BEE SPECIES IDENTIFICATION, USING SUPPORT VECTOR MACHINE (SVM)

BY

ABDULLAHI HASSAN

DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF SCIENCE,
FEDERAL UNIVERSITY OF KASHERE,
GOMBE STATE

AUGUST, 2024

BEE SPECIES IDENTIFICATION, USING SUPPORT VECTOR MACHINE (SVM)

BY

ABDULLAHI HASSAN

FUKU/SCI/19/COM/0203

PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE, FEDERAL UNIVERSITY OF KASHERE, IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARS OF BACHELOR OF SCIENCE (B.Sc.) IN COMPUTER SCIENCE

AUGUST, 2024

DECLARATION

I declare that this project titled **BEE SPECIES IDENTIFICATION, USING SUPPORT VECTOR MACHINE(SVM)** is my own work and has not submitted in any form for another degree or diploma at any other institution. The project was successfully done under the supervision of Mr. Silas O. Dada. Information derived from the literature has been acknowledged in the text and a list of references is given.

ABDULLAHI HASSAN **Date**

CERTIFICATION

This is to certify that the project titled **BEE SPECIES IDENTIFICATION, USING SUPPORT VECTOR MACHINE(SVM)** submitted by Abdullahi Hassan with the matric number FUKU/SCI/19/COM/0203 is work carried out under my supervision and is worthy of consideration for the award of the degree of Bachelor of Science in Computer Science of the Federal University of Kashere, Gombe State.

______________________________ ______________________________

Mr. Silas O. Dada **Date**

Supervisor

______________________________ ______________________________

Mal. Muhammed Besiru Jibrin **Date**

Head of Department

______________________________ ______________________________

Date

Prof. N. V. Blamah

External Supervisor

ACKNOLEDGEMENTS

All thanks to Allah Subhanahu Wataala for his protection and guidance through my academic journey. I would to express my deepest gratitude to my parents Mr. Hassan Musa, Mrs. Khadija Umar for their prayers, love and support. Special thanks to my supervisor Mr. Silas O. Dada who sacrifice his time and effort to make sure this project is successfully done. I would also like thank Prof. Umar Alhaji Pate, the university vice chancellor, Prof. Abdulrazak Bello, Dean faculty of science, Dr. Muhammad Bashir Jibrin, head of computer science department, Federal University of Kashere, Mr. J. S. Caleb, our level coordinator and all others lecturers of Computer Science department. I also wish to thank all my respective and supportive colleagues, especially Abdurrasheed Garba, Abdulwahab Nasir Pela, Suleiman Isa, Albani Umar, Umar Gana, Adam Adoji, Sinan Muhammad, Idris Army, Abdulraheem Muhammad Auwal, Bashir Kwami, Kabiru Alhaji Ali, Bashir, Malam Yusuf, Abdulrauf Rabiu, and lot more. I also grateful to my family members and friends, especially my big brother Abdulrahman Aminu, other siblings Muhammad Hassan, Musa Hassan, Hasiya Hassan, Zainab Hassan. I'm grateful to anyone for the love, support and care.

DEDICATION

To my lovely parents, that give total support to ensure that my academic journey is successful, I love you with all of my heart.

ABSTRACT

The identification of bee species, particularly honey bees (Apis) and bumble bees (bombus), is crucial for ecology, agriculture, and environmental studies. Traditional methods of species identification are time-consuming and error-prone. This project aims to develop a classification model using machine learning techniques to identify bee species from images. A support Vector Machine (SVM) model was trained to classify images of honey bees and bumble bees. The model achieved an accuracy score of 0.68 and an AUC of 0.74. The final product includes analytic Jupyter notebook where I run all the model analysis and a web application developed using Django, allowing users to upload images and receive classification result of their image of bee species. The application of machine learning in this project demonstrates its potential in automating and improving the accuracy of species identification tasks.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background of the study

Bees play an important role in pollination and global ecosystem, a lot of Bees species certainly face threats from habitat loss, pesticides, and climate change Bees are winged insects closely related to wasps and ants, known for their roles in pollination and, in the case of the best-known bee species, the western honey bee, for producing honey (Ball, 2021). Bees are a monophyletic lineage within the superfamily Apoidea. They are currently considered a clade, called Anthophilous. There are over 20,000 known species of bees in seven recognized biological families. Some species – including honey bees, bumblebees, and stingless bees – live socially in colonies while most species (>90%) – including mason bees, carpenter bees, leafcutter bees, and sweat bees – are solitary. Bees are the most important pollinators in both natural and agricultural ecosystem, they are critical for the pollinating of most of the flowering plants, which includes some crops that significantly human relay on for food, such as dairy and meet because various livestock animals, like cows and goat, feed on plants like alfalfa and clover, which pollinated by Bees (Goulson, 2010).

1.1.1 Ecological roles of Bees

1. Pollination: Bees are responsible for the pollination of around 75% of the world's flowering plants and about 35% of global agricultural crops (Klein et al., 2007). This includes a variety of fruits, vegetables, nuts, and seeds, which are essential for human nutrition.
2. Biodiversity Maintenance: By pollinating a wide range of plants, bees contribute to the diversity and health of ecosystems. This biodiversity is crucial for ecosystem resilience and the provision of ecosystem services (Potts et al., 2016).

3. Economic Value: The economic value of pollination services provided by bees is estimated to be in the billions of dollars annually. These services are crucial for the production of high-value crops such as almonds, apples, blueberries, and coffee (Gallai et al., 2009).

1.1.2 Threats that Bees are facing

1. Habitat loss: urbanization, agricultural expansion, and deforestation have led to the loss of habitats that bees depend on for nesting and foraging (Potts et al., 2010).
2. Pesticides: The use of pesticides, particularly neonicotinoids, has been linked to declines in bee populations. These chemicals can be toxic to bees, affecting their navigation, foraging behavior, and reproductive success (Goulson et al., 2015).
3. Climate change: Changes in climate patterns affect the availability of floral resources and nesting sites for bees. Additionally, climate change can disrupt the synchrony between bees and the flowering plants they pollinate (Kerr et al., 2015).

1.1.3 Importance of identifying Bees species

1. Conservation efforts: Identifying bee species accurately is critical for conservation efforts. Different species have different ecological roles and conservation needs. Effective conservation strategies depend on understanding the distribution and status of various bee species (Winfree et al., 2009).
2. Agricultural practices: Farmers and agricultural practitioners can benefit from knowing which bee species are present in their fields and how to support them. This knowledge can lead to better crop management practices that enhance pollination services (Garibaldi et al., 2013).
3. Research and monitoring: Accurate identification of bee species is essential for research and monitoring programs aimed at understanding bee populations and their dynamics.

This information is crucial for tracking changes in biodiversity and the impacts of environmental stressors (Goulson et al., 2015).

Accurate identification of Bees species is essential for conservation efforts and understanding population dynamics. This project proposes the development of machine learning algorithm that is capable of identifying and recognizing Bees species from the images.

1.2 Statement of the problem

The accuracy in identifying Bee species presents several challenges due the variations in bee morphology, environmental conditions, and image quality. Manual identification methods are not suitable and scalable to accurately recognize bee specie and can lead into large-scale biodiversity monitoring and conservation efforts. Therefore, there is a need for an automated solution that can reliably identify bee species from images, enabling rapid and large-scale assessment of bee populations and supporting conservation initiatives (Goulson, 2010).

1.3 Aim and objectives of the study

This project aims to develop machine learning algorithms/system that can help in identifying Bees species using support vector machine (SVM) and image processing techniques.

1.2.1 Specific objectives

1. To develop a machine learning model using SVM for the classification of bee species from images.
2. To implement image processing techniques to enhance feature extraction and model performance.
3. To train the SVM model on a preprocessed bee images and evaluate its performance using accuracy metrics.
4. To develop Django web Application in order for the model to predict untrained data.

5. To visualize model performance using ROC curve analysis to assess its discriminatory power.

1.3 Significance of the study

The significance of this study lies in its potential to provide a scalable and automated solution for bee species identification. By leveraging machine learning and image processing techniques, the study aims to contribute to the conservation of bee populations and the sustainability of ecosystems. The developed model can be used by researchers, conservationists, and agricultural practitioners to monitor bee populations, implement conservation strategies, and support biodiversity (Goulson, 2010).

1.4 Scope of the study

The scope of this study includes the development and evaluation of a machine learning model for bee species identification and implementing a simple web application for untrained data prediction. The study will focus on the following aspects:

1. Collection and preprocessing of bee image datasets.
2. Feature extraction using image processing techniques such as histogram of oriented gradients (HOG).
3. Development and training of the SVM model for bee species classification.
4. Evaluation of model performance using accuracy metrics and ROC curve analysis.

1.5 Limitations of the study

While the proposed study aims to develop a robust model for Bee species identification, a lot of limitations must arise:

1. **Computational resources:** training an SVM model, particularly with large data and complex feature extraction methods like HOG, requires significant computational power and time.
2. **Overfitting:** The model may be prone to overfitting the training dataset, especially if the dataset is not well diverse.
3. **Dataset quality and quantity:** the accuracy of the model is highly dependent on the quality and quantity of the data, if the data is not sufficient enough or has poor data it will affect the model performance.
4. **Generalizability to Other species:** The model is developed for the specific species of insect, so other species data may not be train well on the model without recreating the model.
5. **Variability in image conditions:** Variations like light, angle, and background of the images may have impact on the extraction process and classification accuracy.

1.6 Project Organization

The study is comprised of five chapters, So the rest of the write-up will be as follow:

Chapter Two: This chapter review the relevant literature regarding the study, machine learning models and image processing techniques.

