Image Processing and Recognition

Project – 5

Jothin Vedre & Rahul Pasunuri

1. Introduction:

In this project, we design and application to distinguish between a bird and a squirrel on a bird feeder. There are a lot of applications which recognize animals from images. Our goal is to apply such applications to recognize objects in a video and from a camera feed.

We first implement a simple approach and then build upon it by taking hints from Related Work. We then suggest some other approaches to improve on it. We used the OpenCV library and developed the application in C++.

1. Related Work:

Our work is partially based on Kurt Grandis’ work on a similar problem. His application presented in the PyCon talk, “Militarizing Your Backyard with Python: Computer Vision and the Squirrel Hordes” tries to identify squirrels on the bird feeder and scare them away with a water gun connected to an Arduino.

Although this application is quite efficient, it does not consider camera footage from indoors, facing upwards, with a lot of noise such as trees, sunshine etc., which are prominent in the project sample videos given to us. For example, it sprays water over the area in which the bird feeder is, hoping that the squirrel will retreat. However, we believe it would be more efficient if we were to identify the exact position of the squirrel on the bird feeder.

1. Approach:

We implemented 2 different approaches to this problem. In both these approaches we use prominent computer vision techniques elaborated below:

* 1. Motion Detection:
     1. Simple Motion Tracking:

We detect motion in the video / camera feed by obtaining the individual frames and then differentiating the previous frame (P), current frame (C) and the next frame (N). This includes finding the difference D1 between C and P and the different D2 between N and C. We then apply bitwise “&” operation between D1 and D2 to obtain the pixels changes prominent in both the differences. By doing so, we obtain the common and most frequently changed pixel intensities over 3 frames. We can also improve accuracy of motion tracking by obtaining the pixel intensity changes over multiple number of frames.

We then threshold these changes to obtain a binary image. The binary image is quite noisy as the motion tracking identifies every prominent change in the video / feed, such as the bird feeder itself, the trees, sun etc. To reduce noise by some extent, we apply morphing operations such as erosion and dilation and also before we obtain the area of the motion and threshold it to identify large objects in motion. If the area of change is quite significant, we identify it as a squirrel since squirrels are quite huge in the sample video frame.

* + 1. Background Subtraction:

We also used the Background Subtraction tools available in OpenCV. This technique uses a Gaussian Mixture Model to identify motion efficiently [Zoran]. However, this tool is not very easy to customize for our needs. We intend to use this in future work.

* 1. Object Classification:

We improve upon the approach mentioned in 3.1, by trying to classify the object in motion, based on the training videos. We first obtain the area of motion and the average RGB values of this motion and consider them as features. To do this, we obtain a video file which has footage only of the squirrel. Thus, we can label the obtained set of features as a squirrel. If not, we consider it as a bird.

We now have four features by which we distinguish between a squirrel and a bird and those are the size of motion and the average values of the three color channels (RGB) of the motion. These features are those of a squirrel. Once we obtain these feature values, we label them as a squirrel, add them to a vector and feed it to the SVM classifier of OpenCV. We use the one vs the rest approach for this classification and implement it using a linear kernel. The classifier then learns the features of a squirrel and whenever it identifies similar features in the test footage, it identifies it as a squirrel.

So, in the test video, if the object in motion is predicted as a squirrel, we draw a black square around the object. If not, we just consider it as a bird and do nothing.

1. Experiments:

We have conducted experiments on both versions of our implementation. We noticed a slight improvement in the motion tracking task when we implemented the SVN classifier. However, there are quite a few false positives as the classifier predicts some birds as squirrels. We speculate that this is because of the small size of the training data.

1. Improvements:

We identified key areas in which our implementation can be improved.

* 1. Motion Detection:

The motion detection task can be improved by differentiating more than 3 frames. By doing so, we can track only the most prominent objects in motion. This can help reduce significantly reduce false positives. However, this approach won’t be helpful in time lapse videos.

* 1. Classification:

The classification task can be improved by using more videos as training data. However, this is not very efficient and might take about an hour for a 15 minute video. Instead, we can convert the video into set of images and feed the images to the training data by labelling them as squirrel or bird accordingly.

We can also improve the classification task by using a polynomial kernel or other higher dimensional kernels.

1. References:

OpenCV: BackgroundSubtractorMO2 - <http://www.zoranz.net/Publications/zivkovic2004ICPR.pdf>

<https://blog.cedric.ws/opencv-simple-motion-detection>

<http://dasl.mem.drexel.edu/~noahKuntz/openCVTut3.html#Step%202>

<https://us.pycon.org/2012/schedule/presentation/267/>

<http://pyvideo.org/video/674/militarizing-your-backyard-with-python-computer>

<http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html>

<http://docs.opencv.org/>