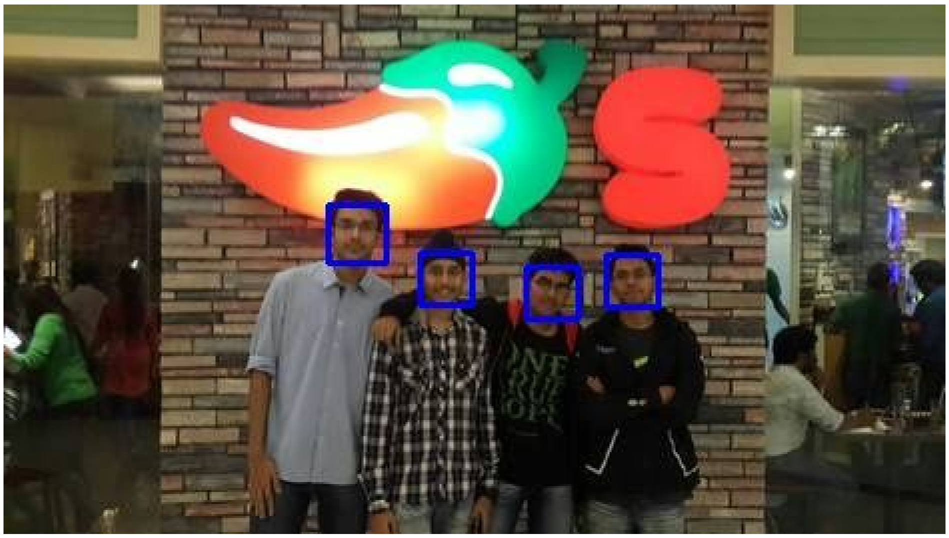
SM in AI Project Report

PROJECT TITLE: OBJECT DETECTION



Jalpreet Singh Nanda (201301146)
Goutam Nair (201301050)
Gaurav Mittal (201301149)

20.04.2016

Project Description

The task is detecting instances of semantic objects of a certain class (such as human Faces or cars) in images and videos.

Object detection has applications in many areas of computer vision, including image retrieval and video surveillance. The challenges include change in lighting conditions, occlusions, viewpoint variance, poor video quality etc.

Background

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.

Object detection systems construct a model for an object class from a set of training examples. In the case of a fixed rigid object only one example may be needed, but more generally multiple training examples are necessary to capture certain aspects of class variability.

Methods typically build a classifier that can discriminate between images (or sub-images) containing the object and those not containing the object. The parameters of the classifier are selected to minimize mistakes on the training data, often with a regularization bias to avoid overfitting.

Theory

Images of objects from a particular class are highly variable. One source of variation is the actual imaging process. Changes in illumination, changes in camera position as well as digitization artifacts, all produce significant variations in image appearance, even in a static scene. The second source of variation is due to the intrinsic appearance variability of objects within a class, even assuming no variation in the imaging process. For example, people have different shapes and wear a variety of clothes. The challenge is to develop detection algorithms that are invariant with respect to these variations and are computationally efficient.

Goal

- Input: It will be an image or a video with a particular kind of objects.
- Output: It should be an image or a video, depending on the input, with some labels for the detected objects.

Challenges

Although has been studied for dozens of years, object detection and tracking remains an open research problem. A robust, accurate and high performance approach is still a great challenge today. The difficulty level of this problem highly depends on how you define the object to be detected and tracked. If only a few visual features, such as a specific color, are used as representation of an object, it is fairly easy to identify all pixels with same color as the object. On the other extremity, the face of a specific person, which full of perceptual details and interfering information such as different poses and illumination, is very hard to be accurately detected, recognized and tracked. Most challenges arise from the image variability of video because video objects generally are moving objects. As an object moves through the field of view of a camera, the images of the object may change dramatically. This variability comes from three principle sources: variation in target pose or target deformations, variation in illumination, and partial or full occlusion of the target.

