Grape Cluster Detection in Grape Farm

Author

Muhammad Osama Shahzad

00000273684

Supervisor

DR ANAS BIN AQEEL

DEPARTMENT OF MECHATRONICS ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

JULY, 2022

Grape Cluster Detection in Grape Farm

Author

Muhammad Osama Shahzad

00000273684

MS-18

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Mechatronics Engineering

Thesis Supervisor:

DR ANAS BIN AQEEL

Thesis Supervisor's Signature:

_____

DEPARTMENT OF MECHATRONICS ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

JULY, 2022

Declaration

I endorse that this research work titled "*Grape Cluster Detection in Grape Farm"* is my own work. The material used in this work from other sources has been properly referenced and it hasn't been presented elsewhere for assessment.

Signature of Student

Muhammad Osama Shahzad
00000273684
MS-18 (Mts-E)

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student

Muhammad Osama Shahzad

00000273684

Signature of Supervisor

Dr. Anas Bin Aqeel

Copyright Statement

Acknowledgements

I am thankful to my Creator **Allah Ta'alaa** to have guided me throughout this work at every step. Indeed, I could have done nothing without **Allah's** priceless support and guidance. Whosoever helped me during the course of my thesis, was **HIS** will, so indeed none be worthy of praise but **HIMSELF** alone.

I am indebted to my mother for supporting me through my darkest times and standing by my side like a beam of light, praying for me and pushing me throughout to work and get through this journey. I am thankful to my late father who gave me life lessons and brought me up, giving me freedom to choose my paths and follow them.

I would also like to express special thanks to my supervisor Dr. Anas Bin Aqeel for creating a passion in me to opt for the field of Robotics and helping me throughout my thesis. Also, he guided and motivated me to enhance my research for the betterment of Robotics community. I am grateful to my previous supervisor, Dr. Waqar Shahid Qureshi who helped me to choose this topic and guided me patiently, helping me turning my weaknesses into strengths. I pay my gratitude to Dr. Hamid Jabbar and Dr. Moshin Islam Tiwana for their tremendous support, guidance and cooperation. Without their help, I wouldn't have been able to complete my thesis.

Finally, I would like to express my thankfulness to all the characters who have rendered valuable assistance to my study and supported me in other possible ways including Saad Ahmed, Faheem Khan, Rizwan Ur Rehman, Wajahat Ali, Ahsan Mehdi and others.

Dedicated to my late father

Abstract

Convolutional Neural Networks and Deep Learning has revolutionized every field since their inception. Agriculture, like all other fields, has also been reaping fruits of developments in mentioned fields. Grapes are one of highest profit yielding and most important fruit related to the juice and wine industry and even dry fruits in form of raisins. The biggest challenge in harvesting grape fruit till date is to detect its cluster successfully. Grape is available in different sizes, colors, seed size and shapes which makes its detection, through simple Computer vision, even harder. Thus, this research addresses this issue by bringing the solution to this problem by using CNN and Neural Networks. A dataset was gathered from a grape farm which consisted of multiple different classes, colors and sizes of grape pictures taken in multiple conditions. It was split in 80/20 format making the larger chunk training dataset while test set consisted of 20% of the data. This dataset was carefully annotated and then fed to a powerful CNN based architecture called YOLO. YOLO is written in Darknet and is a very powerful architecture especially for Image detection. The custom dataset was trained on this architecture and multiple models were created with accuracy ranging from 86%-92%.

Key Words: Grapes, Convolutional Neural Network, YOLO

Table of Contents

List of Figures

Nomenclature

Artificial Intelligence:	AI
Deep Learning:	DL
Machine Learning:	ML
Machine Vision:	MV
You Only Look Once:	YOLO
Convolutional Neural Networks:	CNN
Deep Neural Networks:	DNN
Single Shot Multibox Detector	SSD
Root Mean Square Error	RMSE
Millions	mil
Hectare	ha

