Milestone

Bhargavi Madhunala

11/08/2016

Abstract- Recognizing object by using Sliding Window model is conceptually simple: independently classify all image patches as being object or non-object. Sliding window classification is the dominant paradigm in object detection and for one object category in particular – faces – it is one of the most noticeable successes of computer vision.Mainly the Sliding window function requires three arguments. The First is the image. The second argument is stepSize. The stepsize is nothing but it indicates how many pixels we are going to use. The last argument is windowsize defines the width and height of the windows. Object detection is an important, yet challenging vision task. It is a critical part in many applications such as image search, image auto-annotation and scene understanding; however it is still an open problem due to the complexity of object classes and images. In the context of computer vision (and as the name suggests), a sliding window is rectangular region of fixed width and height that slides across an image.

Introduction

Object recognition is one of the fundamental challenges in computer vision. The main ideology of my project is to Recognize an object. So in my project am planning to add the concept of a sliding window. Sliding windows play an integral role in object classification, as they allow us to localize exactly where in an image an object resides. So by utilizing both a sliding window and an image pyramid we are able to detect objects in images at various scales and locations. By combining a sliding window with an image pyramid we are able to localize and detect objects in images at multiple scales and locations. These techniques, while simple, play an absolutely critical role in object detection and image classification. While both sliding windows and image pyramids are very simple techniques, they are absolutely critical in object detection. Suppose if image is substantially larger than your 64128 window, then you should apply an image pyramid. Object detection involves verifying the presence of an object in image sequences and possibly locating it precisely for recognition. Object tracking is to monitor an object's spatial and temporal changes during a sequence, including its presence, position, size, shape, etc. This is done by solving the temporal correspondence problem, the problem of matching the target region in successive frames of a sequence of images taken at closely-spaced time intervals. These two processes are closely related because tracking usually starts with detecting objects, while detecting an object repeatedly in subsequent image sequence is often necessary to help and verify tracking. Object detecting and tracking has a wide variety of applications in computer vision such as video compression, video surveillance, vision-based control, human-computer interfaces, medical imaging, augmented reality, and robotics. Additionally, it provides input to higher level vision tasks, such as 3D reconstruction and 3D representation. It also plays an important role in video database such as content-based indexing and retrieval.

Background

Several recent approaches have been proposed to integrate scene geometry and detection. Their main goal is to increase precision by selecting consistent detection's. However, scene geometry offers additionally a potential speed increase, if one limits the detector's search region. . Other approaches try to stabilize the camera image by detecting the horizon line or fit a ground plane to stereo measurements. From the computational side, there are two major cost items in the design of a sliding-window classifier: the evaluation of the window classifier itself and the computation of the underlying feature representation. Object is any thing we are interested in it (e.g. person, car, chair, sofa...). The hard part here, we have to find a box/window inside the image matching the target type..NOT using the whole image as it is. Blew is several images, and bounding boxes around the found objects. Mainly we need to develop function Detect(image) that will propose for us some promising boxes inside the image such that probably object exist. Then for each box, evaluate it using Classify method and pick the box that has higher score as our best candidate. Function Detect solved the Object Detection Problem, which is function to guess some boxes to have the object inside.

Methodology

In Point Tracking The detected objects are represented by points, and the tracking of these points is based on the previous object states which can include object positions and motion. Appearance tracking. The object appearance can be for example a rectangular template or an elliptical shape with an associated color. Objects are tracked by considering the coherence of their appearances in consecutive frames. This motion is usually in the form of a parametric transformation such as a translation, a rotation or an affinity. Silhouette tracking: The tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region.I've looked a whole bunch of image recognition approaches. But I'm trying to determine the best for this specific task. Most importantly, the object is made of lines and is not a filled shape. Also, there is no perspective distortion, so the rectangular object will always have right angles in the photograph.

Experiments

The choice of training data is critical for this task. While an object detection system would typically be trained and tested on a single database face detection papers have traditionally trained on heterogeneous, even proprietary, data-sets. As with most of the literature, we will use three databases: (1) positive training crops, (2) non-face scenes to mine for negative training data, and (3) test scenes with ground truth face locations. The FERET program set out to establish a large database of facial images that was gathered independently from the algorithm developers. Dr. Harry Wechsler at George Mason University was selected to direct the collection of this database.

$defsliding_window(image, stepSize, windowSize):$

$foryinxrange(0, image.shape[0], stepSize):$

$forxinxrange(0, image.shape[1], stepSize):$

$yield(x, y, image[y : y + windowSize[1], x : x + windowSize[0]])$

The experiment is fairly straightforward and handle the actual sliding of the window that defines two for loops that loop over the (x, y) coordinates of the image, incrementing their respective x and y counters by the provided step size and then returns a tuple containing the x and y coordinates of the sliding window, along with the window itself.

Analysis

If all goes well the following results can see that for each of the layers in the pyramid a window is slid across it. And again, if we had an image classifier ready to go, we could take each of these windows and classify the contents of the window.

Conclusion

In this paper we see all about sliding windows and their application to object detection and image classification. By combining a sliding window with an image pyramid we are able to localize and detect objects in images at multiple scales and locations. While both sliding windows and image pyramids are very simple techniques, they are absolutely critical in object detection.

References

Danang, V. (2013, April 18). Object Tracking.

K, S. Z. (2005, February). R. Gross, Face Databases, Handbook of Face Recognition. p. 22.

Rosebrock, A. (2015, March 23). Sliding Windows for Object Detection in Machine Learning.