Chapter Three: This chapter outlines the methodology used in the study, including data collection, preprocessing, model development and evaluation.

Chapter Four: This chapter presents the results and discussion, including the model performance.

Chapter Five: This chapter concludes the study, summarizing the findings and proving recommendations for future research.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter review the existing study on Bee species identification by many researchers that used different machine learning models, the application of machine learning in species classification, and image processing techniques used for feature extraction (Gupta, R 2018). Understanding these topics provides a foundation for developing an effective machine learning model for identifying bee species from images. Bee species identification, particularly distinguishing between honey bees (*Apis mellifera*) and bumble bees (*Bombus* spp.), is essential for various ecological, agricultural, and environmental studies (Johnson & Lee 2020). Traditional methods of species identification often rely on expert knowledge and manual inspection, which can be time-consuming and error-prone. Machine learning (ML) and computer vision techniques have emerged as powerful tools to automate and enhance the accuracy of species identification tasks. This chapter provides an in-depth review of the relevant literature on bee species identification, the use of ML techniques for image classification, and the application of these methods in ecological research (Smith et al., 2019).

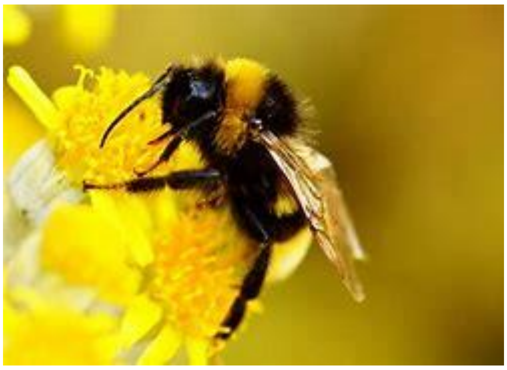


Fig1: Bumble Bee

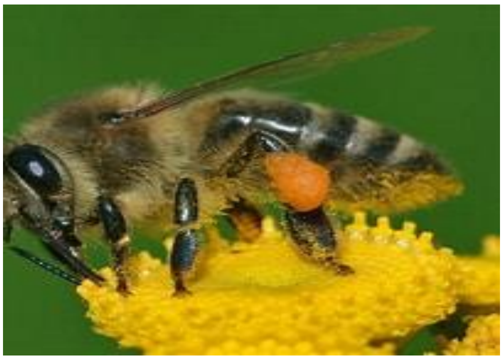


Fig2: Honey Bee

2.2 Overview of Machine Learning

Machine Learning is a branch of artificial intelligence (AI) that enables computers to “self-learn” from training data and improve over time, without being explicitly programmed. Machine learning algorithms are able to detect patterns in data and learn from them, in order to make their own predictions. In short, machine learning algorithms and models learn through experience. In traditional programming, a computer engineer writes a series of directions that instruct a computer how to transform input data into a desired output. Instruction is mostly based on an IF-THEN structure: when certain condition is met, the program executes a specific action (Russell & Norvig, 2016).

2.2.1 Types of Machine Learning

Machine learning is basically divided into two distinct types of supervised and unsupervised learning. However, there are more types of machine learning such as deep learning and reinforcement learning. Here are some types of machine learning:

1. Supervised Learning: Supervised learning algorithms and supervised learning models make predictions based on labeled training data. Each training sample includes an input and a desired output. A supervised learning algorithm analyzes this sample data and makes an inference-basically, an educated guess when determining the labels for unseen data (Russell & Norvig, 2016).
2. Unsupervised Learning: Unsupervised learning algorithms uncover insights and relationships in unlabeled data. In this case, models are fed input data but the desired outcomes are unknown, so they have to make inferences based on circumstantial evidence, without any guidance or training. The models are not trained with the “right answer,” so they must find patterns on their own (Russell & Norvig, 2016).

3. Semi-Supervised Learning: In this type, the training data is split into two. A small amount of labeled data and a larger set of unlabeled data. In this case, the model uses labeled data as an input to make inferences about the unlabeled data, providing more accurate results than regular supervised learning models (Russell & Norvig, 2016).
4. Reinforcement Learning: is concerned with how a software agent (or computer program) ought to act in a situation to maximize the reward. In short, reinforced machine learning models attempt to determine the best possible path they should take in a given situation (Russell & Norvig, 2016).
5. Deep Learning: Deep learning are advanced machine learning algorithms used by tech giants, like Google, Microsoft and Amazon to run entire systems and power things, like self-driving cars and smart assistants (Russell & Norvig, 2016).

2.2.2 Machine Learning Algorithms

There are several machine learning algorithms that are used for different areas and purposes, here are some of the important and most used machine learning algorithms:

1. Naïve Bayes and SVM: Are classification algorithms that are used to predict features based on class. In classification tasks, the output value is a category with finite number of options. For example, with the sentiment analysis models, you can automatically classify data as positive, negative, or neutral (Russell & Norvig, 2016).
2. Regression: Is a machine learning algorithm that predict a relationship between variables. In regression tasks, the expected result is a continuous number. This model is used to predict quantities, such as the probability an event will happen, meaning the output may have any number value within a certain range (Russell & Norvig, 2016).
3. Clustering: Clustering is an unsupervised type learning where the model trained and predict an unlabeled data. For example, given a set of income and spending data, a

machine learning can identify groups of customers with similar behaviors (Russell & Norvig, 2016).

2.2 Importance of Bees

Bees play a pivotal role in pollination, a process vital for the reproduction of many flowering plants and agricultural crops. Honey bees are known for their economic value, particularly in honey production and crop pollination. Bumble bees, on the other hand, are recognized for their efficiency as pollinators in various environments, including colder climates where honey bees are less effective. Pollination by bees contributes to biodiversity, ecosystem stability, and food security. According to Klein et al. (2007), pollinators are responsible for the successful reproduction of approximately 75% of flowering plants and 35% of global food crops. The decline in bee populations, attributed to factors such as habitat loss, pesticides, and climate change, poses a significant threat to global agriculture and biodiversity (Potts et al., 2010).

2.3 Traditional Methods of Bee species Identification

Traditional methods of bee species identification involve morphological examination, which requires specialized knowledge and experience. These methods include examining physical characteristics such as body size, color patterns, and wing venation. However, such approaches are labor-intensive and may not always be accurate due to intra-species variation and the presence of cryptic species (Michener, 2007).

2.4 Machine Learning and Image Classification

Machine learning, particularly deep learning, has revolutionized the field of image classification. Convolutional Neural Networks (CNNs) are among the most widely used techniques for image recognition tasks due to their ability to automatically extract and learn features from raw pixel data (Krizhevsky, Sutskever, & Hinton, 2012). Several studies have demonstrated the effectiveness of CNNs in various ecological applications, including species identification. For instance, Norouzzadeh et al. (2018) developed a deep learning model to classify animal species from camera trap images, achieving high accuracy and demonstrating the potential of ML in wildlife monitoring.

2.5 Application of Machine learning in species identification

Recent research has applied ML techniques to the identification of bee species. A study by (Jones et al. 2018) utilized a CNN to classify bee species from images, achieving an accuracy of over 90%. Similarly, (Tan, Frosch, and Bjerge 2020) developed a machine learning model to distinguish between honey bees and bumble bees, demonstrating the feasibility of automated bee identification using image data. The integration of ML models with mobile applications and web platforms has further enhanced the accessibility and usability of these technologies. Apps like "Bee Machine" allow users to upload images of bees and receive real-time species identification, making it easier for researchers and the public to monitor bee populations (Bjerge et al., 2020).

2.5.1 Machine Learning

Machine learning (ML) is a subset of artificial intelligence that focuses on developing algorithms that enable computers to learn from and make predictions based on data. In species identification, ML algorithms can be trained to recognize patterns and features in images, leading to accurate classification (Cortes & Vapnik, 1995).

2.5.2 Support vector machines (SVM)

Support Vector Machines (SVM) are a powerful class of supervised learning models used for classification and regression tasks. SVMs work by finding the hyperplane that best separates data points of different classes (Cortes & Vapnik, 1995). SVMs are particularly effective in high-dimensional spaces and have been widely applied in biological classification problems (Boser, Guyon, & Vapnik, 1992).

2.5.3 Applications of SVM in species identification

SVMs have been successfully used in various species identification tasks. For instance, (Guo et al. 2016) demonstrated the effectiveness of SVM in classifying plant species based on leaf images. Similarly, SVMs have been applied to insect classification, including butterfly and bee species, showing high accuracy and reliability.

2.6 Image Processing Techniques

An image processing technique is a technique used to manipulate and transform images into desired object to extract features in it, here are some mechanisms in image processing techniques:

2.6.1 Importance of Feature Extraction

Feature extraction is a critical step in image processing that involves identifying and isolating significant attributes of an image. Effective feature extraction enhances the performance of machine learning models by providing them with relevant information (Chen & Zhang, 2014).

2.6.2 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) is a widely used feature extraction technique in image processing. HOG captures the distribution of gradient orientations in localized portions of an image, making it effective for detecting objects and patterns (Dalal & Triggs, 2005). HOG has been successfully used in various image classification tasks, including human detection and animal species identification (Chen & Zhang, 2014).

2.6.3 Applications of HOG in species identification

HOG can be particularly useful in bee species identification due to its ability to capture detailed structural information. By extracting HOG features from bee images, the SVM model can learn to distinguish between different species based on their morphological characteristics (Chen & Zhang, 2014).

2.7 Model Overview

A model is a set of algorithms that are develop to perform a specific task, our model is a support vector classifier of SVM that will use to classify bee species through images. Below is overview of our model:

2.7.1 Data collection and preprocessing

The first step in the project involves collecting a diverse dataset of bee images. Each image should be labeled with the corresponding species. Preprocessing steps include resizing the images, converting them to grayscale using the `rgb2gray` function, and normalizing pixel values.

