CSci 5525 Course Project: Naive Bees Classifier (Group 16)

Abhinav Sambasivan, Sirisha Rambhatla, and Sathishkumar Chellakuttigounderkulandai

December 17, 2015

1 Introduction

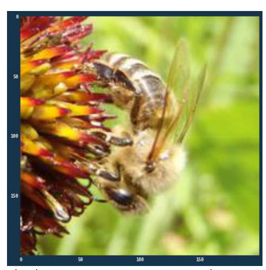
This project is motivated from the "Naive Bees Classifier" challenge by Metis [1] hosted by Drivendata.org. Wild bees are critical to sustain pollination. Currently, identifying the genus of a bee requires expert intervention. Automating this process, such that simply a picture of a bee can be used for classification by a machine learning algorithm, will greatly speed-up the study of the bee populations. As part of this project, we aim to develop classifiers to accurately identify the genus of a bee as either a "Honey Bee" (class 0), or a "Bumble Bee" (class 1) from its photograph.

The report is organized as follows. We first discuss the specifics of the problem, the given dataset and the associated challenges in Section 2. Next we look at various ways to counter the challenges posed by the said dataset in Section 3. We described the feature selection methodology in Section 4. In Section 5, we present our results. Finally, we conclude by presenting the key observations and potential next steps in Section 6.

2 Challenges

Figure 1 shows representative images from the data set and the data distribution table for both classes provided by Metis [1]. Unlike the sample image shown below, the images from the dataset has issues like occlusions, rich backgrounds, difference in scaling, skewed distribution of the data, etc. which are discussed in detail below.

(a) Bumble Bee (Bombus)



(b) Honey Bee (Apis)

Available Data	
Class	Class Size
Bumble Bee	3142
Honey Bee	827

Figure 1: Sample image data and their distribution provided by Metis [1].

2.1 Size of the dataset

The size of the given dataset is very small (3969) as compared to the ambient dimensionality of the image (which was $200 \times 200 \times 3$). In addition, the class distribution of honey bees to bumble bees is about 1:4 i.e. there is a skew in the distribution.

2.2 Challenging images

Owing to the fact that the images were collected from different photographers, both professional and amateur, the quality of the images vary drastically. Among well focused images, it was common for a part of the bee to be hidden in the background or cropped from the photograph. In some cases it is even difficult for humans to spot the bee, leave alone classify it accurately.

2.3 Distinguishing between the two genus

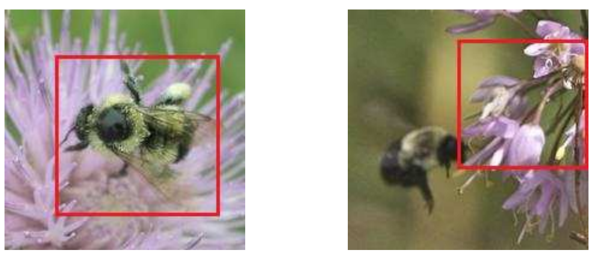
The main feature that distinguishes honey bees from bumble bees is the light brown striations on their abdomen. The bumble bees on the other hand are darker, broad and more rounded. Otherwise, the two genus of bees are not significantly different in appearance from each other.

3 Data Preparation

In order to counteract the challenges mentioned above, we run our classification algorithms on the original dataset and also on two different preprocessed versions of the same. We use the preprocessing techniques, as described below.

3.1 Region of Interest (ROI) Cropping

Since in most cases, the ROI (the bee) occupies a small portion in the whole image, we consider cropping the bee from the image. This way we avoid processing the portion of image which does not contain the bee. To achieve this, we use the region covariance descriptor [2], where we use a set of bee templates (available online) to identify the regions of the image which contain the bee. This method first constructs a suitable feature vector (for both the template image and candidate image segments), and then estimates an empirical covariance matrix for the feature vectors, and finally compares the relative distances of the covariance matrices from that of the template images. Upon further inspection, we observe that this technique isn't always successful in cropping out the bee, owing to the different configurations the bee may be in and presence of rich backgrounds. Fig. 2, shows the ROI cropping obtained by this method.



(a) Successful cropping (b) Unsuccessful cropping

Figure 2: Output of the cropping done using Region Covariance descriptor

3.2 Data Augmentation

As the given dataset is skewed, we employ data augmentation on samples of class 0 to make the distribution of labels to be roughly uniform. We perform data augmentation on the Honey bee (class 0) images by adding noise (zero mean AWGN with variance of 0.01), and by introducing rotations (-90 and 90 degrees). This quadruples the size of class 0 (we get 3308 images from 827). We next process this augmented dataset, which now contains 5955 samples, to obtain the feature vectors as described in the following section.

