Automated Grape Counting with Deep Learning on Big Data

MSc Research Project
Cloud Computing

Naveen Kesavan
Student ID: x19153163

School of Computing
National College of Ireland

Supervisor: Diego Lugones

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Naveen Kesavan
Student ID:	x19153163
Programme:	Cloud Computing
Year:	2021
Module:	MSc Research Project
Supervisor:	Diego Lugones
Submission Due Date:	14/8/2023
Project Title:	Automated Grape Counting with Deep Learning on Big Data
Word Count:	7077
Page Count:	26

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Naveen Kesavan
Date:	17th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Automated Grape Counting with Deep Learning on Big Data

Naveen Kesavan
x19153163

Abstract

This research paper introduces an approach to address the challenge of efficient grape counting in extensive datasets using computer vision and deep learning techniques. The primary motivation behind this study is to develop a system capable of accurately counting individual grapes within images, with specific emphasis on aiding grape crop management in large agricultural areas within developing nations. Leveraging convolutional neural networks and a meticulously annotated dataset, our proposed system demonstrates remarkable proficiency in grape detection and categorisation. The system operates within a cloud-based infrastructure, ensuring accessibility to users across diverse geographical locations. The real-time processing capability of the cloud-based setup is particularly crucial for precision agriculture applications, offering a feasible solution for managing extensive datasets. Notably, this automated grape counting mechanism is uniquely positioned to benefit the farming community in resource-constrained settings. Rather than a universal solution, it is tailored to address the needs of impoverished farmers in developing nations who collaboratively share computing resources within a communal framework. By embracing deep learning for automatic fruit counting, this research presents a promising avenue for enhancing agricultural practices. The system's ability to provide reliable and time-efficient grape counting methods holds significant potential for driving improvements within the farming sector. This work underscores the value of community-oriented solutions in addressing agricultural challenges while harnessing the power of emerging technologies.

1 Introduction

Grape counting plays a pivotal role in various facets of vine yield estimation, serving as a foundational method across agricultural, scientific, and ecological domains. Its significance lies in enabling precise yield estimates, comprehensive evaluations, and informed decision-making processes that collectively contribute to heightened productivity, sustained longevity, and enhanced comprehension of grape-bearing plant systems. Conventionally, the task of grape counting has been a manual endeavour, characterised by its labor-intensive and error-prone nature, particularly when applied to large-scale agricultural contexts. However, recent technological advancements have ushered in a new era of automated grape counting methodologies, aimed at streamlining operations and augmenting efficiency. Automated fruit counting algorithms have gained prominence within the agriculture sector, presenting a cost-effective avenue to enhance precision while curtailing operational expenditures. Traditional fruit counting and detection methods demand significant temporal and labor investments from farmers. Nonetheless, contemporary investigations have explored the feasibility of automating fruit recognition and counting through the fusion of computer vision techniques and unmanned aerial vehicles (UAVs). Notably, the advent of deep learning-based systems, encompassing semantic segmentation and object identification algorithms, has introduced the potential for automated and accurate fruit counting. UAVs, equipped with cameras and sensors, have emerged as potent tools for rapid traversal and high-resolution imaging of expansive farmland, thereby facilitating efficient fruit counting and yield estimation.

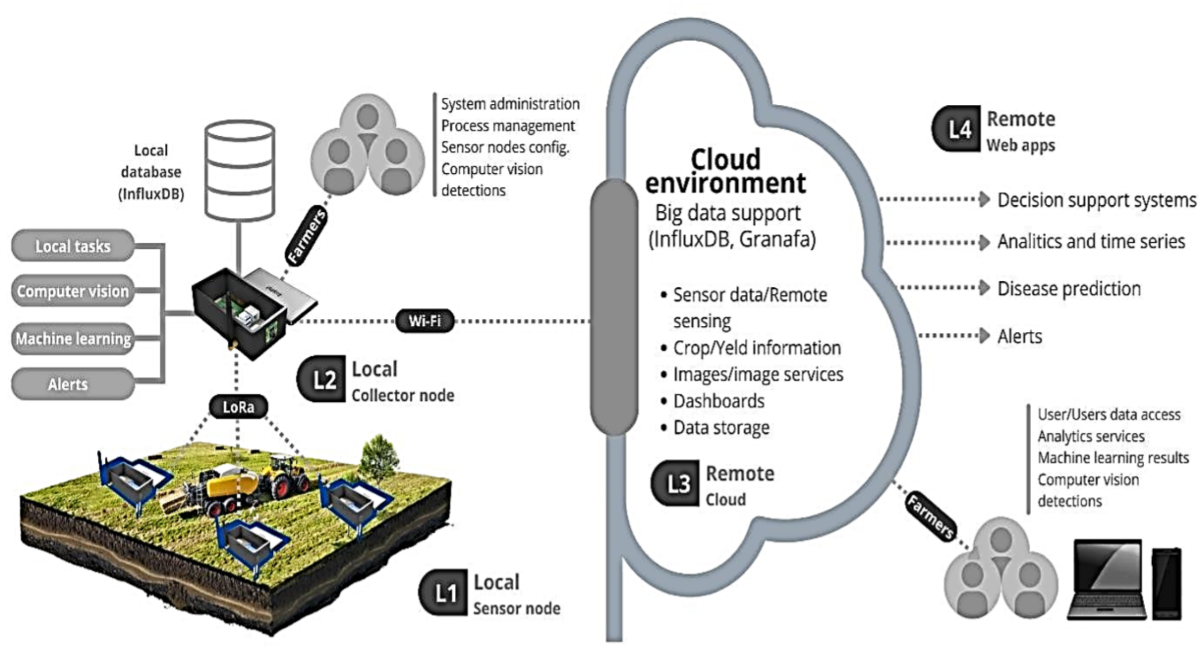

Figure 1: Sensors based strawberry auto counting
Cruz et al. (2022)

Harnessing computer vision methodologies alongside UAV technology for fruit recognition and counting holds transformative potential for agriculture. This potential extends to resource-constrained settings, wherein a communal approach to computing resources is adopted to address grape counting challenges among economically disadvantaged farmers in developing nations. This paper presents a novel communal solution tailored to address grape counting needs within such contexts. Through a comprehensive exploration of sensors-based strawberry counting as a representative automated fruit counting mechanism (depicted in Figure 1), the subsequent sections of this report delve into the intricacies of data collection, storage, machine learning predictions, and visualisation, ultimately contributing to the advancement of scientific understanding in this domain.

1.1 Problem Statement

The primary objective of this research paper is to tackle the issue of quantifying grape bunches in three dimensions, specifically within vineyard environments. Existing methodologies might not be viable for practical applications due to their dependence on controlled lab conditions or specialised equipment. For instance, approaches such as Otsu's binarisation while rotating grape bunches or utilising a stereo camera with fluorescent lighting have limitations in terms of background resource requirements and operational duration. Conventional techniques for grape counting and detection lead to challenges in crop management and yield estimation due to their inefficiency, inaccuracy, and labor intensiveness.

To address these problems, there is a pressing need for the development of accurate, efficient, and cost-effective automated technologies for grape counting and detection. An innovative solution could involve the integration of unmanned aerial vehicles (UAVs) and computer vision algorithms. Nonetheless, significant hurdles exist in handling vast datasets and creating robust deep learning models capable of accommodating variations in grape characteristics like appearance, size, and shape.

The overarching goal of this study is to assist farmers in enhancing crop management and yield optimisation. This can be achieved by furnishing them with reliable and efficient automated systems for grape counting and detection.

1.2 Motivation

The aim of this research is to develop a fully automated system for counting grapes, with the intention of supporting farmers in developing and agriculturally focused nations. The principal objective of the system is to enhance farmers' productivity by assisting in predictive production, planning, and automated field analysis through the implementation of intelligent cloud-based systems. This approach aims to reduce labor expenses, enhance convenience, and extend the system's applicability. The integration of agricultural automation, exemplified by automated grape counting, is anticipated to bolster output precision and volume while reducing labor-related costs. Given that manual grape counting demands considerable time and labor, the pursuit of automated alternatives is of significant interest. These algorithms have the potential to dramatically accelerate the counting process, enabling farmers to allocate their time and efforts to other tasks. The accurate yield estimates generated by automated grape counting additionally support farmers in making well-informed decisions regarding production and marketing. In essence, the agricultural sector can derive advantages from this technology, leading to improved efficiency, increased farmer independence, and sustained economic growth.