CHAPTER 1

INTRODUCTION

Grape is one of the most important fruit in the world with huge profitability and usage. In Botanical terms, grape is considered a berry. It is one of the very few fruits that grows in cluster of 15 to 300. Grapes different colors include red, green and purple; grapes can be seedless, with big seeds and have wide variety of flavor ranging from different degrees of sweet and sour. Grape juice is used in cooking to enhance *umami*. Grapes are used to extract fruit juice, wine and also consumed as toppings, jams, vinegar, grapeseed oil and raisins. Grapes are 81% water and 16% carbohydrates, have negligible fats and a percent of protein and also dietary fiber, which is important part of everyday diet. Grapes are a good source of vitamin C and K. Grapes are cultivated globally at 7 mil ha which makes it one of the leading fruit. Its total production in 2016 was 77.4 mil tones (valued at $68.3 billions) [1]. Red grapes are a major source of resveratrol. Resveratrol has chemo-preventive and therapeutic properties. It is useful in controlling diabetes and has been linked to reduced colon cancer [2]. To improve the grape fruit quality, green shoot thinning is carried out in vineyards where the weaker shoots and fruits are discarded so the vine may support the growth of high quality fruit. This process is also being automated to reduce the labor cost significantly from $650/ha. to $25/ha but the variation in efficiency of shoot removal ranges from 10-85% [3].

Pakistan being an agriculture based economy and a region suitable for grapes cultivation has a huge potential for grape production and not just earn by exports but can also use it to setup and develop its sister industries. By doing so, we will be bringing cash to the farmers which will result in further progression of agriculture sector. Globally Pakistan is ranked 56^{th} in term of production and 96 in term of export [4]. There is a huge potential for it to increase grape productivity by focusing on increasing its yield per ha. Right now Pakistan's yield is just 37% of the average of global yield per hectare [4] and the rate of increase of production in Pakistan is also much lower than the global average. Thus, to increase the grape farming yield and develop local automated solutions it is needed to develop indigenous models that take input from local farms and perform faster and better in the resident environment.

There is a need to develop highly sensitive, fully automated, vision based solutions at home that have applications in crop harvesting, automated spraying, yield estimation and early disease detection etc. Importing such high tech machines from developed countries costs

precious reserves moreover such machines are tested in those countries farms thus their efficiency can vary when used in different conditions. To develop such machines at home, there is a need to first research on the object detection and validating the results. By creating such high tech products, we can not only increase the production at home but also export the finished products in other developing nations.

A lot of research is being carried out relating to grapes and other fruits, mainly to increase their productivity, enhance their taste, approximate the yield and map the fruits using monocular camera and CNN [5]. It's an ancient fruit and archaeological remains suggest that mankind started growing grapes as early as 6500 B.C. [2]. Grape is botanically a berry and is consumed on a wider scale.

1.1 Motivation

With the rapid progress in computing power, field of AI and Deep Learning are touching new heights. Much complex neural networks are being trained and more research is being performed in the world at a rapid pace. World is changing much faster and AI has seeped in a common man's life. It has penetrated in our homes from our mobile phones to Industries and agriculture.

In agriculture, AI and Deep Learning is the forerunner and helping scientists to tackle challenges of food storage, food production and disease management. Population is expected to surge to 250 billion by 2050 and to meet their demands, 70% of increase in food production is needed [6]. In order to increase the crop production and meet rising food requirements, Artificial Intelligence will be the perfect companion to rely on.

Grape has the honor of being the third most valuable crop globally after potatoes and tomatoes [1]. It is used in extracting juice, being fermented to wine and brandy, majorly. Pakistan is a grape exporter but there is a huge untapped potential here to increase production and not just get more revenue but also generate more jobs by setting up juice industry and other related industries. Pakistan can potentially earn billions by increasing its grape production per hectare in the already existing farms. To increase its production, work on the cluster detection is needed. This thesis address the issue of cluster detection in grapes and focuses on increasing the dataset and compare its results with existing models.

Pakistan is among the top three countries that are severely affected by global warming and extreme weather condition according to CRI 2021 report [7]. The average temperature is

increasing in this region which calls for taking drastic measures in every field of life. Pakistan, being an agriculture nation, depends heavily on the products it produce to increase the crop production, Pakistan seek help from the researches being carried out in Europe and other developed countries. But the increasing difference in temperature creates an even more need for original research in home conditions.

