University of Sheffield

Machine learning approaches for distinguishing coloured retroreflective tags on insects

Anindita Tripathy

Supervisor: Michael Smith

A report submitted in fulfilment of the requirements for the degree of MSc in Data Analytics

in the

Department of Computer Science

September 11, 2024

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Anindita Tripathy

Signature: Anindita Tripathy

Date: 11th Sept 2024

Abstract

Tracking bee flight patterns helps researchers understand their abilities to navigate and map their environment. Previous systems could only follow a few bees at a distance of up to 10 meters (1). This study builds on Smith et al.'s (2) use of retroreflective tags by incorporating colored filters to distinguish different tags. This project's approach involves extracting colors from retroreflective tag photos. Colors extracted from images can be used in classifiers to identify tags. This report's methodology successfully differentiated up to 34 out of 40 tags at a distance of 20 m with over 85% accuracy. Combining observations can increase classification accuracy even further. Implementing these approaches on real bees is expected to yield similar results, allowing for multi-bee tracking at greater distances than before. Future studies that utilise coloured retroreflective tags will benefit from the numerous fascinating and valuable impacts seen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Bees are incredible creatures whose behavior has intrigued researchers for years. Upon their initial departure from the hive, they participate in a phenomenon called orientation flight. This critical process allows bees to familiarise themselves with the hive's location in relation to the surrounding environment (3), enabling them to rely on their remarkable orientation skills to return to their hives accurately after foraging distances of up to 3 miles (4). Understanding the intricate behaviour of bees, particularly their flight patterns, plays a vital role in the advancement of research in fields such as ecology, entomology, and environmental science (5).

Factors such as the bee's diminutive size, rapid speed, and lack of easily distinguishable features make manual observation nearly impossible when monitoring their orientation flight over medium spatial scales (6). Nevertheless, traditional methods of tracking bees' flight paths pose significant challenges due to the size and weight of the electronic tags, the cost and complexity of the equipment, and the range of visual perception (7). A key limitation of many existing methods is their capacity to monitor only a single or few bees at a time, therefore significantly restricting the ability to collect comprehensive flight data simultaneously (8). There is a clear requirement for a method that can track multiple bees over longer distances using a simple, light-weight, low-cost technology with better spatial detail.

Prior studies have demonstrated that retroreflective tags are a reliable and effective method for monitoring bees to address these challenges (9). Although these tags have the potential to track individual bees as well as a few bees at the same time, the challenge lies in precisely distinguishing between multiple bees in real time over longer distances. Integrating coloured filters with the light-weight retroreflective tags enables researchers to simultaneously track multiple bees by distinguishing individual tags based on their colour while ensuring simplicity, cost-effectiveness, and precision (2).

1.2 Aim and Objectives

1.2.1 Aim:

The primary goal of this project is to develop a system capable of distinguishing between multiple bees using retro-reflective tags with colored filters. The approach must be able to differentiate among a wide range of tags and produce the best results.

1.2.2 Objective:

1. To design, implement, and test a tracking system that utilizes colored filters on retroreflective tags for distinguishing between individual bees effectively.
2. To develop a system that is capable of distinguishing between as many different tags as possible using colored filters on retroreflective tags.
3. To identify the optimal colored filters that provide the highest accuracy in tracking multiple bees simultaneously.
4. To analyze the collected data to determine the success of the tracking method and assess the feasibility of scaling this approach for broader studies in bee behavior.

These objectives are motivated by the necessity to overcome the constraints of prior methods and to offer a more efficient option for researchers.

1.3 Overview of the Upcoming Chapters

As follows is a summary of the remaining chapters. Chapter 2 provides a detailed review of the literature on current bee tracking techniques, encompassing both conventional and contemporary methodologies. It also explores the technical principles and methodologies explored in this project. Chapter 3 examines the precise specifications of the project, elucidating the theoretical basis for the use of coloured filters. Chapter 4 details the methodology used in this project, including the design, implementation, and testing procedures, by providing insight into the experimental setup and procedures followed. Chapter 5 presents the study's findings and provides a detailed analysis of the tracking system's overall performance. This chapter will assess the project's achievement of its objectives and pinpoint any areas necessitating further investigation. Chapter 6 provides an overview of the project, including detailed analysis of the results and contributions, and proposes possible future research in the field of bee tracking methods.

This research connects theoretical principles with practical applications in my degree program, bringing together expertise from both domains. This integration provides both opportunities and challenges. On the one hand, advanced tracking techniques enable a better understanding of bee behavior. However, it is necessary to strike a balance between technical innovation and practical feasibility, ensuring that the solutions developed are both effective and implementable.

Chapter 2

Literature Review

This chapter conducts a critical analysis of various previous methods used to track bees, identifies the strengths and limitations of each approach, and establishes the methodology for this study in relation to the project's objectives.

2.1 Previous Methods of Bee Tracking

Initially, bee tracking systems were predominantly manual, depending on visual observation and basic labelling (10). As technology advanced, tracking methods became more sophisticated. The evolution from manual observation to contemporary electronic and optical devices demonstrates the growing complexity and accuracy needed to investigate bee behaviour in detail. Furthermore, this development demonstrates an increasing awareness of the constraints imposed by each approach and the ongoing quest for more efficient alternatives.

Over the past 20 years, researchers have relied on harmonic radar to monitor airborne insects (11) (12) (13). The remarkable range of several hundred meters enabled the attainment of these results (14). The technique's primary drawback is that the apparatus is large, expensive, and custom-made, rendering it inaccessible to the majority of researchers, particularly those in low-income countries (15). Typically, precision is on the scale of several meters (11), which is inadequate for certain applications. Carrek et al. (16) developed an effective bee tracking system that uses harmonic radar to detect the motions of bees by attaching lightweight (12 mg) harmonic radar tags to the bees. This allows the bees to carry the tags without disrupting their flight trajectory. It allowed the radar system to detect bees up to a distance of 550 meters from the nest. Although the system proved to be efficient over extended distances, its complexity and high cost, along with the need for specialised radar equipment, rendered it unfeasible for monitoring substantial quantities of bees in various terrains. Nunes-Silva et al. (17) employed RFID tags, which required insects to be within 3 cm of the sensor. Shearwood et al. (18) used insect wing beats for energy with a similar range, but the insect must carry a heavier and more complex payload. While Bender et al. (19) tried with lidar, which can follow untagged bees, but it requires a well-prepared site

and does not allow for individual target tracking. Abdel-Raziq et al. (20) propose employing dead-reckoning to estimate flight route using an on-board sensor and sun orientation, but only using simulated data.

Researchers have employed Very High Frequency(VHF) radiotransmitters to follow bumblebees and butterflies across longer distances than harmonic radar (21) (22) using a light aircraft. Attaching a battery-powered transmitter (200 mg) to the bee can significantly impact behaviour (22), making this strategy unsuitable for many insects. Many electrical direct tracking technologies require an antenna, making them difficult to use with smaller insects or those that need to penetrate narrow areas.

Sun and Gaydecki (1) used the visual tracking method, with two 2.7k video cameras at right angles facing the hive entrance. The movies could differentiate 1.62mm details at 10m, which was adequate to detect 15mm bees. Triangulation allowed for the calculation of bee locations at 60fps. This accurately represented several bees' flying patterns over time. It could discern four flight routes at once. The problem with this method is that to measure more distance, it requires more expensive, higher-resolution cameras to achieve the same precision. Drones, with their mobility and high-resolution cameras, can collect comprehensive photos and movies for real-time bee tracking using computer vision in a wider range of areas, as implemented by Koh and Wich (23). Hardin et al.'s research (24) reveals that drones can also collect environmental data simultaneously, revealing the bees' habitat and behaviour. However, drones face limitations due to battery life and flight regulations, with the risk of disrupting bees' natural behaviours (25).

In low-light or nocturnal circumstances, thermal imaging cameras can track bees by detecting their heat. According to Dyer and Seeley, this method is beneficial for examining bee activity at night or during cooler weather when visual methods are less efficient (26). Thermal imaging can also monitor hive health and detect bee behavior changes that may signal disease or stress (27). Thermal images have lesser resolution than optical cameras, making it difficult to discern bees or capture flight trajectories (28).

In comparison to radar and visual tracking systems, retroreflective tags provide a pragmatic and economical option. Smith et al. (2) at the University of Sheffield conducted a study by affixing lightweight tags on bees and using a combination of cameras and flashes to efficiently detect and monitor bees up to a distance of 35 meters. One of this system's notable advantages is its cost effectiveness compared to other systems. The tags and glue used were virtually free, amounting to £0.02 per bee. The flashes and camera were the only significant expenses, totalling £820. This system's cost-effectiveness, combined with its limited influence on bee behaviour, makes it an excellent choice for extensive laboratory investigations. An inherent constraint of this method, however, has been the challenge of monitoring many bees concurrently because of the homogeneous appearance of the tags. The implementation of a tagging system would enable ways to operate over greater distances and accommodate the use of distinct markers on each tag, hence facilitating the use of various colours in this project.

Ibarra et al. (29) designed a technique to target high-resolution monitoring of bees within a limited region of approximately 50 mm from the hive. This approach completely

avoided the use of tags and instead relied on high-definition video to accurately analyse bee behaviour. The small geographical scope of this technique greatly constrains its usefulness, despite its excellent accuracy. The need to observe a limited region surrounding the hive limits its effectiveness in investigating bees' wider foraging patterns, which frequently span substantially longer distances.