1.3 Research Outcome

Multiple crucial factors essential for the prosperity of agricultural practitioners and the economic viability of their endeavours can be effectively addressed through the creation of a computer vision-enabled, cloud-based deep learning framework designed for the purpose of quantifying grapes. This innovative approach enables real-time and precise yield forecasts, along with the early detection of potential issues such as crop diseases, pest infestations, and nutrient deficiencies. These proactive interventions have the potential to enhance agricultural productivity and decrease overall expenditures. Furthermore, the pivotal practice of grape thinning plays a vital role in the grape production process, significantly influencing grape size, flavour, and overall quality. The task of grape quantification can be automated, providing farmers with improved grape bunch thinning capabilities. This, in turn, optimises the grape yield per bunch, ensuring maximal grape development and maturity. As a result, vineyard management stands to benefit, ultimately leading to heightened profitability.

1.4 Business Objectives

A study is currently underway in the realm of contemporary agriculture, aimed at enhancing grape production by synergising advanced technology with agricultural practices. The central objective of this pioneering effort is to cultivate a sophisticated deep learning system, harnessing the capabilities of computer vision technology, focused solely on the precise enumeration of grapes. This innovative technology holds promise for broad accessibility and scalability, potentially becoming a valuable asset for farmers worldwide. Its design has been thoughtfully tailored to seamlessly integrate within the expansive realm of cloud computing. This inventive approach strategically leverages the untapped potential of cloud infrastructure to forge an economical and readily deployable solution adaptable to diverse devices, meeting the specific requisites of farmers, particularly in underdeveloped regions characterised by sprawling agricultural lands and resource limitations.

The significance of this transformative initiative is particularly profound in the rural expanses of emerging economies, where agricultural productivity intimately influences the sustenance of communities. The accurate assessment of yields has gained substantial prominence within the agricultural narrative, as crops extend their dominion across vast landscapes. The reverberations of this advancement extend extensively, affording farmers a more precise evaluation of their harvest quality and consequently empowering them to make decisions characterised by exactitude. Central to this groundbreaking notion are drones, gracefully navigating the skies to capture multidirectional imagery of fields. This visual mosaic is then compiled into a repository, ripe for insights into the hidden abundance within every land parcel. Woven into the intricate tapestry of this deep learning system are algorithms perpetually engaged in analysis and quantification, presenting an intricate inventory of the opulent crops adorning the fields. The ramifications stretch beyond a mere enumeration, relieving farmers of uncertainty's burden through cost reduction and unbiased monitoring. This paradigm shift offers a crystalline perspective, ushering them into a realm of unparalleled precision, where the authentic nature of their harvests can finally be unveiled. The emergence of cloud-based computer vision heralds a new era wherein agriculture evolves into a blend of art and science, contributing to human sustenance and economic prosperity.

Four discernible stages significantly facilitate the classification of grapes. The initial stage encompasses automated image acquisition via drones, mitigating the need for labor-intensive manual efforts. Subsequently, the second stage entails real-time deep learning regression, estimating grape counts and employing filters for ripeness classification, seamlessly presented on a real-time dashboard.

1.5 Research Question

With the preceding discussion taken into consideration, the present study identifies a set of research inquiries that merit attention:

- **Research Question 1 (RQ1):** Could it be feasible to enhance the efficiency and precision of grape detection and counting in agricultural contexts by refining deep learning architectures, including semantic segmentation and object detection algorithms?

- **Research Question 2 (RQ2):** To expedite fruit counting and yield estimation, what measures can be taken to maximise the effective utilisation of unmanned aerial vehicles (UAVs)? Furthermore, what are the key technical obstacles that necessitate resolution to ensure results that are both accurate and dependable?

- **Research Question 3 (RQ3):** What variables wield substantial influence over the precision and accuracy of algorithms employed for automated grape counting? How can the potential issues stemming from these variables be alleviated to foster dependable outcomes across a diverse array of environments and grape categories?
- **Research Question 4 (RQ4):** To streamline the management of extensive datasets produced by automated grape detection and counting systems, how can cloud-based processing be optimised? Additionally, what strategies can be employed to safeguard data privacy and uphold security standards within cloud-based environments?

Subsequent sections of this paper will expound exhaustively upon various studies pertinent to this subject matter. This will be succeeded by an in-depth exploration of the methodology and the suggested implementation approach. Following that, detailed results under distinct scenarios will be deliberated upon, culminating in comprehensive findings.

2 Related Work

The literature review in this research paper critically examines the landscape of automated berry or fruit counting methodologies, positioning the current study within the broader context of academic research. Automated berry counting is crucial for yield estimation and crop management, especially in fruit farming, where manual counting proves laborious and time-consuming. The convergence of computer vision and machine learning has sparked the development of innovative algorithms in recent years. Autoberry, a drone-deployed algorithm, exemplifies this trend, demonstrating the potential of automated counting methods. Various techniques have emerged for automated berry counting, each with its strengths and limitations.

2.1 Variations in Deep Learning Architectures for Fruit Detection and Counting in Computer Vision Applications

In the realm of deep learning-based methods, Roy et al. (2021) introduced the En-UNet model, achieving superior accuracy in real-time apple peel segmentation compared to traditional UNet architecture. Jiang et al. (2023) enhanced dense crop recognition through MIoP-NMS, yielding significant gains in banana tree detection accuracy. Kang and Chen (2020) developed LedNet for rapid apple detection through multi-scale pyramid labelling, while Mekhalfi et al. (2020) offered a kiwifruit assessment method with streamlined counting procedures and improved accuracy. The works of Buayai et al. (2020), Ponce et al. (2019), Tu et al. (2020), Junos et al. (2021), and others further enriched the field, presenting techniques ranging from grape counting to oil palm fruit harvesting. These studies collectively underscore the advancements brought about by deep learning techniques and their potential applications in various fruit counting scenarios.

2.2 Segmentation and Enumeration of Fruits and Seedlings using Autonomous Unmanned Aerial Vehicles (UAVs)

Automatic fruit counting using Unmanned Aerial Vehicles (UAVs) has gained traction as a powerful approach. Chen et al. (2022) harnessed multispectral photography and LiDAR data to predict apple tree production, highlighting the potential of ensemble learning. Barreto et al. (2021) showcased a UAV-based system for automated plant counting across different crops, simplifying agricultural practices. The integration of UAVs with machine vision, as demonstrated by Vijayakumar et al. (2021), promises more accurate citrus crop yield forecasts and efficient resource utilisation. These studies underscore the efficacy of UAVs in conjunction with machine vision for crop monitoring and counting, emphasising their potential to revolutionise precision agriculture.

2.3 Real-time Monitoring and Edge-Deployed Cloud-Based Approach for Fruit Detection and Counting

Cloud-based real-time monitoring has also emerged as a critical paradigm in fruit detection and counting. Lyu et al. (2022) proposed YOLOv5-CS for green citrus detection, combining object detection and cloud-based processing to achieve high accuracy. Abbas et al. (2022) presented a cloud-enabled system for automated disease detection, enhancing plant health monitoring for farmers. In a broader context, Neelesh Mungoli (2023) delved into the potential of scalable, distributed AI frameworks in cloud computing, offering insights into data storage, management, and training methods. Ningli Xu (2023) evaluated state-of-the-art point cloud registration methods, enhancing understanding and techniques in this domain. These studies collectively underscore the potential of cloud computing in enhancing fruit counting, disease detection, and precision agricultural practices.

2.4 Summary

The amalgamation of deep learning, computer vision, and cloud computing has led to a proliferation of innovative methodologies in fruit detection, counting, and disease diagnosis. Leveraging tools like UNet, En-UNet, MIoP-NMS, and UAVs, researchers have achieved remarkable accuracy across diverse environments. UAVs have emerged as a transformative tool for crop prediction, counting, and harvesting. YOLOv5-CS and cloud-enabled disease detection systems showcase high-precision models with significant practical implications. This body of work collectively signals a new era of productivity, sustainability, and data-driven decision-making in agriculture.

2.5 Research Contributions

The research review delineates avenues for advancing automated fruit counting through deep learning and cloud-based data handling. The suggested avenues include the development of advanced deep learning models capable of handling occlusions, improving data collection and preprocessing techniques, and exploring cloud-based deployment of models. These contributions collectively address the need for improved accuracy, efficiency, and scalability in automated fruit counting, promising to enhance precision agriculture practices and empower stakeholders in the agricultural sector.

The novelty aspects of this endeavor include:

1. Developing a dynamic pipeline capable of automatic adjustments depending on the nature of the input data.

2. Introducing an extensible API (Figure 2) that seamlessly integrates with diverse data acquisition systems such as UAVs and Static Cameras. This API aims to capture images of grapes, assess the quantity of grapes, and subsequently update the central database repository.

3. A government funded communal approach to sharing cloud computing resources and automated grape counting equipment offers several advantages over traditional manual methods:

   (a) **Cost savings:** Hiring workers to manually count grapes is expensive. For 1 acre vineyard, 10 workers for 5 days at 400 INR (4.50 euro) per day costs 20,000 INR (225 euro). In contrast, the upfront cost of automated systems (cameras, drones, cloud servers) can be shared across many farmers, bringing down the per-user cost significantly. The operating costs are also lower as no manual labor is needed.

