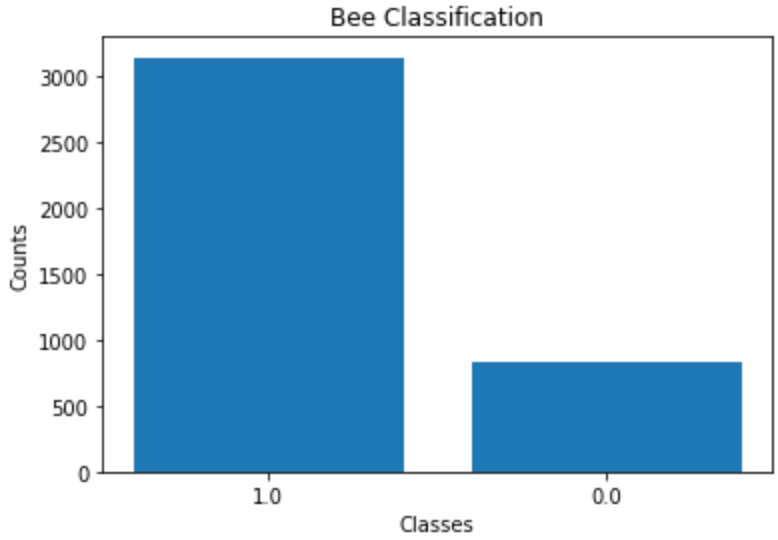
# Bee Classification on Image Dataset of Bees

## Abstract

Three classification algorithms Random Forest Classifier, SVM Classifier, and AdaBoost Classifier were tried for the given dataset of images of honey bees and bumblebees. Though all the algorithms didn't work well on this dataset, SVM worked best with an f1 score of 52.62%.

