Predicting bees species using Support Vector Machine

Manaswi Jaiswal
Information Science and Engineering
01JST17IS026

0.1 Declaration

I, the undersigned, solemnly declare that the project report PREDICTING BEES SPECIES USING SUPPORT VECTOR MACHINE is based on my own work carried out during the course of our study under the supervision of Dr. B S Harish. I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that
I. The work contained in the report is original and has been done by me under the general supervision of my supervisor.
II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
III. It follows the guidelines provided by the university for writing the report.
IV. Whenever some materials (data, theoretical analysis, and text) from other sources have been used, it has been given due credit in the text of the report and their details are provided in the references.

Date: 23 April 2020
Manaswi Jaiswal

Contents

0.2 Abstract

Being able to identify bee species from images is a task that ultimately would allow researchers to more quickly and effectively collect field data. Pollinating bees have critical roles in both ecology and agriculture, and diseases like colony collapse disorder threaten these species. Identifying different species of bees in the wild means that we can better understand the prevalence and growth of these important insects.
This project focuses on designing a model that classifies honey bees and bumble bees. Before the images are classified, they undergo various manipulation techniques and change of forms to get transformed into what can be used as an input to the model.
This report walks us through the various image processing techniques that help in this transformation and classification of these transformed images.

Chapter 1

Introduction

1.1 Problem Statement

Being able to identify bee species from images is a task that ultimately would allow researchers to more quickly and effectively collect field data.
Pollinating bees have critical roles in both ecology and agriculture, and diseases like colony collapse disorder threaten these species. Identifying different species of bees in the wild means that we can better understand the prevalence and growth of these important insects.
Design a model that differentiates between a honey bee and bumble bee.

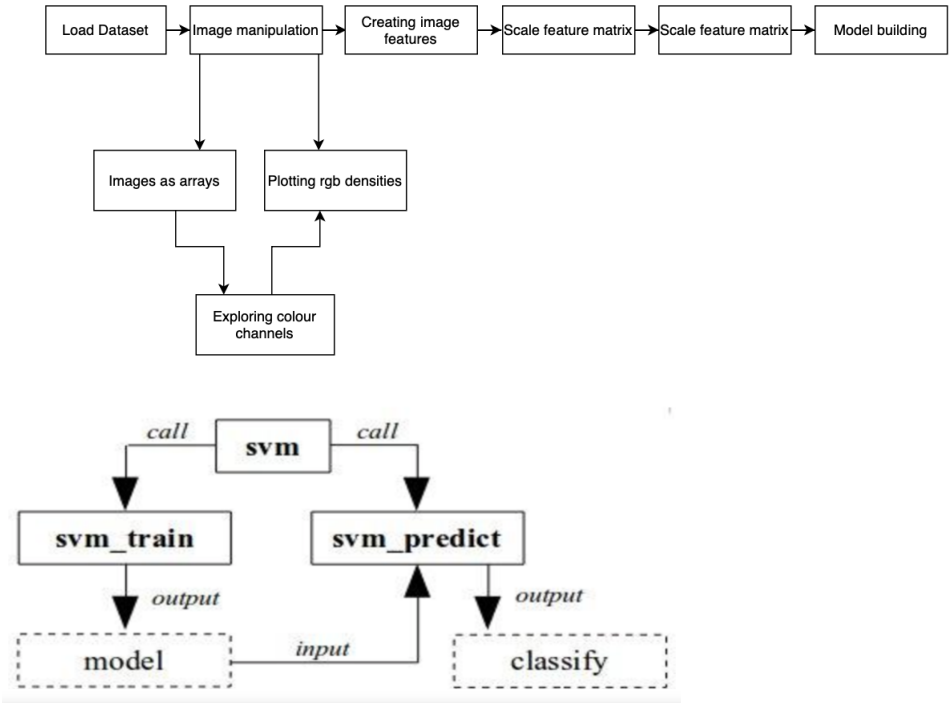
1.2 General Introduction

A honey bee (also spelled honeybee) is a eusocial flying insect within the genus Apis of the bee clade, all native to Eurasia but spread to four other continents by human beings. A bumblebee is any of over 250 species in the genus Bombus, part of Apidae, one of the bee families.

Support Vector Machines (SVMs) are a relatively new supervised classification technique to the land cover mapping community. They have their roots in Sta-

tistical Learning Theory and have gained prominence because they are robust, accurate and are effective even when using a small training sample. By their nature SVMs are essentially binary classifiers, however, they can be adopted to handle the multiple classification tasks common in remote sensing studies.
Classification is a subcategory of supervised learning where the goal is to predict the categorical class labels (discrete, unoredered values, group membership) of new instances based on past observations.
A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

1.3 General Block Diagram



1.4 Applications

1. Face Detection It classifies the parts of the image as face and non-face. It contains training data of n x n pixels with a two-class face (+1) and non-face (-1). Then it extracts features from each pixel as face or non-face. Creates a square boundary around faces on the basis of pixel brightness and classifies each

image by using the same process.

