshould be noted that object detection has not been used much in many areas where it could be of great help. As mobile robots, and in general autonomous machines, are starting to be more widely deployed (e.g., quad-copters, drones and soon service robots), the need of object detection systems is gaining more importance. Finally, we need to consider that we will need object detection systems for nano-robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical.

REFERENCES

[1] AdrianRosebrock , Practical Python and Opencv and case studies 3rd edition for object detection, a computer vision technique.
[2] RossGirshick, Jeff Donahue, Trevor Darrell, Jitendra Malik: Rich feature hierarchies for accurate object detection and semantic segmentation
[3] Kaiming He, Xiangyu Zhang, ShaoqingRen, Jian Sun: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

AUTHOR:

Kopperla. Satya Srinija, B.Tech scholar,
GIET engineering college