MCAST

Spice Detection by using Convolutional Neural Networks

Colton Sammut

Supervisor: Owen Sacco

June - 2024

A dissertation submitted to the Institute of Information and Communication Technology in partial fulfilment of the requirements for the degree of BSc (Hons) Multimedia in Software Development

Authorship Statement

This dissertation is based on the results of research carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

The research was carried out under the supervision of Dr. Owen Sacco

....

Date Signature

Copyright Statement

In submitting this dissertation to the MCAST Institute of Information and Communication Technology, I understand that I am giving permission for it to be made available for use in accordance with the regulations of MCAST and the Library and Learning Resource Centre. I accept that my dissertation may be made publicly available at MCAST's discretion.

....

Date Signature

Acknowledgements

The completion of this study could not have been completed without the guidance of my dissertation supervisor, Dr. Owen Sacco. I would like to humbly thank him for the advice and direction given throughout all the dissertations

I would also like to thank my parents Mr and Ms Sammut for supporting me throughout my undergraduate years, and throughout my dissertation

Abstract

This paper embarks on developing an object detection model for detecting and classifying spices and herbs. Different machine-learning techniques are applied to enhance model metrics. One of the most widely used herbs, which can occasionally be challenging to distinguish. Furthermore, similar to other culinary components like mushrooms, they might be difficult to tell apart from other poisons or dangerous plants. Since its inception, Convolutional Neural Networks (CNNs) have made tremendous advances in the domains of computer vision and image analysis. These varieties of neural networks were created to analyze and classify images. Hence, the study combines both areas to develop a solution that can be used in a common household environment to defect and differentiate spices. Furthermore, a system presented in this research can be utilized in the industry to reduce adulteration and maintain consumer health. The procedure for this study was divided into 3 stages: data collection and pre-processing, model training and evaluation, and finally, user testing and study. The setting of the first phase is the collection process of high-resolution images for the dataset. The dataset was meticulously labelled and augmented to enhance model robustness. The second phase adopted the recently released YOLOv9, which improved on its predecessors with reduced parameter architecture and optimized computational efficiency. The hyper-parameter tuning and transfer learning with pre-trained weights of GELAN have refined the model iteratively. Moreover, Online augmentations such as closed mosaics were deployed during training to enhance model generalization and lessen model overfitting. Model validation was evaluated through mAP, precision, and recall. Through the iterative process in Phase 2, a model with; 85.0% mAP, 80.6% Presion, 78.6% Recall scores was trained. The final phase initiated model deployment in an easy-to-use web application, and afterwards it was tested by 18-25-year-old participants. The user study was done to for qualitative feedback on model performance, its usability and additional features, for instance, the ”Get

Food Recipes" option applicability. The evaluation showed that the model detected highly precise spices and surpassed user expectations in certain areas. But fell slightly behind in classification when compared to the quantitative outcome. This implies that it could be employed in many real-world applications to help the visually impaired recognise basic spices and identify other household items.

Keywords: Computer Vision, Web-Application, Spice Detection, Transfer Learning, Data Augmentation

Table of Contents

List of Figures

List of Tables

List of Abbreviations

NN	Neural Network
ML	Machine Learning
DL	Deep Learning
FCN	Fully Convolutional Network
CNN	Convolutional Neural Network
YOLO	You Only Look Once
mAP	mean Average Precision

Chapter 1: Introduction

This paper aims to explore the state-of-the-art models that can predict and categorize spices in images and other computer vision-related problems in the context of artificial intelligence (AI), which has rapidly grown over the recent past. Interestingly, classification is the main challenge of using this technology to identify spices and herbs. The present study aims to establish a concrete model that can identify a total of different spices and herbs and categorize them systematically through the enhanced features of the YOLOv9 algorithm. Although the specifics of this research are limited to the field of computer vision and apply it towards the automation of culinary work, it can be beneficial to the culinary arts. Moreover, it can aid in reducing the adulteration of spices in the industry, which can cost businesses plenty of funds.

This research is inspired by the recent progress in object detection technologies and the increasing number of people who want to use these tools daily. While it may not be as evident at first AI has already been integrated in multiple areas. Some examples include AI chatbots, number plate security detection systems, Stable Diffusion etc. The emergence of models such as Convolutional Neural Networks has transformed the entire area in terms of how objects are detected, allowing it to process images efficiently while maintaining high levels of accuracy. Most recently, a version of YOLOv9 that includes greatly reduced parameters, leading to improved computational efficiency, was released.

How can we develop an accurate and efficient model for detecting and classifying a variety of spices and herbs? This is the main issue that was addressed during this study. This problem includes a few issues: variability in the appearance of spices and the objects from the set and variability in lighting and backgrounds. The goal is to devise a model capable of performing well in any task and yielding optimal results in any setting. The topic of spices and herbs identification was chosen based on their potential applicability and effectiveness in daily life. Spices or herbs are used extensively in food preparations globally, and using accurate names will improve the effectiveness and precision of cooking. This research is also important to support people with vision-impaired disabilities since a normal life requires eyesight in almost everything.

This study aims to develop a precise spice and herb detection model that can be applied in any real-life context. For this study, ten spices were chosen as classes for the dataset: Anise seeds, Basil Leaves, Black Pepper, Cinnamon, Coriander Seeds, Cumin Seeds, Mint, Parsley, Rosemary, and Whole Cloves. These were chosen both for their distinctive features and similarities as well. For example, both mint and basil were classified based on their leaves, which are similar. The texture is different as mint has more rigid leaves while basil's are smoother. This ensured that the model had a challenge when identifying these differences in the sources provided. This investigation also aims to develop an efficient and accurate model by utilizing sophisticated data augmentation techniques to handle aspects such as variability in the appearance of spices. To reach these goals, the below objectives were identified:

1. Enhancement of the YOLOv9 architecture by optimizing the hyper-parameters to improve object detection capabilities relevant to spices and the common household environment.

2. Assess the effectiveness of various data augmentation techniques, including shear augmentation and closed mosaic, rotational, blurring, and sharpening, in improving the model's accuracy and generalization capabilities.

3. Demonstrate the model's utility in real-world scenarios, including a demo displaying the classification of the parts of a spice used for cuisine and a recipe finder function outlining the practicality of an application.

4. Incorporation of user feedback to refine the application and determine the model's positioning.

To analyze the problem of spice and herb detection, a comprehensive methodology was designed, divided into three distinct phases: data acquisition and preparation, intelligent model formation and assessment, and user experimentation and research. The first stage included the creation of a learning dataset comprising ten spices and herbs. Lighting was varied throughout the data collection process, and pictures of the lower limbs were taken from different perspectives. Further, images were collected from publicly available databases and synthesized with the help of Stable Diffusion. The phase involved gathering a high-resolution set of images and obtaining images under different lighting conditions, as well as various other available images from the repository and synthetic image generation models. The next phase concentrated on creating and perfecting the model

by employing the YOLOv9 architecture, which had been trained using GELAN weights. In the last phase, we developed a web application for practical tests that allowed users to upload images and get food recipes, among other features. Our model performance, such as model accuracy or reliability, was assessed using qualitative feedback from participants ages eighteen to twenty-five during a user study. This methodology was applied to ensure the model was capable; therefore, the most practical approach was chosen.

The user study forms one of the basis of the research work, helping to establish the practical usage and acceptance by the user of the spice and herb detection application. The participants praise several points of the model, while others draw attention to weaknesses, including issues like low false positive rates or improving the interface. It is important to establish that this repetitive cycle of improvement based on the authentic users' experience is crucial for creating a sturdy and efficient application.

However, the opportunities for the future and the integration of this system into other areas are extremely likely. The methods and approaches that were proposed for spice and herb detection can be easily adapted for other object detection problems in many other domains, starting from retail and ending up with healthcare. AI, as well as computer vision, continue to evolve, and this will result in yielding more complex and advanced devices and applications.

