

Bird Species Identification from an Image

Aditya Bhandari,¹ Ameya Joshi,² Rohit Patki³

¹Department of Computer Science, Stanford University

²Department of Electrical Engineering, Stanford University

³Institute of Computational Mathematics and Engineering, Stanford University

This document is the final project report for the CS 229 Machine Learning course at Stanford University. The project aims to quantify the qualitative description of different bird species using machine learning techniques and use it as an effective tool for bird species identification from images.

1 Introduction

Identification of bird species is a challenging task often resulting in ambiguous labels. Even professional bird watchers sometimes disagree on the species given an image of a bird. It is a difficult problem that pushes the limits of the visual abilities for both humans and computers. Although different bird species share the same basic set of parts, different bird species can vary dramatically in shape and appearance. Intra-class variance is high due to variation in lighting and background and extreme variation in pose (e.g., flying birds, swimming birds, and perched birds that are partially occluded by branches).

Our project aims to employ the power of machine learning to help amateur bird watchers identify bird species from the images they capture.

2 Dataset

Caltech and UCSD have gathered data to produce the "Caltech-UCSD Birds-200-2011 (CUB-200-2011)" dataset [3]. The dataset contains 11,788 images of 200 bird species. The list of species names was obtained using an online field guide. Images were harvested using Flickr image search and then filtered by showing each image to multiple users of Mechanical Turk.

3 Features

A vocabulary of 28 attribute groupings and 312 binary attributes (e.g., the attribute group "belly color" contains 15 different color choices) was selected based on an online tool for bird species identification. All attributes are visual in nature, with most pertaining to a color, pattern, or shape of a particular part. Some examples of attributes are:

- *has_back_color::red*
- *has_bill_shape::cone*
- *has_wing_shape::pointed-wings*

4 Algorithms

We realized that the essence of the project was to understand the intricacies of different machine learning algorithms and to learn which algorithm gives good results for which use case. With this philosophy, we wrote our own implementations of KNN and Naive Bayes in MATLAB. An added advantage of not using any library was that we could tweak whatever parameters we wanted to. Looking at the results of these two algorithms, we got a baseline for future techniques that could be implemented using available libraries.

We observed that libraries like *Scikit Learn* allowed us to tweak different aspects of an algorithm, but maybe not to the extent of our own implementation of the algorithm. We faced an inherent trade-off between tweaking ability and the number of algorithms that could be implemented and tested in the time frame of the project. We chose trying out numerous algorithms using the *Scikit Learn* library [2] in **Python**:

1. Naive Bayes
2. Support Vector Machines
3. K-nearest Neighbors
4. Linear Discriminant Analysis (LDA)
5. Decision Trees
6. Random Forests
7. One versus Rest classifiers with Logistic Regression

Based on the results obtained, we chose the best three techniques to improvise on. We used various feature selection and feature reduction techniques to see if we can improve the accuracy further. We started with changing kernels for SVM - Linear and Radial Basis Functions. Next, we did feature reduction using PCA and applied SVM, Logistic Regression and LDA on the reduced features. We

then used feature selection techniques like L1 based method, removing features with low variance, univariate feature selection and tree based feature selection. A slight improvement gave us hope and we decided to play with it more. We used PCA for feature reduction followed by feature selection to obtain a new feature data. On this data, we implemented LDA, Logistic Regression and SVM. This improved the accuracy further.

In the end, we tried including the certainty values of features into our model, that is, we converted the original binary feature data into 8 discrete values between 0 and 1 based on the certainty. On running algorithms on this data, no significant change was observed.

5 Results

We trained and tested our algorithms on the complete data set to start with. Later we randomly separated the data set into training data and test data so that we had samples from each class. 70% of the data was used as training data and 30% was used as test data. The following figures and tables show the results we observed on implementing algorithms as mentioned in the above section.

Figure 1 shows the training versus testing accuracy for different learning methods that we implemented. Figure 2 shows the testing accuracy using different techniques on three of the learning methods - LDA, SVM and Logistic Regression.

Table 1: Results table

Method	Training Accuracy	Testing Accuracy	Using certainty metric	Using PCA	Using Feature Selection	Using PCA + Feature Selection
Naive Bayes	33.07	19.22				
KNN	45.43	31.18				
Decision Trees	99.83	24.35				
Random Forests	99.39	33.58				
LDA	63.56	45.44	46.73	47.81	47.7	47.38
SVM	50.67	43.91	48.15	48.74	49.11	46.93
Logistic Regression	84.42	51.61	52.42	53.31	51.02	53.65