Motion capture has shown enormous potential as a viable alternative to traditional image capture for tracking insects. This technology uses infrared light and numerous retroreflective beads to accurately determine the insect's position and body orientation. Research by Mischiati et al. (30) successfully used the tags, weighing only 7 mg, to monitor dragonflies at distances up to 4 m. However, when dealing with smaller insects, such as bees, their efficacy decreases, particularly over the long distances required to catch complete orienting flights.

Mohan et al. (31) developed bokodes, a data matrix code that uses a photo mask in front of an LED module to encode data that a camera at an infinite distance can interpret. It can store approximately 3 bytes of data and 5 bytes of correction code, in the event that some portion of the tag remains invisible. Researchers have also developed prototypes for passive tags based on retroreflectors and demonstrated that their operational range exceeds 4 meters (31). Although the vast data storage capacity of a Bokode enables the unique tagging of a significant number of bees, the ranges of 4 m are insufficient to monitor a bee's complete learning flight.

2.2 Summary and Future Directions

The technologies previously mentioned provide a diverse array of tools for the purpose of monitoring and researching bees, each with its own set of advantages and disadvantages. The current project aims to enhance retroreflective labelling by utilising coloured filters. The tagging system would enable methods to function over greater distances and would also enable the use of distinctive markers on each tag, which in the context of this undertaking would be various colours. Retroreflective tags continue to be one of the most cost-effective and scalable methods, especially when combined with coloured filters to differentiate between multiple bees at once. This method guarantees the preservation of the bees' natural behaviour, a critical factor in the conduct of precise behavioural studies, by maintaining the tags' minimal invasiveness and lightweight design.

Although this initiative concentrates on retroreflective tags, there is potential to incorporate additional technologies, such as drones for broader area coverage or machine learning for automated tracking, to improve the data collection process. The present invention holds significant promise in significantly improving the precision and effectiveness of monitoring numerous bees across long distances. This would allow for the collection of comprehensive data on their orientation flights and foraging behaviours, as well as environmental issues including habitat loss and climate change.

Chapter 3

Methodology

This chapter provides a detailed description of the requirements, analysis, design, testing, and assessment technique used to accomplish the objectives of the project. The system is sequentially developed, outlining the techniques chronologically, starting with the initial image processing and colour extraction and culminating in tag distinction and classification. To ensure the system's efficiency, the assessment criteria were established that specifically focus on classification accuracy, the ability to distinguish the maximum number of unique tags, and the system's long-distance functionality.

3.1 Dataset Overview

The dataset included in this study was gathered by Leon Joyce, under the guidance of Professor Michael Smith, at the University of Sheffield in 2023. The tags were made of 4x6mm retroreflective rectangles with a tiny peak in the middle. The retroreflectors have a 4x4mm frontal area. The retro-reflector and big colour filter weighed around 5 mg, with the filter being the majority of the weight. The 1mm peak in the middle of the tag maximises visibility from various angles, as tags may not face directly towards cameras. The 40 retroreflectors were placed at 10 cm intervals on the grid, and the selected coloured filters were placed on top of each reflector. One camera module was mounted on a tripod at 1.5 m elevation across the field. Markers were placed every 5 meters on the ground in front of the camera. Initial calibration tests ensured proper camera focus, resulting in 7-pixel blobs in the image. The camera fired every 3 seconds. The camera took two images at each interval, one with and one without flash. The effect of distance was analysed at 15, 20, and 25 m distances.

The initial dataset contains 2520 images in the form of a numpy pickle object. However, the dataset used for this project consists of 82 meticulously selected images, accompanied by required information in JSON metadata. Table 3.1 details the classification of these images based on three distinct distances: 15 m, 20 m, and 25 m. Each image resolution is 2048x1536 pixels, and it includes 40 retroreflective tags that were attached to bees in order to facilitate tracking and identification. The JSON metadata contains the tag information, such as the image file name, the exact tag location (x and y coordinates), and the unique label or identifier

for each tag. These images and tag information function as the main input for the initial data preprocessing phase, followed by the training and evaluation phases.

Table 3.1: Overview of the Dataset

Distance	No. of Images	No. of Tags per Image	Total No. of Tags
15m	7	40	280
20m	40	40	1,600
30m	35	40	1,400
Overall	**82**	**40**	**3,280**

The Figure 3.1 provides an example of how the retroreflective tags appear in the photos and how the system can process these images to identify individual tags. The x and y coordinates correspond to pixel locations, which are critical for locating each tag during the pre-processing phase. In the original raw image 3.1a the tags visibility is quite tiny as the image captures the surrounding. Figure 3.1a reflects the tag grid pattern of illuminated tags w.r.t their tag locations. The Figure 3.1c shows the blob intensity for Tag0 for the same photo object w.r.t the tag location.

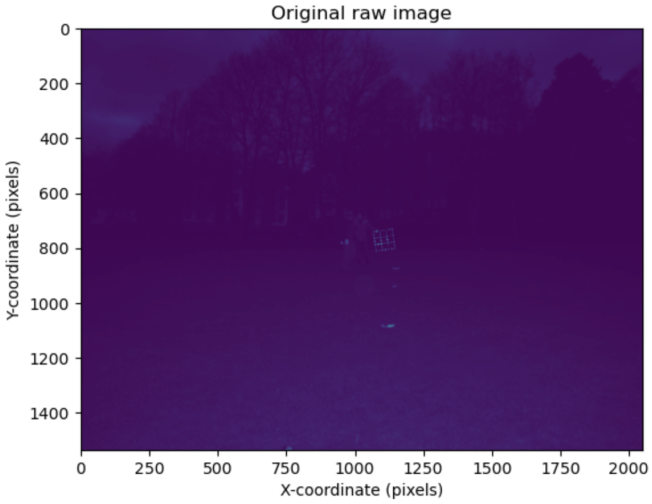
This dataset forms the basis for the design, training, and assessment stages of the bee-tracking model. Making sure that metadata is consistent and that image captures are well documented makes it easier to quickly create and test colour extraction, normalisation, and machine learning classification methods that can tell the difference between different tags. Since the data was collected internally for academic research, it isn't available in open source yet and has been sourced internally for this project work.

3.2 Data Preprocessing

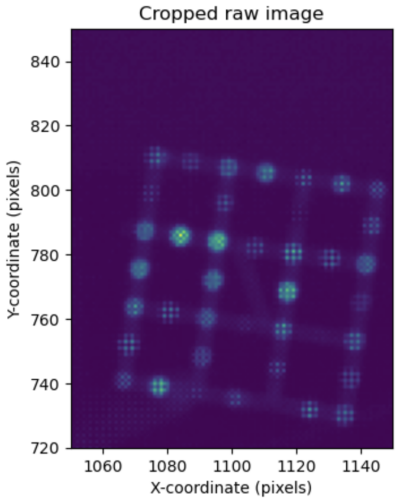
The raw image data is prepared for classification during the data preprocessing phase. This involves taking out the colours and normalising the extracted red, green, and blue (RGB) intensity proportion from the retroreflective tags. It also includes putting together a clean dataset in the form of a CSV file so that the classification model can be developed further. The requirement of this preprocessing phase is to develop an algorithm that can process raw images containing retroreflective tags. These tags reflect light emitted by a flash and must be distinguishable by their unique color. The algorithm should also handle background noise and ensure that the RGB values represent the true color of each tag. The steps implemented are documented in a chronological order as below.

3.2.1 Mapping Photo Objects with JSON Metadata

The JSON metadata includes the file name and x and y coordinates of each tag in the photo object, as well as a dedicated tag label. The metadata files played an important role in accurately identifying the tag locations within the photos (2). An automated Python script



(a) Original Raw Image belonging to 20m distance group w.r.t. pixel coordinates.



(b) Cropped Raw image belonging to 20m distance group w.r.t. pixel coordinates.

(c) Cropped Raw image for Tag0 belonging to 20m distance group w.r.t. pixel coordinates.

Figure 3.1: Various raw images from the 20m distance group: (a) Original raw image, (b) Cropped raw image, and (c) Cropped raw image for Tag0.

was employed to associate the photo objects with their respective metadata. This metadata was then used to crop images and extract RGB values from each tag.

3.2.2 Image Cropping

The image was divided into 8x8 pixel sections, centered on the x and y coordinates of each tag in the photo object. This reduced image region was crucial for achieving precise focus on

each individual tag, free from any interference caused by adjacent tags. In order to maintain consistency, the Bayer pattern (RGGB pattern) was essential at this stage to ensure that each pixel of the cropped image was divided into the appropriate RGB channels, starting with a red channel.

3.2.3 Splitting the Bayer Channel

The image is divided into four channels by the Bayer pattern of the camera sensor: red, green, green duplicated, and blue, which is known as the RGGB pattern. The intensity of each color in the 8x8 cropped image was determined by extracting RGB channels using the Bayer pattern separation technique. The Bayer Pattern and the Point Spread Function (PSF) were indispensable technical components during the preprocessing phase. The Bayer filter is capable of capturing red, green, and blue information; however, it is required to interpolate the absent data. The point spread function guaranteed precision by monitoring the manner in which light dispersed from the retroreflective marks in the images.