This study particularly tackles the challenge of capturing the identity of various species of spices and herbs to fuse the existing gap between abstract model performance and practical utility. The information collected in this research pro-

vides direction for future endeavours so that the technology attains greatness in accuracy through performance improvements over time but also gives both significant and consistent results across multiple uses.

Chapter 2: Literature Review

2.1 Overview

Adulteration of goods has always been an issue in any industry. Industries such as the spice industry are most concerned that they can lose millions over health risks and quality. Nowadays, classification solutions can be handled by computer vision processing systems, and in this paper, we shall focus on CNNs. CNN systems such as the one presented by Redmon et al.(2016) [1] YOLO (You Look Only Once) have high accuracy rates on large datasets, and other systems have been developed using different techniques, kernels and filters, pooling layers, and convolution layers. Spice detection is not new, as [2] has already trained a system to distinguish between 10 commonly used spices. In this chapter, we shall analyze how CNN's work and other solutions to the spice classification system have been developed.

2.2 Convolutional Neural Networks

Zhao et al.(2019) [3] explains, "The goal of object detection is to develop computational models and techniques that provide one of the most basic pieces of knowledge needed by computer vision applications: What objects are where?". The underlying issue with such systems is accuracy, and no system has achieved a proper 100% detection or classification rate. That said, multiple new models have gotten close to this number. To address these challenges, models such as

the Yolo7 presented by [4] have shown how CNNs can solve computer vision issues. In this section, we shall provide an overview of the main benefits of CNN, their inner workings, and what techniques can be incorporated to improve the final model.

2.2.1 Different Models

Kim et al. (2020) [5] and Diwan et al.(2023) [6] brought comparative studies to evaluate these models and their distinct variants. The Author trained the yolov4 (Bochkovskiy et al . 2020) [7], SSD, and Faster-RCNN on an automobile dataset of around 2,600 images to classify five different categories of automobiles. Correspondingly, a test dataset of 560 images was used as unseen data to determine the best model produced by these architectures. The research concluded that the YOLOv4 produced the best metrics holding: a precision of 93%, recall of 98%, and an mAP of 98.19%. [6] issued a study focusing on recent advances in two- and one-stage detectors. The Author carried a deep delve into these types of architectures, outlining their strengths and weaknesses. A notable comparison made during the research is that of YOLO, RCNN, and their respective successors. The Author concluded that the single-shot detector yolo performed better in accuracy and inference time than its two-stage detector counterpart, such as RCNN.

Redmon et al.(2016) [1] presented us with the You Only Look Once (YOLO) object detection model. YOLO's adaptability and real-time capability truthfully make it a state-of-the-art solution to object detection and image classification. The 4th iteration of YOLO published by [7] enclosed updates, one of which was the addition of online data augmentations. These augmentations are used during

training rather than those applied to the dataset. These augmentations included CutMix and Mosaic data augmentation, which proved to be effective in increasing the classifier's accuracy. Yolov4 raised both its AP by 10% while increasing speed during inferencing. Wang et al.(2023) [4] presents the seventh version called Yolov7. This paper explores various techniques to improve and enhance the model's performance, including the investigation of model re-parametrization techniques. They mainly focus on incorporating ensemble strategies, such as model-level ensemble techniques involving the training of multiple models and averaging their weights, as well as module-level re-parametrization methods. The authors introduce a new re-parametrization module, further devising an application strategy tailored for diverse architectures. YOLOv9 [8] recently released building upon YOLO7's foundation. Wang et al.(2024) [8]incorporated the *Generalized Efficient Layer Aggregation Network*(GELAN) architecture which uses programmable gradient information. The expansion focused on lowering parameters by 49% and computations by 43% despite the increase in the AP by 0.6%. This architecture was then trained on the MS-COCO dataset, resulting in pre-trained weights that may be utilized for transfer learning.

2.2.2 Transfer Learning

While it may not be evident, humans transfer knowledge from one task to another, but all algorithms, unless specified, will typically handle tasks from scratch. Torrey & Shavlik (2010) [9] describe the goal of transfer learning as transferring knowledge from source tasks to improve learning in a similar target task. One of the areas that the study mentions is inductive learning. The main focus is to

create a model that generalizes well. To CNNs, transfer learning usually works by having a generic model that can identify simple objects and counters and then train it on a specific task.

Transfer learning has established itself as one of the most effective techniques for object detection tasks. This technique is defined by Naeanjo (2020) [10] as freezing the pre-trained network's layers and adjusting the input and output layer to the task at hand. The main benefits of using transfer learning, as denoted by [10] are the following:

1. Reduce the number of epochs to train the model to its best.
2. Reduce the amount of images needed to train the model.
3. Reduce over-fitting, which may eventually lead to incorrect prediction.
4. Boost training and validation accuracy, particularly in the early stages of training.

This approach saves time and enhances results, as with Sundaram et al.(2022) [2], which adopts this technique to create a solution for spice classification. They [2] utilise the VGG16, AlexNet and GoogleNet pre-trained models to accurate models that could classify ten spices. The authors credit this technique for achieving an outstanding 93.06% validation accuracy and a 95% test accuracy.

2.2.3 Data Augmentation

Data augmentation is an interesting solution to a common issue in the CNN model: Overfitting. Other solutions, such as transfer learning, try to rework the

training process, but data augmentation artificially extends datasets and ultimately provides a CNN architecture with more data. Shorten & Khoshgoftaar(2019) [11] expand upon this concept, producing a survey on the different types and applications of data augmentations. Does the author divide the data augmentation into 5 kinds: geometric transformations, photometric transformations, kernel Filters, image Mixing and random erasing. The authors [11] describe geometric transformations as augmenting the image as a matrix and shifting its positions of pixels. For instance, flipping the image would mean multiplying the pixel positions by a factor of -1. While these are the most simplistic augmentations, they are still commonly used for their effectiveness in supplying a model with different perspectives. Photometric transformations, rather than editing pixel positioning, edit the colour spectrum of the images. For instance, converting the image from RGB or HSV to gray-scale for computational speeds. While both augmentation types have their use cases, Taylor & Nitschke(2018) [12] compared both to determine the most influential towards accuracy. The experiments resulted in crediting geometric transformations, specifically rotation, as having the highest impact on accuracy. While it may seem conflicting, more superficial augmentations are more effective because of augmentation safety, which is mentioned by [11]. Augmentation is described as useful to a model, but [11] points out that certain augmentations degrade accuracy, as the reworked images may not be representative of the original objects. The kernel filter involves a kernel matrix sliding across the image to alter its appearance. Blurring and sharpening are examples of augmentation, and they improve a model's stability upon encountering unseen data with

unclear objects, making the model more robust. The issue with these transformations is that an extreme loss of actual data or a gain of artificial data can occur, which may also not represent the original data. [11] describes image Mixing as one of the most effective augmentations for model development. Rather than augmenting a singular image, these augmentations combine and edit multiple images. While the true reason why these types of augmentations are effective, from a human perspective, the resulting image may be nonsense; researchers speculate that it aids a neural network in identifying spatial relationships. The final type of augmentation is random erasing, in which certain parts of images are removed. The aim is to push the model to learn diverse features of objects rather than relying on a single feature to define a class.

While data augmentation is depicted as being effective, what is its true significance on a model? Perez & Wang(2017) [13] sought to find this by training different models using datasets with the sources of the same images but different augmentations. The two types of augmentations applied were traditional augmentation in the form of geometrical transformations and generative adversarial networks (GAN), which were used to combine images in the dataset. General Adversarial Networks, as illustrated by Zhu et al.(2017) [14] are networks that capture special characteristics of one image collection and figuring out how these characteristics could be translated into another image collection. Regarding [13], it was concluded that even though a mixture of both augmentations could be beneficial, traditional augmentation outclassed images generated by General Adversarial Networks.