   (b) **Time savings:** Manual counting is extremely time consuming, taking 5 days for 1 acre in this example. Automated systems can count grapes in a matter of hours, providing near real-time data to farmers. This allows timelier decision making regarding yields, harvesting, sales etc.

   (c) **Accuracy:** Human counters are prone to errors and variability. Automated systems provide consistent, reliable counts once properly trained. This supports data-driven decision making.

   (d) **Accessibility:** Cloud-based systems can be accessed remotely via cheap mobile devices, opening up grape counting technology to poor, small-holder farmers. A communal approach also reduces barriers to entry.

   (e) **Scalability:** Cloud infrastructure and automated systems can easily scale to handle large vineyards at nationwide or global scale. Manual labor does not scale as efficiently.

   (f) **Sustainability:** Reduced need for massive manual labor forces is more sustainable long-term. Automated systems also promote standardisation and transparency in yield estimates.

   In summary, a government supported communal model for automated grape counting leverages economies of scale while providing farmers an affordable way to access advanced technology and data. This drives efficiency, productivity and transparency in the agricultural sector. The long term benefits outweigh the initial setup costs.

3 Methodology

3.1 Research Method

The challenges associated with grape counting encompassed the following aspects:

1. Identifying an appropriate data source proved to be challenging due to the limited availability of relevant data.
2. The network architecture posed another challenge. Various models were trained across different environments such as AWS, Google Colab, and local setups to assess computational capabilities.

The resultant trained network was intentionally designed to be versatile, allowing it to generate a pickle file or trained model file tailored to the dataset at hand. This adaptability is crucial due to potential shifts in data nature, necessitating corresponding changes in the model. Subsequently, the trained network was deployed on an AWS server[1]. To engage the deployed server, an interface like Postman was employed. Users can submit images containing grapes for counting, which then generates the count of units (Figure 2). Given our primary focus on gauging large-scale grape harvesting on extensive fields or farms, the individual count of grapes holds lesser significance. Instead, a colour-coded system was introduced to provide more insightful information to farmers:

Green: Exceeding 10,000 grapes
Yellow: Ranging from 1,000 to 10,000 grapes
Red: Fewer than 1,000 grapes

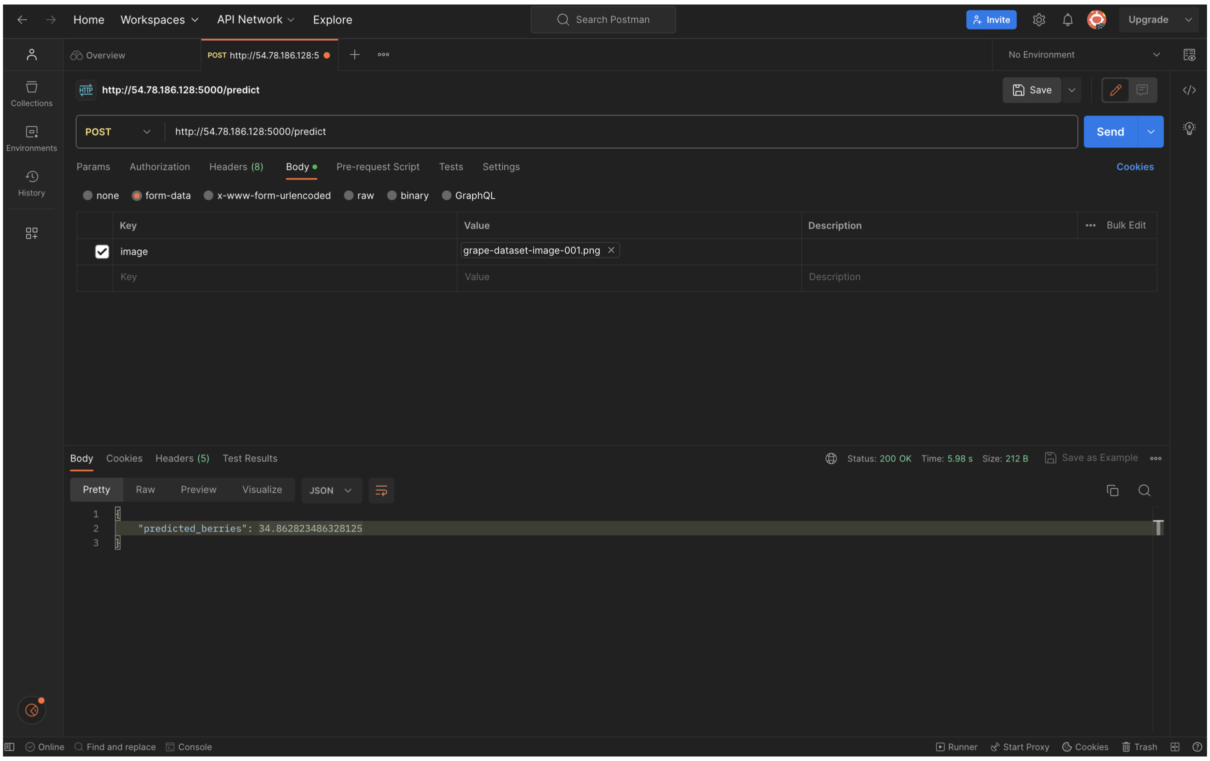

Figure 2: Extensible API

1. **Data Collection Phase:** The initial step of the proposed methodology involves the acquisition of grape images through the utilisation of cameras and drones. These instruments facilitate the systematic collection of images, which are subsequently transmitted to local servers on a daily basis.

[1] Deployment on an AWS server: `http://54.78.186.128:5000/predict`

2. **Image Preprocessing Stage:** Subsequent to data collection, a vital preprocessing stage ensues. This stage encompasses the enhancement of image quality and the reduction of noise. Techniques such as resizing, normalisation, filtering, and segmentation are applied to optimise the images for downstream analysis.

3. **Construction of Dataset:** Following image preprocessing, the dataset is organised into distinct subsets for diverse purposes. The dataset is partitioned into training, validation, and testing sets. To introduce diversity and robustness, the dataset undergoes augmentation through various transformations such as rotation, flipping, and zooming.

4. **Selection of Model Architecture:**The selection of an appropriate deep learning architecture is pivotal for the grape counting task. This selection process is driven by a combination of greed-based criteria and the stability of the model, taking into consideration the inherent trade-off between bias and variance. The identified pipeline incorporates architectures like Convolutional Neural Networks (CNNs) and transfer learning models such as DenseNets and VGGNets. The most optimal architecture is subsequently deployed for real-time evaluation.

   (a) **Training Phase:** The selected deep learning model is trained using the designated training dataset. Refinement of hyper-parameters, including learning rate, batch size, optimiser, and loss function, is undertaken to achieve peak performance.

   (b) **Validation Phase:** The evaluation of the trained model occurs using the validation dataset. Metrics such as accuracy, precision, recall, and F1-score are closely monitored to ensure that the model avoids overfitting.

   (c) **Testing Phase:** Subsequently, the model's performance is assessed on the testing dataset to approximate its efficacy in real-world scenarios.

5. **Deployment Process:** The deployment of the trained model takes place on a cloud platform, with Amazon Web Services (AWS) as a prominent example. This involves the establishment of an API endpoint capable of accepting input images and generating an output count of detected grapes. Additionally, a rigorous assessment of various cloud servers culminates in the selection of the most optimal configuration for deployment.

6. **Ongoing Monitoring and Maintenance:** The final phase of the proposed methodology entails continual vigilance over the deployed model's performance. Regular updates to the model are introduced to accommodate new data and environmental shifts, ensuring its continued efficacy.

By adhering to this meticulously structured algorithm, the methodology achieves comprehensive grape image analysis through systematic data collection, preprocessing, model selection, deployment, and perpetual maintenance.

3.2 Research Resources – Data Science Models

Transfer learning is a machine learning and computer vision approach where a neural network model that was trained on a large dataset for a different purpose is fine-tuned using a smaller dataset that is related to the target task. To train a model for berry or fruit counting, you may take a model that has been pre-trained on ImageNet, remove the classification layer, and then train the model using a dataset consisting of images of berries labelled with their counts. One of the most important factors in evaluating whether or not transfer learning will occur is the similarity between the source and target activity. It works well when there is a limited quantity of data, but additional approaches may be needed if the tasks are too dissimilar.

Figure 3: Cherry Counting using Transfer learning Technique. The output will display how the berry or fruit has been detected

The solutions first identify the objects which in these cases are either berries or grape vines or fruits and draws a bounding boxes. As in the above Figure 3, the cherries are identified individually and marked a bounding box. These bounding boxes then gets counted and a statistical value is fixed based on the training of the model.