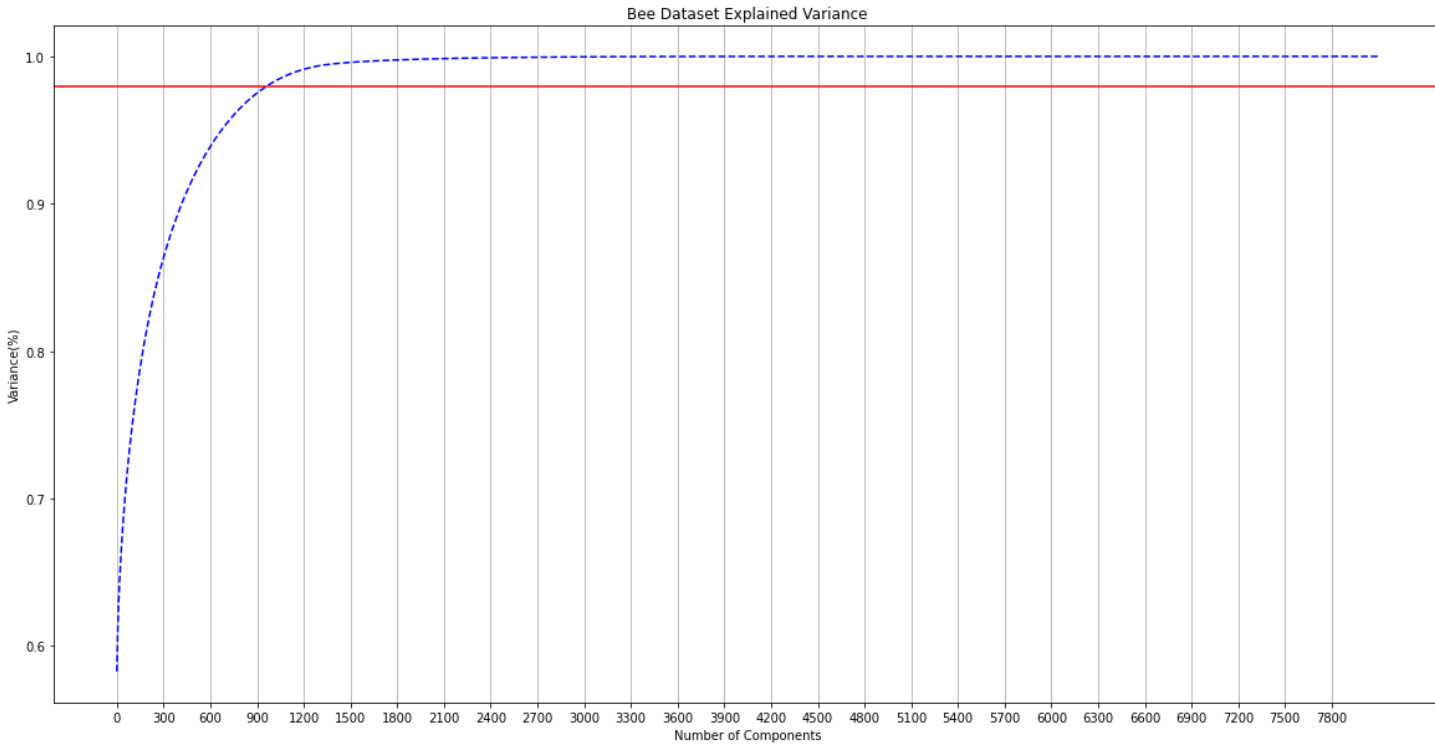
## Data Exploration and Analysis

We were provided with 3969 images in our dataset, among which 3142 were of bumblebees and 827 were of honey bees. Thus our dataset is an imbalanced one. So we use f1 score as the metric for our model. We also had labels.csv file with the labels for each image.




## Extraction of hog features

I stacked images into the list, converted it to grayscale, and applied function to extract hog features. I got 42849 features for each image. I rescaled the image of shape(200,200) to half. Then, I got 8100 features. For further reduction, I used PCA. 1000 dimensions could represent more than 98% of the variance of the original data. So I used 1000 features.




I split the dataset into a train and a test set in the ratio of 3:1.

# Grid Search

We performed a grid search among the Random Forest classifier, AdaBoost classifier, and SVM classifier. The parameters used for each algorithm are listed below.

## 1. Random Forest Classifier

n_estimators is the number of trees built before taking maximum voting or averages of predictions. The most widely used along with default is selected for a grid search. A large number of values were tried for selecting max_depth. The minimum number of samples a node must contain in order to consider splitting is min_samples_split.

| | |
|---|---|
| n_estimators | [5,10,20,50] |
| criterion | [‘gini’, ‘entropy’] |
| max_depth | [3,7,15,19,21,23,25] |
| min_samples_split | [2,4,6,8,10] |

## 2. SVM Classifier

C is the regularization parameter. For C and gamma, the most common values among the state of the art are selected for the grid search. All kernels except polynomial were selected.

| | |
|---|---|
| C | `[0.1,1,10,100,1000]` |
| kernel | `['linear','rbf','sigmoid']` |
| gamma | [1,0.1,0.01,0.001,0.0001] |

## 3. AdaBoost Classifier

n_estimators is the number of trees built before taking maximum voting or averages of predictions. The most widely used along with default is selected for a grid search. The most commonly used learning rates were used.

| | |
|---|---|
| n_estimators | [5, 10, 20] |
| learning_rate | [1,0.5,0.25,0.1,0.01] |

**Algorithms with best parameters and f1 score for best parameters are listed below:**

| **Classifiers** | **Best parameters** | **F1 scores** |
|---|---|---|
| Random Forest | {‘Criterion’ : ’gini’, ‘max_depth’ : 21, ‘min_samples_split’ : 2, ‘n_estimators’ : 5} | 0.4786 |
| SVM | {‘C’ : 1, ’gamma ’: 1, ’kernel’ : ’linear’} | 0.5406 |
| AdaBoost | {‘learning_rate’ : 1, ’n_estimators’ : 5} | 0.4431 |

```
<class 'sklearn.ensemble._forest.RandomForestClassifier'>
0.4786173752084519 {'criterion': 'gini', 'max_depth': 21, 'min_samples_split': 2, 'n_estimators': 5}
<class 'sklearn.svm._classes.SVC'>
0.5405634992662799 {'C': 1, 'gamma': 1, 'kernel': 'linear'}
<class 'sklearn.ensemble._weight_boosting.AdaBoostClassifier'>
0.44307398801272824 {'learning_rate': 1, 'n_estimators': 5}
```

```
[113]  1 classwisef1Score
```

| | Classifier | F1-Score |
|---|---|---|
| 0 | <class 'sklearn.ensemble._forest.RandomForestC... | 0.462337 |
| 1 | <class 'sklearn.svm._classes.SVC'> | 0.526174 |
| 2 | <class 'sklearn.ensemble._weight_boosting.AdaB... | 0.441821 |