Bayer Pattern:

A camera sensor is limited to detecting the intensity of light that reaches a specific location and cannot independently determine the color (32). A camera detects color using a Bayer filter, which is a grid of colored filters positioned above the sensor. This results in each pixel being able to capture only red, green, or blue light, as the filter blocks the preponderance of the other colours (33). After being split into three channels, the data will have significant pauses for each color. Raw image data that is formatted in a Bayer pattern is typically subjected to a decoding algorithm, which interpolates the values of the absent data from the surrounding pixels (34).

Point Spread Function (PSF):

Retro-reflective tags emit light that passes through the camera lens and contacts the sensor, forming a point-spread function. The image features a brilliant central area surrounded by dimmer concentric circles, known as the airy disc. In microscopy and astronomy, the point spread function is commonly used to reduce blur in images of specimens or planets. Measure the smaller concentric rings around a point as focus changes to build a function that can unwind across out-of-focus images to eliminate blur. This technique excels in low-noise, high-resolution applications. The noise and low resolution of the photos make it difficult to see or measure the impact of the concentric rings in this project. The point spread function is useful because it approximates a Gaussian shape with tiny shoulders at the first Airy rings. This explains the Gaussian form of the collected data.

3.2.4 Calculating Mean RGB Intensities

Although there were other methods available for calculating the total intensity of each channel, the decision was made to initially adopt the straightforward averaging technique.

For each channel, the mean intensities of each pixel in each of the cropped images were determined. These intensities indicate the proportion of the total quantity of red, green, and blue light that was reflected from the tags and captured by the camera. The use of harmonic radar tracking by Carrek et al. (16) established a foundation for comprehending the interaction between reflecting surfaces, such as tags, and light, as well as methodologies for capturing and analysing their intensity values.

3.2.5 Normalising the Intensities

Each image showed differences in the total amount of light reflected by the retroreflective tags. These differences are mostly caused by the tag's distance and angle from the camera. To correct for these inconsistencies, the RGB intensities had to be normalised so that the red, green, and blue channels were represented as percentages of the total reflected light. This method was used to account for differences in light conditions and tag distances, allowing for a consistent comparison of RGB values across all tags, regardless of their position within the image. The normalised RGB values were obtained using the Equation (3.1).

$$R_{\text{norm}} = \frac{R}{R+G+B} \times 255, \quad G_{\text{norm}} = \frac{G}{R+G+B} \times 255, \quad B_{\text{norm}} = \frac{B}{R+G+B} \times 255 \tag{3.1}$$

This normalisation ensures that the combined intensity of each tag's RGB values equals 255, adjusting for the effects of the Inverse Square Law. This correction allows for reliable and constant tag comparison, regardless of distance or lighting conditions, and serves as a solid foundation for the subsequent classification procedure.

Inverse Square Law

The Inverse Square Law states that 'The intensity of light diminishes as the square of the distance from the light source.' which can be referred from Equation (3.2).

$$I \propto \frac{1}{d^2} \tag{3.2}$$

Within the scope of this study, the light intensity registered by the camera exhibits exponential reduction as the distance between the camera and the tags grows. If this is not taken into account, tags positioned further away from the camera will exhibit significantly reduced brightness compared to those closer, even if their reflective characteristics are the same.

3.2.6 Storing the Data into CSV File

After processing, the normalized RGB values for each tag were saved in a CSV file for further study. This CSV file contained the photo object's filename, tag label, and the normalised R, G, and B values. The preprocessed data was stored separately for training and evaluation for each distance group, as well as for the overall data.

3.3 Training Classification Models

The purpose is to build and train machine learning models that can precisely categorise retroreflective tags by analysing their normalised RGB intensities obtained during the data preprocessing stage. This stage entails importing the preprocessed dataset for model training, dividing the dataset into training and testing sets, choosing appropriate classification algorithms, and optimising hyperparameters for the target dataset. The techniques employed in this study integrate machine learning methodology, including train-test splits, cross-validation, and grid search, to guarantee both resilience and precision in the classification task. The subsequent steps were followed for each distance group, as well as for the overall data.

3.3.1 Loading and Preparing the Preprocessed Dataset

The preprocessed dataset, containing normalised RGB values of retroreflective tags stored in a CSV file, was loaded into the Python environment using the **pandas** library. The normalised RGB values are the input features, and the matching tag labels are the classification target variables. The dataset was retrieved from the CSV file using the **pd.read_csv()** method. After loading the dataset, the three RGB intensity columns (R, G, and B) were recognised as feature columns. These characteristics reflect the normalised colour intensity values for each tag and will be used as input into classification models.

In multi-class classification tasks, it is critical to keep the dataset balanced, which means that each class or tag label has an equal amount of samples. If some classes are overrepresented, the model may become biased toward them, resulting in poor classification results for underrepresented classes. To overcome this, the dataset was balanced using downsampling. Data was balanced using the **balance_data()** method to reduce over-represented class samples to the same number as the least represented class. This technique ensured that the dataset was balanced and the classification model was not biased towards a specific class. The balanced dataset was then used in the next training and evaluation phases. By balancing the classes, the model was able to generalise better across different tags, resulting in higher classification accuracy for all classes. In Smith et al.'s (2) bee tracking study, preprocessed image data with tag positions was also employed to feed classification algorithms. The systematic handling of preprocessed data guarantees that the learning algorithms are fed consistent and clean inputs.

3.3.2 Train-Test Split

To effectively analyse the model's performance, the dataset was divided into training and test sets using a stratified split. Stratified sampling guarantees that each tag class is proportionally represented in both the training and test sets. A conventional 80/20 split was employed, with 80% of the data for training and 20% for testing. The random state was set to 42 to ensure reproducibility. This method ensures that the model is trained on a representative subset of the data before being evaluated on a separate, unseen test set, resulting in a reliable

evaluation of its generalisation ability. Sun et al. (1) used similar strategies to divide data into training and test sets while tracking bees using video analysis. The stratified train-test split was very important for keeping the class distribution across many subsets of data and preventing bias in the accuracy of classification. The train-test split for different distance groups are as given in Table 3.2.

Table 3.2: Training and Test Splits for Different Distances

Distance	**Train Split**	**Test Split**
15m	5	2
20m	32	8
25m	28	7
Overall	65	17

3.3.3 Feature Scaling

As the features (normalised RGB values) are on different scales, it is critical to standardise them before training models. The features were scaled using **StandardScaler**, which ensured that each feature had a mean of zero and a standard deviation of one. Feature scaling is especially crucial for distance-based models such as K-Nearest Neighbours (KNN), which use the distance between data points to classify. The use of StandardScaler is consistent with machine learning best practices, as illustrated by Ibarra et al. (29), who used feature scaling in insect tracking systems. Without scaling, RGB values could have a disproportionate impact on the distance metric used by KNN and other algorithms.

3.3.4 Classification Model Selection

Two classification methods were chosen and built for the task of distinguishing between numerous tags: K-Nearest Neighbours (KNN) and Random Forest Classifier. These models were chosen because they are simple, successful in multi-class classification, and robust when dealing with noisy data.

K-Nearest Neighbour (KNN)

K-Nearest Neighbour (KNN) is a straight-forward, non-parametric method that classifies data points according to their nearest neighbours' majority class. The simplicity and effectiveness of KNN make it an excellent method for working with low-dimensional data, particularly in cases where features are clearly distinguishable, such as the normalised RGB values used in this project. The k parameter, which represents the number of neighbours, was fine-tuned during the model selection process to attain the best performance. Carrek et al. (16) used KNN in the harmonic radar-based bee tracking system. The algorithm's ability to categorise image-derived data, such as RGB intensities, makes it ideal for this purpose.

Random Forest

The Random Forest Classifier is an ensemble learning method that creates numerous decision trees during training and outputs the majority of class predictions. It is resistant to overfitting and can handle both high- and low-dimensional data effectively, making it appropriate for noisy datasets. The Random Forest model works well in this project because of its ability to generalise across diverse lighting situations and tiny fluctuations in RGB values, which are inherent in the data. Its ensemble nature increases model stability and enables effective noise handling. Smith et al. (2) demonstrated Random Forest's ability in managing huge and noisy datasets in tracking systems. The ensemble method was ideal for assuring model stability, even with minor fluctuations in RGB values due to lighting or tag distance from the camera.

Support Vector Machines (SVM) are commonly used for high-dimensional data and excel at separating non-linearly separable data; however, the normalised RGB values in this project are low-dimensional, making SVM less suitable. Neural Networks (NN), while highly effective for complex, high-dimensional datasets, are computationally expensive and require large amounts of data to avoid overfitting. Given the relatively small dataset in this project, NN would introduce unnecessary complexity and resource demands. Similarly, Naïve Bayes assumes that features are independent, but since the RGB values are correlated, Naïve Bayes would be inappropriate, as it would fail to account for the dependencies between colour intensities. As a result, these models were excluded because they are either too complex or make assumptions that do not fit the nature of the dataset.

3.3.5 Cross-Validation

To guarantee that the both the models performed optimally, a 5-fold cross-validation technique was used during training. Cross-validation subdivides the training data into five parts or folds. For each fold, the model is trained on four subsets and tested on the remaining one. This process is done for each of the five subsets, and the model's accuracy is estimated using the average performance across all folds. Cross-validation reduces overfitting and yields a more trustworthy measure of model performance. The use of cross-validation adheres to best practices established by Sun et al. (1) and Mohan et al. (31), where cross-validation was used in systems to evaluate models and assure generalisability across diverse datasets.