Experiments

Feature Based Object Detection

Detecting a reference object (left) in a cluttered scene (right) using feature extraction and matching. RANSAC is used to estimate the location of the object in the test image. This is an algorithm for detecting a specific object based on finding point correspondences between the reference and the target image. It can detect objects despite a scale change or in-plane rotation. It is also robust to small amount of out-of-plane rotation and occlusion.

This method of object detection works best for objects that exhibit non-repeating texture patterns, which give rise to unique feature matches. This technique is not likely to work well for uniformly-colored objects, or for objects containing repeating patterns.

Test Image

Test Object

Viola-Jones object detection

We develop a simple face tracking system by dividing the tracking problem into three separate problems:

1. Detect a face to track
2. Identify facial features to track
3. Track the face

Detect a face to track

Use the vision.CascadeObjectDetector (Matlab) to detect the location of a face in a video frame. The cascade object detector uses the Viola-Jones detection algorithm and a trained classification model for detection.

Identify facial features to track

Once the face is located in the video, the next step is to identify a feature that will help you track the face. For example, you can use the shape, texture, or color. Choose a feature that is unique to the object and remains invariant even when the object moves.

In our case, we used the skin tone as the feature to track.

Track the face

With the skin tone selected as the feature to track, you can now use the Histogram based tracker for tracking.

Car Detection using Threshold

We tried to detect only light coloured cars to start with. One way to remove the dark-colored cars from the video frames is to use the imextendedmax function. This function returns a binary image that identifies regions with intensity values above a specified threshold, called regional maxima. All other objects in the image with pixel values below this threshold become the background. To eliminate the dark-colored cars, determine the average pixel value for these objects in the image. In the resulting image, still there were extraneous objects remaining, particularly the lane-markings. To remove these objects, you can use the morphological function imopen. This function uses morphological processing to remove small objects from a binary image while preserving large objects.

Car Detection using HOG Features

In this experiment, we used both positive samples and negative samples to train our classifier (Linear SVM).

The basic steps involved are:

1. First step involved in this experiment was to calculate the HOG features of both the positive and the negative features. After the features were calculated, we dumped them into a feature file for future use.
2. Now, using these features, we trained our 2-class Linear SVM. The positive features indicate Cars and the negative features indicate non-cars
3. Now, after we trained the SVM, the model generated was used to detect cars in the Test Dataset with different test cases covered.

Face Detection using Haar Cascade Classifier

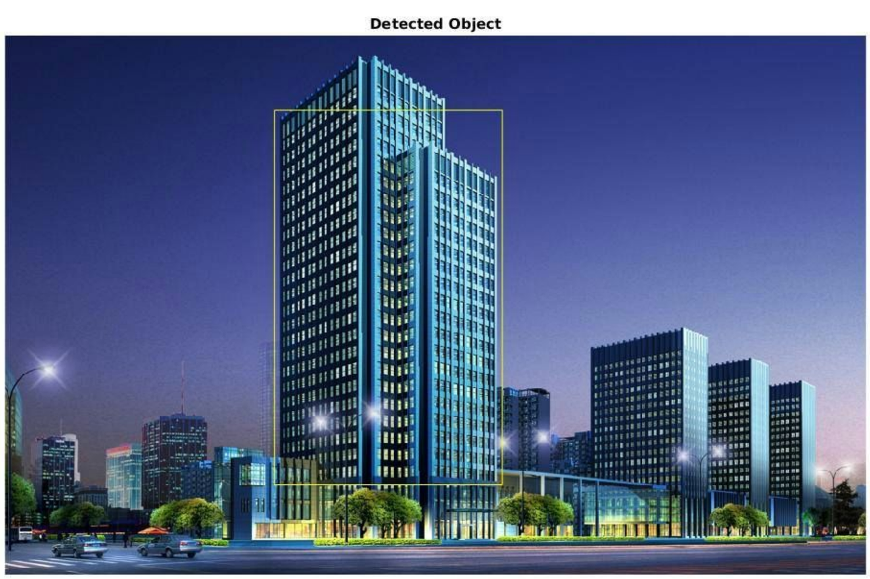
In this experiment too, we used both positive samples and negative samples to train our classifier.

