CSE252: Bird's eye view: Detecting and Recognizing Birds using the BIRDS 200 dataset

Rohan Anil
University of California, San Diego
ranil@cs.ucsd.edu

Abstract

In this project, we benchmark the Naive Bayes-Nearest Neighbor (NB-NN) algorithm on the Birds 200 dataset[6] using C-SIFT feature descriptor. Primary motivation is to provide a comprehensive set of baselines and experiments for further research on the dataset. In addition, we built a bird detector cascade for real time detection of birds in images and videos which is compatible with OpenCV. Finally, We explore on how to include side-information for which we use a log-linear model trained as a post processing step. Our preliminary results using NB-NN with C-SIFT descriptors on a subset of the dataset gives an accuracy of 25%.

1. Introduction

There has been a lot of attention paid to the problem of object recognition and segmentation in the last decade [4] [12] [10] [3] [8] [2] and which resulted in the development of different types of feature descriptors [6,7,8,9,10] during this time. One of the reasons for different types of feature descriptor is because there is no established evidence of superiority of any single feature over the others.And the performance is tied to the specific problem that it is applied to. In this project, we evaluate performance of C-SIFT feature descriptors combined with Naive Bayes- Nearest Neighbor method. We also train a bird detector cascade trained using Ada-boost [7],[12] using OpenCV [5]. Finally we propose an efficient way to include side-information to object recognition pipelines by training a log-linear model with side-information included in the feature vector.

2. Dataset statistics

The Caltech-UCSD Birds 200 (CUB-200) is an image dataset with photos of 200 bird species (mostly North American)[1]. There are 200 categories of birds and a total of 6,003 bird images. The dataset also provides bounding box, rough segmentation and attribute information for each of the image.

Figure 1. Bird Images from Dataset

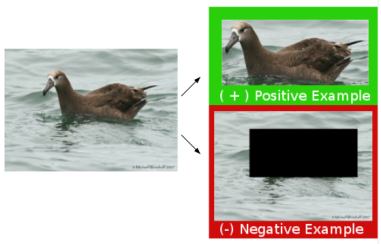

Figure 2. Preprocessing Dataset

3. Dataset preprocessing

Since we are interested in detecting and recognizing birds, we first extract region of interests of birds from images using the bounding box information available in the dataset. We use this as input to the feature descriptor extraction stage for NB-NN method. For training the bird cascade we use these clipped images as the positive examples and the images with clipped regions blacked out as the negative examples as shown in figure 2.

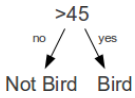

Figure 3. Binary Classifier

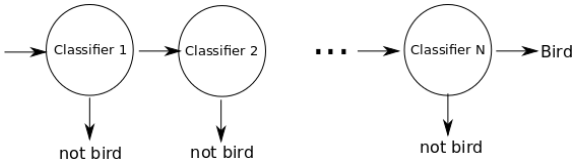

Figure 4. Cascaded classifier

4. Software

The OpenCV[5] haar-training utility was used to train the bird cascade. The C-SIFT (Colored SIFT) descriptor for images were computed using the implementation available here [11]. The Naive Bayes-Nearest neighbor method was implemented in C++.

5. Object Detection

For object detection i.e bird detection we use the Viola & Jones [12] algorithm which uses haar-like features trained using adaboost [7] which trains a cascade of weak classifiers to create a strong one. In this project, we use a binary threshold classifier to train the bird cascade. The features we use are illustrated in the figure 5. The method exploits the use of integral image to calculate features fast and hence the detection runs at real-time.

5.1. Integral Image

Computing the features requires summing pixel values at different regions of the image. Integral Image is trick used to make the computation faster. First, Integral image is computed from the orginal image where

$$\text{Int}(x, y) = \sum_{i=1}^{x} \sum_{j=1}^{y} \text{I}(i, j)$$

where Int is the integral image. Then the sum of pixel values between co-ordinates $((x_1, x_2), (y_1, y_2))$ can be written as Sum $= Int(x_2, y_2) - Int(x_1, y_2) - Int(x_2, y_1) + Int(x_1, y_1)$ which is an $O(1)$ computation. For features which needs sum at angle of 45° a similar derivation is possible called RSAT(Rotated Sum Area Table).

