Amogh Kamal Raj
Vineet Panchbhaiyye
June 12, 2019

Project Report

Using Histogram of Oriented Gradients to detect Burrowing Owl in Camera Trap Images.

1. Introduction

Burrowing owls are diurnal owls which nest in burrows created by other animals. SFBBO(San Francisco Bay Bird Observatory) studies these owls through camera trap images. The camera traps capture images when they sense movement in their field of view. Often these camera traps are triggered due to movement of grasses because of wind or they are set to capture images at a particular interval. The camera traps deployed by SFBBO capture thousands of images in one session. These images may include owls and other animals which share the habitat. SFBBO needs help in detecting various animals present in the images. Detecting Burrowing owls in camera trap images is a difficult task because of camouflage, occlusion due to grass and other objects in the habitat and varied lighting and weather conditions.

2. Dataset

To best of out knowledge there are no labelled datasets of Burrowing owls that are publicly available. We collated our own dataset of 1584 Burrowing Owl images from camera trap images provided by SFBBO and creative commons images available on the internet, mainly flickr.com. These images also include the images created by horizontal flipping. The images from flickr were accessed using the flickr api. Following the paper by Dalal et al, we created a large dataset of negative images by randomly sampling images of the landscape where the camera traps were deployed. The negative dataset we used consisted of 17000 images without burrowing owls. A sample of images belonging to the positive class(with Burrowing Owl) and negative class(without Burrowing Owl) are depicted in Figure 1. The image size for the images in the dataset is 128x192. The image labeling task involved cropping out pixels containing the owl in the camera trap or flickr images. These cropped out samples are then resized to 128x192. The final size of 128x192 has width to height ratio of 2:3 which is more suitable to accommodate a Burrowing Owl perched on ground.

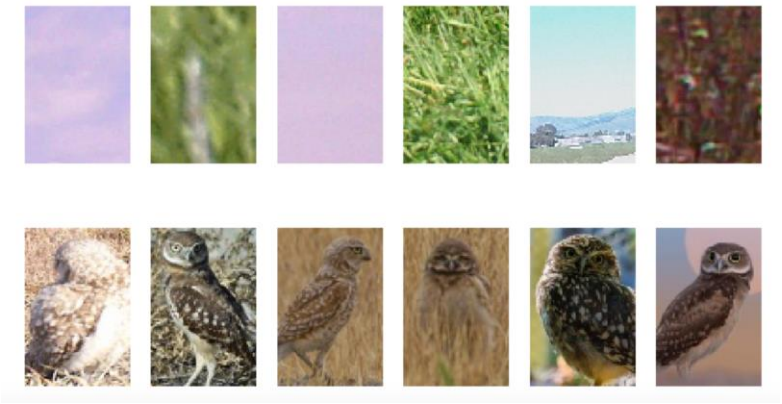

Figure 1: Sample positive and negative class images.

3. Software resources

This section describes the software resources used for the project. We used Canon Digital Professional 4.0 to crop out image sections in camera trap images containing the Burrowing Owl. The algorithm described in the project was written in Python. OpenCV library was used to calculate the HOG features in images, Hog Descriptor in OpenCV was favored over implementation of HOG from scratch as the OpenCV implementation is optimized for speed. SVM classifier available in the SciKit Learn(Sklearn) python library was utilized to train and evaluate SVM classification of HOG features. Finally, Keras with Tensorflow backend was used to implement the Convolutional Neural Network part of the project.

Figure 2: Cropping and resizing of camera trap images.

4. Algorithm

The algorithm used for this project is "Histogram of Oriented Gradients for Human Detection" introduced by Dalal and Triggs in [1]. The algorithm was proposed for Human detection in the MIT pedestrian dataset. In this project we explore if this algorithm can be used Burrowing Owl in images. The method proposed in the paper is based on evaluating well normalized histograms of image gradient orientations in dense grid. The authors assume that local object appearance and shape can be well characterized by local edge directions. An overview of the algorithm is shown in Figure 3.