The basic steps involved are:

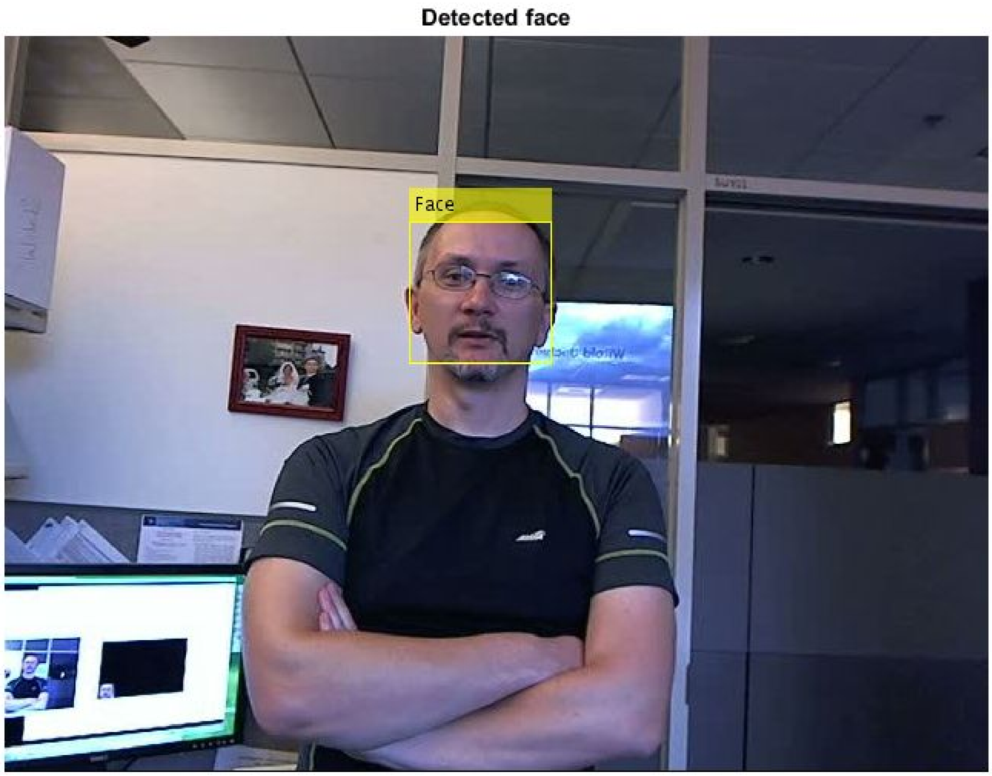
1. First step involved in this experiment was to calculate the Haar features of both the positive and the negative features. After the features were calculated, we dumped them into a feature xml file (openCV) for future use.
2. Now after we have the XML file with the required features, we trained the cascade classifier.
3. Now, the trained model was used to find the location of the face in the test image, if the face exists in the image.