2. Text and Hypertext Categorization Allows text and hypertext categorization for both types of models; inductive and transductive. It Uses training data to classify documents into different categories such as news articles, e-mails, and web pages
Examples:

Classification of news articles into "business" and "Movies" Classification of web pages into personal home pages and others For each document, calculate a score and compare it with a predefined threshold value. When the score of a document surpasses threshold value, then the document is classified into a definite category. If it does not surpass threshold value then consider it as a general document. Classify new instances by computing score for each document and comparing it with the learned threshold.

3. Classification of Images SVMs can classify images with higher search accuracy. Its accuracy is higher than traditional query-based refinement schemes

4. Bioinformatics In the field of computational biology, the protein remote homology detection is a common problem. The most effective method to solve this problem is using SVM. In last few years, SVM algorithms have been extensively applied for protein remote homology detection. These algorithms have been widely used for identifying among biological sequences. For example classification of genes, patients on the basis of their genes, and many other biological problems.

5. Protein Fold and Remote Homology Detection Protein remote homology detection is a key problem in computational biology. Supervised learning algorithms on SVMs are one of the most effective methods for remote homology detection. The performance of these methods depends on how the protein sequences modeled. The method used to compute the kernel function between them.

6. Handwriting Recognition We can also use SVMs to recognize hand-written characters that use for data entry and validating signatures on documents.

7. Geo and Environmental Sciences We use SVMs for geo (spatial) and spatiotemporal environmental data analysis and modeling series.

8. Generalized Predictive Control We use SVM-based GPC to control chaotic dynamics with useful parameters. It provides excellent performance in controlling the systems. The system follows chaotic dynamics with respect to the local stabilization of the target. Using SVMs for controlling chaotic systems has the following advantages-
i. Allows use of relatively small parameter algorithms to redirect a chaotic system to the target.
ii. Reduces waiting time for chaotic systems.
iii. Maintains the performance of systems.

1.5 Challenges

- Unbalanced data
  Some classes have very less data and some have relatively larger amount of data
- Structural datasets
  1. An instance may not be a vector e.g., a tree from a sentence
  2. Labels in order relationships SVM for ranking
- Multi-label classification An instance associated with more than two labels (e.g., a video shot includes several concepts)
- Large-scale Data
  SVM cannot handle large sets if using kernels. There are two possibilities:
  1. Linear SVMs. In some situations, can solve much larger problems
  2. Approximation: sub-sampling and beyond
- Semi-supervised learning
  Some available data unlabeled.

1.6 Motivation

- Both honey and bumble bees have different behaviors and appearances, but given the variety of backgrounds, positions, and image resolutions, it can be a challenge for machines to tell them apart.
- Being able to identify bee species from images is a task that ultimately would allow researchers to more quickly and effectively collect field data.
- Pollinating bees have critical roles in both ecology and agriculture, and diseases like colony collapse disorder threaten these species.
- Identifying different species of bees in the wild means that we can better understand the prevalence and growth of these important insects.

1.7 Objectives

- Examine RGB values in an image matrix
- Analyse the given images by converting them into computational form
- Design a model for differentiating between honey bees and bumble bees
- Explore transfer learning, which harnesses the prediction power of models that have been trained on the dataset.

Chapter 2

Literature Survey

Over the years a lot of work has been done in the fields that are essential to the working and growth of this project.
For this project, we need to review all aspects of image processing, pattern recognition, geometric optics, digital image acquisition and display, hardware, and techniques.

H. C. Andrews, A. G. Tescher, R. P. Kruger, 1972 [2] gave an introduction to current trends in the digital processing and manipulation of images is presented. The paper does not attempt a detailed survey of the field, but outlines instead the motivation behind it, and the relationships between the 'disciplines' such as image enhancement, pattern recognition and scene analysis involved with processing images.

J. Lee,1978 [3] discusses the theoretical and applications aspects of digital processing for two-dimensional images. It also covers the principles and applications of multidimensional digital signal processing. It explains how applying processes like smoothing, sharpening, contrasting, stretching etc on an image can increase its readability and enhance its quality or even transform the image.

In this project we have used pandas dataframes.Pandas DataFrame is nothing but an in-memory representation of an excel sheet via Python programming language.DataFrames are the distributed collection of data. In DataFrame, data is organized into named columns. It is conceptually similar to a table in a relational database. David Wayne Embley,1980 [7] talks about manipulating dataframes. Processing everyday data items constitutes a significant portion of real-world computer applications. Programmers involved with everyday data items confront the drudgery of writing routines to recognize, validate, transform, store, retrieve, manipulate, and display these items. This paper throws light on working with dataframes and the various challenges encountered.