5.2. Cascaded Classifiers

While training, we create a binary threshold classifier as shown in figure 3. at each stage using discrete-adaboost. And for prediction these classifiers are cascaded in the manner shown in figure 4.

6. Object Recogntion

For object recognition, we use the NB-NN [3] together with C-SIFT feature descriptor.

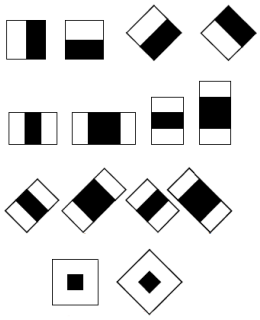
Figure 5. Haar-like features

6.1. C-SIFT Feature Descriptor

The C-SIFT (Colored SIFT) feature descriptor was proposed in [1] which is a variant of the SIFT feature descriptor and incorporates color information. Our dataset contains images of birds for which color is highly predictive attribute for recognition. Hence, we chose C-SIFT since it includes color information and is also invariant to color and photometrical variations [1]

7. Naive Bayes-Nearest Neighbor (NB-NN)

The NB-NN is a non-parametric method which has been successfuly employed for image retrieval tasks [3] and out perform bag of words based approaches on certain datasets. There are two main advantages of NB-NN mthod i) it is non-parametric and ii) it does not cause any of lose of information which bag-of-words quantization causes. In the NB-NN method there is no learning stage, and for prediction - NB-NN uses Image-to-Class distance. The descriptors found in the test image is matched against all the descriptors in each of the class and the minimum distance is recorded. The final prediction is given as the following

$$\hat{C} = \text{argmin}_C \sum_{i}^{n} ||d_i - NN_C(d_i)||^2$$

where d_i , i = 1..n are the descriptors computed from the test image. $NN_C(d_i)$ is nearest neighbor the distance function. We use euclidean distance in our experiments.

Class	1	2	3	4	5	6	7	8	9	10
1	**14**	0	0	0	0	0	0	0	7	0
2	6	**1**	1	1	0	0	0	0	2	0
3	5	0	**2**	1	0	0	0	0	3	0
4	3	0	0	**6**	0	0	0	0	11	0
5	1	0	0	3	**0**	0	0	0	6	0
6	3	0	0	0	1	**0**	0	0	9	0
7	1	0	0	1	3	0	**0**	0	13	0
8	8	0	0	0	0	0	0	**5**	4	0
9	3	0	0	1	0	0	0	0	**7**	0
10	4	0	0	0	0	0	0	0	7	**1**

Table 1. Confusion Matrix for NB-NN with C-SIFT

8. Experiments & Results

8.1. Bird Detection

The cascade[1] was trained using 100 positive examples and 100 negative examples. The first 100 images from training was used to generate these examples. Although we were able to generate the cascade, We could not evaluate its performance in time which we have left for future work.

8.2. NB-NN using C-SIFT

In this experiment, we use a subset which contains the first 10 categories of birds from the entire dataset. We follow the same training and testing splits used in [6]. The confusion matrix in table 1 was obtained and an overall accuracy of 25%.

8.3. Side Information

For including side-information, we train a log-linear model as a post processing step on the training examples as illustrated in figure 6. This work is similar to [9] where authors have used log-linear model to encode side-information in a collaborative filtering problem. Log linear model has the following form

$$p(c|\vec{x}) = \frac{\exp(\hat{w_c} \cdot \vec{x})}{Z}$$

where $Z = \sum_c \exp(\hat{w_c} \cdot \vec{x})$, c is the object class and $\vec{x}$ - feature vector encodes the side information and the probability distribution for each of the classes which we calculate from the previous stage of the object recognition pipeline for each of the training example .