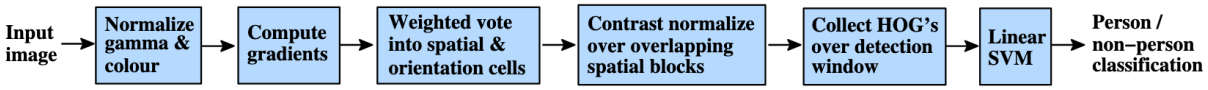

Figure 3: Steps in HOG for Human detection algorithm

The first step after the training image is normalized is to compute the gradients. Sobel filter is used to compute the gradient magnitude and orientation at each pixel in the image. In the next step the 128x192 image is divided into cells of size 8x8 as shown in Figure 4a. A histogram of gradients calculated for each cell. The gradients are binned into 9 values from 0 to180 degrees, for example 0, 20,4 0 .. so on. Both gradient magnitude and direction are considered to create the histogram. A bin is selected based on the direction and the vote, a value that determines to which bin it goes into, is selected based on the magnitude. Figure 4b. clearly explains this process. The pixel encircled in blue has a magnitude of 2 and an angle of 80 degrees, as 80 falls in the 5^{th} bin, the algorithm places 2 in that bin. Similarly, for the pixel encircled in red, as degree 10 falls in the bins 0 and 1 (i.e., 0 to 10 and 10 to 20), the corresponding magnitude 4 will be split into two and will be placed in both the bins. Calculating a histogram over a patch makes the algorithm more robust to noise as individual gradients may have noise but a histogram over 8 x 8 image makes if less sensitive to noise.

The gradient which was calculated in the previous steps is very sensitive to lighting. If the image is made darker or lighter the corresponding gradient magnitudes will change and therefore altering the histogram values. In order to make the algorithm sensitive to these lighting changes block normalization of the histogram is performed. Usually, a vector is normalized by dividing individual elements of the vector by its magnitude. Same process is applied to compute the normalized values of 9x1 histogram. The normalization process is carried out on a 16x16 block highlighted by a red square in Figure 4a.A 16×16 block has 4 histograms which can be concatenated to form a 36 x 1 element vector which can be then be normalized. The window is then slide by 8 pixels highlighted by a blue square in Figure 4a, and a normalized 36×1 vector is calculated over this window and the process is repeated for the whole image.

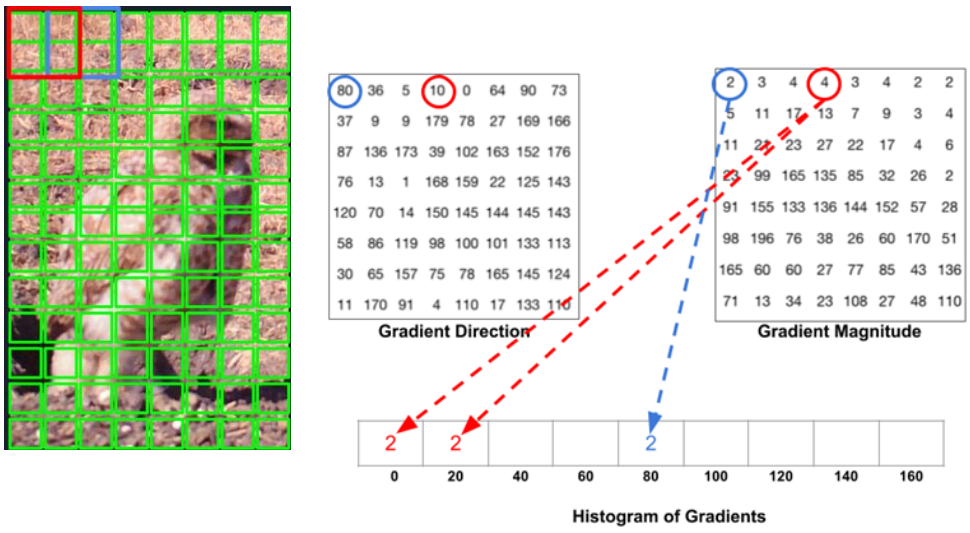

Figure 4a[left] 4b[right]: 8x8 cells shown on for a sample training image.

Final step is to calculate the HOG features which form the inputs to the SVM classifier. All the 36 x 1 feature vectors for a 16 x 16 block which were calculated in the previous step are concatenated into a single [23x15x36] = 12420-dimension vector. We used these HOG features to train and evaluate the SVM Classifier. The results are presented in section 5 below.

5. Owl detection in camera trap images

Most camera trap image resolution is in Megapixel scale. The dimension of the images in out held-out test set is 2592 x 1944. Our HOG-SVM classifier expects an input of 128x192, we use a sliding window approach to detect the owl in camera trap images. In this approach, we crop 128x192 sized sections of the image starting from the top left corner and passing it to the trained HOG-SVM classifier to predict if the particular section contains an owl or not. The next crop is obtained by sliding the crop window by 32 pixels along the row and 48 pixels down after the completion of a row. Once the whole image has been covered, we scale the crop width and height by 2. Also, the slide length along the rows and columns is doubled. This is done until either crop width or crop height is half of the input(camera trap) image width or height respectively.