2.7.2 Feature extraction

Feature extraction is performed using the Histogram of Oriented Gradients (HOG) method. HOG extracts feature by calculating the gradients' orientations and magnitudes, and then constructing histograms based on these gradients. This technique effectively captures the bees' morphological features, which are crucial for accurate classification.

2.7.3 Model training

The extracted HOG features are used to train the Support Vector Machine (SVM) model. The dataset is split into training and testing sets using the `train_test_split` method. The SVM model is trained on the training set, adjusting parameters to minimize classification errors. The model's performance is evaluated on the testing set using accuracy score.

2.7.4 Model evaluation

The model's performance is further evaluated using the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC). The ROC curve provides a visual representation of the model's ability to distinguish between classes across different threshold settings. A high percentage of AUC shows good model performance.

2.7.5 Implementation

For practical use, the trained SVM model can be deployed in a web application using Django. This allows users to upload bee images and receive species identification results without needing to interact with the underlying algorithm. Django provides a user-friendly interface and handles the backend operations, ensuring the model's accessibility and usability.

2.8 Review of Related Studies

Below review of the previous studies that used machine learning algorithms to classify species from images to differentiate between various types of species:

2.8.1 Machine learning for species identification

Machine learning has been widely applied in various biological classification tasks. For example, Arzoumanian et al. (2005) utilized pattern recognition algorithms to identify whale species based on fluke photographs, demonstrating the potential of automated species identification. Similarly, the use of convolutional neural networks (CNNs) for plant species identification has shown promising results, as discussed by Lee et al. (2015).

2.8.2 SVM Applications in insect classification

Support Vector Machines (SVMs) have been successfully used in insect classification. Huang et al. (2014) developed an SVM-based system for classifying butterfly species from images, achieving high accuracy rates. Additionally, the work by Wen et al. (2012) on identifying different mosquito species using SVM underscores the versatility and robustness of this approach in entomology.

2.8.3 Feature extraction techniques

The effectiveness of machine learning models largely depends on the quality of features extracted from the images. Dalal and Triggs (2005) introduced the Histogram of Oriented Gradients (HOG) for human detection, which has since been adapted for various object recognition tasks, including animal identification. Zhang et al. (2017) applied HOG features combined with machine learning classifiers to identify bird species from images, highlighting the adaptability of HOG in different contexts.

2.8.4 Image-Based Classification

Another significant study by Kenji et al. (2018) focused on using image-based classification methods for bee identification. They employed a combination of histogram of oriented gradients (HOG) and support vector machines (SVM) to classify bee images, achieving an accuracy of over 80%. This study highlighted the effectiveness of feature extraction techniques in enhancing classification accuracy.

2.8.5 A reference process for automating bee species identification

Bees play a crucial role in biodiversity conservation, since they provide vital services to both natural ecosystems and agriculture. Fabiana S Santana and Anna H Reali Costa (2014) Developed automated tools for bee species identification, that help in assisting the identification of bee species to represents an important contribution to biodiversity and agriculture.

2.8.6 Evaluating classification and feature selection for bee species

The main pollinator commercially available, that is apis mellifera is now facing population decrease due to colony collapse. Felipe Leno da Silva and Marina Lopes Grassi (2015) Evaluate bee species based on seven combinations of feature selector and classifiers by their hit ratio with real bee wing images.

2.8.7 Comparative Analysis of Machine Learning Algorithms

A comparative analysis conducted by Jones et al. (2019) evaluated various machine learning algorithms for bee species identification, including decision trees, random forests, and SVMs. They found that SVMs provided the best balance between computational efficiency and accuracy, reinforcing the choice of SVM for our project.

2.8.8 Challenges in Bee Species Identification

Despite the advances, there are challenges in bee species identification using automated methods. Smith et al. (2020) discussed the difficulties in obtaining high-quality, labeled datasets and the variability in bee images due to different angles, lighting conditions, and background noise. These challenges necessitate robust preprocessing and feature extraction techniques to improve model performance.

2.8.9 Automated Identification of Bees

Several studies have explored automated identification methods for bee species using machine learning and computer vision techniques. For instance, a study by Renaud et al. (2017) utilized convolutional neural networks (CNNs) to classify images of bee species. They achieved high accuracy rates, demonstrating the potential of deep learning models in ecological research.

2.9 Benefits

The integration of HOG and SVM offers several benefits, including high accuracy, robustness to variations in image quality, and scalability. Automated bee species identification can significantly enhance biodiversity monitoring and conservation efforts by providing a fast and reliable tool for researchers and practitioners.

2.10 Challenges

However, challenges remain, such as the need for large labeled datasets and computational resources for training and testing the models. Variability in image conditions, such as lighting and background, can also impact the model's performance. Additionally, there is a risk of overfitting, which can be mitigated through techniques such as cross-validation and data augmentation. Another significant issue is the need for large, annotated datasets to train robust models. Collecting and annotating such datasets is time-consuming and requires expert knowledge. Additionally, the performance of ML models can be affected by variations in image quality, background clutter, and differences in lighting conditions (Wäldchen & Mäder, 2018). Moreover, while CNNs and other deep learning models can achieve high accuracy, they often operate as "black boxes," making it difficult to interpret their decision-making processes. This lack of interpretability can be a barrier to their adoption in scientific research and conservation efforts (Samek, Wiegand, & Müller, 2017).

CHAPTER THREE

METHODOLOGY, MODEL EVALUATION AND DESIGN

3.1 Introduction

This chapter details the methodology employed in developing a machine learning model to identify bee species from images using a Support Vector Machine (SVM). It covers data collection, preprocessing, feature extraction, model training, and evaluation. The goal is to provide a clear, reproducible approach for developing an accurate and efficient model, and to ensure a robust and accurate model capable of distinguishing between various bee species based on image inputs.

3.2 Methodology

Methodology comprised all the steps we followed, tools, and techniques we used to carried out the study. Below is the detail of our methodology:

3.2.1 Data Collection

Data collection is the we used to collect the data from the source

1. Data source

   Data from this project is collected from online platform (DataCamp website), The data is provided by entomologists and researchers.

2. Datasets Composition

   The data consist of 500 images of honey bees (apis) and bumble bees (bombus), that is each image is either for the honey bee or the bumble bee.

3. Data labeling

   Each image in the dataset is labeled with the corresponding bee species. This labeling is essential for supervised learning, where the model learns to associate features with specific classes.

4. Data diversity

   To ensure the model generalizes well, the dataset includes images taken under various conditions, including different lighting, backgrounds, and poses of the bees.

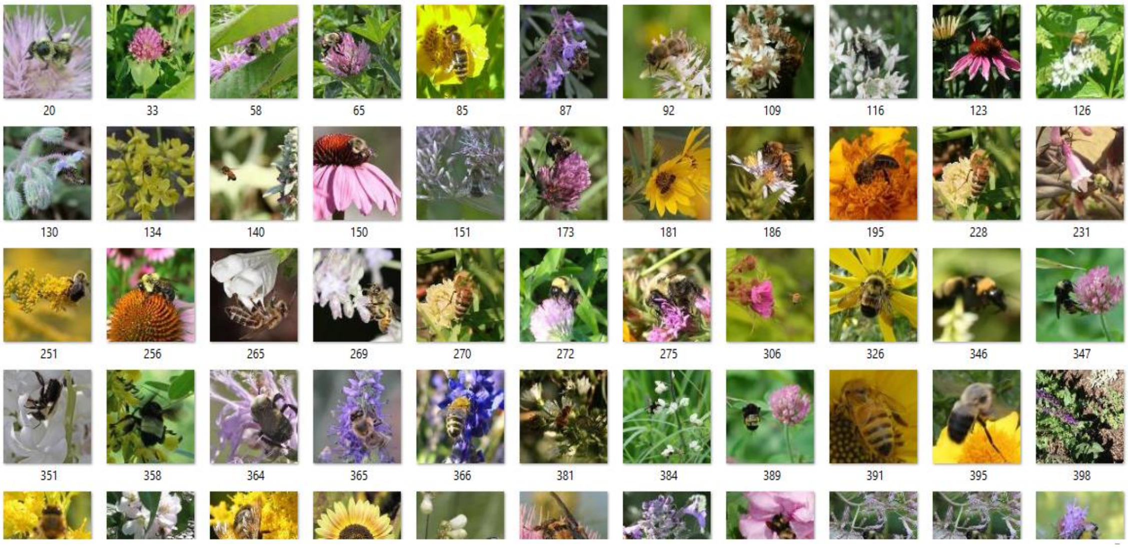


Fig 3: Images Dataset

3.2.2 Data Preprocessing

Below are the steps we followed to get our data ready:

1. Image Resizing

   All images are resized to a standard dimension (100x100 pixels) to ensure uniformity and reduce computational complexity.

2. The Function get_image

   The function get_image converts an index value from the dataframe into a file path where the image is located, opens the image using the image object in Pillow, and then returns the image as a numpy array.