2.3 Computer Vision Concerning Spice Classification and Detection

In the context of recognizing different spices, Fatima et al.(2022) [15]introduced a computer vision system utilizing a siamese network. This approach aimed to distinguish between papaya seeds and black peppercorns using deep learning techniques. The methodology involved training the model on two sets of images: one comprising pure black peppercorn and the other containing black peppercorn mixed with papaya seeds. The resulting 4,000 image dataset demonstrated an accuracy ranging from 92% to 96% in identifying distinctive features. This closely compares the human validation rate of 97%, which

Similarly, Nasution & Gusriyan(2019) [16] employed a computer vision approach to identify nutmeg based on its visual characteristics, addressing the challenges associated with manual review. The authors incorporated image processing techniques to extract visual information, including size, shape, and colour. The system underwent testing with a 600-image dataset, achieving a high accuracy of 93.33% using a Support Vector Machine (SVM) algorithm. While a K-Nearest Neighbors (KNN) algorithm was also employed, it yielded a slightly lower precision accuracy of 88.33% analogised to the SVM approach.

The study conducted by Sundaram et al.(2022) [2], a comprehensive exploration of spice classification, was undertaken to assess the viability of accurately categorizing ten widely used spices through computer vision technology. The research leveraged the VGG16 CNN architecture and extensively trained on a substantial dataset comprising 2000 images representative of the ”Spice 10” dataset. [2] using several techniques, most notably transfer learning, developing a model

with an mAP of 93.06%. Furthermore, during the testing phase, the VGG16 model demonstrated remarkable efficacy by achieving an accuracy rate of 95%. This study signifies a significant stride in demonstrating the robust capabilities of convolutional neural networks in the intricate task of spice recognition.

In the realm of spice recognition using computer vision, several studies have tackled the intricate task of differentiating spices. [15] employed a Siamese network to distinguish papaya seeds from black peppercorns, achieving an accuracy of 92% to 96%. Similarly, [16] focused on nutmeg identification, reaching a high accuracy of 93.33 per cent using a Support Vector Machine (SVM) algorithm. A comparison with K-Nearest Neighbors (KNN) showed SVM's superiority, with a slightly lower precision accuracy of 88.33 per cent for KNN. In a study conducted by A. Sundaram et al. (2022) [2], spice classification was explored using the VGG16 CNN architecture and transfer learning on a Spice 10 dataset, resulting in an impressive average accuracy of 93.06 per cent. VGG16 exhibited robust performance during testing, achieving an accuracy of 95%, emphasizing the efficacy of CNN-based approaches for accurate spice classification.

2.4 CNN Study with a Mixed Method Approach

Yoshiyuki & Keiji (2013) [17], developed a system to track and classify food on mobile phones and tested it on participants to get feedback on the efficiency. This mixed-method approach is very similar to the approach used in our methodology. The sustained on a data set of 6,781 images, sustained on a data set of 6,781 images, and maintained accuracy was 81.55 %. A user study with five

students was conducted to demonstrate their solution further during the interview. This removed the dependability of proto-type metrics so a more comprehensive and credible evaluation could be held. The authors [17] mention that despite positive comments about usability, accuracy was not as efficient for practical use.

2.5 Summary

In this chapter outlined the main aspects of the study and its current standing point currently. Via comparative studies such as Kim et al. (2020) [5], and Diwan et al. (2023) [6], it was determined that for this study, YOLO would be used for its remarkable speed and accuracy. Through additional investigation, techniques such as transfer learning were introduced and shall be used during the methodology to enhance model metrics. Moreover, acknowledgement of Yolo's hyper-parameters to further augment photographs was noted by Bochkovskiy et al.(2020) [7] as this will be used. When mentioning data augmentation, Perez & Wang(2017) [13] indicated that traditional means of augmentations were to be used over other complex methods. Moreover, Shorten & Khoshgoftaar(2019) [11] via the deep dive into augmentation and their effectiveness, influenced what types of augmentation would be used as not to degrade the system upon adding them. Sundaram et al. (2022) [2] displayed an instance of spice classification and heavily influenced the way forward due to their remarkable accuracy, suggesting to follow the research's foot-steps. Finally, Yoshiyuki & Keiji Y2013) [17] displayed that even though model metrics may look agreeable when put to practical use, a model may find it difficult to classify different objects. This is why a survey

shall evaluate the model's main capabilities.

Chapter 3: Research Methodology

As shown in Fig.3.2, our methodology is divided into three phases, each involving distinct data collection and processing techniques to achieve our research objectives.

3.1 Chosen Methodology

Answering our hypothesis with the quantitative data is not enough. While theoretically, a model can have efficient validation metrics, how does the model fair in case of multiple instances of unseen data? While this can be done manually through a test split, it was an inadequate approach to the research questions. To design a methodology that can be effective, the example of the user study from Kawano & Yanai(2013) [17] was heeded. Instead of interviews, a descriptive survey denting the main expectations of the model picked as the data gathering method. Questions would typically base themselves on the metrics of the model, such as bounding box precision, classification and unexpected detection. The proposed methodology would lead to a comprehensive and detailed evaluation of a multinational solution can be accomplished.

3.2 Phase 1: Data Collection and Pre-Processing

Before data collection was begun, a dataset needed to be defined, and during this study, the dataset was hosted online on Roboflow [18]. This system allowed

Figure 3.1: *Dataset Images Examples*

easy modifications to the dataset and a more manageable way to augment images with their corresponding labels. Moreover, it has straightforward pre-processing and resizing images with an image size of 1024 by 1024. The initial step in occupying the dataset involved capturing high-resolution images of 10 different spices and herbs. Initially, all spices, meaning powders, crushed and dried, would be collected, but fresh spices were the main focus as other forms had appearance issues. These issues included a lack of object, texture, shape, and colour, all required for a CNN to identify the spices presented. Fig 3.1 displays the classes chosen with examples of all types of images and their respective sources. The majority of our images are Augmented images, as they are all the other images combined and edited to create alternative versions for our dataset. These images were developed after the labelling process as transformations; in some instances, transformations needed to occur even on labels. Images are collected

through physical photography, the second most common image source. A 48-megapixel camera was used to capture these quality images with fine details of our species. Various household lighting conditions were simulated by adjusting the light sources, including natural sunlight, artificial indoor lighting, and mixed lighting conditions. Images were also taken from multiple angles and distances to capture the spices' variability in appearance. Excluding the control over environmental factors, images collected from these means also had a consistent image size. Another way of image collection was Stable Diffusion, using a model developed by Rombach et al.(2022) [19]. By supplementing training data with AI-generated scenarios, variability and volume were added to the dataset, eventually leading to a generalised model. Moreover, by adjusting the hyper-parameters of the Stable-Diffusion model, images with greater consistency in quality and size were generated. These images were then uploaded and labelled according to the dataset. To enhance the diversity of the dataset, some images were collected from open-source websites. However, this method was the least preferred due to factors such as inconsistent image size, unrealistic lighting conditions, image type, and object positioning, which could potentially degrade our final model. The primary goal was to obtain data that could effectively train a model to perform well under diverse conditions. These also would go through the same process as the other image types.