6. Results

We trained and evaluated our HOG-SVM classifier using Linear SVM and SVM with a polynomial lifting function with degrees 2,3 and 4. These trained classifiers where evaluated using test set (391

images in positive class and 3750 in negative class). The results of these evaluation are presented in Tables 1,2, 3 and 4 below.

	Prediction	
True label	No Owl (0)	Owl(1)
No Owl (0)	4194	60
Owl (1)	88	304

Linear	precision	recall	f1-score	support
No owl = 0	0.98	0.99	0.99	4502
Owl = 1	0.84	0.8	0.82	394

Table 1: AP Matrix and Precision - Recall for Linear SVM classifier

	Prediction	
True label	No Owl (0)	Owl(1)
No Owl (0)	4218	36
Owl (1)	84	308

POLY (Degree =2)	precision	recall	f1-score	support
No owl = 0	0.98	0.99	0.99	4254
Owl = 1	0.90	0.79	0.84	392

Table 2: AP matrix and Precision - Recall for SVM classifier with Degree 2 polynomial lifting function

	Prediction	
True label	No Owl (0)	Owl(1)
No Owl (0)	4234	20
Owl (1)	84	308

POLY (Degree =3)	precision	recall	f1-score	support
No owl = 0	0.98	1	0.99	4503
Owl = 1	0.95	0.8	0.87	395

Table 3: AP matrix and Precision - Recall for SVM classifier with Degree 3 polynomial lifting function

	Prediction	
True label	No Owl (0)	Owl(1)
No Owl (0)	4234	16
Owl (1)	73	323

POLY (Degree =4)	precision	recall	f1-score	support
No owl = 0	0.98	1	0.99	4250
Owl = 1	0.95	0.83	0.88	396

Table 4: AP matrix and Precision - Recall for SVM classifier with Degree 4 polynomial lifting function

Both the classifier has nearly same number of support vectors, but the precision achieved for the polynomial SVM classifier is higher than the linear SVM classifier.

Figure 5 and 6 show some successful and failed owl detection results. The right image in Figure 5 shows that HOG-SVM detects grass at a number of locations as owl. Similarly, in the image on left the owl though occluded is detected but there are other false positive detections.

In Figure 6, right side image the classifier fails to detect the owl and has labelled surrounding patches as owl.

Besides the Linear SVM classifier and Polynomial SVM classifier, we also experimented with Gaussian or Radial Basis Function SVM classifier, but these classifiers failed to converge.

Figure 5: Sample results of owl detection on Camera trap images

Figure 6: Sample results of owl detection on Camera trap images.

Figure 5 and 6 show some successful and failed owl detection results. The right image in Figure 5 shows that HOG-SVM detects grass at a number of locations as owl. Similarly, in the image on left the owl though occluded is detected but there are other false positive detections.

In Figure 6, right side image the classifier fails to detect the owl and has labelled surrounding patches as owl.

Besides the Linear SVM classifier and Polynomial SVM classifier, we also experimented with Gaussian or Radial Basis Function SVM classifier, but these classifiers failed to converge.

7. CNN classifier

Along with HOG-SVM based classifier we also explored a Convolutional Neural Network(CNN) based classifier to achieve better performance. We retrained VGG16 to build a classifier which classifies an image as owl or no owl, this process is also known as transfer learning[2]. VGG16 is a popular CNN for image classification. VGG16 architecture has 16 trainable layers. Figure 7 show various layers in the VGG16 architecture.