Alvarez L., Guichard F., Lions P. L., and Morel .1. M., 1993 [5] throws light on techniques and methodologies for validating the authenticity of digital images and testing for the presence of doctoring and manipulation operations on them has recently attracted attention. It consists of a review of three categories of forensic features and discuss the design of classifiers between doctored and original images. The performance of classifiers with respect to selected controlled manipulations as well as to uncontrolled manipulations is analyzed. The tools for image manipulation detection are treated under feature fusion and decision fusion scenarios.

Statistical methods are mainly useful to ensure that the data is interpreted correctly. And that apparent relationships are really significant or meaningful and it is not simply happen by chance. The statistical analysis helps to find meaning to the meaningless numbers.V. Vapnik,1995 [13] discusses the fundamental ideas which lie behind the statistical theory of learning and generalization. It considers learning as a general problem of function estimation based on empirical data. Omitting proofs and technical details, the author concentrates on discussing the main results of learning theory and their connections to fundamental problems in statistics. These include the setting of learning problems based on the model of minimizing the risk functional from empirical data, a comprehensive analysis of the empirical risk minimization principle including necessary and sufficient conditions for its consistency, non-asymptotic bounds for the risk achieved using the empirical risk minimization principle, principles for controlling the generalization ability of learning machines using small sample sizes based on these bounds, the Support Vector methods that control the generalization ability when estimating function using small sample size.

E. Osuna, R. Freund, and F. Girosi, 1997 [17] presents a decomposition algorithm that can be used to train SVM's over large data sets. The main idea behind the decomposition is the iterative solution of sub-problems and the evaluation of, and also establish the stopping criteria for the algorithm. It presents previous approaches, as well as results and important details of our implementation of the algorithm using a second-order variant of the Reduced Gradient Method as the solver of the sub-problems.

Y. Tang, B. Jin, Y. Sun and Y. Zhang,2004 [15] take us through an innovative learning model called granular support vector machines for data classification problems by building just two information granules in the top-down way. The experiment results on three medical binary classification problems show that granular support vector machines proposed in their work provides an interesting new mechanism to address complex classification problems, which are common in medical or biological information processing applications.

SVMs (Support Vector Machines) are a useful technique for data classification. A training method to increase the efficiency of SVM has been presented by Yiqiang Zhan,D. Shen, 2005 [14] for fast classification without system degradation. Experimental results on real prostate ultrasound images show good performance of their training method in discriminating the prostate tissues from

other tissues and they claim that their proposed training method is able to generate more efficient SVMs with better classification abilities.

Mangasarian OL, Wild EW, 2005 [16] propose a new approach to support vector machine (SVM) classification wherein each of two data sets are proximal to one of two distinct planes that are not parallel to each other. Each plane is generated such that it is closest to one of the two data sets and as far as possible from the other data set. Each of the two nonparallel proximal planes is obtained by a single MATLAB command as the eigenvector corresponding to a smallest eigenvalue of a generalized eigenvalue problem. Classification by proximity to two distinct nonlinear surfaces generated by a nonlinear kernel also leads to two simple generalized eigenvalue problems. The effectiveness of the proposed method is demonstrated by tests on simple examples as well as on a number of public data sets.

A HOG descriptor is computed by calculating image gradients that capture contour and silhouette information of grayscale images. Gradient information is pooled into a 1-D histogram of orientations, thereby transforming a 2-D image into a much smaller 1-D vector that forms the input for machine learning algorithms such as random forests, support vector machines, or logistic regression classifiers. N. Dalal, B. Triggs,2005 [9] shows a study of feature sets for robust visual object recognition; adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, it shows experimentally that grids of histograms of oriented gradient (HOG) descriptors significantly outperform existing feature sets for human detection. The study shows influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results.

In machine learning, to catch useful indicators and obtain a more accurate result, we tend to add as many features as possible at first. However, after a certain point, the performance of the model will decrease with the increasing number of elements. This phenomenon is often referred to as "The Curse of Dimensionality." The curse of dimensionality occurs because the sample density decreases exponentially with the increase of the dimensionality. When we keep adding features without increasing the number of training samples as well, the dimensionality of the feature space grows and becomes sparser and sparser. Due to this sparsity, it becomes much easier to find a "perfect" solution for the machine learning model which highly likely leads to overfitting. To avoid these problems, we use PCA. Dimensionality reduction is the process of reducing the dimensionality of the feature space with consideration by obtaining a set of principal features. Dimensionality reduction can be further broken into feature selection and feature extraction. V. Rokhlin, A. Szlam, and M. Tygert,2009 [11] explain Principal component analysis (PCA), which is an important technique to understand in the fields of statistics and data science. It explains, in detail, how to reduce the dimension of the feature space (called "dimensionality reduction"). It focuses on the many ways to achieve dimensionality reduction,Feature Elimination,Feature Extraction. It answers questions like 'When to use PCA?',

'How does PCA work?','why does PCA work in various situations?'. It also briefly discusses the extensions to PCA.