The data labelling procedure began directly after the image collection process concluded. Supplying precise labelling is necessary for training a supervised learning model. Each graphic was labelled with the relevant spice label with

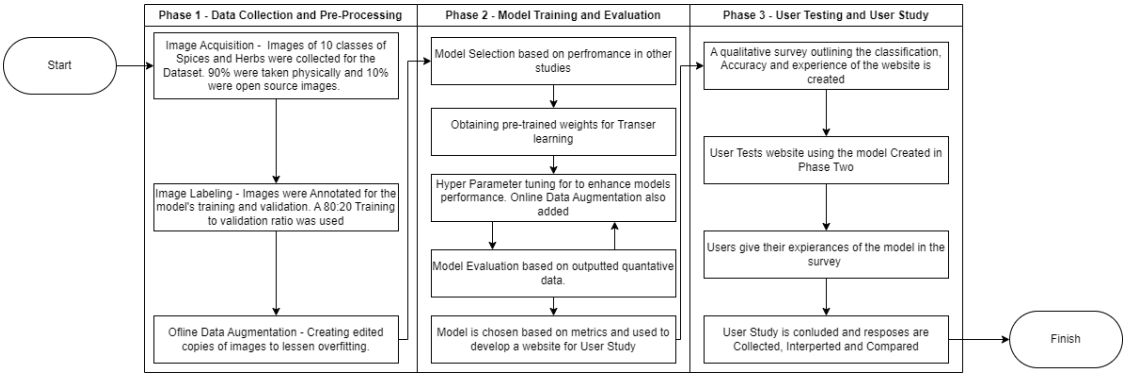


Figure 3.2: *Research Pipeline*

specialised annotation tools. When trying to recognise spices, one of the main details is shape. For example, as displayed in Figure 3.1, the main difference between the basil and the mint classes is the shape, as basil has a more rounded shape. This is why polygons were used to label the area of spices instead of using a more simplistic label choice, such as the case with bounding boxes. The standard split ratio of 80% training and 20% validation was used. This division guarantees the models have sufficient data to learn from, with different collections for adjusting hyper-parameters and assessing efficacy. By changing the original photos, an algorithm can augment data synthetically and boost the overall size of the dataset used for training. During experiments, the common augmentations used were adjustment of exposure, rotation and Adjustment of Brightness. Exposure and Brightness were added to increase resilience against environmental factors such as lighting. Rotation was added so that the model could recognise spices in different positions. With the final model in mind, the shear augmentation was also used to help the model identify our species from various angles, such as with mobile phones. As discussed in Section 2, data augmentation is crucial for improving the model's robustness and generalisation ability.

3.3 Phase 2: Model Training and Evaluation

Popular models like Faster-RCNN, YOLO, and SSD are well known for their effectiveness in object detection tasks. A process to select one of these architectures needed to take place, and from Kim et al.(2020) [5] results and evaluation of these models, the model chosen for training was Yolo for its excellent precision at a steady speed. More precisely, YOLOv9 presented by Wang et al.(2024) [8] shall be used because it has significantly fewer parameters than other state-of-the-art object detection models. The reduced parameters translate to a lighter model, which is faster to train and deploy. Alongside fewer parameters, YOLOv9 employs optimised calculation strategies that further enhance efficiency. Since pre-trained weights were required for transfer learning, GELAN pre-trained weights trained by [8] were opted for. The weights enable a more diverse model to detect patterns with better accuracy and speed. An issue that arises when training any neural network is computational capability. In this analysis, the NVIDIA A100 Tensor Core GPU was used to conduct the experiments for spice detection models. This alternative allows for faster training at a more extensive batch size and would prevent termination of training due to insufficient computational units.

Optimisation of hyper-parameters such as learning rate, batch size, number of epochs, and regularisation parameters is crucial to produce a model with high results. During training, online data augmentation was used to reduce over-fitting further and create variances between epochs. The most commonly used augmentation during training is the closed mosaic with a 15%. This technique stitches together multiple images to form a mosaic and can help the model learn better

spatial relationships. Other online data augmentation techniques that were used include random rotation, translation, and scaling of images. Early stopping was eventually added to reduce unnecessary training and decrease over-fitting. The model was given a 10-epoch ”patience” and a maximum of 100 epochs to train, which would most likely stop between epoch 40 and epoch 60. The iterative training, validation, and refinement processes provided a comprehensive approach to achieving a high-performance spice detection system. It would generate a model integrated into the web application in phase 3.

Generally, the following actions had to be taken to ensure a better model in the subsequent iteration:

1. Class Re-balancing: If the evaluation metrics indicated that the model was biased towards certain classes, techniques such as class re-balancing were employed, i.e. Adjusting the dataset to even the distribution between classes.

2. Label Reconfiguration: Mislabeling is a common contributing factor to poor performance, and these labels were reviewed and corrected. This step ensured that the training data was accurate and that the correct labels of the spices were correct.

3. Hyper-parameter Optimisation: Further fine-tuning of hyper-parameters was conducted based on the model’s performance. This iterative process involved adjusting the learning rate, batch size, number of epochs, and regularisation parameters to find the optimal configuration for the model.

4. Additional Data Augmentation: Experimentation with different data augmen-

tation was done to enhance the model's robustness. These included random cropping, Gaussian noise addition, and brightness/contrast adjustments. The goal was to simulate real-world conditions where the spices might be photographed.

The main metrics that shall be used for evaluation are mAP, precision, Recall, F1 and loss. High precision implies that the model's errors do not mistake the background as a spice for the actual data. In object detection, the model performs well by avoiding such cases where the model gets confused and classifies the background. A higher mAP denotes the optimized identification of objects across the various classes and confidence levels. The high recall indicates few false negative errors where the model can misclassify a spice. In object detection, it excels in not overlooking any object. Recall and precision are levels of model quality, and the F1 score ranges from 0 to 1, where the increased F1 score represents higher model performance. This is very helpful when many students are in one class and scanty in the other. A lower box loss means the model is improving on the localization of objects within the image domain. The lower the class loss, the better the model distinguishes between one object class and another. Table3.1 display how some mentioned metrics are calculated. AP refers to the average precision of a singular class, and as can be detected below, the map is the average of all classes. In this study, n = 10 is the number of classes the datasets hold. TP refers to True positives, meaning the number of predictions of the model that are truly in the image, while FP refers to false positives, which are predictions that the model made, but truly no object was

there. Finally, FN refers to False negatives, which is the when the misses and objects are supposed to be detected.

Table 3.1: *Key Metrics in Object Detection*

Metric	Mathematical Expression
mAP	$\frac{1}{n}\sum_{k=1}^{n} AP_k$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1 Score	$\frac{2 \cdot Precision \cdot Recall}{Precision+Recall}$

By employing such an approach to hyper-parameter optimisation, data augmentation, and iterative refinement, an effective spice detection model was in the making. A model which is theoretically capable of accurately identifying and classifying spices under various conditions, making it well adapted for real-world applications.

3.4 Phase 3: User Testing and User Study

The user study was the final data collection process for the study, and it aimed to evaluate the practical performance of the spice detection model trained in phase 2. The collected qualitative data comprehensively assesses the model's effectiveness and usability in a real-world application. To maintain a consistent experience on all devices regardless of operating system, a web application was chosen as the interface of our model. The website was designed to be simplistic, with the most minor input required for inferring. Users can add images by taking them in real-time or linking a URL to images. Afterwards, a button is clicked that infers the data and returns an inferred image in a container on the web page. Alternatively, users also can generate a list of foods that include these

spices in their recipes. This was done to display the practicality and adaptability of our returned data. The application sent an asynchronous HTTP request to the Roboflow (2024) [18] API to which the model trained was uploaded. During the prediction process, the user would be prompted to wait until the model has been done predicting, which, on completion, an image with bounding boxes and a label would appear in the container. Alternatively, a user could receive a list of recipes using a spice detected in the image. Instead of returning an image, the API would return a JSON in which the name of the spice would be extracted and sent to another API to return a list of recipe names. This function was done to demonstrate the versatility and capabilities of the model.

The primary purpose that the user study was trying to serve is the assessment of how well the detection of spice operates when in use in real-life environments. Specifically, participants were asked to judge the accuracy of bounding boxes placed around the detected spices, the accuracy of the recognised spices, instances of missed spices, which are known as false negatives, and instances of spices that were identified when they were not present, which are known as false positives. The collected data would aid in understanding how effective the model is in real-life situations, notably where the lighting, background, and angles may differ from the ones used in creating the model. Discussion on how the technology used to detect spices could be pulled over to detect other items in use in the home, like ingredients, medicine or utensils. The uncertain and vast nature of such queries pursued the model's versatility and the possible situations in which it could be applied other than those described in these directions. Another crucial

aspect of the qualitative study was to assess how the spice detection model could contribute to the effectiveness of the end-users with vision impairment. Regarding the applicability of such a technology, participants were asked to explore whether a tool could help if a person has a problem identifying spices and herbs. This feedback is essential to determine how far the invention is applicable, needed and will influence society. This was done as advice in developing features meant to enhance the interaction of the visually impaired users, such as the audio description or the haptic feedback mechanisms.