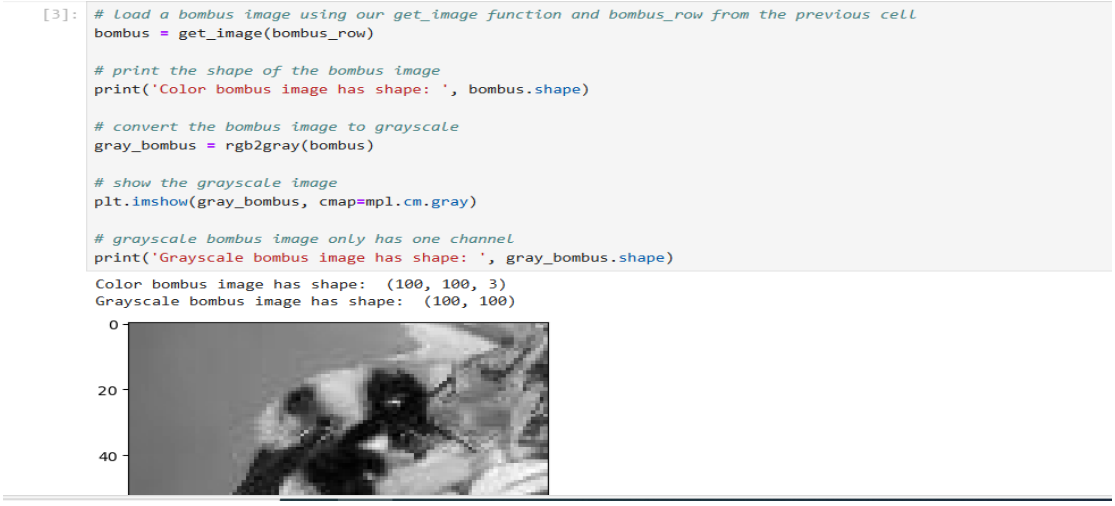
3. Grayscale Conversion

   Images are converted to grayscale using the `rgb2gray` function. Grayscale images simplify the data and focus the model on structural features rather than color.

The rgb2gray function computes the luminance of an RGB image using the following formula Y = 0.2125 R + 0.7154 G + 0.0721 B.

4. Normalization

   Pixels values are normalized into scale of (100x100x1) to standardize input data and improve model’s performance during training.



```
[3]: # load a bombus image using our get_image function and bombus_row from the previous cell
     bombus = get_image(bombus_row)

     # print the shape of the bombus image
     print('Color bombus image has shape: ', bombus.shape)

     # convert the bombus image to grayscale
     gray_bombus = rgb2gray(bombus)

     # show the grayscale image
     plt.imshow(gray_bombus, cmap=mpl.cm.gray)

     # grayscale bombus image only has one channel
     print('Grayscale bombus image has shape: ', gray_bombus.shape)
```

```
Color bombus image has shape:  (100, 100, 3)
Grayscale bombus image has shape:  (100, 100)
```

Fig 4: Image Processing

3.2.3 Feature Extraction

Feature extraction performed to get the features from images of the dataset

1. Histogram of Oriented Gradients

   An image is divided in a grid fashion into cells, and for the pixels within each cell, a histogram of gradient directions is compiled. To improve invariance to highlights and shadows in an image, cells are block normalized, meaning an intensity value is calculated for a larger region of an image called a block and used to contrast normalize all cell-level histograms within each block. The HOG feature vector for the image is the concatenation of these cell-level histograms.

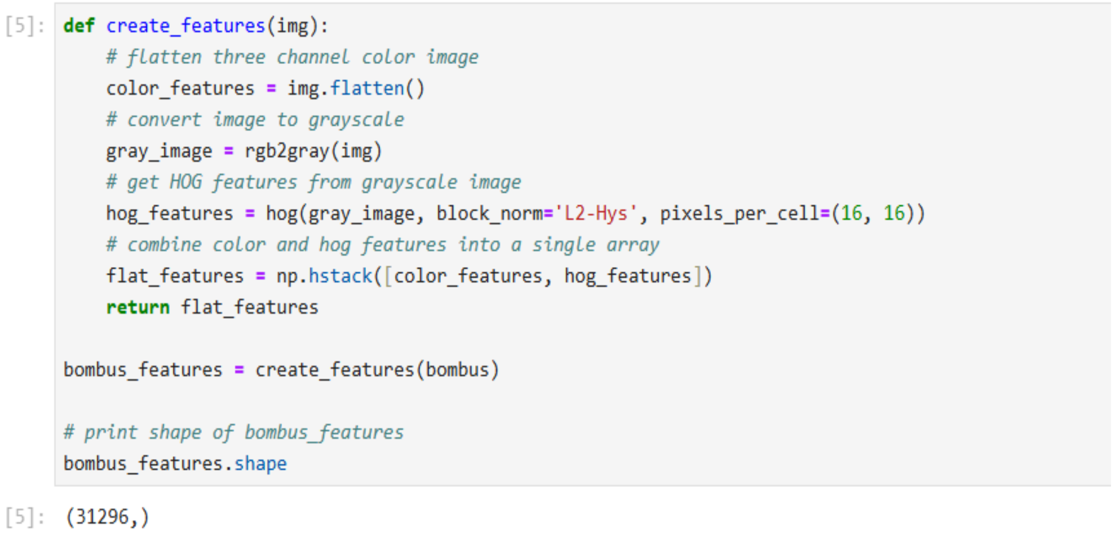
2. HOG parameters

Table 1: Key parameters for HOG

Cell Size	16x16 pixels
Block Size	L2-Hys
Visualize	True

3. Image features and Row flatten

An algorithm is created to format the data where rows correspond to images and columns correspond to features. This means that all the information for a given image needs to be contained in a single row. We want to provide our model with the raw pixel values from our images as well as the HOG features, we just calculated. To do this, we will write a function called create_features that combines these two sets of features by flattening the three-dimensional array into a one-dimensional (flat) array.



```
[5]: def create_features(img):
         # flatten three channel color image
         color_features = img.flatten()
         # convert image to grayscale
         gray_image = rgb2gray(img)
         # get HOG features from grayscale image
         hog_features = hog(gray_image, block_norm='L2-Hys', pixels_per_cell=(16, 16))
         # combine color and hog features into a single array
         flat_features = np.hstack([color_features, hog_features])
         return flat_features

     bombus_features = create_features(bombus)

     # print shape of bombus_features
     bombus_features.shape

[5]: (31296,)
```

Fig 5: Feature extraction

3.2.4 Model Training

We trained our model on the training dataset. Below are the steps:

1. Data Splitting

   The dataset is split into training and testing sets using an 70-30 split, ensuring that 70% of the data is used for training and 30% for testing.

2. Scale Training and Test Features

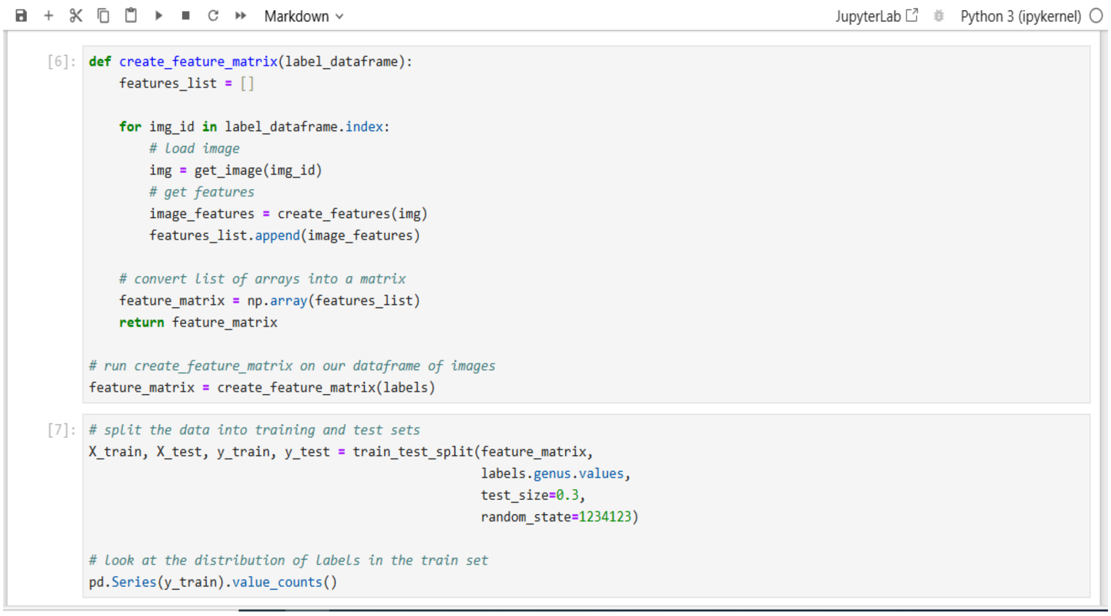
   The data is rescaled to work well with the model using “StandardScaler”. Sklearn provide this feature to made the model robustly work with the data.

3. Principal Component Analysis

   PCA is performed to reduce the number of features. We have 31296 features for each image and we have 500 images. PCA is a way of linearly transforming the data such that most of the information in the data is contained within a smaller number of features called components. We keep 350 components. This means our feature matrices train set and test set will only have 350 columns, rather than the original of 31,296.