4 Feature Selection

Our feature extraction methodology is motivated from the sample preparation template provided by Metis [1]. It is essential to create features which capture the striation like patterns (also the distinguish the two classes) in the image. Hence in addition to utilizing the raw RGB pixel values, we add informative features as described below.

4.1 Feature extraction

First, we convert the RGB image to grayscale, and apply histogram equalization to each image. Then, we obtain features using histogram of oriented gradients (HOG) and DAISY feature descriptor [3] (based on HOG). In addition we use the raw pixel values (RGB) to form our features.

4.2 Dimensionality reduction

The size of the feature set obtained from the previous step is very large compared to that of the dataset (168046 features). So, we performed dimensionality reduction using PCA and extract the most significant features for each image. The number of principal components choose to keep plays a very crucial role in the classification performance, this is discussed in detail in the following sections.

5 Results

In this section, we present the quantitative results of our project. For our analysis we choose the following three classifiers - kernel support vector machine (k-SVM) with rbf kernel, sparse logistic regression (S-LR), and random forest (RF). We report and analyze the performance (AUC and train/test error rates, where applicable) of the said classifiers on each dataset (the original, augmented, and cropped), and compare it with the benchmark provided by Metis [1]. Note that this benchmark is a result of k-SVM on original dataset with all PCA features. Further, we analyze the effect of number of features (principal components, in our case) on the AUC and train/test error rates.

5.1 Error metrics

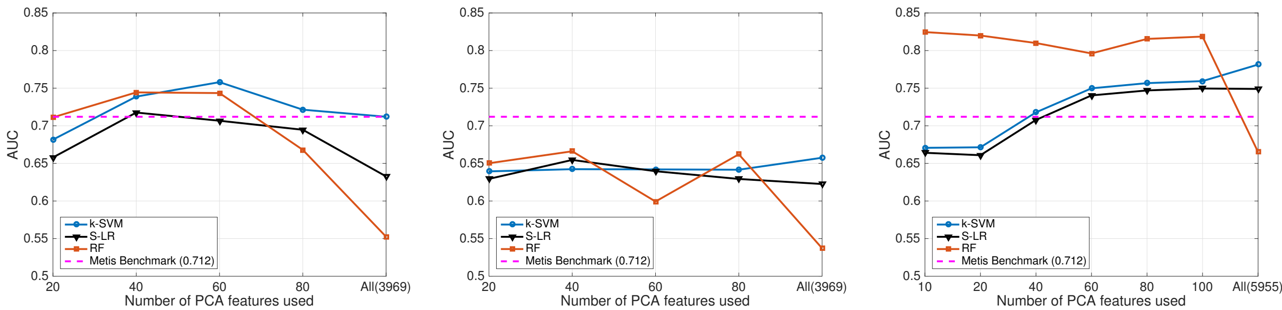
As mentioned previously, training data available for each class is very skewed (honey bees to bumble bee ratio is about 1:4). Thus, error rate is not an indicator of the performance of a classifier (at least for the original data and cropped data case, where dataset is skewed). Hence, in addition to error rates, we use the area under the receiver operating characteristics (ROC) curve, henceforth referred to as AUC, as a performance metric. ROC is a 2-D plot of True Positive Rate(TPR) against False Positive Rate(FPR), and the area under the ROC curve is indicative of how good the classifier is, also this metric is insensitive to skewed datasets.

5.2 Experimental set-up

We use Python's machine learning package, `scikit-learn`'s [4] implementations of k-SVM, S-LR, and RF. We use cross-validated grid-search over a parameter grid to find the best parameters for each classifier. Also, we use code from `opencv-covariance-features` [5] to implement the region covariance descriptor [2]. Further, we use `minimal bag of visual words image classifier` [6] to analyze the k-SVM with bag of image words algorithm.

5.3 Analysis

Panels (a)-(c) of Fig. 3 shows the performance of classifiers (k-SVM, S-LR, and RF) on original, cropped, and augmented datasets, respectively, with varying number of features (principal components). In Fig. 3 (a), the original data case, we observe that all classifiers beat the Metis baseline at some point, with k-SVM achieving the best performance of `0.758` operating on `60` PCA features. It is interesting



(a) (b) (c)

Figure 3: AUC performance of the classifiers (k-SVM, S-LR, and RF) on (a) the original dataset, (b) the dataset formed by cropping the bees, and (c) the augmented dataset. The baseline AUC of 0.712 was provided by METIS. The best AUC reported in the progress report was 0.63.

to note that RF's performance is pretty close to the baseline even with 20 features, we'll discuss the significance of this result shortly. We also report the training and test set error-rates in the supplement for this case (see Fig. 5 in the Supplement), noting the fact that these error-rates are not indicative of the performance as the dataset is skewed.