For this purpose, we developed a web application that utilised the CNN model to detect and classify spices. The application was hosted on a web server to ensure easy accessibility for all participants, allowing them to interact with the model from their households. The users would then begin the survey, which queried the following and would give a rating from 1(Strongly Disagree) to 5 (Strongly Agree):

1. Before using the web application ”Spice and Herb Detection”, I was aware that this type of technology is commonly used.

2. While using the web application ”Spice and Herb Detection”, I had no cases of Spices or Herbs being detected where they were not supposed to.

3. While using the web application ”Spice and Herb Detection”, all spices and herbs in the image were inside a bounding box.

4. While using the web application ”Spice and Herb Detection”, I had no cases of spices or herbs being categorised as the wrong type of spice.

5. I believe that I would use the ”Get Food Recipes” feature daily.

6. I believe that the technology displayed in ”Spice and Herb Detection” can be used for all household items(i.e. ingredients, medication, etc...) to aid in identification and detection.

7. After using the web application ”Spice and Herb Detection”, I think that people with vision issues may benefit from an application using this technology.

Qualitative data was collected through structured surveys featuring a Likert scale, which was hosted on Google Forms [20]. This gives participants easy access to the study and would output formative graphs of all responses for more reasonable evaluation. The surveys focused on several key aspects corresponding to the model metric outputted in phase 2. The main focus of the participants is initial expectations regarding the accuracy and usefulness of the spice detection application, experiences while interacting with the application, including ease of use and real-time detection accuracy, and overall post-usage satisfaction with the application. By comparing these qualitative insights with the quantitative results obtained in phase 2, we can validate the model's performance and identify areas for further improvement. The user study provides valuable data for refining the model but helps understand the spice detection system's practical implications and potential real-world impact.

3.5 Ethical Considerations

During the methodology, numerous ethical considerations had to be made to ensure that copyright and GDPR were obeyed. Copy might mainly concern collecting images that belong and are copyrighted by a person or business. To prevent this and risk legal action, the dataset was built from scratch, and photographs collected were either sourced by physical photography, downloaded from open-source websites, or used in a stable diffusion model for which the developers have been credited. The second instance where ethical considerations had to be regarded was during the online survey. All questions identifying an individual would not be integrated into the questionnaire and would be unlawful without a signed agreement. While this would be an acceptable option, there was no justifiable reason to collect such data as the survey mainly needed the collection of opinions regarding the proposed application. Moreover, the applicants would not need to log in to access the questionnaire, meaning e-mails were not collected. The final ethical consideration is data collection through images uploaded by users. A standard tool in CNN is active learning, which would have benefitted the model; this would also mean that the application would collect images. While this could have been used by having an agreement before entering the web application, it would have dismayed the participants. Thus, no active learning or image collection was permitted to keep phase 3 as simple as possible for any user.

Chapter 4: Analysis of Results and Discussion

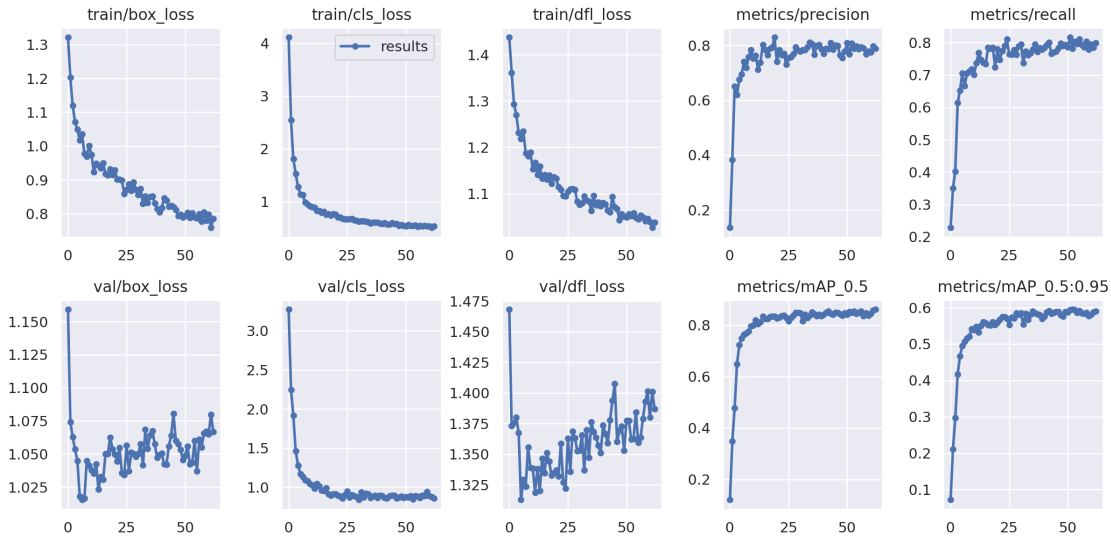


Figure 4.1: *Metric Curves for final model*

4.1 Quantitative Results

4.1.1 Analysis of Model Metrics

In phase 1, a dataset was built to train and quantitatively assess the trained models. With this data set, numerous models were iteratively developed using the recently released Yolov9 architecture. Every iteration would output the model metrics, which were examined to determine the changes needed to enhance the next model. Table 4.1 displays the timeline of models developed with their respective metrics and improvements.

One of the crucial early changes that persisted in the final version was the implementation of Online Data Augmentation. This involved introducing a 15%

chance of generating close-mosaic images during each training epoch. However, it became evident that the augmented training data was causing significant issues with the validation part of the training process, prompting a substantial reduction in the amount of augmentation. New image types, such as synthetic figures and open-source images, were introduced to the dataset for model generalization and diversification. Although this led to a decline in metrics, it was imperative to address the class imbalance in the dataset. A patience modifier of 10 epochs was added for early stopping. This technique dynamically stops the training to reduce over-fitting and unnecessary training. The final enhancement to the model was, reducing the learning rate from 0.01 to 0.0001. This allowed the model to learn at a slower pace, which resulted in a minimum loss.

Model List				
Version	**mAP**	**Precision**	**Recall**	**Changes**
v4	81.3%	82.9%	68.4%	Added Online Augmentations
v10	81.2%	81.1%	68.7%	Reduced Offline Augmentations
v12	83.4%	75.6%	80.8%	Added Open-Source data to address Over-fitting
v16	77.0%	74.9%	80.4%	Added Synthetic data to address class unbalancing
v22	78.8%	72.3%	74.3%	Added Early Stopping to reduce Over-fitting
v24	81.8%	72.9%	80.9%	Adjusted Label positioning to Increase Accuracy
v30	85.0%	80.6%	78.6%	Reduced Learning rate to reduce model mistakes during training

Table 4.1: *Models Milestone Table*

The high mAP deduces that throughout all the classes, a high average precision was maintained. This results in a model that can predict most spices with high confidence, which is displayed in Figure 4.3 as this figure displays an example of the model predicting spices in images. Furthermore, the digit next label

denotes the confidence, and as can be seen in the majority of cases the value is ample. The precision denotes that the model is capable of detecting spices without missing any of them in the images. The lowest of the metric recall shows that the model is slightly over-fitting as there are cases of the model detecting spices where, instead, it should have been the background. This can also be seen in the confusion matrix in Figure4.2 as the background is detected as spices in all classes. Finally is the F1 score, which is 79.6%. This measures the model's overall performance which means the model strikes a good balance between precision (80.6%) and recall (78.6%). This means it's not overly focused on either minimizing false positives or false negatives, but performs reasonably well on both fronts.