3.3.6 Hyper-Parameter Tuning

To improve the classifiers' performance, hyperparameter tweaking was done using GridSearchCV, a method that exhaustively examines a set of hyperparameters and picks the best model based on cross-validation accuracy. The tuning technique was used for both the Random Forest Classifier and the K-Nearest Neighbours (KNN) algorithm.

KNN:

The KNN classifier's hyperparameters were tested as follows:

- **n_neighbors**: The number of neighbors to consider when voting, with values ranging from 3 to 11.
- **Weights**: The technique of weighing neighbors' votes, with options ['uniform', 'distance'].
- **Metric**: The distance metric to use, with options ['Euclidean', 'Manhattan'].

Random Forest:

The Random Forest classifier was customised by specifying a set of hyperparameters to evaluate. The following parameters were examined:

- **n_estimators**: The forest's tree count, represented by the values 100, 200, and 300.
- **max_depth**: Each tree's maximum depth, with values of 5, 10, and 15.
- **min_samples_split**: The minimum number of samples needed to split an internal node, with values of [10, 20, 30].
- **min_samples_leaf**: The minimum number of samples required to reach a leaf node, with values [5, 10, 15].
- **max_features**: The number of features to examine when determining the optimal split, with values ['sqrt', 'log2'].

3.3.7 Training Classification Models

After determining the optimum hyperparameters using grid search, both the classifiers were trained on the entire training dataset for each of the distance groups as well the overall distance. The input features were the scaled and normalised RGB values, while the target variable for classification was the tag labels. The training technique followed Mischiati et al.'s (30) multi-class classification approach to distinguish between insect flight trajectories. Normalised RGB intensities ensure that the classifiers can reliably identify between the tags, much like they were used to distinguish between distinct tracking locations in insect research.

3.4 Testing & Evaluation

The testing and evaluation phase is critical for determining the effectiveness and generalisability of the classification models created during this project. This phase entails testing the trained classifiers on test data, using appropriate metrics to determine their performance in correctly classifying retroreflective tags. The testing procedure was developed to verify that the models are strong, accurate, and dependable for use in multi-bee tracking.

3.4.1 Model Evaluation on Test Data

After training the classifiers with the best hyperparameters, the models were assessed on the test dataset, which made up 20% of the total data reserved during the train-test split. The test set allows for an unbiased evaluation of the model's performance on previously unknown data by mimicking real-world circumstances in which the model encounters new samples. The evaluation was performed for each of the classification models for each distance group as well as the overall distance to analyse and compare the results obtained.

3.4.2 Handling Potential Overfitting

Overfitting arises when a model incorporates both known patterns and noise from the training data, resulting in inadequate ability to generalise to new data. To prevent this issue, the following strategies were implemented:

Regularisation for K-Nearest Neighbours (KNN):

- **n_neighbors**: Adjusting the number of neighbours ensured that the model struck a balance between sensitivity to the data and generalisation.
- **Weights**: Distance weighting was used to prioritise closer, more relevant neighbours, thereby reducing overfitting.
- **Metric**: The selection of a distance metric, such as Euclidean or Manhattan, ensures the accurate calculation of distances between points.

Regularisation for Random Forest:

- **max_depth**: Limiting tree depth kept overly complex trees from fitting noise in the data.
- **min_samples_split** and **min_samples_leaf**: These options ensured that splits and leaf nodes captured general trends rather than overly specific patterns.
- **max_features**: Limiting the number of features utilised in splits promoted model variety while reducing overfitting.

Cross-Validation and Validation Curves

Cross-validation was used to make sure the model worked well with a lot of different sets of data, and validation curves helped keep the model's complexity in check by showing how it was overfitting or underfitting across different hyperparameters.

Overfitting prevention methods, such as regularisation and cross-validation, have been extensively reported, including in Mohan et al. (2009). These methods contributed to the success of this project's models in generalising to previously unexplored data.

3.4.3 Evaluation Metrics

The classification models' performance was evaluated using a variety of indicators that reveal how effectively they generalize to new data and handle multi-class classification problems. The metrics employed in this project were as follows:

Accuracy:

Accuracy is the fraction of successfully classified tags among the total number of predictions and can be represented by the Equation (3.3).

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{3.3}$$

Although, accuracy is a generic measure of model performance and may not fully capture performance if the dataset has class imbalances, it is helpful in this project since it is previously ensured that the data is balanced.

Precision:

Precision is defined as the ratio of accurate positive predictions to the total number of positive predictions which follows the Equation (3.4).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.4}$$

Achieving precision is crucial in this project to reduce false positives, especially when accurately detecting specific tags is vital, as incorrect predictions could impact the accuracy of tag-based tracking.

Recall (Sensitivity):

Recall, also known as sensitivity, quantifies the ratio of real positives to the overall number of observed positives, including both true positives and false negatives. Refer Equation (3.5)

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.5}$$

When the objective is to accurately identify as many tags as feasible, recall is of utmost importance. This project serves to ensure that the models effectively capture all occurrences of each tag, minimizing the probability of any tags being overlooked during the classification process. Recall, also known as sensitivity, quantifies the ratio of real positives to the overall number of observed positives, including both true positives and false negatives.

F1 Score:

The F1-Score is the harmonic mean of precision and recall, offering a balanced statistic for minimising both false positives and false negatives as shown in Equation (3.6)

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.6}$$

In this case, the F1-score is helpful because it ensures that the models are not only producing accurate classifications but also minimizing mistakes related to missed or misclassified tags by balancing accuracy and recall. This is more critical because of the dataset's multi-class nature.

Confusion Matrix

The confusion matrix breaks down true positives, true negatives, false positives, and false negatives for each class. It enables the project to determine which tags are misclassifying the models, providing insight into where improvements may be required. Analysing the confusion matrix reveals which tags are commonly mistaken with others, allowing the models to be refined further. It also provides information on how many tags were succesfully classified by both the models for each distance group.

To fully understand the model's performance in this project, these evaluation measures were used. Since the task requires identifying numerous tags based on RGB values, the models must generate accurate predictions, minimise misclassifications, and capture all instances of each tag. Precision, recall, and the F1-score reveal how successfully the models balance erroneous positives and false negatives, ensuring that all tags are correctly detected without overfitting to specific patterns. The confusion matrix helps identify categorisation issues, improving models repeatedly. Similar tracking systems, such as Sun et al. (1) and Mohan et al. (31), used these multi-class categorisation criteria. This provides a complete performance evaluation, validating model robustness in real applications.

Chapter 4

Results & Discussion

4.1 Data Preprocessing

When it came to preparing the raw images and extracted characteristics for the purpose of model training, data preprocessing was an extremely important step. Imagery of retroreflective tags was the primary component of the collection. These photos were acquired at three distinct focus distances: 15 meters, 20 meters, and 25 meters. The pictures were processed in order to extract the features that were pertinent, specifically the intensities of the RGB channels. As a result of this preprocessing, the models were provided with clean and normalised data, which allowed for accurate classification.

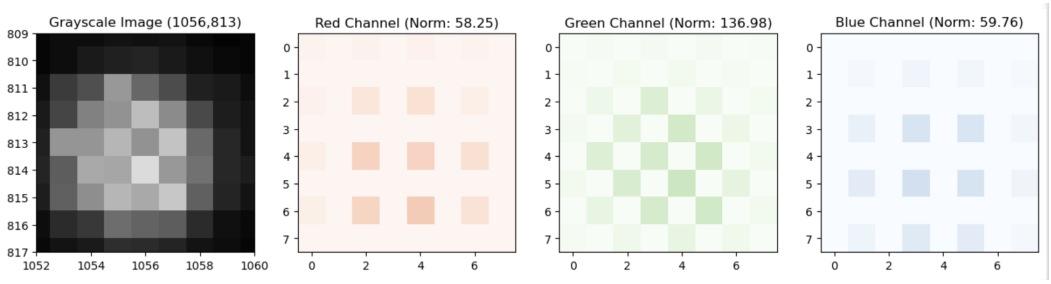


Figure 4.1: Greyscale, Red, Blue and Green Channel Intensities of a 8x8 cropped photo image of 20m distance.

The greyscale image that was captured for one photo object at a focus distance of 20 m, as depicted in 4.1, demonstrates how the image is processed by cropping a small region that is 8 by 8 pixels in size and is centred around the tag. It is possible to visualise the overall brightness distribution of the tag using the greyscale representation. On the other hand, the intensities of the RGB channels—red, green, and blue—are the most important factors in discriminating between the various tags. It also displays the normalized RGB channel intensities for the image that has been cropped to 8 meters by 8 meters. The normalized intensity of the red channel is 49.04, the green channel is 116.44, and the blue channel is 89.51. Each tag's variances in light reflection are essential characteristics that are used for

classification, and these unique values bring attention to those differences. Pixel intensities were normalised during preprocessing to make RGB values comparable across images. This normalisation helps to reduce variations caused by varying lighting conditions or tiny shifts in the location of tags.