Figure 3 (b) shows the variation in performance of the classifiers on the cropped dataset with the number of PCA features. We notice that the performance of the classifiers, in this case, worsens across the board, as compared to the original data case. We attribute this drop in the performance to the effectiveness of our bee-finding methodology. It seems that cropping the ROI from the image (which might sometimes miss the bee, altogether), proves to be detrimental. We can draw this conclusion by observing the performance of RF when operating on 20 PCA features, which shows that quality of the first 20 features has dropped. In essence, RF operating on a few features seems to be a good judge of the quality of features. Again, we report the training and test set error-rates in the supplement (see Fig. 6 in the Supplement), noting the fact that the dataset is skewed in this case and as a result these error-rates are not indicative of the performance.

Finally, in Figure 3 (c), we present the performance results of the classifiers operating on augmented dataset. We notice that the performance of all the classifiers has significantly improved as compared to the two cases presented above (original and cropped datasets). Interestingly, there is a large improvement in the performance of RF when operating on 20 features, this is the reason we added an additional experiment on 10 features. The jump in RF's performance seems to be a result of improvement in the quality of features, as a result of data augmentation. Also, we observe a diminishing returns trend with increasing features, for both k-SVM and S-LR. This behavior can be linked to how fast the singular values of the features decay. As a result, we can potentially use significant singular values as a way to select the number of features we desire to keep. We show a plot of scaled singular values of the PCA (applied to the features) in the (see Fig. 7 in the Supplement). For the augmented data case, we also report the train and test error-rates in Fig. 4, as this dataset is well balanced in terms of the distribution of samples. We notice that both for k-SVM and S-LR, the train and test error rates are very similar, except when all PCA features are considered. This (similarity in train and test error rates) is a desirable trend we seek for deployment-ready classifiers. Also, we observe that test error rate is lowest for RF, as compared to k-SVM and S-LR. We report RF operating on 10 features from augmented dataset to be the best classifier in our analysis, with an AUC of 0.825.