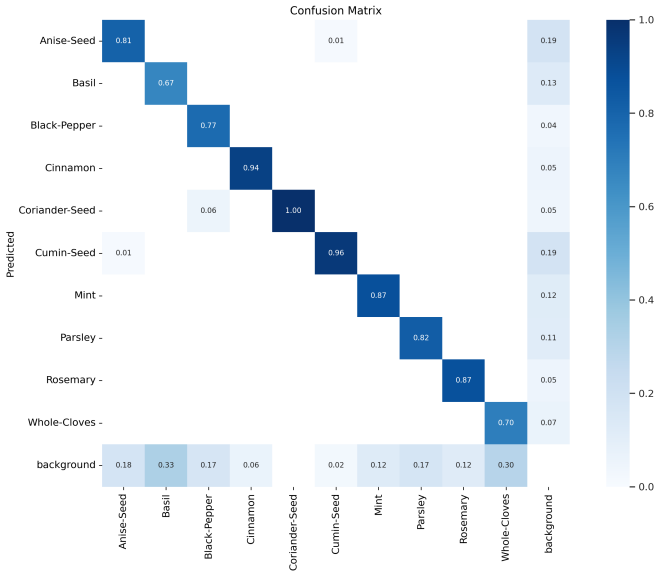
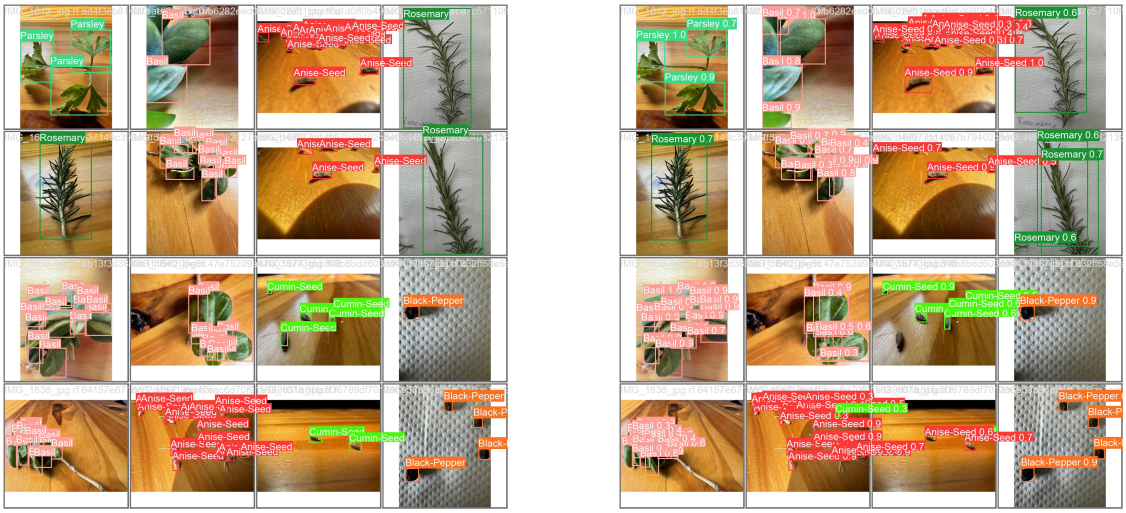


Figure 4.2: *Confusion Matrix for final model*

4.1.2 Challenges and Mitigations

The confusion matrix referenced in 4.2 devices our metrics in specific classes. With this data, a more comprehensive evaluation of misclassification and faulty detection is made. The weakest link in our models is the basil class, yielding the lowest percentage of true positives. While basil is not the most misclassified, it is the least detected, as 33% of validation instances were detected as background. As depicted in figure 4.3, a clear view of Over-fitting can be seen. Validation images with the basil class included have had parts of instances detected, which, while it benefits the model, also lessens the models' metrics. This is because the model is noticing a number of false positives that depend on the perspective it can be seen prediction can be seen a correct. However, the issue with arranging these labels would lead to the model misidentifying certain features that are not present in the partial instance of the basil. The lack of a concise shape and texture would convince the model that other plants (Even unclassified ones) with a similar shape to basil would be classified as basil. While not as prevalent as in the basil call, other classes had these issues, such as the Rosemary and the cumin seed, as depicted in figure 4.3. This balance between generalization and class definition has proven to be a major challenge during training and changes.

Class imbalance was another issue that had been addressed during training. The reason why this occurred was because the objects in the images had different variety in instances. To balance the classes and reduce under-representation in the model, synthetic images using a Stable diffusion model were added. These images proved efficient in balancing the categories of our dataset and further adding



(a) *Validation labels* ***(b)*** *Model Prediction*

Figure 4.3: *Batch Validation Examples*

data upon which to train. Moreover, image size and specification were controlled to match the images in the dataset. This discourages the model from displaying bias towards classes with better representation due to a number of the chosen categories having similar features, which resulted in a consistent number of true positives through all confusion matrices. Moreover, this change stabilized the loss of the trained models, as a wider range of balanced data had been given.

In early iterations, an issue of over-fitting was prevalent. Reducing this issue included manipulation, multiple techniques and hyper-parameters to provide better results. The most impactful was data augmentation, which inflated our data set by a multiplier of 3. Data augmentation techniques tested included a variety of transformations such as rotation, scaling, translation, flipping, exposure adjusting, shearing, and Brightness adjustment. The dilemma with certain augmentations is that they augment the illustration results, which no longer reflect the original spice. Ultimately, offline augmentations were limited to rotation by 90, 180 and 270 degrees. Moreover, during validation, the closed mosiac hyper-parameter was

overloaded, so images were cut out and meshed together to develop new images. This helps the model learn about spatial relationships and enhances its ability to detect spices in a complex arrangement.

While not obvious on the surface, training for 100 epochs was unnecessary as the model's loss would stop earlier. Therefore, the model would not learn past a certain threshold. This would lead to longer training sessions, Over-fitting, and wasted computational units. To solve this issue, early stopping in the form of the patience parameter was added. The setting would allow the article to keep an eye on the loss, and if the loss has not had any significant change, it would stop training and export the results. During training, these parameter effects were studied and evaluated, and it was concluded that a patience value of 10 was required. This would ensure that the model would not stop too early, affecting performance, but it would still be effective in detecting when the model learning rate stops rising.

The final improvement, and one of the most effective, was reducing the learning rate. By default, in yolov9, this hyper-parameter is set to a rate of 0.001. While this rate was active, during evaluation through graphs such as referenced in figure 4.1, it was noted that the model would learn at an accelerated rate. Additionally, it was recognised that the model was making several mistakes and, therefore, misinterpreting spice features. This was mitigated by dividing the learning rate by 10, meaning a new learning rate of 0.0001. This resulted in version 30, the final model used in phase 2 for the user study.

4.2 Qualitative Results

Phase 3 required a collection of participants' opinionated responses. Selected participants aged 18 to 25 were tasked with using the website developed, which hosted the spice detection model. The website was designed for easy accessibility, allowing participants to upload images and receive real-time detection results. The user interface was kept simple to ensure a seamless user experience, with minimal input required for image uploading and inference, as complications could hinder the user experience. Through this process, a collection of 15 responses was compiled as referenced in table 4.2. These questions mainly aimed to compare the qualitative metrics.