3.2.1 VGGNet

To use VGGNet for grape counting, we must first adopt the VGGNet pre-trained deep neural network architecture (Figure 4). It was with the ImageNet Large Scale Visual Recognition Challenge in mind that this architecture was first conceived. The approach of adapting VGGNet for the task of grape counting makes advantage of transfer learning. Obtaining a pre-trained VGGNet model, which is a version of the network already taught to detect features using a large-scale dataset like ImageNet, is the first stage in the process. The last layers of classification in the VGGNet have been replaced with a new layer optimised for regression. This adjustment allows the model to make predictions about continuous variables like the total number of grapes in a given image. A dataset for grape counting should comprise several different grape species, as well as different lighting, viewing angles, and backgrounds. Having reliable ground truth counts is crucial for both training and grading reasons. During fine-tuning, only the newly added regression layer is modified; all other pre-trained layers of the VGGNet remain static. The model is then retrained on the grape dataset, but only the output layer's weights are modified (the other layers' weights are left at their default values). To evaluate the model's berry-counting accuracy, it is applied to a separate validation or test dataset, where metrics like mean squared error and mean absolute error are used. The goal of this exercise is to improve the model's efficiency by adjusting its hyper-parameters. Post-processing techniques can be employed to further refine the output if necessary.

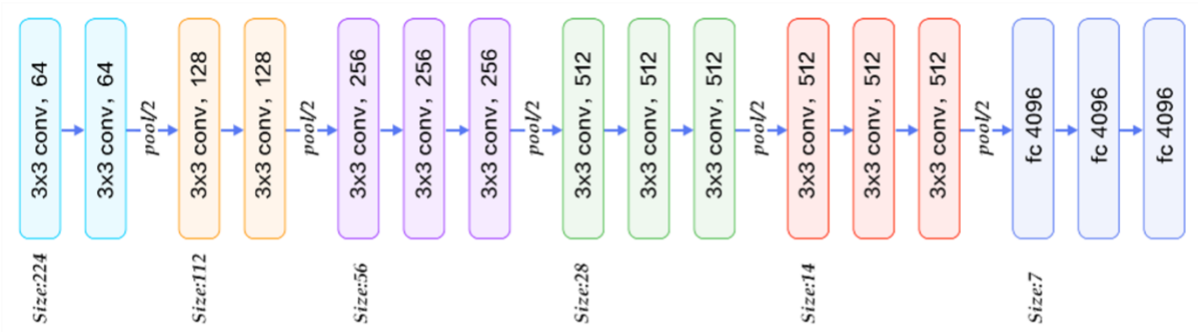

Figure 4: Architecture of VGGNet (VGG-16)

VGGNet and transfer learning make it feasible to do grape counting with only a little quantity of labelled data by reusing the knowledge from a much larger dataset. An accurate and efficient grape counting model might be constructed using this method as its basis.

3.2.2 MobileNet

MobileNets are a family of neural network architectures developed specifically for use in the field and in embedded systems (Figure 5). These designs are highly efficient and lightweight. Ordinary convolutions are broken down into depth-wise and point-wise convolutions to create depth-wise separable convolutions. This drastically reduces the amount of processing time and effort required, and simplifies the model without sacrificing anything in the way of accuracy. This makes MobileNets well-suited for low-power systems like smartphones and IoT gadgets. To get started using MobileNets for grapes counting, we may choose a model that has already been trained on a large dataset like ImageNet or COCO. If you do this, you may use MobileNets to tally the grapes. Using a model that has been pre-trained to distinguish generic traits from photographs can speed up the process of learning how to count grapes. A dataset of grapes image files with associated counts should be compiled for training purposes. Make sure the dataset includes several types of grapes, lighting conditions, and backgrounds so the model can respond to a wide range of settings. While tuning the MobileNet, only the weights of the newly added regression layer are updated; the weights of the other layers remain unchanged. Try with different values for the model's hyper-parameters like its learning rate, batch size, and epoch count to see what gives you the best results when you apply it to the grapes counting task. After the model has been fine-tuned, it should be tested on a separate validation or test dataset and its performance measured using metrics like mean squared error (MSE) or mean absolute error (MAE). This is crucial for evaluating the model's ability to anticipate fruit yields.

MobileNets is successful on mobile and embedded devices because of its small footprint. This allows for accurate grapes counting in real time with low computational requirements. Using MobileNets' transfer learning technique, you can capitalise on the insights obtained from massive datasets by fine-tuning the model to the narrow task of grapes counting using comparatively fewer labelled data. You'll be able to get the job done more rapidly as a result of this. In cases when resources are scarce, MobileNets may

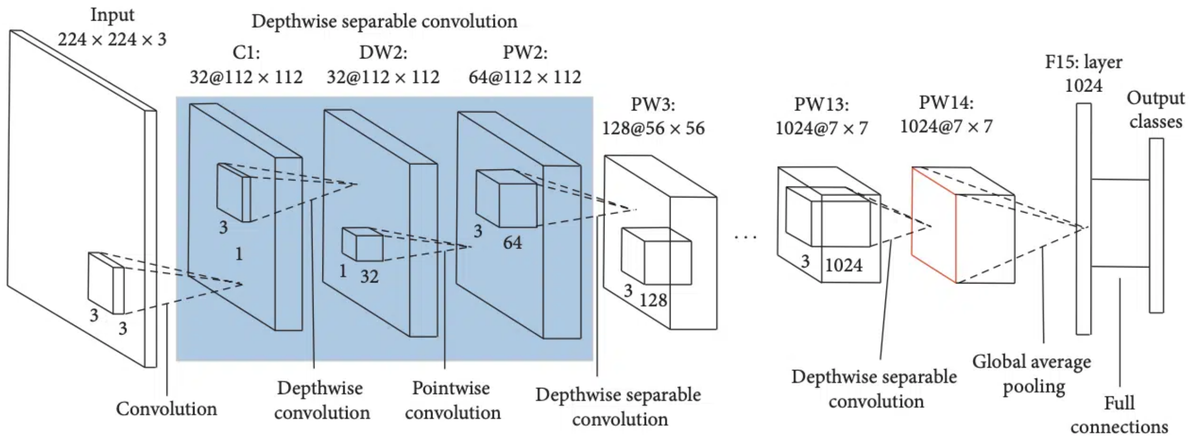

Figure 5: Architecture of MobileNets (Dense MobileNets)

provide an accurate and efficient solution for grapes counting applications; however, this assumes that the system has been thoroughly assessed and fine-tuned.

3.2.3 DenseNet

In this part, we show our work developing an automated grape counting system using the DenseNet deep learning architecture (Figure 6). Accurate grape counting is essential for many uses in agriculture and grape harvesting, and we want to address this issue as part of our research. To begin, we amassed a sizeable and diverse data set of grape images including a wide range of species, lighting conditions, and perspectives. Each image has been annotated with ground truth counts for the purposes of training and evaluation. To take advantage of transfer learning, we used a DenseNet model that had previously been trained on ImageNet. Because of this, we were able to conserve resources. Because of this, the model could begin training with the ability to recognise generic images, which sped up its education for the grape counting task. To arrive at an accurate forecast of the grape count, we replaced the final classification layer with a regression layer during the tuning phase. We put in a lot of work optimising the model's hyper-parameters like learning rate, batch size, and number of epochs to get the best possible results. The evaluation was carried out using the performance measures of mean squared error (MSE) and mean absolute error (MAE). Our trials showed that the DenseNet-based method was effective, as it yielded grape count estimations that were highly consistent with manual counts.

Our implementation of the grape counting system with DenseNet gives a practical means of automating the necessary grape counting procedures. Our research has the potential to shed new light on the intersection of agricultural automation and computer vision. More progress will be made in grape counting and other areas as we explore new research directions, such as exploring new architectures and methods for augmenting data. Those are only a couple such instances.