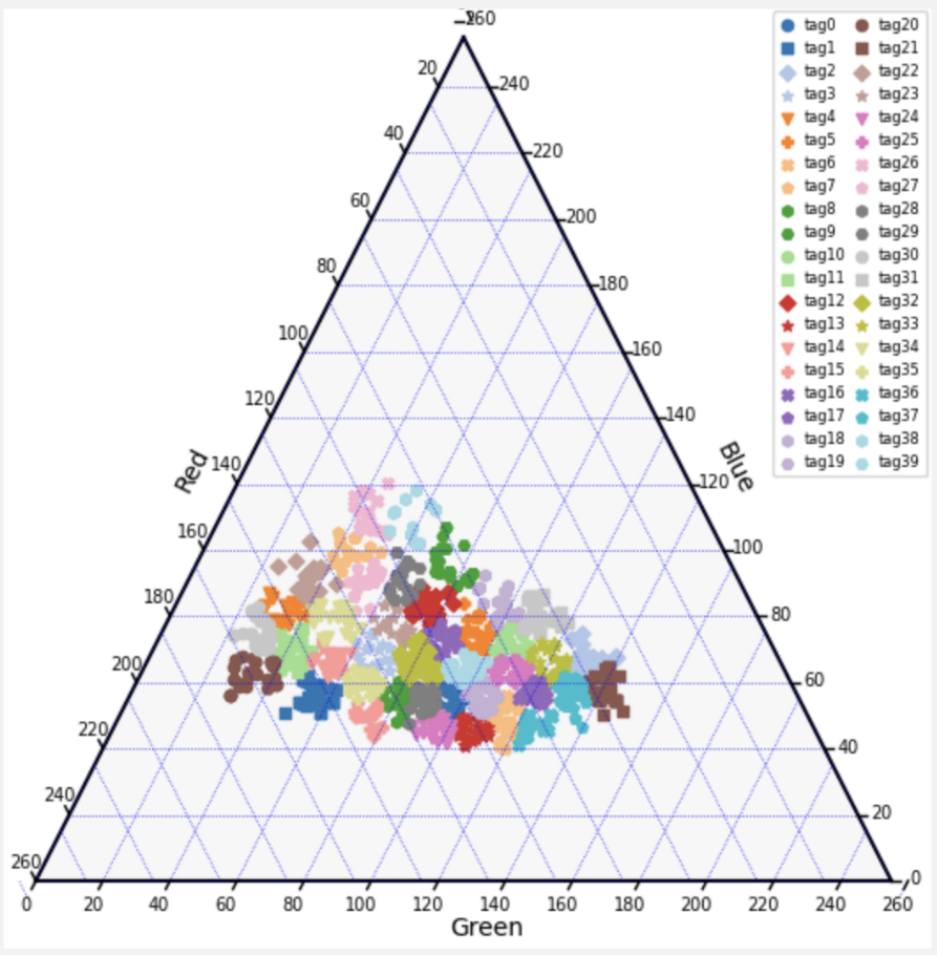


Figure 4.2: Ternary plot in the RGB triangle for the preprocessed 25m data.

The ternary plot shown in 4.2 illustrates the RGB value distribution for the tags located 25 meters away. The position of each tag on the plot is controlled by the relative contributions of the red, green, and blue channels. Each point on the plot corresponds to a tag. The layout, which is triangular, offers a clear visual representation of the way in which the RGB intensities differ between tags. For the purpose of ensuring that the RGB data were retrieved and normalised in an exact manner across all focus distances, preprocessing was an enormous contributor. While the RGB values offered sufficient variance to differentiate between the tags, even at a longer distance of 25 meters, the clustering of tags in distinct parts of the ternary plot demonstrates that this variance was sufficient. The various colour markers can be thought of as representing the individual tags, with each tag holding a distinct place

according to the RGB intensities it possesses.

The first thing that was done in the preprocessing stage was to crop the raw photos. To create a region that is centered around the tag, each image was cropped into an 8x8 pixel region. Through cropping, it was made certain that only the pixels that directly represented the tag were analysed. This was accomplished by deleting any unnecessary components of the image, such as background noise or sections that were not important. This narrowed the analysis's focus to the primary region of interest, reducing noise and increasing the precision of subsequent feature extraction. In addition, uniform input data was provided by cropping all of the images in the same way. This is an essential component for training machine learning algorithms that are dependent on structured input.

Following the cropping of the photos, the next step involved channel separation and intensity separation. In order to extract the distinct light intensities that were reflected by each tag, the RGB channels, which can be broken down into red, green, and blue, were separated. This step was crucial because each tag reflects a different quantity of red, green, and blue light, which are critical characteristics for distinguishing between tags. Instead of focusing on a composite image, the machine learning model was able to concentrate on these individual colour characteristics thanks to channel separation capabilities. When the intensities were retrieved, they gave a distinct collection of characteristics for each tag. These qualities created the core features that were utilised for classification. It would have been difficult to differentiate between the tags if the RGB channels had not been separated, particularly considering the small changes in the way that each tag reflected light.

Following the phase of intensity extraction, the RGB data were next subjected to normalisation, as demonstrated in 4.1. It was required to do this in order to guarantee that the RGB intensities were equivalent across a variety of photos, which may have been captured under a variety of lighting circumstances or at a variety of distances from the camera. The values were corrected through the process of normalization so that the overall intensity of each tag's RGB channels would add up to a standard value. This ensured that the values were consistent throughout the dataset. This phase was especially critical for minimizing the brightness variations caused by the amount of distance that separated the camera from the tag or by differences in illumination between different photographs. As a result of scaling the intensities to a common range, the models were able to concentrate on the relative variations in colour rather than being misled by disparities in brightness. Through the use of this normalisation, the classification models were able to function with standardised and trustworthy data, which ultimately resulted in more accurate predictions.

4.2 Model Evaluation

The results showed that both the K-nearest neighbour (KNN) and Random Forest (RF) classifiers could correctly identify retroreflective tags based on their RGB intensities. A total of three distances—15 meters, 20 meters, and 25 meters—as well as the entire dataset were used to evaluate each model individually. The performance of the model was evaluated using

key evaluation metrics, which include accuracy, precision, recall, and F1 score. The specifics of this evaluation are presented in Table 4.1.

Table 4.1: Performance metrics of K-nearest neighbor and Random Forest for different focus groups.

Model	Distance	Accuracy	Precision	Recall	F1 Score
K-nearest neighbor (KNN)	15m	**0.925**	0.932	0.928	0.923
	20m	0.919	0.927	0.914	0.922
	25m	0.902	0.913	0.911	0.918
	Overall	0.874	0.893	0.873	0.873
Random Forest (RF)	15m	**0.912**	0.929	0.924	0.916
	20m	0.894	0.904	0.911	0.908
	25m	0.873	0.895	0.891	0.884
	Overall	0.849	0.853	0.862	0.859

4.2.1 KNN Classifier:

As can be shown in Table 4.1, the KNN classifier consistently demonstrated a high level of performance across all distance types. KNN achieved the maximum accuracy at the 15-meter distance, where it achieved 0.925, with a precision of 0.932, recall of 0.928, and an F1 score of 0.923. This represents the highest accuracy that it has ever attained. In light of these findings, it can be concluded that the model successfully classified the tags, hence reducing the number of false positives and false negatives. As was highlighted in the literature review, this strong performance is consistent with earlier research that has praised KNN for its effectiveness in low-dimensional data categorisation. For example, normalised RGB intensities are one example of the type of data that falls under this category. Given that the RGB values in this dataset were distinct without being unduly complicated, the KNN algorithm was a good choice because of its simplicity.

A solid performance was maintained by KNN at the 20-meter distance, with an accuracy of 0.919. This represents only a minor decrease from the values obtained at the 15-meter distance. In addition, the precision (0.927) and recall (0.914) remained high, which indicates that the model was able to manage the greater distance effectively. This was the case, even though the reflective tags became more difficult to discern as the distance increased. This is consistent with the ideas reflected in Methodology section where it discussed that the normalisation of RGB intensities helped reduce the influence of noise and ensure consistency in classification even at wider distances.

With an accuracy of 0.902, the KNN model showed an additional modest decline in performance when it was applied to the 25-meter distance. The F1 score stayed at 0.918, showing that the model continued to perform well at extended distances, despite the fact that there was a slight decrease in both precision (0.913) and recall (0.911). This further supports the usefulness of the KNN algorithm in keeping a good balance between precision

and recall, as the methodology had predicted. This is especially true after normalisation and feature extraction.

KNN's overall performance across all distances reached an accuracy of 0.645, which was lower than its individual distance-specific accuracies. This was the case regardless of the distance. The fact that this is the case implies that although KNN is quite good at categorising tags at specified distances, integrating datasets from multiple ranges may bring additional complications or noise that test the performance of the model.

4.2.2 Random Forest Classifier:

In addition, the Random Forest classifier performed admirably, but it fell slightly short of the KNN classifier across all distances assessed. The RF model came in with an accuracy of 0.912, a precision of 0.929, a recall of 0.924, and an F1 score of 0.916 when it was tested at a distance of 15 meters as given in Table 4.1. However, the data indicate that RF was significantly less resilient in terms of minimising false positives in comparison to KNN. This despite KNN-like performance in this range. This is most likely due to the fact that RF relies on decision trees, which have the potential to get overfitted to particular patterns, resulting in slightly increased categorical variability. The technical review section of the methodology addresses this issue.

When the RF classifier was placed at a distance of 20 meters, its accuracy decreased to 0.894, while its precision was 0.904 and its recall was 0.911. Despite the fact that the F1 score remained high (0.908), it is evident that RF had a more difficult time than KNN when it came to greater distances. This is most likely due to the increased noise and decreased intensity produced by the reflecting tags. The results are consistent with other research findings that are comparable to this one, such as those addressed in Smith et al. (2020), in which it was demonstrated that RF is sensitive to noise in datasets that involve many different types of complex characteristics.