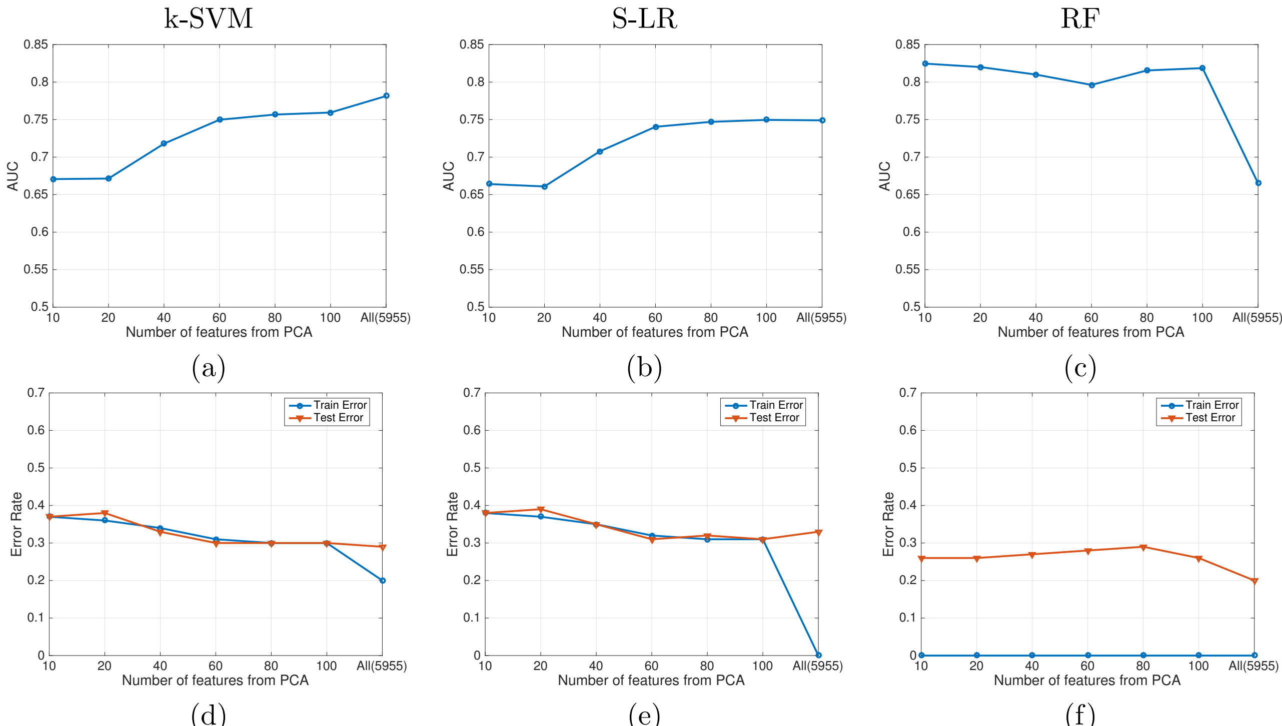


Figure 4: AUC and error-rate performance of the classifiers on augmented dataset with increasing number of features (principal components). Panels (a)-(c) show the AUC for each classifier, k-SVM, S-LR and RF, respectively, and panels (d)-(f) show the corresponding error rates for each classifier. The baseline AUC of 0.712 was provided by METIS. The best AUC reported in the progress report was 0.63.

5.4 A note on bag of image words

We also explored the k-SVM on bag of image words classifier based on SIFT features. Although, the results on augmented data set were not promising (AUC of `0.65`), the features generated by the algorithm could potentially be used in the current model. We leave it as a future work.

6 Conclusions

The Metis bee challenge exposed us to a range of issues encountered in practical machine learning. The given data suffered from a variety of issues, including but not limited to small training set, skewed distribution of classes, presence of rich backgrounds, and occlusions.

Data augmentation turned out to be a very simple yet very effective way of artificially increasing the number of samples, and offsetting the effects of a skewed distribution across classes. Next, we learnt that the random forest classifier (operating on the PCA features) can work as a good indicator of the quality of features we have, i.e. how informative the features are. Further, our analysis of the performance of k-SVM and S-LR on augmented dataset reveals that they exhibit diminishing returns. This means that standard methods like singular value decomposition (SVD) can be employed to decide on the number of features to be used.

For next steps, the promising performance of RF leads us to believe that boosting could be a good contender. Further, as mentioned in the progress report, although the initial goal of this project was to also explore convolutional neural networks(ConvNets), we opted against it. However, owing to the exciting results obtained by augmenting the data, we are hopeful again and will continue our pursuit of exploring ConvNets.

References

[1] The Metis Challenge: Naive Bees Classifier, “The Metis Challenge: Naive Bees Classifier,” 2015.

[2] O. Tuzel, F. Porikli, and P. Meer, “Region covariance: A fast descriptor for detection and classification,” in *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, Berlin, Heidelberg, 2006, ECCV’06, pp. 589–600, Springer-Verlag.

[3] E. Tola, V. Lepetit, and P. Fua, “DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, May 2010.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] Mark Asbach, “OpenCV covariance features, `https://bitbucket.org/mark.asbach/opencv-covariance-features`,” 2013.

[6] L. S. Hackenberg, “Minimal bag of visual words image classifier, `https://github.com/shackenberg/ Minimal-Bag-of-Visual-Words-Image-Classifier`,” 2014.

A Supplement

A.1 Error rate performances of k-SVM, S-LR, and RF on original and cropped datasets