In this context, a huge data is needed to be gathered and trained using a reliable and working model with satisfactory performance parameters. This has been the primary motivation to detect the cluster so that it can be further used in future in cluster harvesting, to spray pesticides and to only take pictures of grape clusters automatically in grape farm environment to be observed by the agriculture engineers or surveyors. Data can be used from the already available datasets but the motivation to create a newer dataset comes from the fact that the available data is from the farms in Europe or South America and doesn't represent the local environment. Thus there exist a need to create a dataset from scratch that truly represents the local conditions (i.e. lighting, leaves cover, pruning practices etc.).

1.2 Objectives

This research pursues a few major and multiple minor objectives related to data, object detection and Deep Learning. The major objective regarding data was to collect an unbiased data that is enough in numbers to not let the model face the phenomenon of underfitting.

Most of the already available datasets are biased, implicating that the images are taken in a way to increase the accuracy and tailor the results to perform better in certain conditions. To collect such a dataset, one use a single lens to capture images, use identical camera settings, take images at a matching angel and keep the distance from object of interest and camera same, By following the set principles, the model is able to converge soon and accuracy shoots up but such models perform poorly at the unseen data, so in this thesis an opposite approach is being taken regarding data by not following any such practice and take images by following an open ended approach.

Other major objective was to train the model with not just high accuracy but also high precision and recall (i.e. high F1 score) to make it actually viable to be used in industrial environment. Most of the researches only follow the high accuracy approach without doing much to improve F1 score which makes that research difficult (sometimes impossible) to be implemented in the real-time environment. So, this research will discuss the precision and recall relation in details and also provide a viable solution to their trade-off. Apart from the

mentioned objectives, the combined goal of this research is to create a deep learning model that is able to perform well in multiple environments and is able to integrate well with other frameworks.

In Pakistan, the grape farms conditions are different from the farms in Europe and other countries as Pakistan lies in one of the regions severely affected by environment change and global warming. Thus to work in this environment it is needed to develop solutions at home. One of the objectives is to develop a local object detection system on which further research (e.g. berries harvesting, yield detection etc.) for local farms can be carried out.

1.3 Problem Identification

Following is the problem statement this research intends to provide answers to:

"Training an indigenous, reliable, fast and operational Machine learning model to detect the maximum clusters in multiple occasions is the need of the time and this problem needs to be addressed for making any further advances in a grape farming."

To improve the process of cultivation and spraying on grapes clusters, newer algorithms and researches are needed to be implemented on a generalized data that is collected not only to

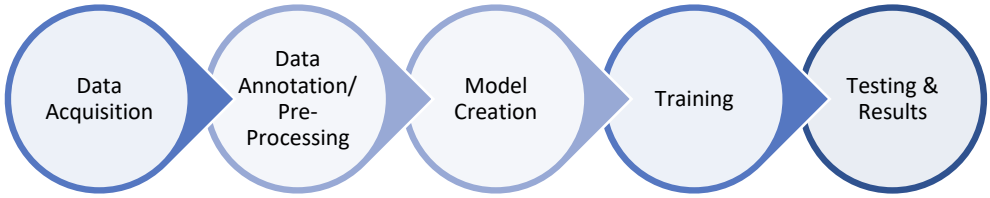


Figure 1 Generalized Block Diagram for Grapes Cluster Detection

improve the accuracy but also must be feasible to be implemented in the actual grape farm environment.

For this we need a dataset that is generalized with data collected from multiple sources and under many different conditions. A generalized block diagram, showcasing this whole process, is shown in Figure 1.

1.4 Research Background

Machine Learning and Deep Learning are the developing fields of the present that are greatly impacting our surroundings directly and also indirectly. It has changed the way we see things today and has affected every single part of current lives.