With an accuracy of 0.873, precision of 0.895, and recall of 0.891, RF's performance continued to deteriorate to the point where it was measured at a distance of 25 meters. With a score of 0.884 on the F1 test, it appears that RF was still successful, although it was less trustworthy than KNN when applied over longer distances. The decrease in performance demonstrates that RF is more sensitive to problems that are connected to distance, which is consistent with the idea that Random Forest might not generalise as well as KNN in this particular classification job.

The RF model's accuracy dropped to 0.761 over the entire dataset, which is a significant decrease compared to its performance in terms of distance-specific accuracy. This suggests that the model, despite being effective at short distances, had difficulty maintaining its robustness over combined data from varied ranges. This is a reflection of the possible limitations of RF when dealing with aggregated datasets.

4.2.3 Comparative Analysis

Both versions demonstrated excellent performance at 15 meters and 20 meters, with only a little variation in how accurately and precisely they performed. The KNN model, on the other hand, outperformed the Random Forest model both at 25 meters and across the entire dataset. This demonstrates that the KNN model is more resilient to the difficulties caused by increased distance. The KNN algorithm, which is based on distance-based classification, proved to be more effective in managing the RGB characteristics, particularly after normalisation. This was in line with what was anticipated from the section on methodology. Both the results and the technical factors that were described in the literature review are in agreement with one another. In the literature review, the usefulness of KNN in low-dimensional datasets that are comparable was highlighted.

In spite of the fact that it is effective, the Random Forest model exhibited a stronger sensitivity to increased distances and a greater tendency towards performance degradation when dealing with mixed datasets. This is consistent with the findings of earlier research, such as that conducted by Mohan et al. (2009), which showed that RF has difficulty dealing with datasets that exhibit increased noise or variance over greater ranges.

Although both KNN and RF are effective classifiers, KNN regularly beat Random Forest, particularly at greater distances, and displayed more stable overall performance. Random Forest was also seen to perform less consistently overall. For applications that require high precision and dependability across a range of distances, the KNN model emerged as the more ideal alternative for classifying retroreflective tags based on RGB intensities. This is especially true for applications that require the information to be accurate.

4.3 Hyper-Parameter Tuning & Its Impact

In order to tune the hyperparameters for both the K-nearest neighbour (KNN) and Random Forest (RF) classifiers, GridSearchCV with 5-fold cross-validation was used. This procedure assisted in determining the ideal hyperparameters for each distance as well as for the entire dataset, as demonstrated by the final model performance that was discussed before.

The selection of these hyperparameters was an extremely important factor in improving the accuracy, precision, recall, and F1 score of both models. The distance measure selection, the number of neighbors for KNN, the depth of trees, and the number of estimators for Random Forest all had a significant impact on the accuracy with which the models classified the retroreflective tags. Tables **??** and **??** provide a concise summary of the outcomes that were encountered during the tuning process.

4.3.1 KNN Hyper-Parameters

According to the data presented in Table 4.2, the optimal performance at a distance of 15 meters was accomplished by utilising the Manhattan distance measure and adjusting the weighting based on the distance. Because of this selection, the model was able to give

greater priority to neighbours that were closer to it. This resulted in an improvement in accuracy since it prevented the model from incorrectly classifying tags based on neighbours that were further away and perhaps louder. At tighter ranges, where even tiny fluctuations in RGB intensities could have led to skewed Euclidean computations, the Manhattan distance, which estimates the absolute differences between points, most certainly performed better than the Euclidean distance. The optimal setup for the model consisted of 11 neighbours with Manhattan distance and distance weighting at both 20 meters and 25 meters between them. As the distance between the camera and the tags increased, the model required a greater number of data points in order to generate appropriate predictions. This is indicated by the fact that the number of camera neighbors increased. This reflects the challenges posed by tags that are separated from each other. Noise and weaker signal reflections could have resulted in the need for a greater number of neighbors in order to guarantee accurate classification. Because it is more resistant to fluctuations in individual RGB channels, the Manhattan distance continues to perform better than the Euclidean distance. With the help of 11 neighbors and distance weighting, the model was able to obtain the best possible performance for the entire dataset by utilizing Euclidean distance, which is a method that estimates the distance between points in a straight line. Perhaps the larger data variance prompted the decision to use Euclidean distance for the entire dataset. The Euclidean measure, which is more straightforward, may have been more suited for balancing the impact of different distances on the classification process.

Distance	**Metric**	**n_neighbors**	**Weights**
15m	Manhattan	9	Distance
20m	Manhattan	11	Distance
25m	Manhattan	11	Distance
Overall	Euclidean	11	Distance

Table 4.2: Best KNN Hyperparameters for Different Distances

4.3.2 Random Forest Hyper-Parameters

Detailed information regarding the selected hyperparameters for each distance is provided in Table 4.3, which pertains to the Random Forest classifier. At a distance of 15 meters, the model performed best with 100 trees, a maximum tree depth of 15, and a minimum sample split of 10. This helped control overfitting by preventing trees from getting too deep. The use of sqrt as the maximum feature option (that is, selecting characteristics to examine for each split in a random fashion) meant that the trees were diversified, which improved generalization and assisted the model in avoiding placing an excessive amount of emphasis on certain features, such as a single RGB channel. The number of trees was increased to 300 at 20 and 25 meters, while the maximum depth was reduced to ten. This resulted in a more generalised model that was better adapted to deal with the increased noise and volatility in the data. Furthermore, this allowed the model to avoid overfitting to certain distances

and assisted in maintaining a high degree of precision, even as the distance between the tags and the camera increased. Once again, the best performance was attained with 100 trees and a maximum depth of 15, which was applied to the entire dataset. The use of these hyperparameters ensured that the model was able to strike a balance between collecting the overall patterns in the data and managing the noise and variation introduced by the tags at different distances.

Distance	n_estimators	max_depth	min_samples_split	min_samples_leaf	max_features
15m	100	15	10	5	sqrt
20m	100	10	10	5	sqrt
25m	300	10	10	5	sqrt
Overall	100	15	10	5	sqrt

Table 4.3: Best Random Forest Hyperparameters for Different Distances

4.4 Assessment of Number of Successful Tags

The confusion matrices provide a useful visual representation of how well KNN and Random Forest (RF) classifiers perform across different distances and the entire dataset. At 20 meters, the KNN classifier works well, properly categorising 34 of 40 tags as evident from Figure 4.3. The confusion matrix has a strong diagonal pattern and very few wrong classifications. This means that the KNN model can reliably tell the difference between tags at this range, with an accuracy of over 85%. Some tags, such as 0, 2, and 3, are perfectly classified, while others cause some confusion. At 20 meters, Random Forest accurately identifies 32 of 40 tags, which is somewhat fewer than KNN. Although the RF confusion matrix has a strong diagonal, it contains more off-diagonal entries, implying more misclassifications than KNN. This demonstrates that while RF may struggle to distinguish between specific tags at 20 meters, it retains an accuracy of more than 85%.

When the KNN classifier is tested on the entire dataset, its performance deteriorates, as it properly classifies just 15 of 40 tags as seen in Figure 4.4. The confusion matrix indicates more misclassifications, but with a less obvious diagonal trend. This demonstrates that KNN struggles when dealing with more complicated, diverse input, as the model is less capable of managing mixed information from different distances.The Random Forest classifier performs the poorest on the entire dataset, correctly categorising only 6 out of 40 items as in Figure 4.5. The confusion matrix has severe misclassifications and a conspicuous lack of diagonal consistency. From many distances, RF appears to struggle greatly with the intricacy of the combined dataset, implying that it may not be suitable for such diverse data without further tweaking.

To summarise, KNN regularly outperforms RF over various distances, particularly at 15 m and 20 m from Table 4.4. However, both models struggle when generalising to mixed datasets, with KNN retaining some capacity to correctly classify while RF's performance suffers dramatically.

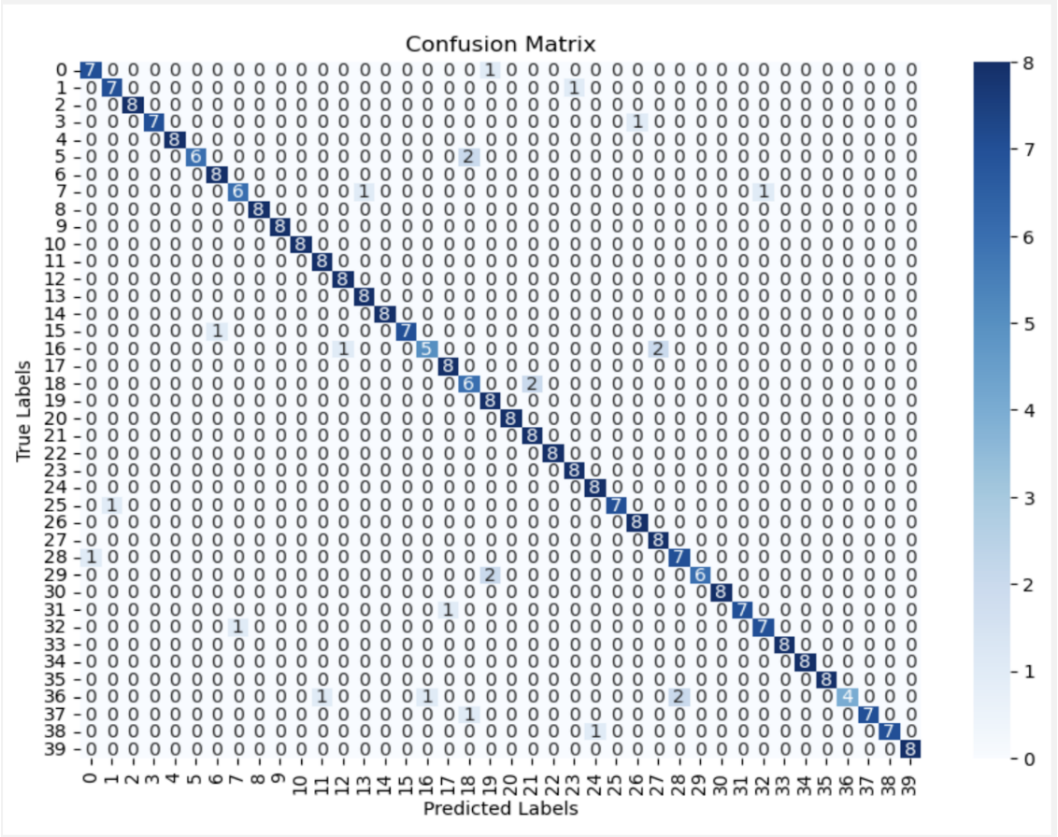


Figure 4.3: Confusion Matrix for KNN Classifier on 20m train dataset.