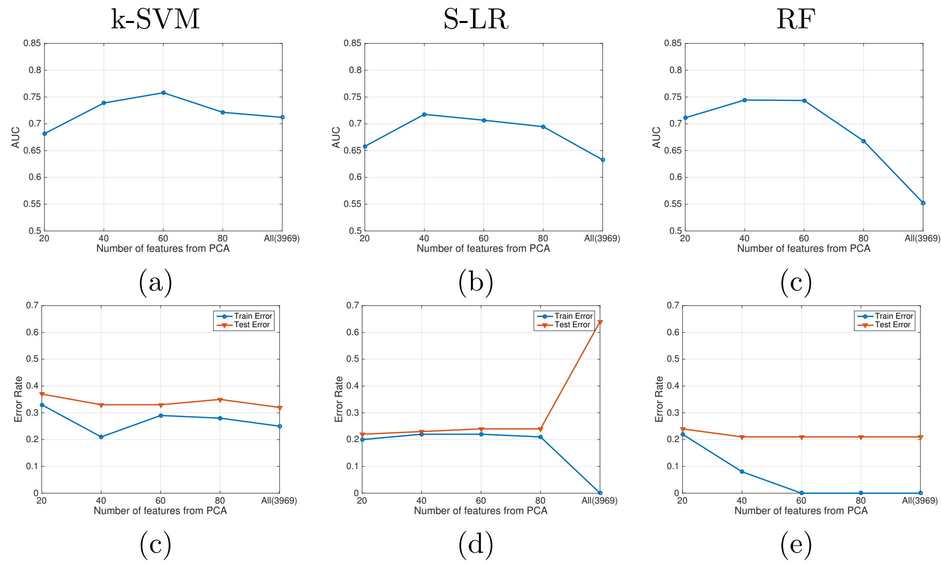


Figure 5: AUC and error-rate performance of the classifiers on original data with increasing number of features (principal components). Panels (a)-(c) show the AUC for each classifier, k-SVM, S-LR and RF, respectively, and panels (d)-(f) show the corresponding error rates for each classifier. The baseline AUC of 0.712 was provided by METIS. The best AUC reported in the progress report was 0.63.

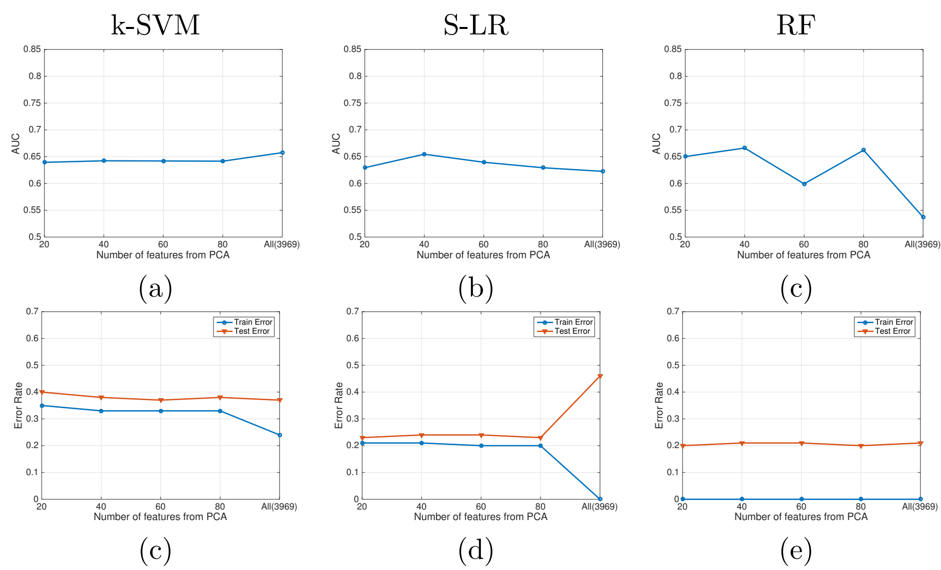


Figure 6: AUC and error-rate performance of the classifiers on cropped data with increasing number of features (principal components). Panels (a)-(c) show the AUC for each classifier, k-SVM, S-LR and RF, respectively, and panels (d)-(f) show the corresponding error rates for each classifier. The baseline AUC of 0.712 was provided by METIS. The best AUC reported in the progress report was 0.63.

A.2 Decay of singular values of the PCA features for original, cropped, and augmented datasets

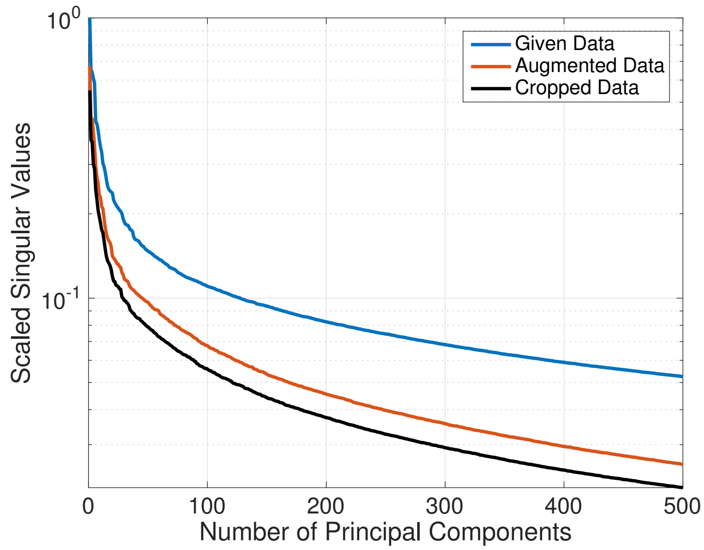


Figure 7: Decay of singular values of the PCA features for original, cropped, and augmented datasets.