Since our data is also in the form of dataframes, we require a knowledge of manipulating dataframes which is provided by Wes McKinney,2011 [8] discusses pandas, a Python library of rich data structures and tools for working with structured data sets common to statistics, finance, social sciences, and many other fields. It provides integrated, intuitive routines for performing common data manipulations and analysis on such data sets. It aims to be the foundational layer for the future of statistical computing in Python. It serves as a strong complement to the existing scientific Python stack while implementing and improving upon the kinds of data manipulation tools found in other statistical programming languages such as R.

Umesh P,2012 [1] explains how PIL can be used to display image, create thumbnails, resize, rotation, convert between file formats, contrast enhancement, filter and apply other digital image processing techniques etc. It also throws light on powerful image processing and graphics capabilities. It breaks down the concepts of in-built function to perform operations like - load images, save, change format of image, and create new images.

After the images are processed, there is a need to display them. Visualizations are the easiest way to analyze and absorb information. Visuals help to easily understand the complex problem. They help in identifying patterns, relationships, and outliers in data. It helps in understanding business problems better and quickly. It helps to build a compelling story based on visuals..This need is met by the matplotlib module in python.To communicate information clearly and efficiently, data visualization uses statistical graphics, plots, information graphics, and other tools.Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. John Hunter and Darren Dale, 2012 [6] introduce various applications of matplotlib describing its output formats and its interaction with the graphical environment.It focuses on the plotting capabilities of matplotlib ranging from plotting basic lines and legends to plotting subplots to existing plots and complex scales.It helps visualize data thereby aiding in making statistical inferences. This paper throws some light on how matplotlib has a similar feel to Matlab's graphical plotting and how it gives you control over every aspect of a figure.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu,2014 [4] present scikit-image, an image processing library that implements algorithms and utilities for use in research, education and industry applications. This paper highlights the advantages of open source to achieve the goals of the scikit-image library, and showcases several real-world image processing applications that use scikit-image.

To get better results, it is important to choose various highly correlated attributes and dropping the insignificant ones so as to get those attributes that affect the dataset significantly. So our purpose is to perform feature slection

and then move on to dimensionality reduction. Zubair Nabi, 2016 [10] focuses on Feature Selection. Many machine learning methods are built to work best with data that has a mean of 0 and unit variance. This work provides a guide to using simple ways to rescale data so that it works well.

Now that we have the significant attributes, we move on to splitting the dataset. Mayukh Bhattacharyya, 2019 [12] is a small article that successfully explains the minor details not to be overlooked before splitting the dataset into test and train sets to train the model. It talks about something called stratification which will lock the distribution of classes in train and test sets. It also talks about how split by the indices of the dataset is done for better results and how preprocessing plays an essential role to get good results.

Chapter 3

Proposed Method

3.1 Design

A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. As per the design of the project , the following is the sequentially chart of the project workflow:

1. Image manipulation with PIL
Pillow has a number of common image manipulation tasks built into the library.Operations that Pillow provides include resizing, cropping, rotating, flipping, converting to greyscale (or other color modes).
Often, these kinds of manipulations are part of the pipeline for turning a small number of images into more images to create training data for machine learning algorithms. This technique is called data augmentation, and it is a common technique for image classification.

2. Representing images as arrays of data
Most image formats have three color "channels": red, green, and blue (some images also have a fourth channel called "alpha" that controls transparency). For each pixel in an image, there is a value for every channel.
The way this is represented as data is as a three-dimensional matrix. The width of the matrix is the width of the image, the height of the matrix is the height of the image, and the depth of the matrix is the number of channels. So, the height and width of the image are both 100 pixels. This means that the underlying data is a matrix with the dimensions 100x100x3.

3. Explore the color channels
Color channels can help provide more information about an image. A picture of the ocean will be more blue, whereas a picture of a field will be more green. This kind of information can be useful when building models or examining the differences between images.
The kernel density estimate for each of the color channels on the same plot

are examined to understand how they differ. When we make this plot, we'll see that a shape that appears further to the right means more of that color, whereas further to the left means less of that color.

4. Plotting rgb densities to differentiate between the two bee species
The rgb densities of the two images are plotted to see the variations in the red, blue and green channels. The differences can be noted once the graphs are plotted.

5. Saving manipulated images
To use these images in the future, they have to be saved so that they can be used in the future.