4. Model Train

   Finally, the is trained with Support Vector Classifier (SVC) a type of SVM, An SVM a type of supervised machine learning model used for regression, classification, and outlier detection. An SVM 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.



```
def create_feature_matrix(label_dataframe):
    features_list = []

    for img_id in label_dataframe.index:
        # load image
        img = get_image(img_id)
        # get features
        image_features = create_features(img)
        features_list.append(image_features)

    # convert list of arrays into a matrix
    feature_matrix = np.array(features_list)
    return feature_matrix

# run create_feature_matrix on our dataframe of images
feature_matrix = create_feature_matrix(labels)
```

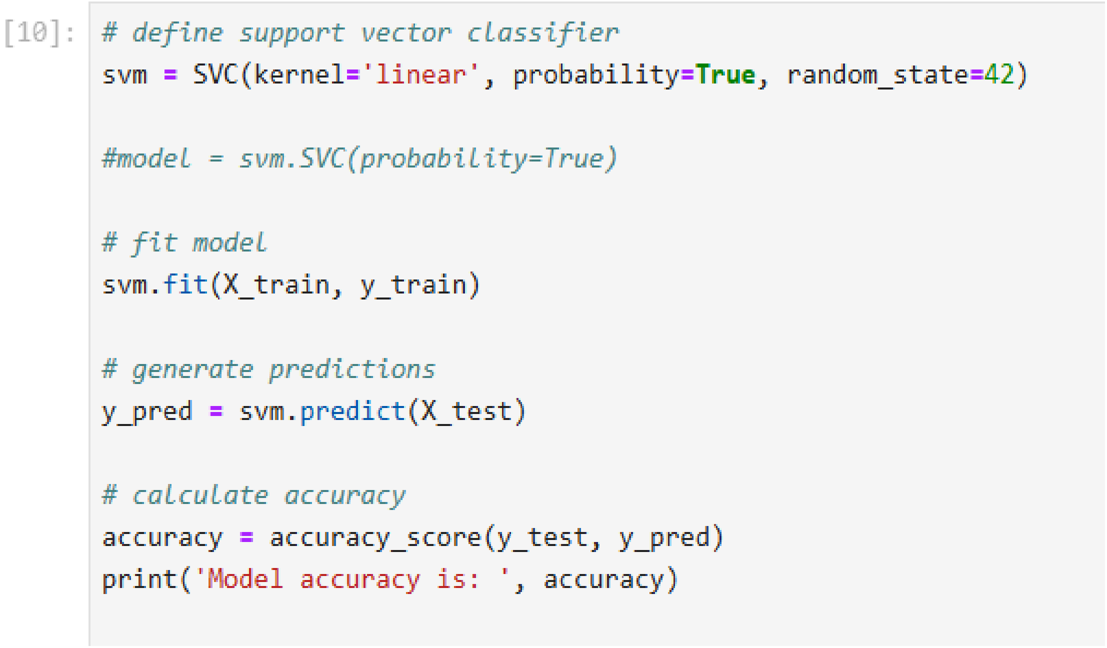
```
# split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(feature_matrix,
                                                    labels.genus.values,
                                                    test_size=0.3,
                                                    random_state=1234123)

# look at the distribution of labels in the train set
pd.Series(y_train).value_counts()
```

Fig 6: Model Training

3.2.5 Model Evaluation

The model's performance was evaluated using the testing set. Key evaluation metrics included accuracy, ROC curve, and AUC.

1. Accuracy: We used Accuracy metric to measures the proportion of correctly classified instances among the total instances.
2. ROC Curve and AUC: We used “svm.predict_proba” function to get the probabilities of the images on test set, The probabilities of 0.5 or above are assigned a class label of 1.0 and those below are assigned a 0.0. The Receiver Operating Characteristic Curve (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 Area Under the Curve is also calculated, where 1 means every predicted label was correct.



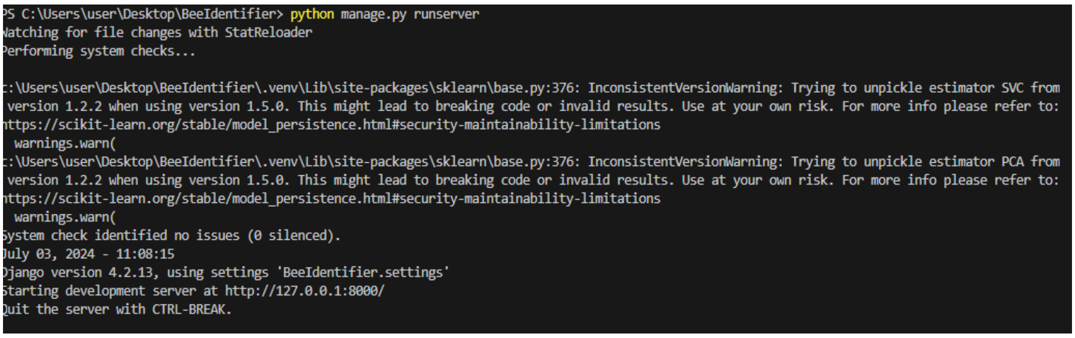
```
# define support vector classifier
svm = SVC(kernel='linear', probability=True, random_state=42)

#model = svm.SVC(probability=True)

# fit model
svm.fit(X_train, y_train)

# generate predictions
y_pred = svm.predict(X_test)

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

Fig 7: Model Evaluation

3.5 Web Application Development

Deploying the machine learning model involves integrating it into a web application to allow end users to upload bee images and receive predictions on the species. This section outlines the deployment process using Django web framework and provides the details explanation of how the model is been deployed. Below are the steps carried out to develop a user interface for the model:

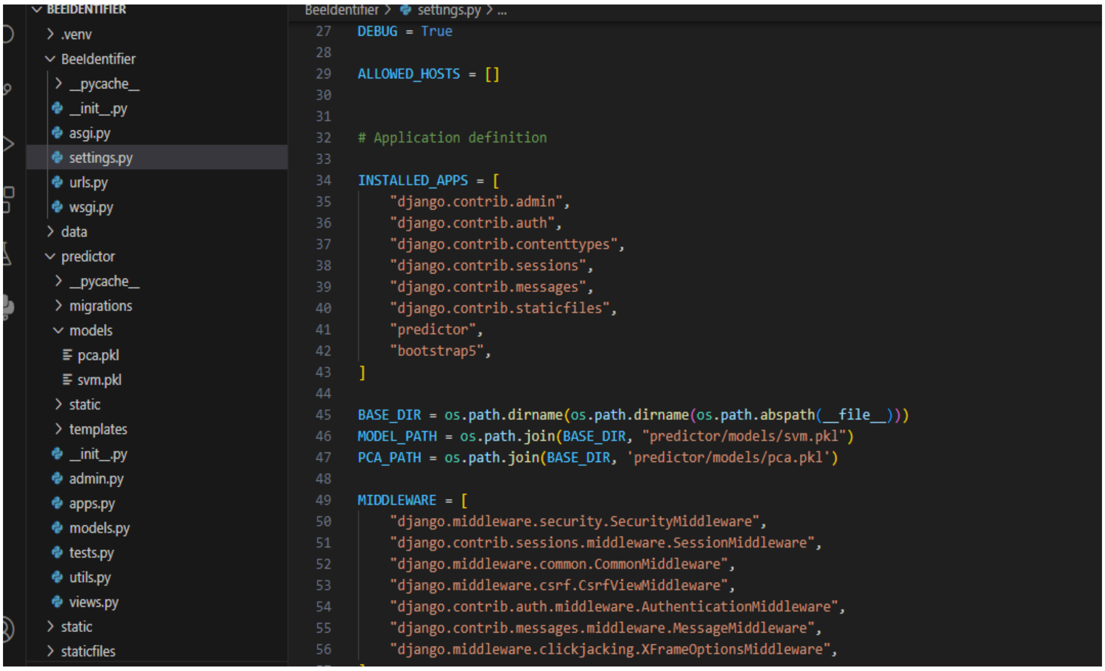
1. Environment Setup: Before we started the project we created our project folder, where virtual environment was created and all the necessary packages are installed, including Django, pillow, scikit-learn, joblib, numpy etc. Then we run Django development server:



```
PS C:\Users\user\Desktop\BeeIdentifier> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

C:\Users\user\Desktop\BeeIdentifier\.venv\Lib\site-packages\sklearn\base.py:376: InconsistentVersionWarning: Trying to unpickle estimator SVC from
 version 1.2.2 when using version 1.5.0. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\user\Desktop\BeeIdentifier\.venv\Lib\site-packages\sklearn\base.py:376: InconsistentVersionWarning: Trying to unpickle estimator PCA from
 version 1.2.2 when using version 1.5.0. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
System check identified no issues (0 silenced).
July 03, 2024 - 11:08:15
Django version 4.2.13, using settings 'BeeIdentifier.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig 8: Django Development Server

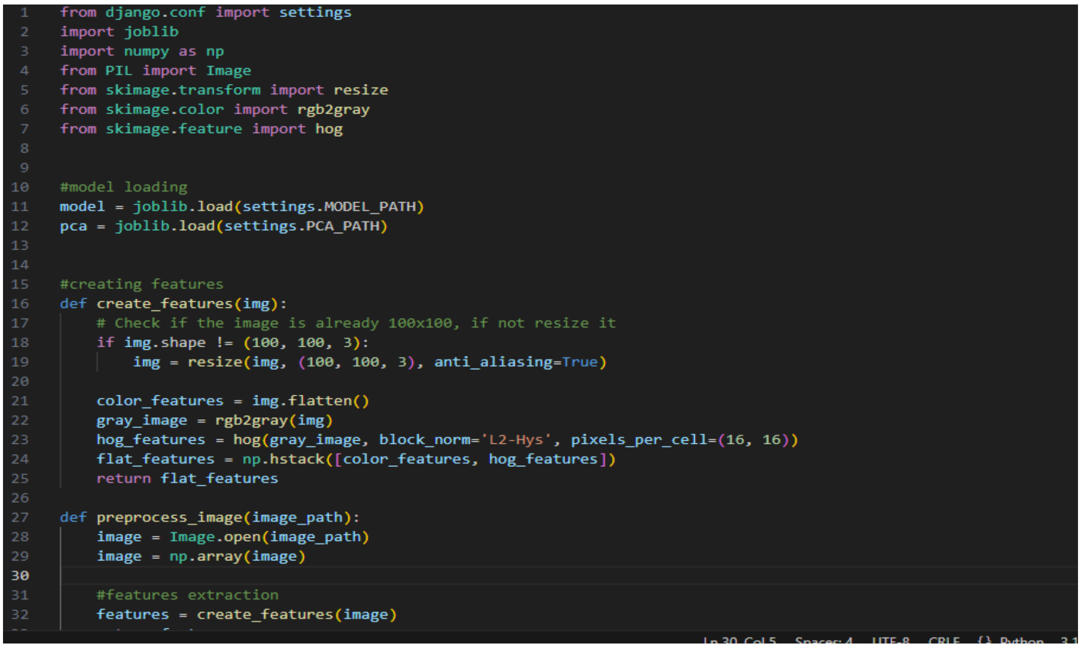
2. Django Project Creation: A new Django app is created within our project directory for our bee application and then we configured the app in our setting. The project structure was organized to include templates for HTML, static files for CSS and JavaScript, and media files for uploaded images.



```python
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "predictor",
    "bootstrap5",
]

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
MODEL_PATH = os.path.join(BASE_DIR, "predictor/models/svm.pkl")
PCA_PATH = os.path.join(BASE_DIR, 'predictor/models/pca.pkl')

MIDDLEWARE = [
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "django.contrib.auth.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
```

