Image Processing and Recognition Project – 5

By
Jothin Vedre (jvedre@indiana.edu)
Rahul Pasunuri (rahupasu@indiana.edu)

Introduction:

In this project, we design an 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 [6,7] and developed the application in C++.

Related Work:

Our work is partially based on Kurt Grandi's 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.

Our Approach:

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

Approach-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 D_1 between C and P and the different D_2 between N and C. We then apply bitwise “&” operation between D_1 and D_2 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. Erosion helps in shrinking the boundary region pixels of the foreground object and dilations

helps in enlarging the boundary region pixels. So, combining both the operations will help us in getting rid of the noisy parts in the object blob.

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.

Approach-2: Object Classification:

We improve upon the approach mentioned in the previous section, by trying to classify the object in motion, based on the hand labeled images.

Data Set collection:

We have used the tool 'ffmpeg' to get screen-shots from images and manually labeled those images to one of the two categories (bird and squirrel). We have collected and marked a total of 100 images for each category for the task of training.

Some sample squirrel images:

Some sample bird images:

Features Used:
1) Area of motion.
2) Average pixel values of each of the RGB channels in the motion.
So, we get a total of 4 features for every motion.
The color information of the motion (squirrels/birds) is captured by the 2nd feature type.

Classifiers Used:
We have used SVM classifier of OpenCV for our classification purpose. We have tried both linear and polynomial kernels to measure the performance of the model.

Results:

We observed that approach-2 (SVM classifier) outperforms approach-1 (simple motion detector). Also, we observed that the polynomial kernel SVM performs better than the linear kernel SVM in our case. This might be because the feature space is not linearly separable. We mark a squirrel using a red colored rectangle and mark a bird using a blue colored rectangle.

Our model is able to detect a squirrel with an accuracy of 40% in the test set. The precision on the birds was good, but has a very low recall. We expect that training with a relatively larger number of images should improve the results a lot

Below are some correct detections made by the polynomial SVM kernel.

Below are some Mis-classifications

Improvements/Future Work

We identified some 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 significantly reduce false positives. However, this approach won't be helpful in time lapse videos.

2) Increasing Training Data Size:

The classification task can be improved by using more labeled images as training data. However, this takes a lot of human effort. Instead we can try to generate more training images from a small sample of labeled images. One such approach would be to get the principal components of bird and squirrel images separately and generating more images by adding small amounts of their corresponding principal components.

3) Background Subtraction:

We have also used the Background Subtraction tools available in OpenCV. This technique uses a Gaussian Mixture Model to identify motion efficiently[1]. However, this tool is not very easy to customize for our needs. We intend to use this in future work.

4) Cost Matrix:

Also, we can use a cost matrix for our predictions, based on the requirements and needs of the user. For example, a person might be completely paranoid about having squirrels on his bird feeder. Similarly, a person might allow some false positives for bird, but can keep a high penalty on false negatives (like missing a bird). All these can be incorporated with the help of a cost matrix, and can be taken as an input from the user. Our predictions can be modified based on the expected cost.

4) Feature Selection:

4.1) Contour Corners:

We currently use four features to distinguish between a bird and a squirrel and those are the area of the motion, the red channel value, blue channel value and green channel value. However, these features are not very efficient when there are illumination and wind factors. Hence we propose to use the number of points in the contours of the object in motion. A contour is the outline of an object in an image, which in this case, is an object in motion.

Once we obtain the object in motion using background subtraction, we can outline the object and find the number of corners in the contour. Ideally, birds have more contour corners than squirrels, given their beak and tail. Once we obtain the number of corners, we can use this value as a feature.

4.2) SIFT:

We can also use SIFT features to identify whether it is a squirrel or not. However, it remains to be seen if SIFT can be applied on object in motion. It is not possible to have a SIFT feature match for every bird or squirrel in town. We can look at the similarity of the features by calculating the 128 dimensional L2 – norm distance and threshold it.

5) Classifiers:

We achieved considerable assumption that using SVM classifiers. But we ignored the fact that the consecutive images in the video are dependent on each other. So, the I.I.D. assumption that SVM is based on is clearly not valid here. So, a graphical models like a Bayesian Network or a Hidden Markov Model can be a better choice here, as these can capture such kinds of dependancies.

Also, recently work shows the effectiveness of deep CNN's on the object detection tasks. So, running deep CNN's on this task might be worth considering. Also, a parts based model like latent SVM might be useful since, latent features can be inferred from existing features.

6) Handling varying Illumination:

As, the bird feeder resides outdoor, it is subjected to a lot of variations of illumination, which results in different levels of shadow strengths of the birds and squirrels. As, we have used color values as features, the changes in illumination and the shadow strengths poses a serious issue. As, our training set is too small, it doesn't capture images of all kinds of illuminations. So, our model doesn't work, if the illumination varies by a lot from the training data images. One way to solve this problem, is to have a sufficient set of training images which can help SVM learn all kinds of illumination patterns. An another way to solve this problem would be to do some kind of contrast or brightness adjustments.

Source Code

The source code of our implementation is publicly available in the below location.

https://github.iu.edu/rahupasu/p5

References:

1. OpenCV:BackgroundSubtractorMO2 - http://www.zoranz.net/Publications/zivkovic2004ICPR.pdf
2. https://blog.cedric.ws/opencv-simple-motion-detection
3. http://dasl.mem.drexel.edu/~noahKuntz/openCVTut3.html#Step%202
4. https://us.pycon.org/2012/schedule/presentation/267/
5. http://pyvideo.org/video/674/militarizing-your-backyard-with-python-computer
6. http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
7. http://docs.opencv.org/
8. http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm
9. http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm
10. http://www.cs.berkeley.edu/~rbg/latent/index.html