6. Make a pipeline
To create an image processing pipeline, we have all the tools in our toolbox to load images, transform them, and save the results. In this pipeline, the following is done:

- Load the image with Image.open and create paths to save our images to
- Convert the image to grayscale
- Save the grayscale image
- Rotate, crop, and zoom in on the image and save the new image

7. Display image of each bee type
The data is loaded into a dataframe , where the index is the image name (e.g. an index of 1036 refers to an image named 1036.jpg) and the genus column implies the bee type. genus takes the value of either 0.0 (Apis) or 1.0 (Bombus).
We create a function that converts an index value from the dataframe into a file path where the image is located, opens the image, and then returns the image as a numpy array.
8. Image manipulation with rgb
The rgb2grey function computes the luminance of an RGB image using the following formula Y = 0.2125 R + 0.7154 G + 0.0721 B.
Image data is represented as a matrix, where the depth is the number of channels.

9. Histogram of oriented gradients
The idea behind HOG is that an object's shape within an image can be inferred by its edges, and a way to identify edges is by looking at the direction of intensity gradients (i.e. changes in luminescence).

10. Create image features and flatten into a single row
The model has to be provided with the raw pixel values from images as well as the HOG features we just calculated.To do this, a function that combines these two sets of features by flattening the three-dimensional array into a one-dimensional (flat) array is created.
This generates a flattened features array for the bombus image.Features for each image are created and then the flattened features arrays are stacked into a big

matrix we can pass into the model. Another function is created to perform the following tasks:

- Load an image
- Generate a row of features using the create_features function above
- Stack the rows into a features matrix

11. Scale feature matrix and PCA
Many machine learning methods are built to work best with data that has a mean of 0 and unit variance. To use a SVM, our model of choice, the number of features need to be reduced.For this, PCA is used.

12. Splitting into train and test sets
Now the data needs to be converted into train and test sets. The ratio used is 70% of images as training data and 30% as testing dataset.
13. Training model
A support vector machine (SVM) is used to train the model. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

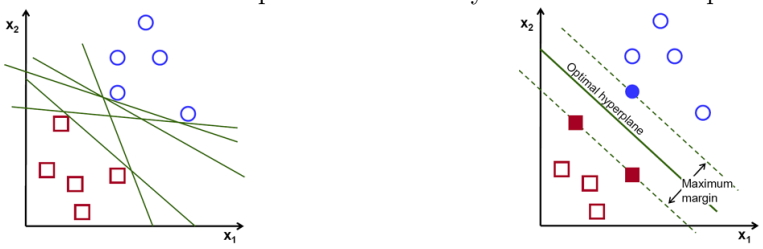
14. Score model
The trained model is used to generate predictions for the test data. To see how well our model did, the accuracy is calculated by comparing predicted labels for the test set with the true labels in the test set.

15. The Receiver Operating Characteristic curve and the Area Under the Curve
The ROC curve plots the false positive rate and true positive rate at different thresholds. ROC curves are judged visually by how close they are to the upper lefthand corner.
The AUC is also calculated, where 1 means every predicted label was correct. Generally, the worst score for AUC is 0.5, which is the performance of a model that randomly guesses.

16. Classification report
A classification report is built showing the main classification metrics like precision, recall, f1-score etc.)

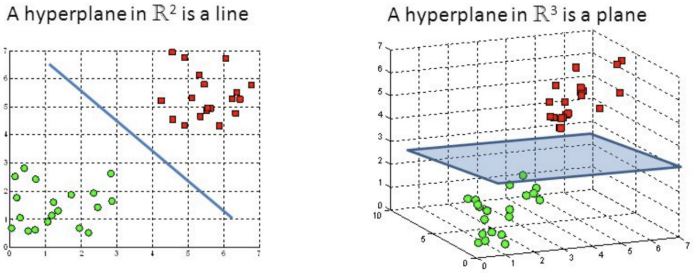
3.2 Algorithm and the required mathematical equations

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

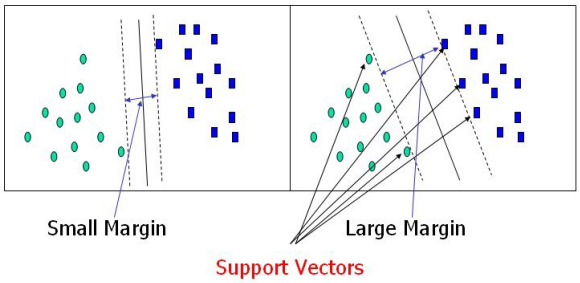


To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyperplanes and Support Vectors



Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.



Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we

maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.
In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

Hinge loss function (function on top can be represented as a function below)

$$c(x, y, f(x)) = (1 - y * f(x))_+$$

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function.
The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \left(1 - y_i \langle x_i, w \rangle\right)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

When there is no misclassification, i.e our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.

$$w = w - \alpha \cdot (2\lambda w)$$

When there is a misclassification, i.e our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update.