Figure 1: Training vs Testing Accuracy

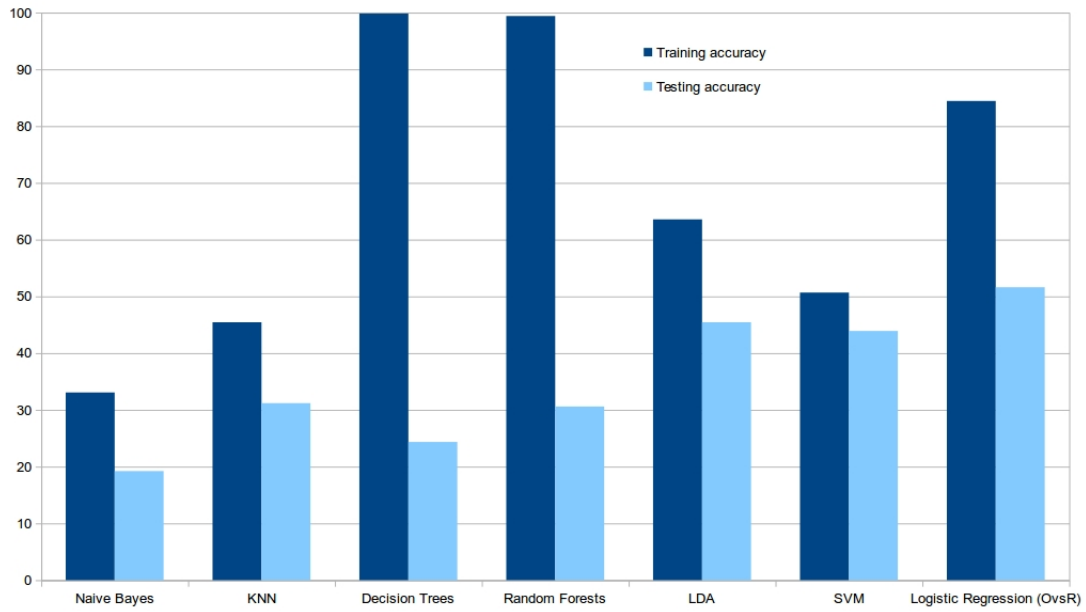
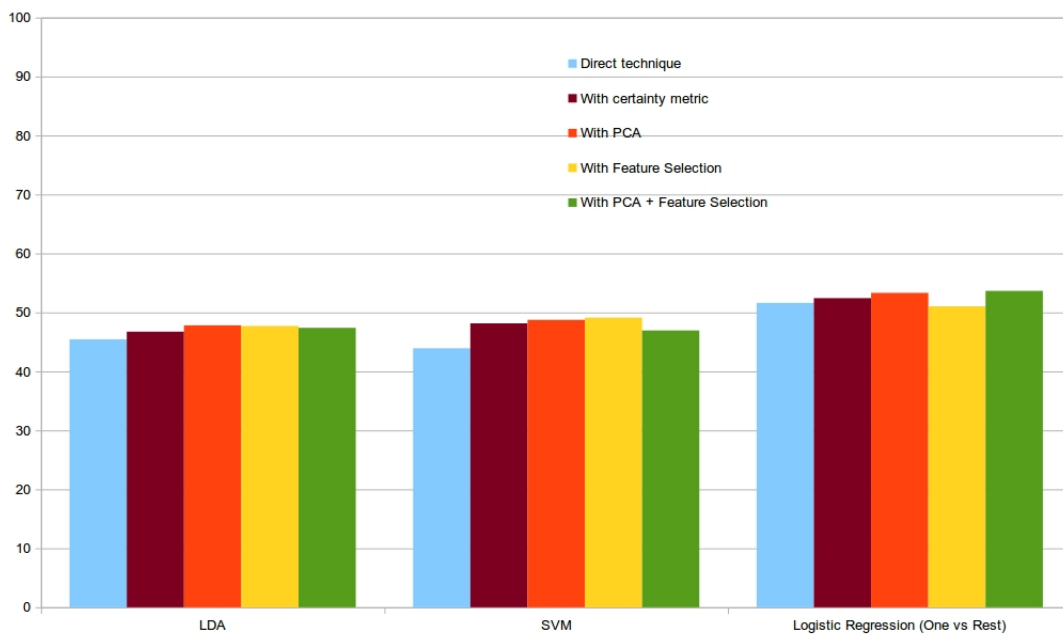


Figure 2: Testing Accuracy wth different techniques



6 Discussion

We initially observed low accuracy with basic implementation of Naive Bayes and KNN in MATLAB. We then observed improved accuracy with library implementations of SVM, LDA and Logistic Regression. Feature selection and feature reduction improved the accuracy to 53%. We believe such an accuracy for a 200 class classification problem is fairly decent.

Table 2: Comparison with related published work [1]

Feature Extraction Method	Learning Method	Percentage Accuracy
MTurks	Logistic Regression	53.65
Computer Vision	SVM	51.0
Computer Vision	Logistic Regression	65.0
Computer Vision	SVM+CNN	75.7

7 Future Work

1. We implemented Neural Networks and when we ran it on our machine for just 5 hidden neurons, it went out of memory and could not complete. So, we can try to run Neural Networks on high performance computing machines.
2. Computer vision algorithms can be used for automatic feature extraction.
3. We can develop an Android/iOS application that identifies a bird in real time on clicking its photo.

References

- [1] Steve Branson et al. “Bird Species Categorization Using Pose Normalized Deep Convolutional Nets”. In: *CoRR* abs/1406.2952 (2014). URL: <http://arxiv.org/abs/1406.2952>.
- [2] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [3] C. Wah et al. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.