3.2.4 ResNet

ResNet, an architecture for deep neural networks (Figure 7), to carry out the robotic grapes counting. Accurate and efficient techniques for counting grapes are the focus of

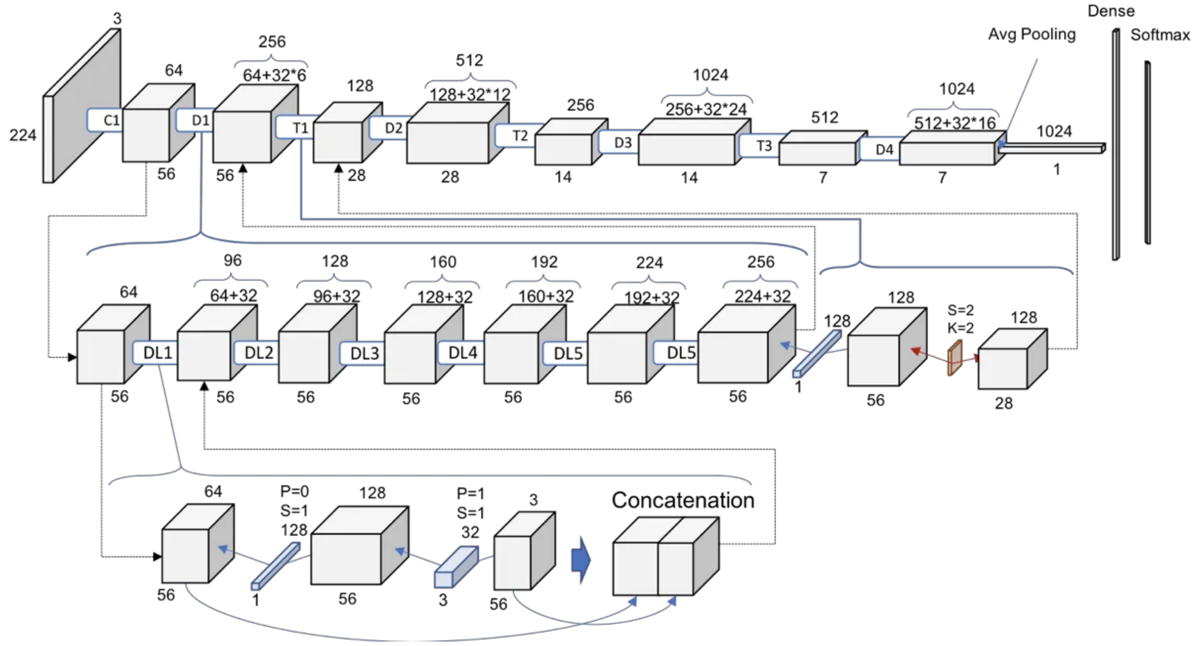

Figure 6: Architecture of DenseNet

this study; such counting is useful in many areas of agriculture and grapes harvesting. The study used a large and diverse dataset of grapes image pairings, each of which had associated ground truth counts for use in training and evaluation. ResNet's residual learning framework will be used to finish the task because of its efficacy when dealing with deeper designs. To speed up the model's fine-tuning process, a pre-trained ResNet model was modified using transfer learning so that it could make use of characteristics it had already learnt. ImageNet was used to first train the model. In order to create a precise forecast of the grapes count, we swapped out the final classification layer with a new regression layer while keeping the pre-trained layers fixed. Several hyper-parameters, such as the learning rate, batch size, and epochs, were tweaked to get the greatest performance out of the model.

In conclusion, the ResNet-based approach offers a potentially valuable choice for automated grape counting. This study has important implications for agricultural automation and computer vision, and it may find further use in other counting tasks. The potential for growth and new discoveries in grape counting and related topics might be explored in future research.

3.2.5 Fully Connected Layers

Here, we'll go through how to use Transfer Learning to choose features for a three-layer ANN (Figure 8) trained on regression data in order to estimate grape counts. This approach aims to improve grape count forecasts by making use of the knowledge gathered from a pre-trained deep learning model.

Using a three-tiered artificial neural network (ANN) and Transfer Learning for feature selection, we hope to create a reliable grape count prediction model based on regression. The strategy uses a deep learning model that has already been trained to increase feature representation and prediction accuracy.

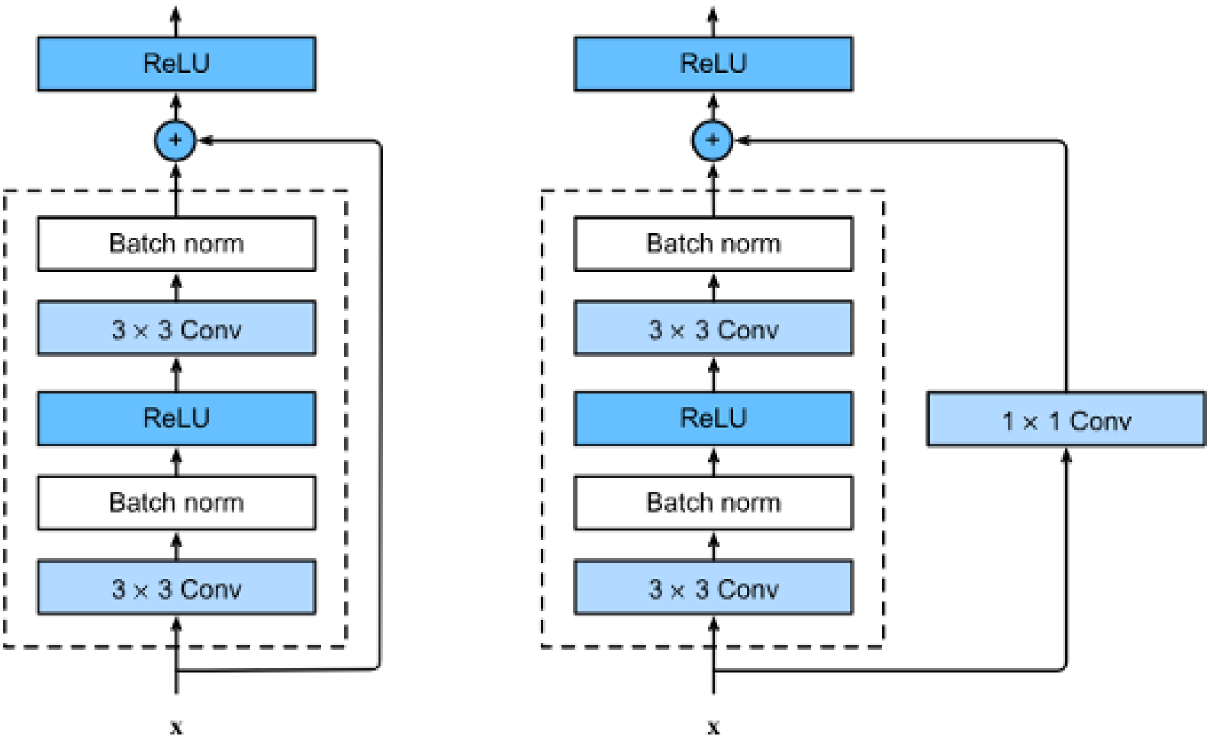

Figure 7: Architecture of ResNet

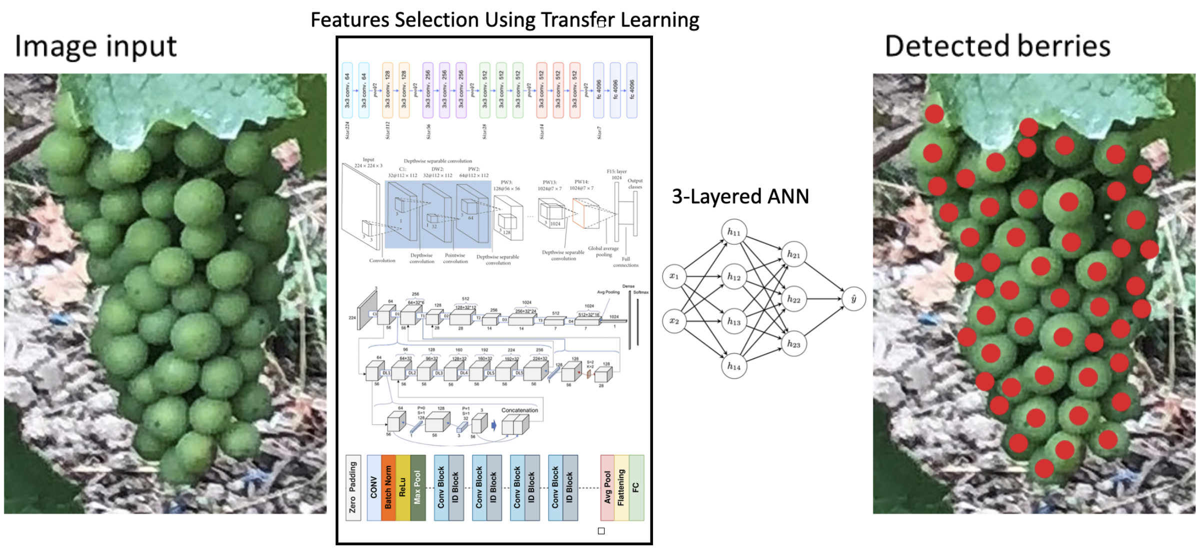

Figure 8: ANN for the Fully Connected layer

3.3 Research Resources – Cloud Computing

The process of utilising AWS for grape counting involves the following sequential stages (Figure 9):

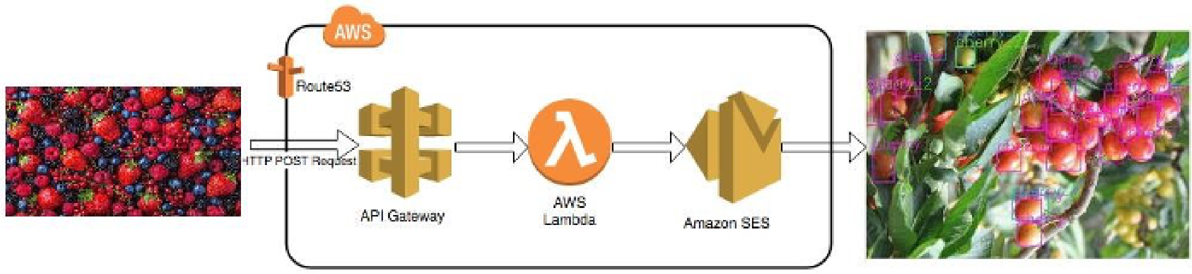

Figure 9: ANN for the Fully Connected layer

1. **Image Collection:** The initial step entails amassing an extensive dataset of grape images encompassing diverse angles, lighting scenarios, and backgrounds.