In the field of agriculture, ML and DL have brought a revolution. These are core technologies driving AI based robots, autonomous spray drones, mapping and autonomous harvesting. Deep learning based models are predicting the crop maturity index, health and yield, also huge amount of agricultural data is being gathered to further improve the field of agriculture. AI has dramatically increased the crop yield and profit margin of the farmers. Berenstein found out that usage of sprays can be reduced by 30% if we detect and spray 90% of the clusters of grapes [8]. This will not only save huge amount of resources but increase farmer's profit margin and also lessen the pollution that is being caused owning to excessive usage of pesticides.

Zabawa & Kicherer [9] worked on the detection of single grapevine berries. They annotated 32 images into three classes naming berry, edges and background. Every berry was surrounded by edge while the remaining image was termed as background. They were able to achieve accuracy ranging from 84% to 95%; however 28% of detected berries were False positive (Type II error); the problem was tackled by incorporating different methods including image filtering.

Aquino & Millan [10] work in grape yield prediction is also noteworthy and mentionable. They integrated a camera with a vehicle specifically designed to be used in a grape farm. Camera was triggered by the movement of the vehicle. The data was collected during nighttime using LED as the artificial light source. They were able to limit the average square error to 0.16 kg per vine and RMSE was 0.48kg for an image segment consisting of three vines. Another such work was by Ralph Linker & Kelman [11] who also worked on yield estimation, but of apples, in night time. They used the specular light (light reflected from apple surface) during night time to detect the fruit, favorable results for which were achieved by aligning camera with the light source. Nellithimaru et al. [12] presented a FAST R-CNN based model of grape counting and 3D reconstruction of vineyard algorithm that used a camera

equipment with an air blower to accurately model plant by hindering leaves movement from object of interest. Many others devised models for grape detection and yield estimation exist but few notable works include Font et al. [13] work on yield estimation using artificial illumination at night time, Huerta et al. [14] work on creating a 3D model from images and using it to estimate yield and lastly Nuske et al. [15] work on the berry detection during night time. Most of these works cover the detection problem during night or training the dataset during day time is less. Nevertheless, these methods provide a non-invasive and automated way of grape detection and is being used in different applications from spraying to yield estimation etc.

Object detection in orchards is not a new thing and is spread across many fruits and crop such as Bargoti et al. [16] worked on detection and yield estimation of apples, Huang et al. [17] worked on citrus detection system using a mobile platform, Lim et al. [18] worked on detecting the kiwi fruit flowers in orchard environment, Borianne et al. [19] worked on detection of mangoes detection, detection of immature peaches by Kurtulmus et al. [20], Motohisa Fukuda et al. worked on the fruit growth monitoring system using RGB images supported by Deep Learning [21] and trees diameter estimation, mapping and segmentation by Steven W. Chen et al. [22] are only a few to be named. Yasuhiro Miura et al. work on estimating the mass of vegetables on conveyer belt, using monocular camera [23] and Bilal Arshad et al. work in tracking and counting wild animals [24] are great examples of Deep Learning changing the human lives for the better by making the work more efficient and cheap. Object detection is generally performed using the CNN or DNN techniques such as Fast R-CNN [25], RNN and YOLO [26]. Though a lot of work is being carried out in fields of object detection and yield estimation but no reliable method has yet been discovered with enough accuracy. Generally the yield estimation algorithms work fine against the average grape cluster but show bias when they encounter a weak cluster. Moreover, the scarcity of good and enough data is another reason of less efficient or overfitted models.

CHAPTER 2

LITERATURE REVIEW

2.1 Data and its Significance

Data is the backbone of any model. In the new world, it will be right if we say that data is the product and humans are an asset to produce this precious product. World powers are securing the data by investing hundreds of billions of dollars and safeguarding it as a national asset that can either make or break nations. It is one of the biggest national asset. In this age of technology, if we want to see which country is most powerful, one of the things we'll factor in will be the data that country has access to or how many data warehouses it has?

In simpler terms, computer process data in binary number system and the most basic form of that number system is a bit (shown in Figure 2). All forms of data including images, videos, text files and audios are first transferred into binary form in order for the computing machine to understand it and process it to get some results.