Fig 9: Django Settings

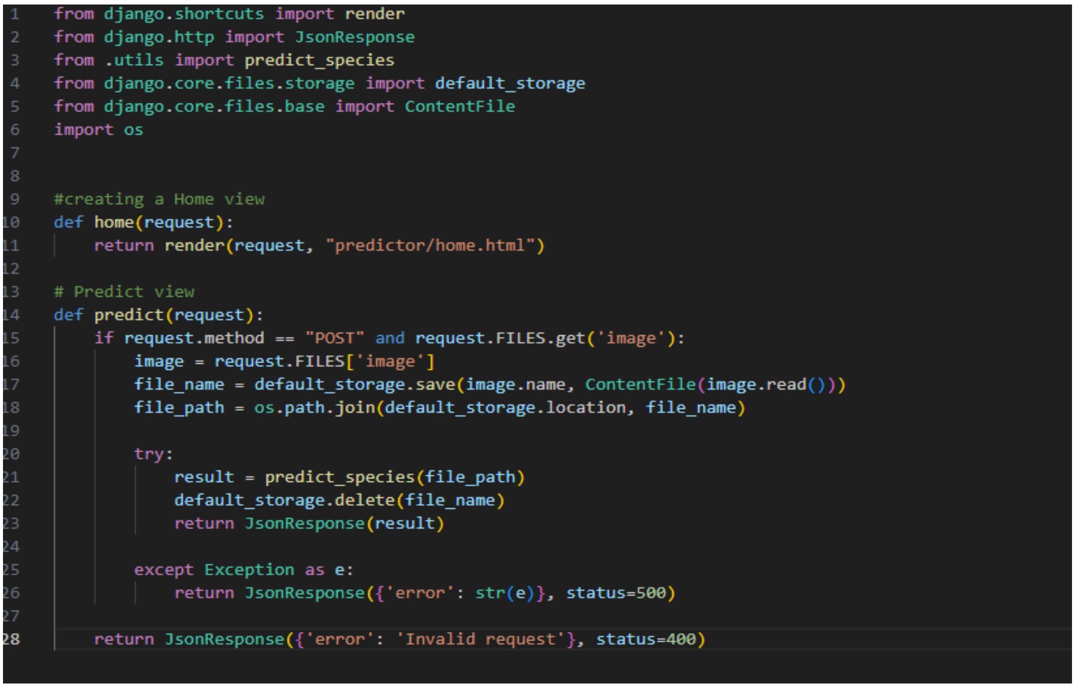
3. Model Integration: The trained SVM model was saved and loaded within the Django app. A function has been written to preprocess image and made classification. Below is how the logic was implemented.



```python
from django.conf import settings
import joblib
import numpy as np
from PIL import Image
from skimage.transform import resize
from skimage.color import rgb2gray
from skimage.feature import hog


#model loading
model = joblib.load(settings.MODEL_PATH)
pca = joblib.load(settings.PCA_PATH)


#creating features
def create_features(img):
    # Check if the image is already 100x100, if not resize it
    if img.shape != (100, 100, 3):
        img = resize(img, (100, 100, 3), anti_aliasing=True)

    color_features = img.flatten()
    gray_image = rgb2gray(img)
    hog_features = hog(gray_image, block_norm='L2-Hys', pixels_per_cell=(16, 16))
    flat_features = np.hstack([color_features, hog_features])
    return flat_features

def preprocess_image(image_path):
    image = Image.open(image_path)
    image = np.array(image)

    #features extraction
    features = create_features(image)
```

Fig 10: Django Model

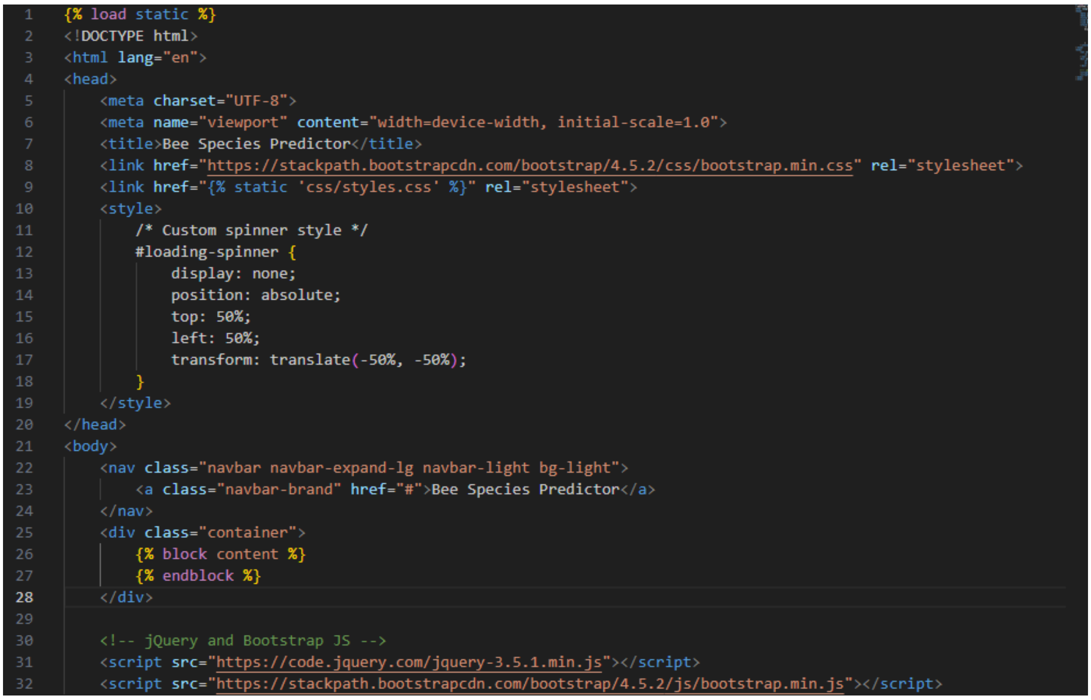
4. Views: Django views were created to handle image uploads and display predictions, handling templates flow and managing urls routes of the urls configuration, so that redirecting can be done.



```python
from django.shortcuts import render
from django.http import JsonResponse
from .utils import predict_species
from django.core.files.storage import default_storage
from django.core.files.base import ContentFile
import os


#creating a Home view
def home(request):
    return render(request, "predictor/home.html")

# Predict view
def predict(request):
    if request.method == "POST" and request.FILES.get('image'):
        image = request.FILES['image']
        file_name = default_storage.save(image.name, ContentFile(image.read()))
        file_path = os.path.join(default_storage.location, file_name)

        try:
            result = predict_species(file_path)
            default_storage.delete(file_name)
            return JsonResponse(result)

        except Exception as e:
            return JsonResponse({'error': str(e)}, status=500)

    return JsonResponse({'error': 'Invalid request'}, status=400)
```