2. **Image Upload to S3:** Post data collection, the acquired images can be seamlessly transferred to an Amazon S3 repository. This facilitates effortless accessibility of images through various AWS services.

3. **EC2 Instance Setup:** Subsequently, an Amazon EC2 instance is established to execute the deep learning model. The choice of instance type is contingent upon the computational prerequisites of the model.

4. **Library Installation:** Following instance activation, indispensable libraries and dependencies, such as TensorFlow, Keras, and OpenCV, are installed.

5. **Model Training:** With the essential libraries in place, the YOLO model is trained utilising the grape image dataset. This procedure can be executed using AWS Deep Learning AMIs, which offer pre-configured environments tailored for deep learning tasks.

6. **Result Storage:** The culmination of this process involves the retention of grape counting outcomes. This can be accomplished by storing the results in either an Amazon RDS database or an Amazon DynamoDB NoSQL database, affording the opportunity for subsequent in-depth analysis.

7. **Comprehensive Solution:** Collectively, the utilisation of AWS for grape counting presents a scalable and cost-effective solution for efficiently processing substantial volumes of image data.

In summary, leveraging AWS for grape counting offers an adaptable and economically efficient resolution for handling substantial quantities of image data.

4 Implementation

4.1 Data Acquisition

The dataset[2] utilised in this study, as illustrated in Figure 10, encompasses a collection of image filenames (denoted as 'ImgName') paired with their corresponding counts of berries (referred to as 'berryCount'). This dataset captures quantities of berries or fruits depicted in diverse images, rendering it amenable for tasks involving image-based regression or classification focused on estimating berry or fruit counts. Exemplary representations of images within the dataset are provided below.

```
data = pd.read_csv('count - count.csv.csv')
data.head()
```

	ImgName	berryCount
0	20170227_130321_HDR.jpg	15
1	20170227_130555_HDR.jpg	25
2	20170227_130817_HDR.jpg	35
3	20170227_131334_HDR.jpg	45
4	20170227_133249_HDR.jpg	51

Figure 10: Dataset Description

As exemplified in Figure 11, images of grapes have been selected for training the models. A comprehensive analysis of the histogram (Figure 12)unveils that the majority of observations indicate a count of 75 grapes, albeit there are a few instances of outliers with notably higher counts. The statistical metrics of the dataset reveal that the mean and median grape counts are 100 and 125, respectively. However, it is the mode grape count (75) that offers a more accurate measure of central tendency.

4.2 Data Pre-processing

The data pre-processing stage entails a critical series of operations to ensure the suitability of the input data for subsequent modeling procedures. The designated function, denoted as 'preprocess_image', is devised to undertake these preparatory tasks. Given an image path, the function expertly addresses potential file extension inconsistencies. Subsequently, the image is loaded and resized to dimensions of (224, 224), followed by conversion to an array format. Adjustments to dimensions are executed, and VGG16-specific preprocessing is applied. The culmination of these operations yields the preprocessed image, optimised for downstream analysis.

In line with the pre-processing protocol, the implemented code iterates through the 'data' DataFrame, extracting the 'ImgName' and 'berryCount' values for each entry. The complete image path is constructed, and the image is subjected to the 'preprocess_image' function. The preprocessed image is then appended to the set denoted as 'X', while the corresponding berry or fruit count is appended to the set 'y'.

[2]Dataset: `https://periakiva.github.io/finding_berries/datasets.html`

Figure 11: Images of the grapes which will be rendered as the streaming video through API

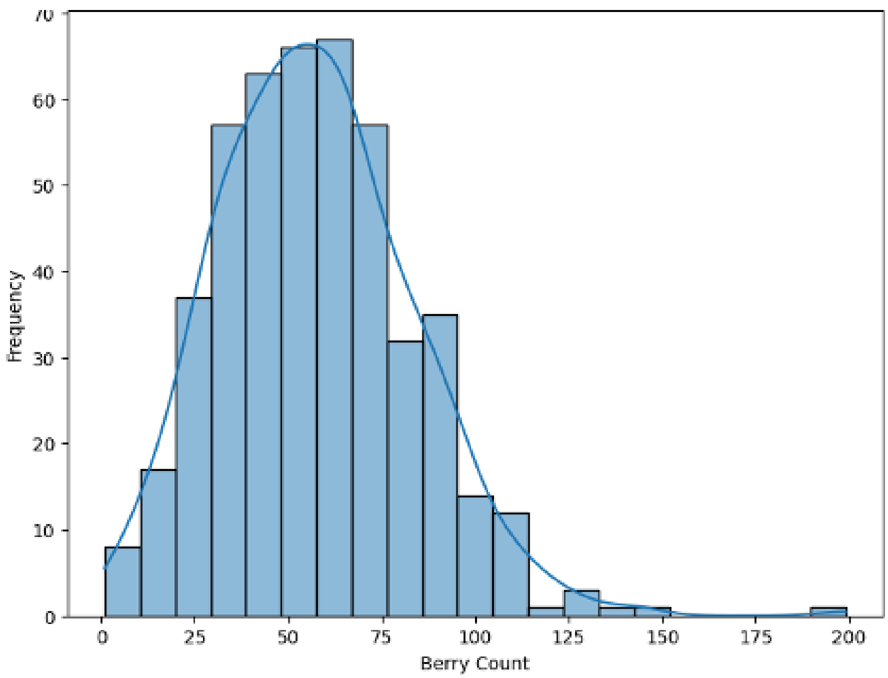

Figure 12: Distribution of the data

Furthermore, the pixel values of the training and evaluation data, represented by 'X_train' and 'X_eval' respectively, undergo a normalisation procedure. This normalisation involves scaling the pixel values to reside within the range of [0, 1], which is achieved by dividing the pixel values by 255.

To facilitate model training and evaluation, the dataset (X, y) is partitioned into distinct training and evaluation subsets, denoted as (X_train, y_train) and (X_eval, y_eval), respectively. This partitioning is executed using an 80-20 split ratio, and a consistent random state is employed to ensure replicability. Recognising the importance of data augmentation in scenarios characterised by limited training data, the code integrates data augmentation techniques. These techniques introduce diversity into the training dataset by applying transformations such as rotation, shifting, flipping, and more. The augmentation process diversifies the dataset, enabling the models to learn from a wider array of examples, thereby enhancing generalisation and mitigating overfitting. To operationalise this, the code configures an image data generator utilising the Keras 'ImageDataGenerator' class. The augmentation techniques include rotation, width and height shifts, shearing, zooming, horizontal flipping, and filling mode adjustments. The integration of these augmentations augments the richness of the training dataset, thus fostering improved model generalisation.

4.3 Modelling Implementation

The core of the modeling phase involves the construction of a convolutional neural network (CNN) architecture tailored to the demands of the problem domain. The model is designed to undergo multiple layers of convolution followed by max-pooling, aimed at extracting discriminative features. The process culminates with global average pooling, which facilitates spatial aggregation. The architecture also encompasses two fully connected (dense) layers, fortified with regularisation mechanisms, alongside a dropout layer to counteract overfitting effects. The ultimate dense layer generates a single output value. The model configuration is oriented towards image classification tasks, characterised by an input shape of (224, 224, 3).

The model architecture comprises sequential stacking of various layers, each contributing to the intricate process of feature extraction and classification. Key components encompass a base model, a flatten layer, a dense layer equipped with regularisation and dropout strategies, and the final dense layer responsible for generating the ultimate output. This template serves as the foundation for subsequent model iterations, albeit with adjustments in initialisation specifics to accommodate the desired model variations.