# Model Evaluation and Comparison

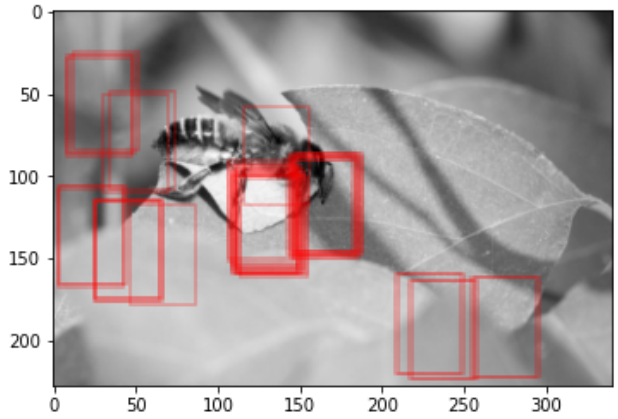
After selecting the best parameters for each algorithm using a grid search, we calculated the f1 score in test data for each algorithm.

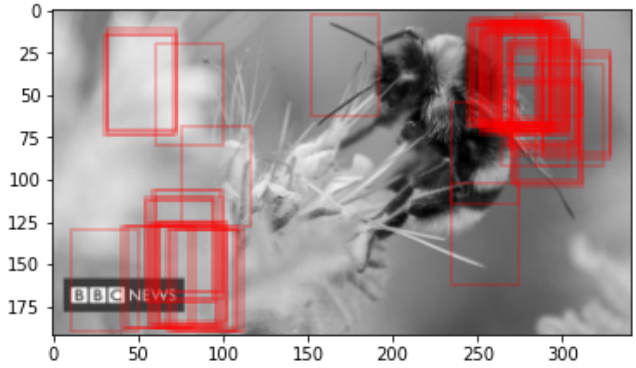
| **Classifier** | **F1 Score** |
|---|---|
| Random Forest | 0.4623 |
| SVM | 0.5262 |
| AdaBoost | 0.4418 |

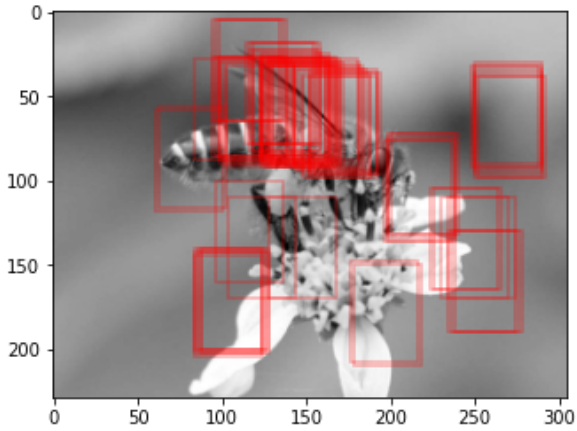
SVM classifier is found to work best on the given test data with an f1 score of 0.5262. Thus SVM is the final model for our classification task.

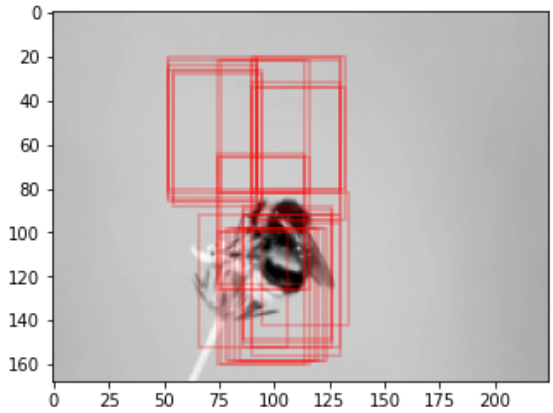
# Model Usage on unseen examples

Some images which contained either honey bee or bumblebee were taken from the web. We used our model to predict those images. The result is shown below:







# Conclusion

All three models didn’t work well on this dataset. Among the trained models, SVM worked best with an f1 score of 52.62%.