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

Chapter 4

Experimental Analysis

4.1 Dataset

The project consists of two dataset, both of which are pictures of bees(honey and bumble). The first dataset consists of four pictures and these pictures are used to understand image manipulation using Python Imaging Library. Various functions like rotation, cropping and flipping. These processed images are saved into a new folder.
The second dataset contains 4960 images and corresponding labels. The labels dataset is a csv file that stores information regarding 500 pictures (whether they are honey or bumble bees).

4.2 Experimental Settings

To test the efficacy of the proposed model, we have used the bees dataset. One dataset contains four pictures that are used for manipulation and another dataset consists of 4960 images used for prediction.To test the efficacy of the proposed model, we have conducted extensive experiments by varying the size of the training set and testing set.

test and train split
Splits arrays or matrices into random train and test subsets.The following parameters are passed:

- bees_pca(input):Allowed inputs are lists, numpy arrays, scipy-sparse matrices or pandas dataframes.
- test_size:If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the test split.
- random_state:random_state is the seed used by the random number generator

The train_test_split is passed as follows:
X_train, X_test, y_train, y_test = train_test_split(bees_pca, labels.genus.values,

test_size=0.2, random_state=1234123)
C support vector classifier
The model uses Support Vector Machine. The following parameters are passed:

- c:Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.
- break_ties:If true, decision_function_shape='ovr', and number of classes ¿ 2, predict will break ties according to the confidence values of decision_function; otherwise the first class among the tied classes is returned. Please note that breaking ties comes at a relatively high computational cost compared to a simple predict.
- cache_size:Specify the size of the kernel cache (in MB).
- class_weight:Set the parameter C of class i to class_weight[i]*C for SVC. If not given, all classes are supposed to have weight one.
- coef0:ndependent term in kernel function. It is only significant in 'poly' and 'sigmoid'.
- decision_function_shape:Whether to return a one-vs-rest ('ovr') decision function of shape (n_samples, n_classes) as all other classifiers, or the original one-vs-one ('ovo') decision function of libsvm which has shape (n_samples, n_classes * (n_classes - 1) / 2). However, one-vs-one ('ovo') is always used as multi-class strategy.
- degree:Degree of the polynomial kernel function ('poly')
- gamma:Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.
  if gamma='scale' (default) is passed then it uses 1 / (n_features * X.var()) as value of gamma
  if 'auto', uses 1 / n_features.
- kernel:pecifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable
- max_iter:Hard limit on iterations within solver, or -1 for no limit.
- probability:Whether to enable probability estimates. This must be enabled prior to calling fit, will slow down that method as it internally uses 5-fold cross-validation, and predict_proba may be inconsistent with predict.
- random_state:The seed of the pseudo random number generator used when shuffling the data for probability estimates.
- shrinking:Whether to use the shrinking heuristic.
- tol:Tolerance for stopping criterion.
- verbose:Enable verbose output. Note that this setting takes advantage of a per-process runtime setting in libsvm that, if enabled, may not work properly in a multithreaded context.

The support vector classifier is defined as follows:
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear', max_iter=-1, probability=True, random_state=42, shrinking=True,tol=0.001,verbose=False)

4.3 Result Tables

Dataset Size(train,test)	Accuracy in %	Distribution of labels(train,test)
(60,40)	0.605	(175,175)
(70,30)	0.66	(153,147)
(80,20)	0.72	(201,199)

	Precision	Recall	f1-score	support
0.0	0.73	0.71	0.72	51
1.0	0.71	0.73	0.72	49
accuracy			0.72	100
macro avg	0.72	0.72	0.72	100
weighted avg	0.72	0.72	0.72	100

precision is the fraction of relevant instances among the retrieved instances and is defined as:

precision = tp / (tp + fp) or (true positives)/(prediced positives)

recall is the fraction of relevant instances that have been retrieved over total relevant instances in the image, and is defined as

recall = tp / (tp + fn) or (true positives)/(actual positives)

Where, tp = true positives, fp = false positives anf fn = false negatives. Recall in this context is also referred to as the true positive rate or sensitivity, and precision is also referred to as positive predictive value (PPV).

f1-score: is a measure of a test's accuracy. It considers both the precision and the recall to compute the score. The f1-score can be interpreted as a weighted average of the precision and recall, where an f1-score reaches its best value at 1 and worst at 0.

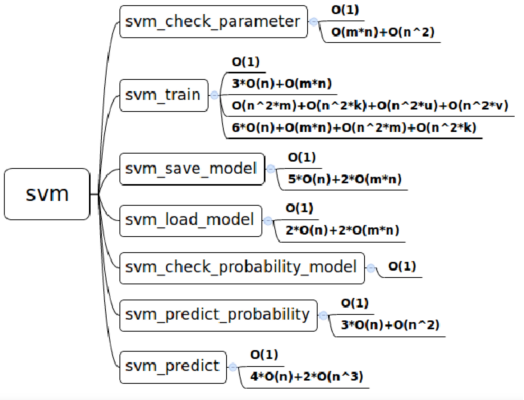
The general formula is: 2.(precision.recall)/(precision+recall)

The f1-score regulates bot precision and recall.