5 Evaluation

In this research study, a comprehensive evaluation of various convolutional neural network (CNN) architectures is conducted using three distinct evaluation metrics: Intersection over Union (IoU), Mean Average Precision (mAP), and Mean Squared Error (MSE). These metrics have been selected to holistically assess the performance of the models in object detection tasks while considering accuracy, precision, and convergence. **Intersection over Union (IoU)** is utilised to quantify the degree of overlap between predicted bounding boxes and actual ground truth boxes. This measure captures the accuracy of the spatial localisation of objects within the images. The IoU score is calculated by dividing the intersection area by the union area of the bounding boxes. **Mean Average**

Precision (mAP) is a widely accepted metric that balances precision and recall across different IoU thresholds. This metric provides insight into the models' ability to accurately detect objects of varying sizes and spatial configurations. By calculating the average precision over different IoU thresholds and taking the mean across all object classes, mAP offers a comprehensive view of detection performance. **Mean Squared Error (MSE)** is employed to quantify the extent of deviation between the predicted object counts and the actual counts. This metric allows an assessment of the models' accuracy in predicting the number of objects present in an image. The choice of evaluation metric depends on the specific task requirements, including the desired accuracy level and the trade-off between precision and recall. By leveraging these metrics, the research seeks to provide a thorough understanding of the models' capabilities and limitations in object detection scenarios. The study conducts experiments using six different CNN architectures: Basic CNN, Shallow CNN, DenseNet, MobileNet, ResNet, and VGGNet. Each architecture is evaluated based on the aforementioned metrics to discern their strengths and weaknesses. Written Code loads images from the "data/data" directory, randomly selects 5 images, and displays them in subplots. For each image, it predicts using a model and chooses green or red based on the prediction threshold. It adds a colored box at the bottom with a predicted value in white text. The chosen color represents prediction confidence, and the text is centered within the box. Subplot titles are set to image filenames, and axes are turned off for cleaner display.

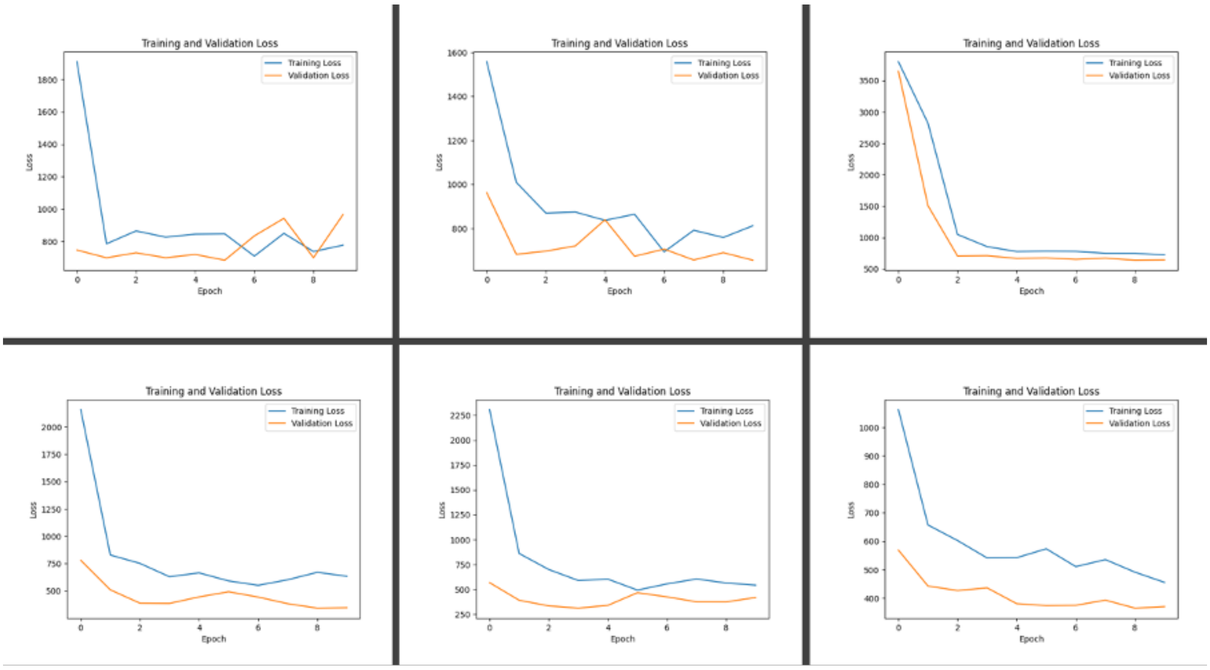

Figure 13: From left to right respectively: Training and Validation loss of Basic CNN, Shallow CNN, DenseNet, MobileNet, ResNet and VGGNet

5.1 Experiment / Case Study 1: Basic CNN

The training performance of the Basic CNN demonstrates erratic behavior in loss (Figure 13), and validation loss remains persistently high (Figure 14). This suggests that the model struggles to converge efficiently and generalise to unseen data (Figure 15).

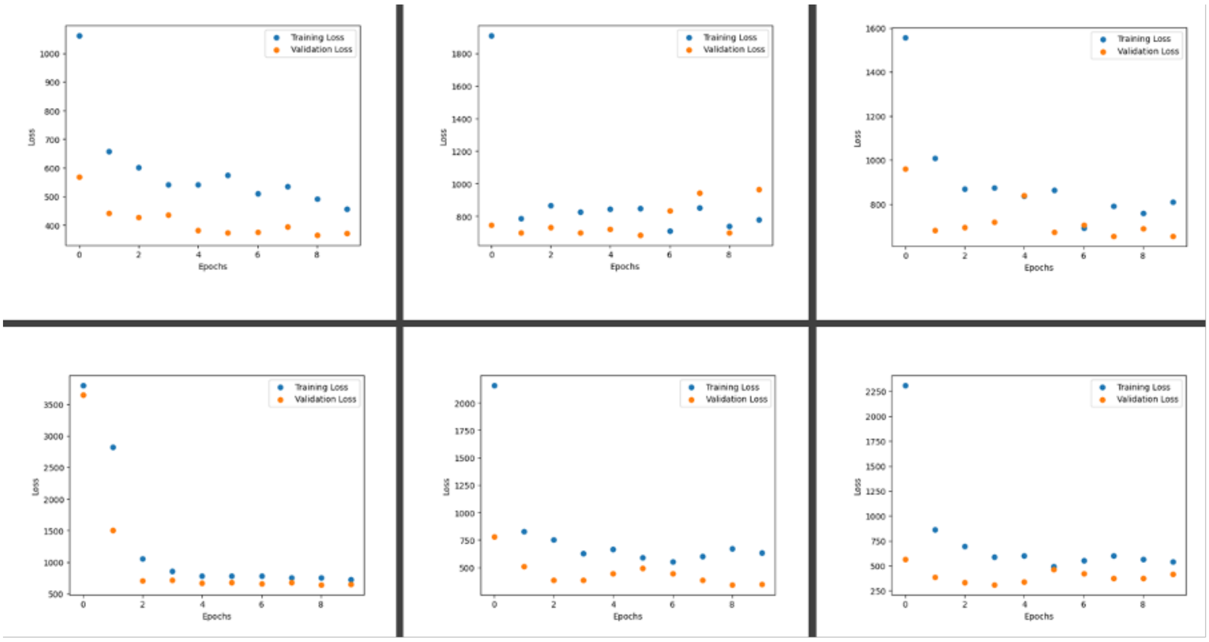

Figure 14: From left to right respectively: Prediction Scatter Plot loss of Basic CNN, Shallow CNN, DenseNet, MobileNet, ResNet and VggNet

Figure 15: Prediction of Basic CNN

5.2 Experiment / Case Study 2: Shallow CNN

Figure 16: Prediction of Shallow CNN

The Shallow CNN exhibits a training pattern of gradually decreasing loss (Figure 13)with relatively consistent validation loss trends (Figure 14). This points towards better convergence and generalisation capabilities compared to the Basic CNN (Figure 16).

5.3 Experiment / Case Study 3: DenseNet

Figure 17: Prediction of DenseNet

The DenseNet architecture presents fluctuating performance trends, with training loss exhibiting variations lacking a clear pattern (Figure 13). Validation loss similarly varies (Figure 14), indicating potential challenges in achieving stable convergence and generalisation (Figure 17).

5.4 Experiment / Case Study 4: MobileNet

Figure 18: Prediction of MobileNet

The MobileNet model displays fluctuating trends in performance, but with decreasing loss during training (Figure 13). Validation loss also exhibits variations but generally follows a decreasing trajectory (Figure 14). This suggests a comparatively stable convergence and generalisation process (Figure 18).

5.5 Experiment / Case Study 5: ResNet

Figure 19: Prediction of ResNet

The ResNet's training performance shows decreasing loss over epochs (Figure 13), while validation loss experiences initial decrease followed by subsequent increases (Figure 14). This hints at the possibility of overfitting to training data and insufficient generalisation (Figure 19).

5.6 Experiment / Case Study 6: VGGNet

Figure 20: Prediction of VGGNet

The VGGNet's training performance showcases decreasing loss over epochs (Figure 13), indicating enhanced performance. Validation loss also demonstrates a decreasing pattern (Figure 14), suggesting better generalisation capabilities (Figure 20).