A significant portion of participants disagreed or were neutral about their awareness of such technology. This suggests that spice detection technology is not widely known among the user base. However, 46.3% of participants were at least somewhat aware, indicating moderate prior exposure or knowledge levels. This neutrality in knowledge would imply that expectations of the model were mixed. A notable 53.4% of the user base encountered incorrect detection, indicating room for improvement in the model's accuracy. The 40% who disagreed or were neutral suggest that while the model performs adequately in some cases, it struggles with consistent reliability. This indicates the necessity of further refining the detection algorithms to minimize false positives and negatives. With 70% of users agreeing that the bounding boxes were accurate, this aspect of the application appears to be performing well. This directly matches up with the highest metric of the model being mAP. The 13.4% who disagreed or strongly

Table 4.2: *Survey Results*

Statements	**Response**				
	1	2	3	4	5
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Before using the web application ”Spice and Herb Detection,” were you aware that this type of technology is commonly used?	6.7%	26.7%	20%	33.3%	13%
While using the web application ”Spice and Herb Detection,” did you encounter any instances where spices or herbs were detected incorrectly?	6.7%	33.3%	6.7%	26.7%	26.7%
Did the bounding boxes accurately encapsulate all spices and herbs in the image?	6.7%	6.7%	13.3%	30%	40%
Were there any cases where spices or herbs were misclassified as a different type?	20%	6.7%	13.3%	26.7%	33.3%
Do you believe the ”Get Food Recipes” feature is useful for daily use?	0%	6.7%	13.3%	46.7%	33.3%
Do you think the technology displayed in ”Spice and Herb Detection” can be applied to other household items for identification and detection?	0%	0%	13.3%	26.7%	60%
After using the web application, do you think people with vision impairments could benefit from an application using this technology?	0%	0%	26.7%	26.7%	46.7%

disagreed highlight a minority of cases where bounding boxes were imperfect, suggesting some edge cases or specific conditions where the model could improve. Approximately 60% of participants reported misclassification, which is a significant concern. This indicates that while the bounding boxes may be accurate, the classification algorithm still needs substantial refinement. A higher Recall and Precision score was required to get better feedback on this question.

But, this contradicts the results derived from the model as the precision and recall were approximately double that of 40%. This shows that the model is not as accurate in different environments and has a harder time predicting correlation when unseen data varies plenty from the data given for training. The 26.7% responses suggest that misclassification was not a consistent issue for all users, but it remains a critical area for improvement. An encouraging 80% of users found the ”Get Food Recipes” feature useful, highlighting its practical value and appeal. Only 6.7% disagreed, indicating that this feature significantly enhances the user experience by providing added functionality beyond spice detection. 86.7% of participants believe the technology could be applied to other household items. This high level of agreement underscores the perceived versatility and potential of the detection technology. It suggests that users see value in expanding the application's scope beyond just spices, possibly to other areas such as ingredient detection, medication identification, or tool recognition. 86.7% of participants agreed that the technology could be used for other household items. This high level of agreement indicates branch use of the detection technology and suggests the potential of the technology to be applied to different purposes beyond spices. This may include the application of identifying ingredients, types of medications, or tools. An even higher percentage of the participants, 73. 4% agreed that the technology could be helpful to people with vision impairment. Such a high level of correlation insinuates that the application has the potential to aid the visually impaired in gaining a measure of independence and convenience. 27% of the participants are neutral about the impact, suggesting that while its usefulness is

apparent, implementing more refinements to boost efficiency or features particular to accessibility may be helpful.

The main concerns arising out of practicality in the survey are in line with research insights derived from other studies of object detection. In a similar context of comparing the model achieved metrics with the user's responses, which can be specific to classifying tasks, Kawano & Yanai (2013) [17]have depicted similar issues. This trend hints that evaluation models rely on mAP, precision, and recall. Still, these metrics often do not accurately reflect how the models work in practice or how effective and robust they are when used in practical applications. Consequently, including the qualitative data collected from the model users to fill the gaps with critique is vital for any evaluation process.

In addition, differences in the lighting falling within a scene and the positions of the images captured play a critical role in determining the model's performance. While collecting the data, images were taken in different lighting and positions to mimic the environment. Nevertheless, the variety of states in these conditions might remain insufficient to cover a wide range of real-life scenarios. For example, if images depict certain shadows or reflections, the model is likely to misidentify features of spices and herbs, thus delivering incorrect labels.

Finally, in terms of evaluation, the "Spice and Herb Detection" application has received good feedback and positive results; however, the survey results drawn from the testing suggest that future enhancements are necessary concerning false positives and misclassification. These difficulties can be explained by finding certain failures' root causes and examining modern approaches towards model en-

hancement in practice. This way, the application will improve its stability and effectiveness in real-life situations. All these need to be addressed in future research and development endeavours to improve the practicality and efficiency of the spice and herb-detecting technology.

4.3 Comparison of Results

While the results may seem contradictory, the reality is that qualitative results showed the effectiveness of the model, but the qualitative data showcases the instability of such a model. The model's accuracy is reduced when presented with different cases of unseen data under unexpected conditions. This means that if a model is used in the industry to predict spice adulteration, it can theoretically work if the training data's environment and conditions are comparable to the actual environment. When trying to produce an application that is used widely, meaning that the model predicts under different conditions, an issue of classification arises. Future model iterations should address these practical limitations to bridge this gap. Moreover, the integration of user feedback clearly displayed the model's weak point, which in the area of computer vision should be utilized more, as a gap of mixed approaches to computer vision tasks was found.

Chapter 5: Conclusions and Recommendations

The research question posed in this research was, How feasible and how effective is a computer vision system for detecting spices and herbs? Furthermore, the quantitative data showed the possibilities of the model, and detailed accuracy levels in a standard setup were shown. However, reviewing the users' responses, it became clear that the model was inconvenient in different conditions since lighting discrepancies and object positioning significantly affected the results. These differences in results and straightforward testing imply that while the core concept of the technology is useful, more work needs to be done to give the device a solid, functional foundation. Hence, while the study achieved its objective of proving the feasibility of such a system, it ought to have also extended out areas that require further research and development to overcome the real-life hurdles of such different strains of spices or coloured lighting.

The research was structured into three phases: data gathering and data preparation, model construction and performance assessment, and user interaction and research. During these phases, a strong methodology to build the efficient model for spices and herbs identification was elaborated with the help of a state-of-the-art YOLOv9 structure, which is enhanced by the beforehand selected data augmentation methods.

5.1 Summary of Research

The first phase dealt with constructing a dataset sufficient for the problem at hand. Ten different spices and herbs were photographed in high definition with different lighting and perspectives. Extra images were collected from open platforms on the internet and generated by the Stable Diffusion model. This information was necessary to train the model and its ability to respond to many real-life situations. Pre-processing, including rotational augmentations and online augmentations, namely, closed mosaic, was used in this study to artificially increase the size of the dataset in a bid to boost the generalization capability of the model.

In the model training and evaluation stage, YOLOv9's GELAN architecture was used primarily because of the computationally efficient strategies and decreased number of parameters. Moreover, since training requires computational power, the NVIDIA A100 Tensor Core GPU was used in this section. The model was iteratively growing and tested depending on the features like mAP, precision, and recall. Such important increments consisted of the inclusion of online data augmentation and synthetic data for handling the class imbalance issue and lowering the learning rate, all of which aimed to increase the model's acuracy and minimize cases of overfitting. The last model, which was used to generate the final solution, exhibited enhancements in performance that brought out an mAP of 85. 0%, with a precision of 80.6%, and a recall of 75. 6% and an F1 score of 79.6%.

In the user testing phase, the model was implemented into a web application where survey participants would upload images and get real-time detection

results. The survey structure complements the above quantitative assessment to provoke qualitative responses regarding the app's performance and usability. The web application was described as very useful and accurate by most of the participants but was lacking when classifying the various spices. Further comments praised the additional option of 'Get Food Recipes' The following areas were noted for improving the usefulness of the application: As the final limitation, objects were noted as false positives and misclassified, signified that this model had to be modified in the future to make it more realistic to use in practical life situations.

5.2 Future Works

Although the current work has provided promising findings, a number of future research directions and development ideas appear to improve the performance of the "Spice and Herb Detection" application. The following potential directions are constructed based on what has been learned from both the assessment of quantitative models and the collection of noted user feedback; it hopes to overcome the discussed limitations in this research and to envision new opportunities for expanding the application.

For future work, the below are recommended:

1. Separate classes for different strains, i.e., parsley, would be divided into Chinese, Flat leaf or Italian, and English parsley. This will ensure that models have accurate class definitions for different strains and diminish the incorrect classification during testing

2. Further, focus on qualitative data. The conduction of a deeper qualitative methodology can be implemented to identify further the causes of misclassification and the expectations of users towards the system

3. Near the end of this study, the Yolov10 Wang et al.(2024) [21] was released, which reduced the number of parameters, resulting in increased average precision and lower latency. Similar Development using the new architecture may lead to faster and more accurate predictions.