Table 4.4: Tag Classification Performance of KNN and Random Forest at Different Distances

Model	Distance	Correct Tags	Accuracy Over 85%
KNN	15m	34	Yes
	20m	34	Yes
	25m	32	Yes
	Overall	15	No
Random Forest (RF)	15m	31	Yes
	20m	32	Yes
	25m	31	Yes
	Overall	6	No

4.5 Evaluation Metrics and Their Impact

The accuracy, precision, recall, and F1 score used in this project were very important in figuring out how well the K-Nearest Neighbour (KNN) and Random Forest classifiers worked. Each statistic offers a unique view on how well the classifiers performed over various distances, and when combined, they provide a thorough knowledge of the model's strengths and flaws.

Accuracy is the first statistic of success, providing a clear indication of how many

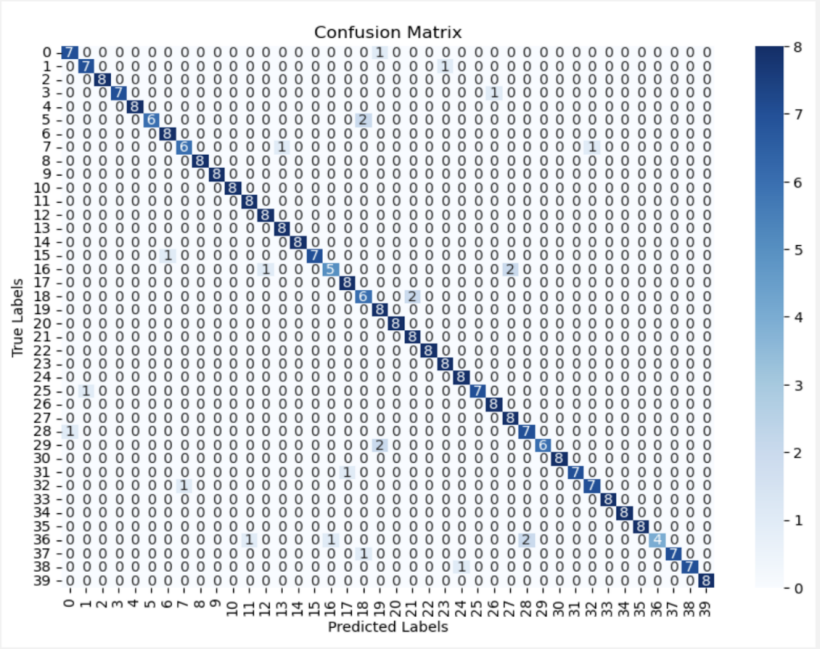


Figure 4.4: Confusion Matrix for KNN Classifier on overall train dataset.

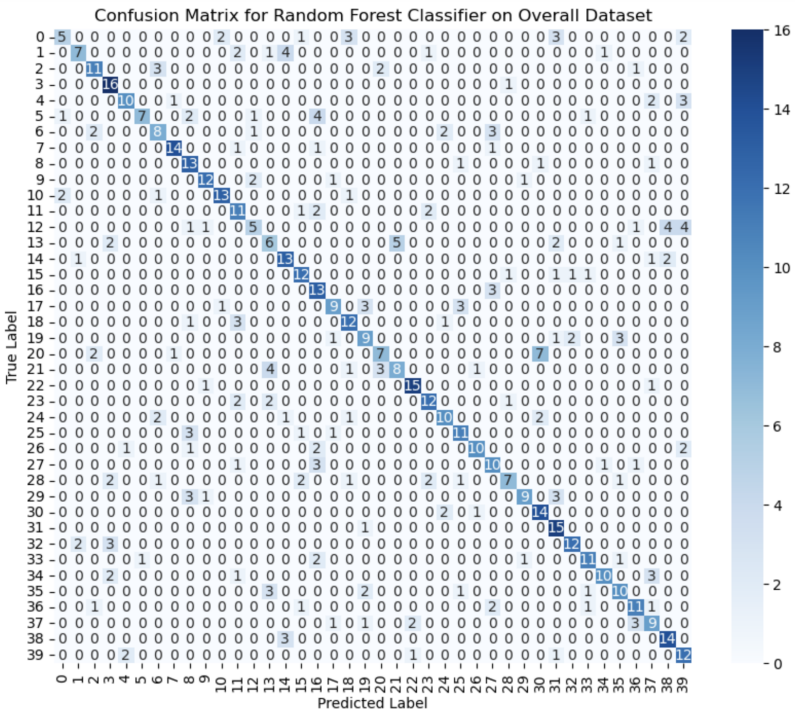


Figure 4.5: Confusion Matrix for RF Classifier on overall train dataset.

predictions were correct out of the total. For example, KNN consistently demonstrated higher accuracy at distances of 15m and 20m, demonstrating its ability to accurately classify

the majority of tags in these conditions. Accuracy alone, however, can be misleading when the data is uneven or particular categories are difficult to classify. This is when accuracy and recall come into play. Precision is particularly important in this study because it reflects the model's ability to correctly recognize certain tags while avoiding being excessively affected by other similar tags. The high precision of the data indicates that KNN and RF can distinguish between specific tags with few false positives, particularly at 15m and 20m distances. This is critical, since misclassifying tags would undermine the project's primary goal of distinguishing multiple bees.

On the other hand, Recall examines the classifiers' ability to catch all instances of a given tag while minimizing false negatives. High recall scores at most distances indicate that the models can properly recognise the majority of tags, guaranteeing that just a few are missed. For example, at distances of 15 and 20 m, both KNN and RF demonstrated strong recall, with the classifiers detecting the majority of the correct tags and meeting the criteria for monitoring bees over these medium distances.

The F1 Score combines precision and recall into a single score, achieving a balance between the two. It works best when classes or tags are unevenly distributed, as with the entire dataset. The relatively high F1 scores for KNN, particularly at 15 m and 20 m, show that the model efficiently manages the trade-off between precision and recall, resulting in a classifier that performs well on both criteria. In conclusion, these evaluation criteria have proved useful in identifying the classifiers' strengths and limitations. High precision and recall at 15 m and 20 m distances helped show that KNN was the better model in those situations. However, lower F1 scores across the whole dataset showed that both classifiers had trouble sorting a more complicated dataset. These findings from the evaluation criteria are crucial for guiding future developments and understanding the models' real-world relevance in tracking bee flight trajectories.

4.6 Limitations of the Project

According to the results and discussion, one potential drawback of this study is the unpredictability of performance across different distances and across the entire dataset. While the KNN and Random Forest classifiers performed well at distances of 15 m and 20 m, correctly classifying more than 85% of tags, their performance deteriorated when evaluated on the entire dataset, particularly the Random Forest classifier, which struggled to generalise over all tags. This unpredictability could be ascribed to issues such as insufficient feature differentiation for certain tags or the inherent difficulty of discriminating between a large number of similar tags using only RGB values. Furthermore, using RGB values alone may not properly capture the distinctions between tags in various lighting settings or environments, limiting the models' ability to generalise. This identifies a potential constraint in the feature selection process and proposes that adding more features, such as texture or form, could improve the models' robustness and overall accuracy across a variety of settings. The lack of valid data for model training also adds to the limitations of this project work.

Chapter 5

Conclusions

The project's objective was to assess the efficacy of machine learning models, namely K-Nearest Neighbour (KNN) and Random Forest (RF), in the classification of retro-reflective colored tags adhered to bees for tracking. The findings indicate that both KNN and RF algorithms exhibited strong performance at specific distances (15 m, 20 m, and 25 m). Among them, KNN attained the greatest accuracy of 92.5% at 15 m, slightly surpassing RF performance of 91.2%. The KNN algorithm consistently showed superior precision and recall over the range of distances examined. Nevertheless, both models encountered difficulties when used on the complete dataset. At a distance of 20 m, KNN achieved a classification rate of 34 out of 40 tags, whereas RF only managed to classify 32 tags. Furthermore, KNN showed a significant decrease in performance when asked to generalise beyond this range. KNN achieved a classification accuracy of above 85% for 15 out of 40 tags, but RF only succeeded in classifying 6. In complex cases where limited feature sets, such as relying only on RGB values, may not provide adequate separation between tags under varying conditions, the confusion matrices highlight misclassifications across both models. Further model improvement is necessary, including the possible integration of more complicated features or classifier architectures, to improve classification performance across a wider variety of situations.