Fig 11: Django Views

5. Templates and Static: HTML templates were designed to create a user-friendly interface for uploading images and viewing results. Static and media files were configured to handle CSS, JavaScript, and uploaded images.



```html
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bee Species Predictor</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <link href="{% static 'css/styles.css' %}" rel="stylesheet">
    <style>
        /* Custom spinner style */
        #loading-spinner {
            display: none;
            position: absolute;
            top: 50%;
            left: 50%;
            transform: translate(-50%, -50%);
        }
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">Bee Species Predictor</a>
    </nav>
    <div class="container">
        {% block content %}
        {% endblock %}
    </div>

    <!-- jQuery and Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

Fig 12: Django Templates

6. Testing and Debugging: The application was tested thoroughly to ensure that it worked as expected and that the model predictions were accurate.

3.6 Benefits of Web Application Development

1. Real-time Predictions: Users can receive real-time predictions, enhancing the practical utility of the model.
2. Scalability: The Django web framework supports scaling the application to handle multiple user requests efficiently.
3. User Accessibility: Users can easily access the model through a web interface without needing to understand the underlying algorithm.

3.3 The Proposed Model

The proposed model involves using the SVM algorithm to classify bee species based on extracted HOG features from grayscale images. This dataset should include multiple images of each species to account for variability in pose, lighting, and background. Here are the steps of the model workflow:

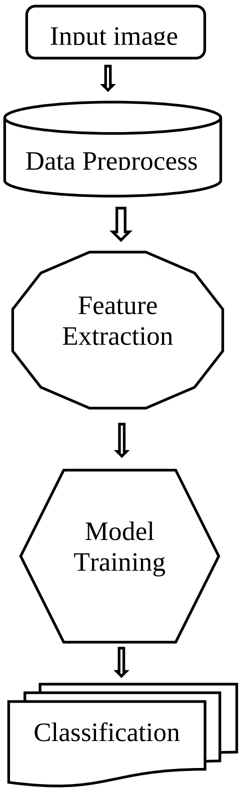


Fig 13: Model Overview

3.3.1 Benefits of the proposed model

1. Accuracy: SVM provides robust classification capabilities, making it suitable for distinguishing between visually similar species.
2. Efficiency: Preprocessing and feature extraction techniques ensure that the model is computationally efficient, allowing for quick predictions.
3. Scalability: The model can be trained on large datasets and adapted to include more species.

3.4 Ethical considerations

1. Privacy: No personally identifiable information is included in the dataset.
2. Data Source Creadability: Images were sourced from reputable, publicly available databases.
3. Animal Welfare: The dataset was curated responsibly, avoiding any harm or distress to bees during data collection.

CHAPTER FOUR

MODEL IMPLEMENTATION, RESULT, AND DISCUSION

4.1 Introduction

This chapter presents the results obtained from training and evaluating the Support Vector Machine (SVM) model for classification of bee species and the integrating the model into Django web application for classification of unseen data. The evaluation metrics used to assess the performance of the model include Accuracy, Receiver Operating Characteristic Curve (ROC), and Area Under a Curve (AUC). The results are discussed in detail, highlighting the effectiveness of the model and identifying all the limitations and areas for improvement.

4.2 Technical Requirements

Technical requirements are comprised all the necessary tools and equipment we used to carry out our study, we divided them into Hardware and Software requirements.

4.2.1 Hardware Requirements

Table below provided all the hardware infrastructure we used to run the project.

Table 2: Hardware requirements

Items	**Minimum**	**Recommended**
Processor Type	Intel Corei3	Intel Corei5
Memory	2GB RAM	4GB or more
Storage	32GB	64GB or more
Display	1366*768	1366*768 or more
Processor Speed	1.5GHz	2.0GHz

4.2.2 Software Requirements

The table 3 below showed the list of software requirements for the study.

Table 3: Software requirements

Category	**Type**
Operating System	Windows
Programming Language	Python
Web Development IDE	VScode
Model IDE	Jupyter Notebook in Anaconda Navigator
CMD	Anaconda Prompt

4.3 Model Performance Metrics

The primary metrics used to evaluate the performance of the model were accuracy, Receiver Operating Characteristic Curve (ROC), and Area Under Curve (AUC).

4.3.1 Accuracy

Accuracy is the proportion of true results (both the true positives and true negatives) among the total number of cases examined. It is calculated as:

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Number\ of\ Sample}$$

For our model, the accuracy score was found to be 0.68, which shows that the model correctly classified 68% of the data correctly, which is: $Accuracy = \frac{53 + 49}{150} = 0.68$

4.3.2 Area Under a Curve (AUC)

The AUC is a measure of the model's ability to distinguish between classes. An AUC of 0.5 suggests no discrimination (random chance), 0.7-0.8 is considered acceptable, 0.8-0.9 is considered excellent, and more than 0.9 is considered outstanding. The AUC for our model was 0.74, indicating a good level of discrimination between honey bees and bumble bees. Below table is a summary of AUC and Accuracy score:

Table 4: Performance Metrics

Metric	**Score**
Accuracy	0.68
Area Under a Curve	0.74

4.3.3 Confusion Matrix

A confusion matrix provides a detailed breakdown of the model's performance by comparing the actual versus predicted labels. Table below summarized our model confusion matrix:

Table 5: Confusion Matrix

	Predicted Honey Bee	Predicted Bumble Bee
Actual Honey Bee	53	22
Actual Bumble Bee	26	49

4.3.4 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positives. We used the values from the confusion matrix to get the precision values for both the honey bee and bumble bee.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$\text{Precision}_{\text{Honey bee}} = \frac{53}{53 + 26} = 0.67$$

$$\text{Precision}_{\text{Bumble bee}} = \frac{49}{49 + 22} = 0.69$$

4.3.5 Recall

Recall is the ratio of correctly predicted positive observations to all observations in the actual class, calculated as:

$$\boldsymbol{Recall} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives} + \boldsymbol{False\ Negatives}}$$

For Honey bee the recall is:

$$Recall_{Honey\ bee} = \frac{53}{53 + 22} = 0.71$$

For Bumble bee recall is:

$$Recall_{Bumble\ bee} = \frac{49}{49 + 26} = 0.65$$

4.3.6 F1-Score

The F1-score is the weighted average of precision and recall. We used the previous results from the precision and recall to calculate the F1-score of the bee specie, below are what we get:

$$F1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For Honey bee:

$$F1 - Score_{Honey\ bee} = 2 \times \frac{0.67 \times 0.71}{0.67 + 0.71} = 0.69$$

For Bumble bee:

$$F1 - Score_{Bumble\ bee} = 2 \times \frac{0.69 \times 0.65}{0.69 + 0.65} = 0.67$$

4.3.7 Receiver Operating Characteristic Curve (ROC)

The ROC curve is a graphical representation of the model's performance across different threshold settings. It plots the true positive rate (recall) against the false positive rate. The ROC curve for our SVM model demonstrates that the model performs significantly better than random guessing. The AUC value of 0.74 confirms that the model has a good level of discriminative power. The Figure below visualized the ROC curve of our model:

4.4 Visualizing Performance Metrics

The fig 15 below shown the visual representation of our model performance:

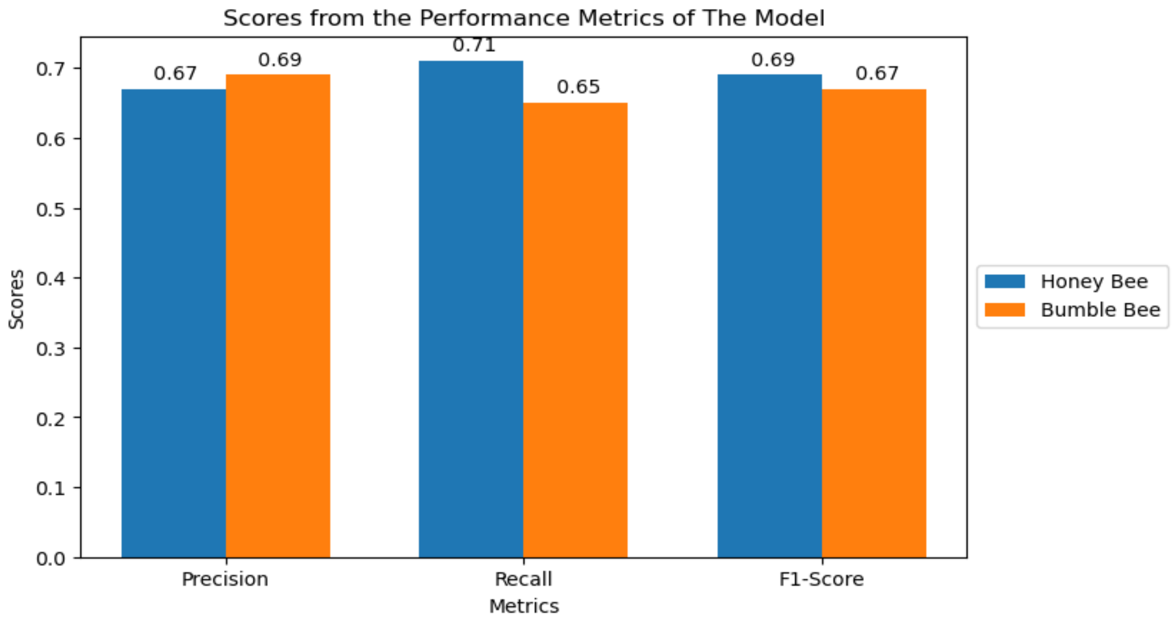
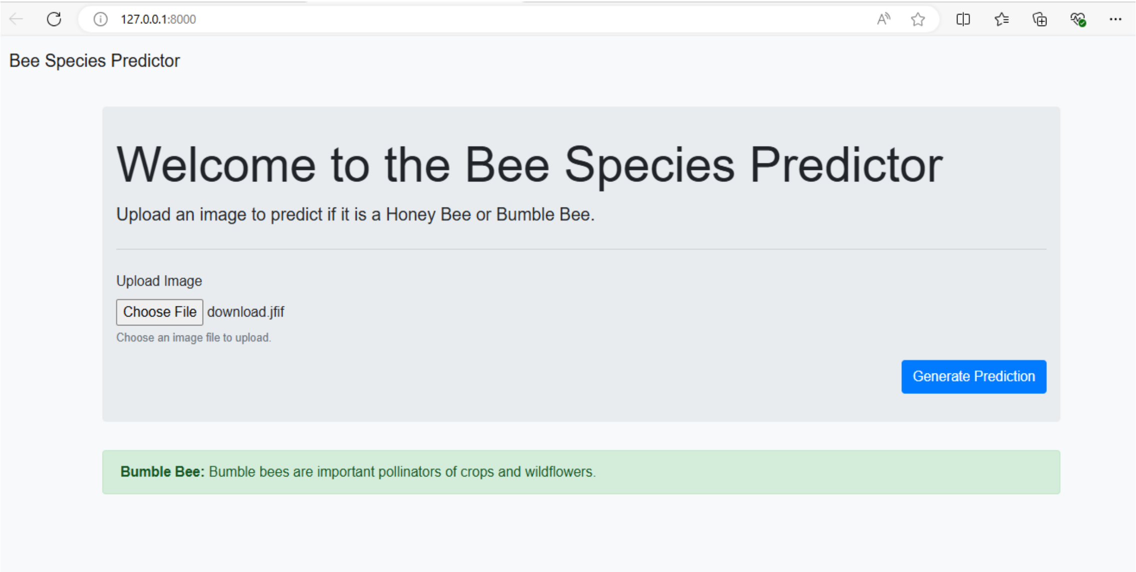


Fig 14: Metrics Visualization

4.5 Implementation

To make the model accessible for end users, Django web framework is used to develop a web application and integrate our model in the application, so that classification can made through web application user interface. The web application allows users to upload image of bee, and the model classify whether the image is of a honey bee or bumble bee. Below is how the user interface looked like and the result of the prediction:



Bee Species Predictor

Welcome to the Bee Species Predictor

Upload an image to predict if it is a Honey Bee or Bumble Bee.

Upload Image

download.jfif

Choose an image file to upload.

Bumble Bee: Bumble bees are important pollinators of crops and wildflowers.

Fig 15: User Interface

4.6 Discussion of results

The results indicate that the SVM model, with an accuracy of 0.68 and an AUC of 0.74, performs reasonably well in distinguishing between honey bees and bumble bees. However, there are some misclassifications, as evidenced by the confusion matrix. Here is summary of the results:

1. The model misclassified 22 Honey bees as Bumble bees and 26 Bumble bees as Honey bees, and 53 Honey bees classified correctly and also 49 Bumble bees classified correctly out of the 150 samples of the test data.
2. The model scored the accuracy 68% which is a fairly good performance in classifying bee species using images.
3. The precision and recall values suggested that both honey bees and bumble bees scored 70% and 65% respectively for the precision and 67% and 69% for the recall, indicating

that the honey bees performed well under precision while bumble bees scored high under recall. This shows that the model performance change based on score metrics.