[1] http://cseweb.ucsd.edu/ ranil/birds/cascade.xml

8.3.1 Training

For learning the weight vectors we can use stochastic gradient descent (SGD) to minimize the negative log-likelihood for each of the training examples.

$$L(X) = -\sum_i \log(p(c_i|\vec{x_i}))$$

The update rules for SGD for the training example (x_i, c_i) is

$$w_{c_i,k} = w_{c_i,k} - \lambda((p(c_i|\vec{x_i}) - 1) \cdot x_{ik} + \nu \cdot w_{c_i,k})$$

$$w_{c_{j\neq i},k} = w_{c_{j\neq i},k} - \lambda((p(c_{j\neq i}|\vec{x_i})) \cdot x_{ik} + \nu \cdot w_{c_{j\neq i},k})$$

where k ranges from one to the length of the feature vector.

8.3.2 Issues with Training

There are a few issues related to over-fitting that can arise using this method. One is if the side information is unique to a particular class, It is most likely that weights for those features will high. There are two ways to deal this problem - one way is to have high regularization parameter ν for side information features - the second way is to carefuly select those features as side-information that is not unique to particular class. There could also be a case where side-information is unique to a class and highly discriminatory in which case over-fitting is preferred. Currently we are experimenting with using annotation as side-information.

9. Conclusion and Future work

In this project, we worked on two areas i.e bird detection using haar-like features and NB-NN with C-SIFT descriptor for bird recognition. The NB-NN gave comparable performance to the current state-of-art results on the dataset. We also discussed on how to include side-information in the recognition pipeline. We have left the comparision of NB-NN against bag-of-words based methods and creating and evaluating bird detection cascade using the entire dataset as future work.

10. References

References

[1] A. E. Abdel-Hakim and A. A. Farag. Csift: A sift descriptor with color invariant characteristics. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 1978–1983, Washington, DC, USA, 2006. IEEE Computer Society.

[2] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

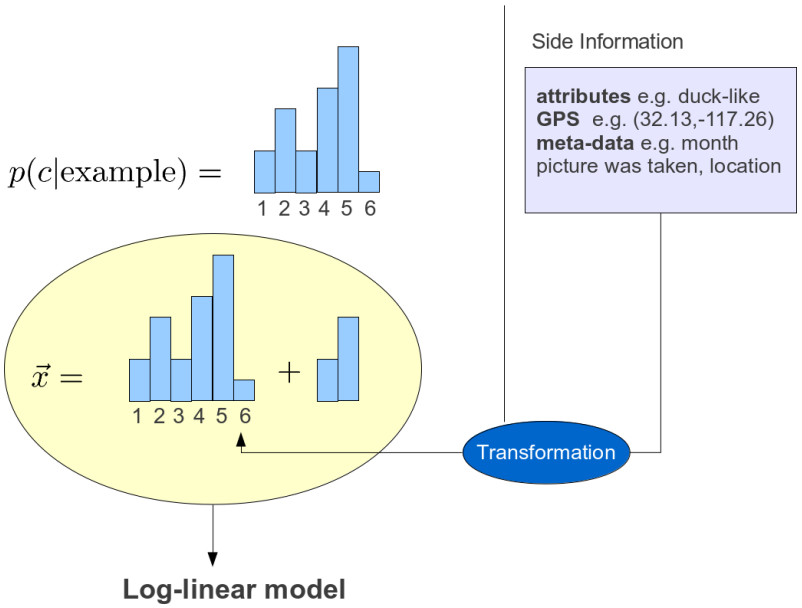

Figure 6. Side Information

[3] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*. IEEE Computer Society, 2008.

[4] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision (ECCV)*, Heraklion, Crete, Sept. 2010.

[7] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.

[8] D. Lowe. Object recognition from local scale-invariant features. pages 1150–1157, 1999.

[9] A. K. Menon and C. Elkan. A log-linear model with latent features for dyadic prediction. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 364–373, Washington, DC, USA, 2010. IEEE Computer Society.

[10] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context, 2007.

[11] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[12] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001.