5.7 Discussion

From an academic perspective (Table 1), the research underscores the importance of selecting appropriate CNN architectures for object detection tasks. The findings emphasise that while certain models exhibit promising training trends, their ability to generalise to unseen data varies significantly. These insights can guide future research into refining CNN architectures for improved convergence and robust generalisation.

From a practitioner standpoint, the study offers valuable guidance for selecting CNN architectures based on task-specific requirements. By understanding the strengths and weaknesses of each architecture, practitioners can make informed decisions to achieve optimal object detection performance in real-world scenarios.

Table 1: Performance Comparison.

Model Used	R^2 Score	Mean Squared Error
Basic CNN	-0.42	965.96
DenseNet121	0.49	345.43
MobielNet	0.38	415.74
ResNet50	0.037	654.66
VGG16	0.45	369.81

5.8 Answering Research Questions

- **Research Question 1 (RQ1):** Could it be feasible to enhance the efficiency and precision of grape detection and counting in agricultural contexts by refining deep learning architectures, including semantic segmentation and object detection algorithms?

  Answer: Deep learning models employing transfer learning architecture in conjunction with Fully Connected layers aim to automate grape counting, thereby minimising the need for manual labor. These deep learning models will take the form of regression models, initially segmenting individual grapes on vines and subsequently determining the identified bounding boxes. The cumulative count of grapes will be derived from this process. This approach is versatile, as the pipeline functions smoothly across various counting scenarios, including:

  Grapes: Tested
  Population: Tested
  People: Tested

- **Research Question 2 (RQ2):** To expedite fruit counting and yield estimation, what measures can be taken to maximise the effective utilisation of unmanned aerial vehicles (UAVs)? Furthermore, what are the key technical obstacles that necessitate resolution to ensure results that are both accurate and dependable?

  Answer: Within this context, the proposed resolution involves developing an API using Flask (Figure 2), serving as the current framework, which can be adaptable to alternative technologies for seamless integration with the data acquisition process. This API will facilitate automated grape quantity calculations. In cases where the automated calculation falls short of accuracy, manual intervention will be implemented selectively, establishing a feedback loop essential for enhancing the learning process of the model.

- **Research Question 3 (RQ3):** What variables wield substantial influence over the precision and accuracy of algorithms employed for automated grape counting? How can the potential issues stemming from these variables be alleviated to foster dependable outcomes across a diverse array of environments and grape categories?

  Answer: The crucial elements to consider are:

  1. The type of data, including its source from various plantations. This is significant because the specific plants present can affect the detectability of grapes by the automated image acquisition platform.

2. The choice of device employed for conducting the image acquisition process.

$$Error = \sum_{i=1}^{n}(Actual_i - Forecast_i)$$

Error will function as a multiplicative factor.
Final Prediction = α * Actual Prediction
Where α is directly proportional to Error.

- **Research Question 4 (RQ4):** To streamline the management of extensive datasets produced by automated grape detection and counting systems, how can cloud-based processing be optimised? Additionally, what strategies can be employed to safeguard data privacy and uphold security standards within cloud-based environments?

  Answer: Local computer lacked the capacity to effectively train the expansive architectures of transfer learning models. Consequently, a more capable solution was needed, and this was successfully addressed through the utilisation of both Google Cloud Platforms and AWS platforms. Additionally, given the substantial volume of real-time incoming data, a robust storage solution was imperative. This requirement is met by leveraging AWS storage. Furthermore, for efficient model training, GPU-based servers were employed, significantly enhancing model computations and real-time predictions. These servers also facilitate the creation of endpoints, which can be seamlessly integrated into numerous systems, enabling parallel processing. To ensure data security, a distinct login is assigned to each user, preventing data sharing among users and thus safeguarding the data.

6 Conclusion and Future Work

In summary, this study underscores the importance of meticulous assessment of CNN architectures through comprehensive metric analysis. The systematic juxtaposition of model performance enhances the collective comprehension of object detection techniques, facilitating progress in both academic and practical contexts.

Among the various neural network architectures examined, the DenseNet model exhibited superior performance when deployed within AWS's real-time cloud infrastructure. Its successful application to grape counting not only showcases its remarkable precision in detecting and quantifying grapes, but also accentuates the capacity of cloud computing to bolster scalability and productivity.

The synergistic integration of DenseNet and AWS embodies cutting-edge advances in amalgamating machine learning and cloud technologies, marking a pivotal stride in optimising agricultural methodologies. This advancement holds substantial potential for accurate yield estimation, quality control, crop management, harvest optimisation, cost reduction, time efficiency, and research refinement. The promising trajectory of precision agriculture shines brighter as we harness the synergy of artificial intelligence and cloud computing. This symbiosis promises refined resource allocation and precise yield prognostication, nurturing berries and other crops season after season. The continuous accumulation of dataset images captured by drones or comparable means will continually refine upcoming models, forging a collective solution for the benefit of farmers in developing nations.

However, commercialisation of this solution is not the optimal route. Instead, it is advisable for the government to sustain and support the resources as an open-source service, catering to the needs of economically disadvantaged farmers. This strategy ensures perpetual enhancements and prevents exploitation stemming from inadequate computer literacy. Given that computations are only required thrice annually, the government can efficaciously leverage existing cloud infrastructure like AWS for remote execution, rather than investing in on-premises systems.

In terms of future endeavours, an intuitive web application stands as an extension of this project, disseminating automated and efficient grape counting results to farmers via SMS or local community centres. With an expanded dataset, the project's horizon extends toward identifying fungi that curtail yields by a third during winter seasons, promising further agricultural advancement.

References

Abbas, S. H., Vashisht, S., Bhardwaj, G., Rawat, R., Shrivastava, A. and Rani, K. (2022). An advanced cloud-based plant health detection system based on deep learning, *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, IEEE, pp. 1357–1362.

Barreto, A., Lottes, P., Yamati, F. R. I., Baumgarten, S., Wolf, N. A., Stachniss, C., Mahlein, A.-K. and Paulus, S. (2021). Automatic uav-based counting of seedlings in sugar-beet field and extension to maize and strawberry, *Computers and Electronics in Agriculture* **191**: 106493.

Buayai, P., Saikaew, K. R. and Mao, X. (2020). End-to-end automatic berry counting for table grape thinning, *IEEE Access* **9**: 4829–4842.

Chen, R., Zhang, C., Xu, B., Zhu, Y., Zhao, F., Han, S., Yang, G. and Yang, H. (2022). Predicting individual apple tree yield using uav multi-source remote sensing data and ensemble learning, *Computers and Electronics in Agriculture* **201**: 107275.

Cruz, M., Mafra, S., Teixeira, E. and Figueiredo, F. (2022). Smart strawberry farming using edge computing and iot, *Sensors* **22**(15): 5866.

Jiang, Q., Huang, Z., Xu, G. and Su, Y. (2023). Miop-nms: Perfecting crops target detection and counting in dense occlusion from high-resolution uav imagery, *Smart Agricultural Technology* **4**: 100226.

Junos, M. H., Mohd Khairuddin, A. S., Thannirmalai, S. and Dahari, M. (2021). Automatic detection of oil palm fruits from uav images using an improved yolo model, *The Visual Computer* pp. 1–15.

Kang, H. and Chen, C. (2020). Fast implementation of real-time fruit detection in apple orchards using deep learning, *Computers and Electronics in Agriculture* **168**: 105108.

Lyu, S., Li, R., Zhao, Y., Li, Z., Fan, R. and Liu, S. (2022). Green citrus detection and counting in orchards based on yolov5-cs and ai edge system, *Sensors* **22**(2): 576.

Mekhalfi, M. L., Nicolò, C., Ianniello, I., Calamita, F., Goller, R., Barazzuol, M. and Melgani, F. (2020). Vision system for automatic on-tree kiwifruit counting and yield estimation, *Sensors* **20**(15).
URL: *https://www.mdpi.com/1424-8220/20/15/4214*

Ponce, J. M., Aquino, A., Millan, B. and Andújar, J. M. (2019). Automatic counting and individual size and mass estimation of olive-fruits through computer vision techniques, *IEEE Access* **7**: 59451–59465.

Roy, K., Chaudhuri, S. S. and Pramanik, S. (2021). Deep learning based real-time industrial framework for rotten and fresh fruit detection using semantic segmentation, *Microsystem Technologies* **27**: 3365–3375.

Tu, S., Pang, J., Liu, H., Zhuang, N., Chen, Y., Zheng, C., Wan, H. and Xue, Y. (2020). Passion fruit detection and counting based on multiple scale faster r-cnn using rgb-d images, *Precision Agriculture* **21**: 1072–1091.

Vijayakumar, V., Costa, L. and Ampatzidis, Y. (2021). Prediction of citrus yield with ai using ground-based fruit detection and uav imagery, *2021 ASABE Annual International Virtual Meeting*, American Society of Agricultural and Biological Engineers, p. 1.