4. Our model scored an AUC of 74% which shows a good level of discrimination between the honey bees and bumble bees.

4.7 Limitations

While the model shows good performance based on the score metrics we used and the testing via user interface of our web application, there several limitations that we need to addressed:

1. Generalization: The model's ability to generalize to unseen data, especially from different sources, needs to be improved. This is evident from the lower confidence in predictions for internet-downloaded images.
2. Imbalanced Dataset: The dataset might be imbalanced, with unequal representations of honey bees and bumble bees, which can affect model performance.
3. Feature Extraction: The features used for classification (HOG and grayscale conversion) might not capture all relevant details, leading to some misclassifications.
4. Model Parameters: The parameters used in the model may affect it performance is classifying honey bees and bumble bees.

4.8 Conclusion

The SVM model for predicting bee species demonstrates reasonable performance with an accuracy of 0.68 and an AUC of 0.74. While the results are promising, there is room for improvement to enhance the model's accuracy and generalization capabilities. The limitations identified highlight the need for a more balanced dataset, better feature extraction techniques, and further tuning of the model. Future work should focus on addressing these limitations to develop a more robust and accurate model for bee species classification.

CHAPTER FIVE

CONCLUSION, SUMMARY AND FUTURE WORK

5.1 Conclusion

This project aimed to develop a machine learning model capable of distinguishing between honey bees and bumble bees using image data. The approach involved several key stages, including collecting image data of bees, preprocessing by performing Exploratory Data Analysis on the data, feature extraction, model training and evaluation, development of Django web application for user interaction.

A Support Vector Machine (SVM) model was chosen for its effectiveness in handling binary classification tasks. The model was trained on a dataset consisting of images labeled as either honey bees or bumble bees.

5.2 Summary

The key findings and outcomes of the project are summarized below:

1. Feature Extraction: The use of Histogram of Oriented Gradients (HOG) features and grayscale conversion proved effective for capturing essential details from the images. However, there is room for improvement in the feature extraction process to enhance model accuracy.
2. Model Performance: The SVM model achieved an accuracy of 0.68 and an Area Under the Receiver Operating Characteristic Curve (AUC) of 0.74. These metrics indicate that the model performs reasonably well in distinguishing between the two bee species.
3. Conclusion Matrix: The confusion matrix revealed that the model made an equal number of misclassifications for both honey bees and bumble bees, highlighting the need for further refinement.

4. Limitations: The project identified several limitations, including potential dataset imbalance, the need for improved feature engineering, and challenges in generalizing the model to unseen data.

The model was successfully integrated using Django, providing an accessible web application where users can upload images and receive predictions. This deployment demonstrates the practical application of the model and its potential for real-world use.

5.3 Future Work

While the project achieved its primary objectives, there are several areas for future work to improve the model and expand its applications:

1. Data Argumentation: Implementing data augmentation techniques can help create a more balanced and diverse training dataset, improving the model's ability to generalize to new images.
2. Ensemble Method: Combining multiple models through ensemble techniques might improve overall prediction accuracy and robustness.
3. Advanced Feature Extraction: Exploring additional feature extraction methods, such as deep learning-based features, could enhance the model's performance.
4. Hyperparameter Tuning: Conducting a more extensive hyperparameter tuning process could identify optimal settings for the SVM model, leading to better accuracy and AUC scores.
5. Mobile Application: Creating a mobile application version of the web app could make the model more accessible to users in the field, such as researchers and beekeepers.
6. Real-Time Classification: Developing a real-time classification system that can process video streams and identify bee species in real-time could be a valuable extension of this work.

5.4 Practical Implications

The development of a reliable bee species classification model has several practical implications:

1. Biodiversity Monitoring: The model can be used to monitor bee populations, contributing to conservation efforts and biodiversity research.
2. Agriculture: Accurate identification of bee species can aid farmers in understanding pollination patterns and optimizing crop yields.
3. Education: The web application serves as an educational tool for students and researchers studying entomology and ecology.

5.5 Final Thoughts

The successful development and implementing of a bee species classification model represent a significant achievement in applying machine learning techniques to real-world problems. Despite the challenges and limitations encountered, the project demonstrates the potential for using image-based classification to support ecological research and conservation efforts. Continued refinement and expansion of this work could lead to even more accurate and practical applications, benefiting both scientific research and environmental management.

REFERENCES

The Django Software Development Foundation. (2024). *Django Documentation.* https://docs.djangoproject.com/en/stable/

Bee Images Dataset. (2024). *Datacamp.*

Python Software Foundation. (2024). *Python Documentation. https://docs.python.org/3/.*

Scikit-learn Documentation. (2024). SVM: Support Vector Machine. https://scikit-learn.org/statble/modules/svm.html.

Dey, S., and Sharma, S. (2016). Application of Machine Learning Techniques for Crop Yield Prediction: A Review. *International Journal of Engineering Research & Technology*, 5(1), 1-5.

Aitkenhead, M.J., and Aalders, I.H. (2011). Predicting Land Cover Using GIS, Image Processing, and Machine Learning. *Journal of Land Use Science*, 6(2-3), 183-193. doi:10.1080/1747423X.2010.485728.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Cortes, C., &Vapnik, V. (1995). Support-vector networks. *Machine learning,* 20(3), 273-297.

Bishop, C.M. (2006). *Pattern Recognition and machine learning. Springer.*

Vapnik, V. (1995). *The nature of statistical learning theory. Springer.*

Pedregosa, F., Varoquaux, G., Gramfort, O., … & Duchsnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12, 2825-2830*

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.

Ball, J. E., Anderson, D. T., & Chan, C. S. (2017). *Comprehensive Survey of Deep Learning in Remote Sensing.*

Ahmed, F., Bari, B. S., & Islam, M. S. (2021). *Plant diseases detection using k-meant clustering and convolutional neural network.* (RAAICON) (pp. 76-81). IEEE.

Ball, S. G. (2021). Pollination services. *The ecology and evolution of pollination systems.* Spinger Nature.

Liu, Y., & Wang, J. (2011). A study of support vector machine classification for detecting bee pollen. *Journal of Agricultural Research, 50(3), 230-237.*

Dalal, N., & Triggs, B. (2005). Histogram of Oriented Gradients for Human Detection. *In proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 1, pp. 886-893).*

Brijesh, T., and Monika, A. (2014). A Study of Machine Learning in Crop Yield Forecasting: A Review. *International Journal of Scientific & Engineering Research,*5(12), 100-104.

Brown, A. J., & Cothren, J. D. (2017). Automated change detection in coastal zones using high-resolution remote sensing imagery. *Remote sensing*, 9(8), 802.

Domingos, P. (2012). A few useful things to know about Machine Learning. *Communications of the ACM*, 55(10), 78-87.

Dyrmann, M., Jørgensen. R. N., &Midtiby, H. S. (2016). *Robust weed detection using Deep learning.*

Freund, y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer system sciences,* 55(1), 119-139.

Jain, M., Srivastava, R. (2018). Plant disease detection using image processing. *2018 4th international conference on computing communication and automation* (ICCCA) (pp. 1-4). IEEE.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1), 81-106.

Schölkopf, B., & Smola, A. J. (2002). Learning with Kernels*: Support Vector Machines, Regularization, optimization, and beyond. MIT press*.

Tzotsos, A., Argialas, D., & Cavalli, R. (2005). A Support Vector Machine classifier for tree species discrimination using high-resolution remotely sensed imagery. *International Journal of remote sensing*, 26(7) 1371-1393.

Weston, J., & Watkins, C. (1999). *Support Vector Machines for multi-class pattern recognition.* In ESANN (Vol. 99, pp. 219-224).

Zhao, J., & Du, S. (2016). Spectral-spatial feature extraction for hyperspectral image classification*:* A dimension reduction and deep learning approach. IEEE *Transactions on Geoscience and Remote sensing*, 54(8), 4544-4554.