4.4 Discussions on Results

The model classifies an image as a bumble bee or a honey bee with 72% accuracy with the experimental settings as mentioned above.
This means that when an image is given as an input to the algorithm, it predicts the genus as apis or honey bee(0.0) or bombus or bumble bee (1.0) with an accuracy of 72%.

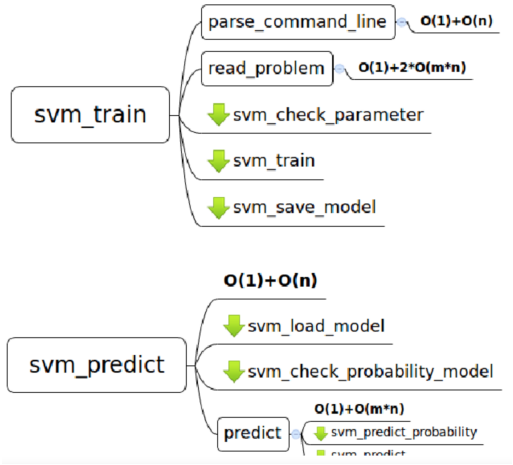
4.5 Complexity of the algorithm



Num.	Methods	Complexity (Big-O)
1	parse_command_line	O(1)+O(n)
2	read_problem	O(1)+2*O(m*n)
3	svm_predict (main) dan predict	O(1)+O(n)+O(n*m)
4	svm_check_parameter	O(1)+O(m*n)+O(n^2)
5	svm_train	O(1)+9*O(n)+2*O(m*n)+ 6*O(n^2*m)
6	svm_save_model	O(1)+5*O(n)+2*O(m*n)
7	svm_load_model	O(1)+2*O(n)+2*O(m*n)
8	svm_check_probability_model	O(1)
9	svm_predict_probability	O(1)+3*O(n)+O(n^2)
10	svm_predict	O(1)+4*O(n)+2*O(n^3)

The complexity of SVM is

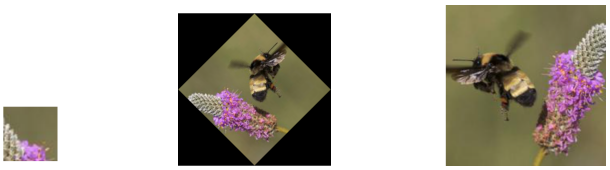
O(max(n,d) min (n,d)^2)
where n is the number of points and d is the number of dimensions



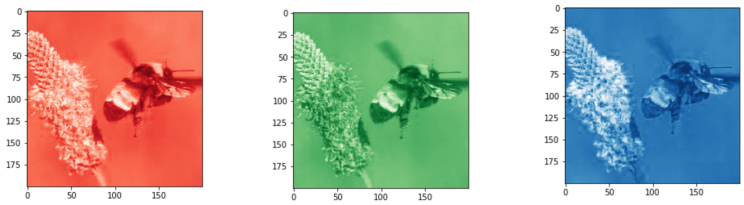
The image processing takes 9.218374013900757 seconds to run. The classification model takes 14.60240912437439 seconds to run.

4.6 Snap shots of the results

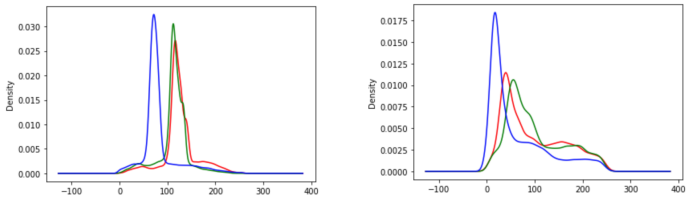
- Cropped,rotated and flipped images



- Red, green and blue channel plots



- rgb density plot of honey and bumble bee images

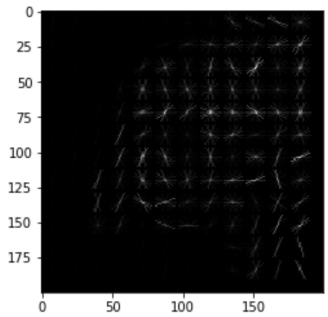


- Histogram of Oriented Gradients

```
In [4]: # run HOG using our greyscale bombus image
        hog_features, hog_image = hog(grey_bombus,
                                      visualize=True,
                                      block_norm='L2-Hys',
                                      pixels_per_cell=(16, 16))

        # show our hog_image with a grey colormap
        plt.imshow(hog_image, cmap=mpl.cm.gray)
```

```
Out[4]: <matplotlib.image.AxesImage at 0x1a25f51ed0>
```