Results

Feature Based Object Detection

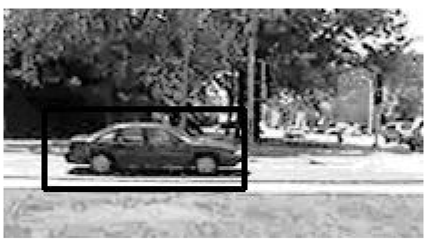


Viola-Jones object detection

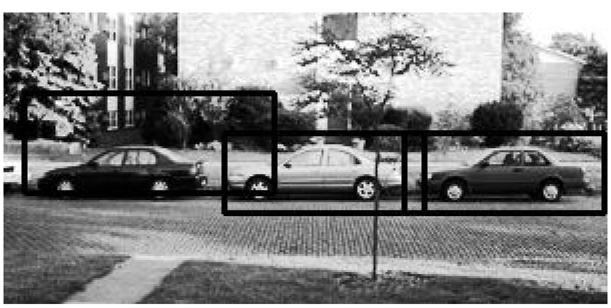


Car Detection using Threshold

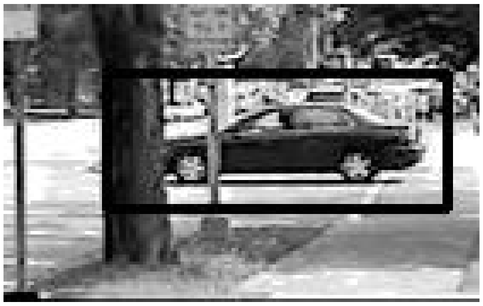
Car Detection using HOG Features



Single Car



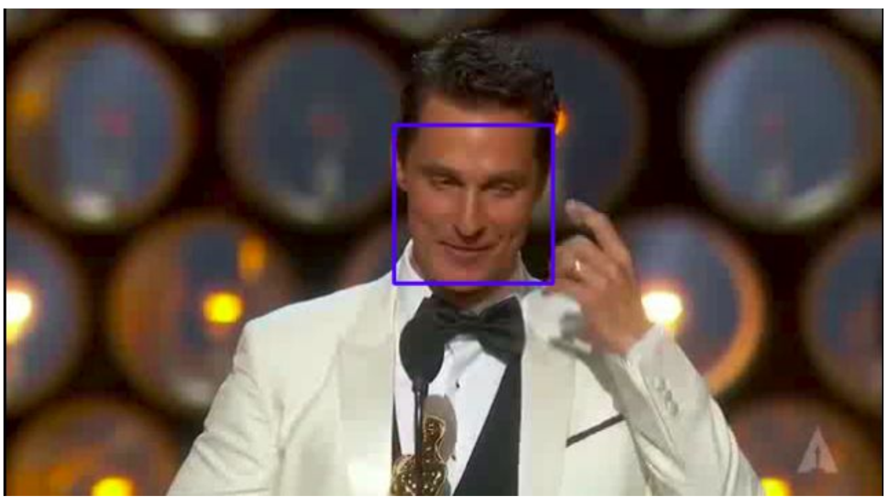
Multiple Cars

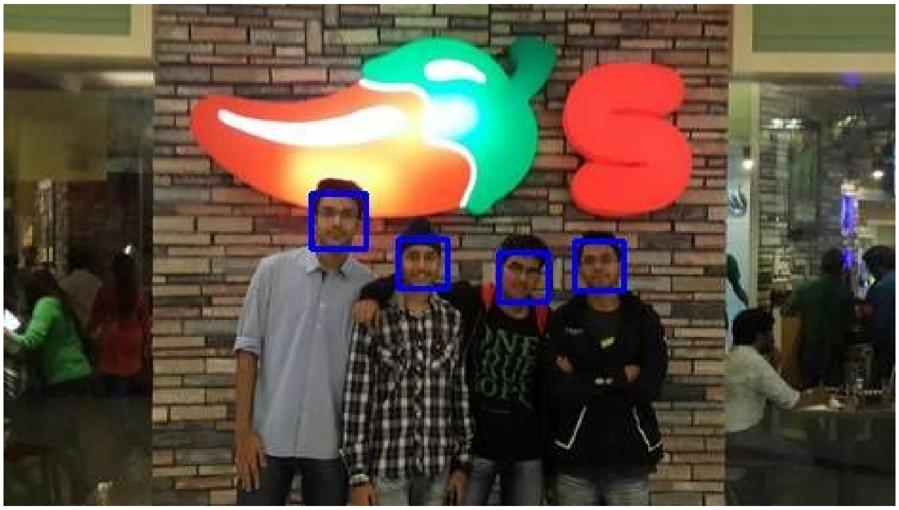


Occlusions Test case

Face Detection using Haar Cascade Classifier

Single Face





Multiple Faces