4. Similarly to Kawano & Yanai (2013) [17], a mobile application uses the model to predict spices. This was the first idea as an inferencing platform for this study. Still, since smartphones have different processing speeds, Operating systems and memory a website was used instead, so the model had a wider range of participants.

5. Given the physical limitation, some parts of the study could have been expanded upon, for example, dataset variety. Adding further classes to the dataset can train an expanded model with more definition, which can indirectly assist in producing better.

These areas will be investigated in future studies to improve the results of this work and establish a more practical and reliable approach for recognizing spices and herbs. In conclusion, it is evident that the end result is the development of an effective and usable application targeted at a wide population, including the visually impaired, towards easing convenience and utility in the cookery and general lives of the population.

List of References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[2] Arunachalam Sundaram, Abdullahi Masud, Ali AlMarhoon, and Bhaskarjit Sarmah. Transfer learning approach for classification of widely used spices. *Yanbu Journal of Engineering and Science*, 19(2):1–21, 2022.

[3] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[4] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[5] Jeong-ah Kim, Ju-Yeong Sung, and Se-ho Park. Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition. In *2020 IEEE international conference on consumer electronics-Asia (ICCE-Asia)*, pages 1–4. IEEE, 2020.

[6] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. Object detection using yolo: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications*, 82(6):9243–9275, 2023.

[7] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[8] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024.

[9] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[10] José Naranjo-Torres, Marco Mora, Ruber Hernández-García, Ricardo J Barrientos, Claudio Fredes, and Andres Valenzuela. A review of convolutional neural network applied to fruit image processing. *Applied Sciences*, 10(10):3443, 2020.

[11] Khoshgoftaar T.M Shorten, C. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[12] Luke Taylor and Geoff Nitschke. Improving deep learning with generic data augmentation. In *2018 IEEE symposium series on computational intelligence (SSCI)*, pages 1542–1547. IEEE, 2018.

[13] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[14] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[15] Noor Fatima, Qazi Mohammad Areeb, Irfan Mabood Khan, and Mohd Maaz Khan. Siamese network-based computer vision approach to detect papaya seed adulteration in black peppercorns. *Journal of Food Processing and Preservation*, 46(9):e16043, 2022.

[16] IS Nasution and K Gusriyan. Nutmeg grading system using computer vision techniques. In *IOP Conference Series: Earth and Environmental Science*, volume 365, page 012003. IOP Publishing, 2019.

[17] Yoshiyuki Kawano and Keiji Yanai. Real-time mobile food recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7, 2013.

[18] B Dwyer, J Nelson, J Solawetz, et al. Roboflow (version 1.0)[software]. *URL: https://roboflow. com. computer vision*, 2022.

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[20] Google forms: Online form creator | google workspace.

[21] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.

Chapter A: Introduction of Appendix

This appendix provides supplementary materials to support the main text of the research paper. It includes additional details and information that may be helpful for a more comprehensive understanding of the study's methodology, results, and analysis. The appendix is organized into the following sections:

Chapter B: Sample Code

B.1 YOLOv9 notebook

```
# -*- coding: utf-8 -*-
"""Yolo9 Spice Detection.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1EPOKn2ErOxGqNde9l2mVmE0_FZ_8fscw
"""

# View CPU/GPU Specs
!nvidia-smi

# Installing and Importing dependencies
!pip install roboflow
import os
import shutil
from IPython.display import Image
from roboflow import Roboflow

#Connecting to google drive for a more permnet model
from google.colab import drive
drive.mount('/content/drive')
os.chdir('/content/drive/MyDrive/yolov9')

# Setting a Variable home as a shot cut to the directory
HOME = os.getcwd()
print(HOME)
```

```
# Commented out IPython magic to ensure Python compatibility.

#Cloing Yolov9 REPO

!git clone https://github.com/SkalskiP/yolov9.git

# %cd yolov9

shutil.move('/content/yolov9', '/content/drive/MyDrive')

Set your working directory to YOLOv9 in your Google Drive

Install requirements

!pip install -r requirements.txt -q

!pip install -q roboflow


#Fetching pre-trained weights for Yolo

!mkdir -p {HOME}/weights

!wget -P {HOME}/weights -q

↪  https://github.com/WongKinYiu/yolov9/releases/download/v0.1/yolov9-c.pt

!wget -P {HOME}/weights -q

↪  https://github.com/WongKinYiu/yolov9/releases/download/v0.1/yolov9-e.pt

!wget -P {HOME}/weights -q

↪  https://github.com/WongKinYiu/yolov9/releases/download/v0.1/gelan-c.pt

!wget -P {HOME}/weights -q

↪  https://github.com/WongKinYiu/yolov9/releases/download/v0.1/gelan-e.pt

!ls -la {HOME}/weights


from roboflow import Roboflow

rf = Roboflow(api_key="API-KEY")

project = rf.workspace("workspace name").project("Project Name")

version = project.version(30) # Dataset Version

dataset = version.download("yolov9") # Version of Download


# Running the Train.py with overloaded Hyper-parameters

!python train.py \

--batch 8 --epochs 100 --img 1024 --device 0 --min-items 0 --close-mosaic 15 --workers

↪  4 --patience 10\

--data "/content/drive/MyDrive/yolov9/Spice-and-Herbs-dataset-30/data.yaml" \

--hyp hyp.scratch-high.yaml \

--cfg models/detect/gelan-c.yaml \
```

```
--weights /content/drive/MyDrive/yolov9/weights/gelan-c.pt \

# Resuming the Train.py with overloaded Hyper-parameters
!python train.py \
--resume --batch 8 --epochs 100 --img 1024 --device 0 --min-items 0 --close-mosaic 15
↪  --workers 4 --patience 10\
--data "/content/drive/MyDrive/yolov9/Spice-and-Herbs-dataset-30/data.yaml" \
--hyp hyp.scratch-high.yaml \
--cfg models/detect/gelan-c.yaml \
--weights /content/drive/MyDrive/yolov9/weights/gelan-c.pt \

# Upload of the model to the Roboflow Dataset
rf = Roboflow(api_key="API-KEY")
project = rf.workspace("workspace name").project("Project Name")
version = project.version(30)
version.deploy(model_type="yolov9", model_path=f"path/to/your/model")
```

B.2 Stable Diffusion Notebook

```
# -*- coding: utf-8 -*-
"""Stable diffusion.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1pPE7tG-BnbvZSHbh5OifseUr1OXsTYoe
"""

# Commented out IPython magic to ensure Python compatibility.
# %%sh
# pip install -q --upgrade pip
# pip install -q --upgrade diffusers transformers scipy ftfy huggingface_hub

from huggingface_hub import notebook_login
```

```
# Required to get access to stable diffusion model
notebook_login()

import torch
from diffusers import StableDiffusionPipeline

pipeline = StableDiffusionPipeline.from_pretrained(
    "runwayml/stable-diffusion-v1-5", torch_dtype=torch.float16, revision="fp16"
)

pipeline = pipeline.to("cuda")

import os

from IPython.display import Image, display


def generate_images(
    prompt,
    num_images_to_generate,
    num_images_per_prompt=4,
    guidance_scale=8,
    output_dir="images",
    display_images=False,
):

    num_iterations = num_images_to_generate // num_images_per_prompt
    os.makedirs(output_dir, exist_ok=True)

    for i in range(num_iterations):
        images = pipeline(
            prompt, num_images_per_prompt=num_images_per_prompt,
            ↪  guidance_scale=guidance_scale
        )
        for idx, image in enumerate(images.images):
```

```
            image_name = f"{output_dir}/image_{(i*num_images_per_prompt)+idx}.png"
            image.save(image_name)
            if display_images:
                display(Image(filename=image_name, width=128, height=128))

# 1000 images take just under an 1 hour on a V100

generate_images("Coriander Seeds", 50, guidance_scale=4, display_images=True)

from google.colab import drive
drive.mount('/content/drive')
```