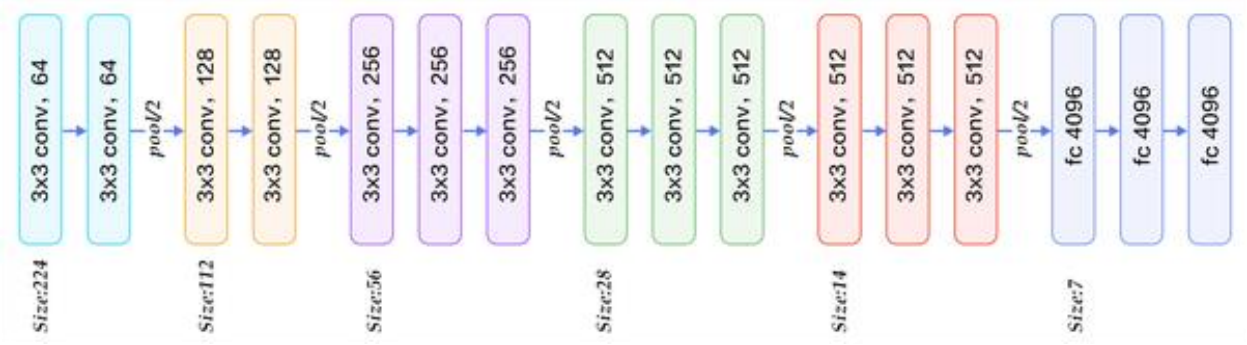

Figure 7: VGG16 Architecture

For our evaluation we used pretrained VGG16 weights, the weights. We modified the original VGG16 to fit the Owl Detection task. Since the input size of image to VGG16 is 224x224, the cropped owl images are resized to 224x224. Along with this, we use only one fully connected layer of size 128 instead of 3. In all, the last two Conv layers and the 128 sized fully connected layer were fine-tuned during the process. We achieved a training accuracy of 97.27% , overall the CNN solution performed better than HOG-SVM solution as it used color information in images.

Figure 8: Sample output of VGG16 based owl detector.

8. Conclusion and future work

Burrowing owls have evolved to camouflage well in their habitat. As a result, often the general characteristic orientation of edges formed by the owl along with its speckled plumage is similar to the objects found in its surrounding like clumps of grass, dry earth and rocks. Human detection in an urban landscape does not have this challenge. Figure 9 compares HOG descriptor of an owl and patch of grass.

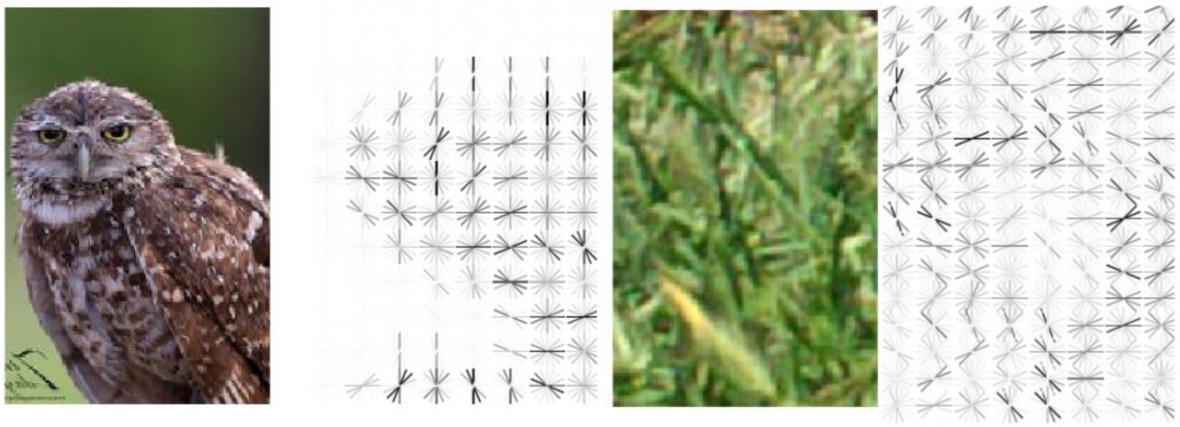

Figure 9: HOG descriptor of a Burrowing owl and a patch of grass.

During the preparation of our dataset, we observed that the average width of the cropped images was 326.46 pixels while the average height was 490.29 pixels. Considering this we decided to train a HOG-SVM classifier for 128x192 image size. We also trained a classifier for 64x96 images, though this classifier has better metrics on the validation data, but it performs poorly in the held-out dataset. This results from the down sampling of image patches to 64x96.

Figure 10 compares output of 128x192 and 64x96 owl detection on the same input image, both the SVMs were trained with the same parameters, the 128x192 HOG descriptors lead to better classification results.

Since HOG descriptor is calculated for on grayscale image the color information is lost in the process. As suggested during the final presentation providing color information to the SVM classifier can improve the classification accuracy. This can be achieved by converting the image to YUV 4:2:2 or 4:2:0 color space and computing the HOG for each channel independently. These can then be concatenated, and the resulting feature vector can be used to train the SVM classifier to obtain better performance.

Figure 10: Left 128x192 HOG-SVM detector vs Right 64x96 HOG-SVM detector.

9. References

1. Histogram of Oriented Gradients for Human Detection, Naveen Dalal and Bill Triggs, CVPR 2005, https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf
2. Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015, https://arxiv.org/abs/1409.1556