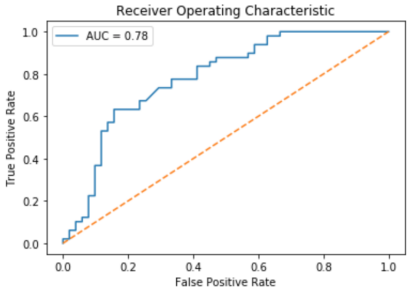
- Accuracy

```
# generate predictions
y_pred = svm.predict(X_test)

# calculate accuracy
accuracy = accuracy_score(y_pred, y_test)
print('Model accuracy is: ', accuracy)
```

```
Model accuracy is:  0.72
```

- ROC Curve



- Classification Report

	precision	recall	f1-score	support
0.0	0.73	0.71	0.72	51
1.0	0.71	0.73	0.72	49
accuracy			0.72	100
macro avg	0.72	0.72	0.72	100
weighted avg	0.72	0.72	0.72	100

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

This paper gives a brief introduction to the basics of image manipulation classification used in the field of image processing. The proposed approach using SVM as a classifier for classification of bees provides a classification efficiency of 72% . The proposed approach is computationally less expensive and also yields result.

5.2 Future Scope

The outcomes in this research are based on results that involve only sample datasets. It is necessary that additional datasets should be considered for the evaluation of different classification problems as the information growth in the recent technology is extending to heights beyond assumptions. Recent field of technology is growing and data are by nature dynamic. Hence, further classification of the entire system needs to be implemented right from the scratch since the results from the old process have become obsolete. The model, with incremental learning, can be used in categorization process to improve the following aspects in each type of problems.

- The scheme of restoring the blurred image is currently implemented on grey level. This model can be enhanced in future, by making it suitable for real-time images, in an image restoration system. This model can also be used in videos, to classify the genus of a bee that is briefly presented on a running video.
- This model can be built into a simple deep learning model that can automatically detect honey bees and bumble bees and then load a pre-trained model for evaluation.
- The image classification model can be enhanced in future, by including more low-level features such as shape and spatial location features apart from optimizing the weights and learning rate of the neural network.

References

[1]Umesh P.Image Processing in Python.2012.Department of Computational Biology and Bioinformatics,University of Kerala.CSI Communications.pp22-24.
[2]H. C. Andrews, A. G. Tescher, R. P. Kruger, "Image processing by digital computer".1972.IEEE Spectrum. vol. 9, no. 7, pp. 20-32.
[3]J. Lee, "Digital image processing by use of local statistics", Proc. 1978.IEEE Computer Soc. Conf. on Pattern Recognition and Image Processing.pp. 55-61, 1978.
[4]St efan van der Walt, Johannes L.Sch onberger, Juan Nunez-Iglesias, Fran cois Boulogne, Joshua D. Warner, NeilYager, Emmanuelle Gouillart, Tony Yu.2014.scikit-image: image processing in Python.PeerJ
[5]Alvarez L., Guichard F., Lions P. L., and Morel .1. M. Axioms and fundamental equations of image processing. Archive for Rational Mechanics and Analysis.1993.123(3):199-257.
[6]John Hunter and Darren Dale. The Matplotlib User's Guide.2012.
[7]David Wayne Emb-ley.Programming with data frames for everyday data items.1980.acm.org
[8]WesMcKinney.pandas: a Foundational Python Library for Data Analysis and Statistics.2011.dlr.de.
[9]N. Dalal, B. Triggs.Histograms of oriented gradients for human detection.2005.IEEE.
[10]Zubair Nabi.Machine Learning at Scale.2016.Pro Spark Streaming.
[11]V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. SIAM Journal on Matrix Analysis and Applications.2009.31(3):1100–1124.
[12]Mayukh Bhattacharyya.3 Things You Need To Know Before You Train-Test Split.2019.Towards Data Science.
[13]V. Vapnik. The Nature of Statistical Learning Theory.1995.NY: Springer-Verlag.p 123–180.
[14] Y. Zhan and D. Shen, "Design efficient support vector machine for fast classification", Pattern Recognition.2005.vol. 38, pp. 157-161.
[15] Y. Tang, B. Jin, Y. Sun and Y. Zhang, "Granular support vector machines for medical binary classification problems", Proceedings of 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology.2004.CIBCB '04, pp. 73-78.
[16] Mangasarian OL, Wild EW Multisurface proximal support vector machine classification via generalized eigenvalues.2005.IEEE Trans Pattern Anal Mach Intell 28(1):69–74.
[17] E. Osuna, R. Freund, and F. Girosi. Support vector machines: training and applications. AI Memo 1602.1997.MIT.