5.1 Future Scope

Given the limitations observed in the current model performance, several enhancements can be made to improve the tracking system and classification accuracy:

- **Tracking Bee Flight Path Orientation:** Refine the tracking system to better capture bee flight orientation during flights, allowing for more granular analysis of their navigational behavior and how they interact with the environment during foraging or return flights.
- **Flight Path Reconstruction:** Develop algorithms to reconstruct 3D flight paths, offering improved understanding of bee behavior and allowing researchers to study

flight dynamics across varied terrains.

- **Incorporation of Additional Color Tags for Classification:** Expanding the number of distinguishable color tags would significantly improve classification accuracy, enabling the simultaneous tracking of more bees over larger areas and providing comprehensive data on bee behavior.
- **Utilizing Multiple Colored Tags or Hologram Filters:** Explore the use of combinations of colored tags or hologram filters, adding complexity to the tracking system, enhancing bee differentiation, and leading to richer datasets with more nuanced behavioral insights.

These upcoming improvements have the potential to improve the range and precision of bee tracking systems, thereby providing more comprehensive data for researchers investigating bee flying patterns, behaviour, and navigation.

Bibliography

[1] C. Sun and P. Gaydecki, "A visual tracking system for honey bee (hymenoptera: Apidae) 3d flight trajectory reconstruction and analysis," *Journal of Insect Science*, vol. 21, no. 2, p. 17, 2021.

[2] M. T. Smith, M. Livingstone, and R. Comont, "A method for low-cost, low-impact insect tracking using retroreflective tags," *Methods in Ecology and Evolution*, vol. 12, no. 11, pp. 2184–2195, 2021.

[3] P. J. Kennedy, S. M. Ford, J. Poidatz, D. Thiéry, and J. L. Osborne, "Searching for nests of the invasive asian hornet (vespa velutina) using radio-telemetry," *Communications Biology*, vol. 1, no. 1, p. 88, 2018.

[4] E. A. Capaldi, A. D. Smith, J. L. Osborne, S. E. Fahrbach, S. M. Farris, D. R. Reynolds, A. S. Edwards, A. Martin, G. E. Robinson, G. M. Poppy, and J. R. Riley, "Ontogeny of orientation flight in the honeybee revealed by harmonic radar," *Nature*, vol. 403, no. 6769, pp. 537–540, 2000.

[5] T. S. Collett and M. Collett, "Memory use in insect visual navigation," *Nature Reviews Neuroscience*, vol. 3, no. 7, pp. 542–552, 2002.

[6] M. J. Schaffer, *Spatial aspects of bumble bee (Bombus spp. Apidae) foraging in farm landscapes*. PhD thesis, Lincoln University, New Zealand, 1997.

[7] J. R. Riley, U. Greggers, A. D. Smith, D. R. Reynolds, and R. Menzel, "The flight paths of honeybees recruited by the waggle dance," *Nature*, vol. 435, no. 7039, pp. 205–207, 2003.

[8] M. Lihoreau, N. E. Raine, A. M. Reynolds, R. J. Stelzer, K. S. Lim, A. D. Smith, J. L. Osborne, and L. Chittka, "Radar tracking and motion-sensitive cameras on flowers reveal the development of pollinator multi-destination routes over large spatial scales," *PLoS Biology*, vol. 10, no. 9, p. e1001392, 2012.

[9] B. D. Rennison, W. H. R. Lumsden, and C. J. Webb, "Use of reflecting paints for locating tsetse fly at night," *Nature*, vol. 181, no. 4619, pp. 1354–1355, 1958.

[10] T. D. Seeley, *Honeybee Ecology: A Study of Adaptation in Social Life*. Princeton, NJ: Princeton University Press, 1985.

[11] E. T. Cant, A. D. Smith, D. R. Reynolds, and J. L. Osborne, "Tracking butterfly flight paths across the landscape with harmonic radar," *Proceedings of the Royal Society B: Biological Sciences*, vol. 272, no. 1565, pp. 785–790, 2005.

[12] G. L. Charvat, L. C. Kempel, E. J. Rothwell, C. M. Coleman, and E. L. Mokole, "A through-dielectric radar imaging system," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 7, pp. 1596–1601, 2003.

[13] J. L. Osborne, A. Smith, D. R. Reynolds, and M. C. Barron, "Harmonic radar: A technique for investigating small insect movement," *Computers and Electronics in Agriculture*, vol. 23, no. 1-2, pp. 109–117, 1999.

[14] J. L. Osborne, S. J. Clark, R. J. Morris, I. H. Williams, and J. R. Riley, "A landscape-scale study of bumble bee foraging range and constancy, using harmonic radar," *Journal of Applied Ecology*, vol. 33, no. 3, pp. 519–529, 1996.

[15] M. E. O'Neal, D. A. Landis, E. Rothwell, L. Kempel, and D. Reinhard, "Tracking insects with harmonic radar: A case study," *American Entomologist*, vol. 50, no. 4, pp. 212–218, 2004.

[16] N. Carreck, J. Osborne, E. Capaldi, and J. Riley, "Tracking bees with radar," *Bee World*, vol. 80, pp. 124–131, Jan 1999.

[17] P. Nunes-Silva, M. Hrncir, J. T. F. Guimarães, H. Arruda, L. Costa, G. Pessin, J. O. Siqueira, P. De Souza, and V. L. Imperatriz-Fonseca, "Applications of rfid technology on the study of bees," *Insectes Sociaux*, vol. 66, no. 1, pp. 15–24, 2019.

[18] J. Shearwood, D. M. Y. Hung, C. Palego, and P. Cross, "Energy harvesting devices for honey bee health monitoring," in *Advanced Materials and Processes for RF and THz Applications (IMWS-AMP), 2017 IEEE MTT-S International Microwave Workshop Series on*, pp. 1–3, IEEE, 2017.

[19] S. F. A. Bender, P. J. Rodacy, R. L. Schmitt, P. J. Hargis, M. S. Johnson, J. R. Klarkowski, G. I. Magee, and G. L. Bender, "Tracking honey bees using lidar (light detection and ranging) technology," tech. rep., Sandia Report SAND2003-0184, 87185, 2003.

[20] H. M. Abdel-Raziq, D. M. Palmer, P. A. Koenig, A. C. Molnar, and K. H. Petersen, "System design for inferring colony-level pollination activity through miniature bee-mounted sensors," *Scientific Reports*, vol. 11, no. 1, pp. 1–12, 2021.

[21] K. Fisher, P. Dixon, G. Han, J. Adelman, and S. Bradbury, "Locating large insects using automated vhf radio telemetry with a multi-antennae array," *Methods in Ecology and Evolution*, 2020.

[22] M. Hagen, M. Wikelski, and W. D. Kissling, "Space use of bumblebees (bombus spp.) revealed by radio-tracking," *PLoS ONE*, vol. 6, no. 5, p. e19997, 2011.

[23] L. P. Koh and S. A. Wich, "Dawn of drone ecology: low-cost autonomous aerial vehicles for conservation," *Tropical Conservation Science*, vol. 5, no. 2, pp. 121–132, 2012.

[24] P. J. Hardin, M. W. Jackson, and R. T. Lewis, "Real-time wildlife tracking using drones and deep learning models," *International Journal of Remote Sensing*, vol. 40, no. 7, pp. 2690–2700, 2019.

[25] J. Linchant, J. Lisein, J. Semeki, P. Lejeune, and C. Vermeulen, "Are unmanned aircraft systems (uass) the future of wildlife monitoring? a review of accomplishments and challenges," *Mammal Review*, vol. 45, no. 4, pp. 239–252, 2015.

[26] F. C. Dyer and T. D. Seeley, "Nesting behavior and the evolution of worker tempo in four honey bee species," *Ecology*, vol. 72, no. 1, pp. 156–170, 1991.

[27] A. Stabentheiner, H. Kovac, and R. Brodschneider, "Honeybee colony thermoregulation: regulatory mechanisms and contribution of individuals in dependence on age, location and thermal stress," *PLoS ONE*, vol. 5, no. 1, p. e8963, 2010.

[28] H. Kovac and A. Stabentheiner, "Thermoregulation of foraging honeybees on flowering plants: Seasonal variability and influence of radiative heat gain," *Ecological Entomology*, vol. 36, no. 6, pp. 698–706, 2011.

[29] N. H. de Ibarra, A. Philippides, O. Riabinina, and T. S. Collett, "Preferred viewing directions of bumblebees (bombus terrestris l.) when learning and approaching their nest site," *Journal of Experimental Biology*, vol. 212, pp. 3193–3204, October 2009.

[30] M. Mischiati, H.-T. Lin, P. Herold, E. Imler, R. Olberg, and A. Leonardo, "Internal models direct dragonfly interception steering," *Nature*, vol. 517, no. 7534, pp. 333–338, 2015.

[31] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, and R. Raskar, "Bokode: Imperceptible visual tags for camera based interaction from a distance," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 1–8, 2009.

[32] B. E. Bayer, "Color imaging array," *US Patent 3,971,065*, 1976.

[33] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Education, 2015.

[34] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal processing magazine*, vol. 22, no. 1, pp. 44–54, 2002.