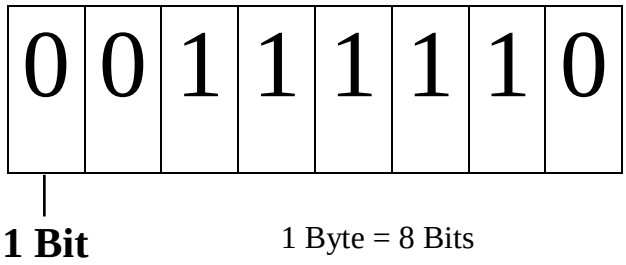


Figure 2 showcasing the most basic unit of Data

With the increase in computation power the size of computers also began to shrink and technology started getting cheaper. It made our computing systems more accessible for a common man. Moore's Law is playing its part and is holding out till now. With the increase in computation power and decrease in sizes, new technologies have been developed and new processes have been initiated. In this Information age, the 4th Industrial revolution has already began. Internet of things (IOT) and Cyber / Physical systems drive the 4th Industrial revolution. Machine to Machine (M2M) communication has started other than the normal Man to Machine Interface. To drive all this, huge amount of data is needed to process and analyze useful

information. Meeting the increasing demands of data became a challenge and huge data centers were established throughout the world. Only Facebook has 30,000 data servers and 25 TB of data is logged into its servers daily. Seagate average capacity of hard disk drives has increased from 1TB to 5.6 TB from 2015 to 2021 respectively. To control this huge influx of data, a separate field came into existence named *Big Data*. In this field, Data Analysts try to find meaningful information from the huge chunk of data.

2.1.1 Data in Artificial Intelligence

Artificial Intelligence is the study to develop abilities in machines to perform a certain task with equal to or more than human's accuracy. AI embodies but is not limited to fields including Machine Learning (ML), Natural Language Processing (NLP), Deep Learning, Computer Vision etc. Figure 3 represents basic hierarchy of Artificial Intelligence.

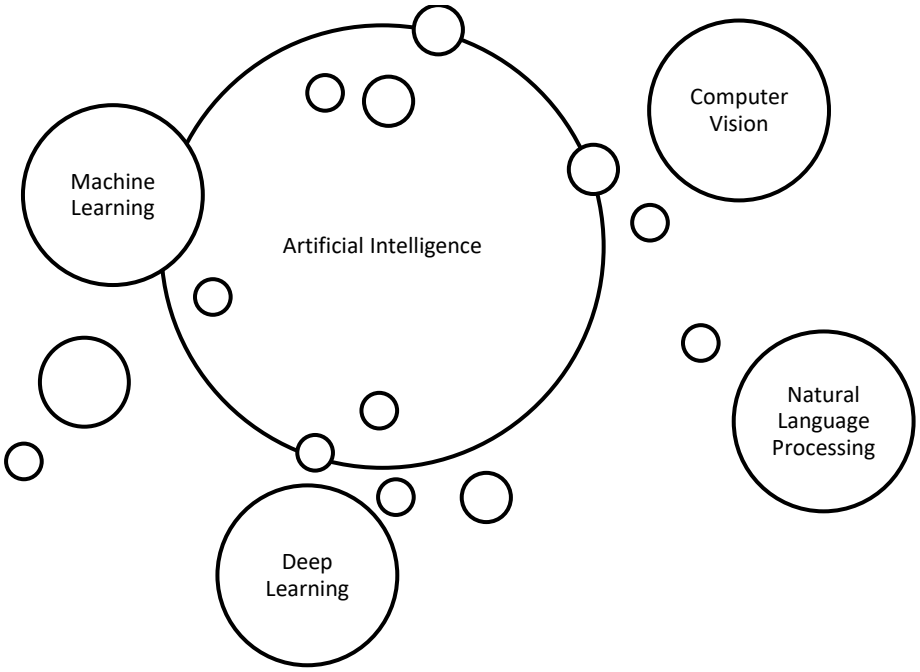


Figure 3 Hierarchy of Artificial Intelligence

All of the mentioned fields require some sort of data to be used as an example or label. Data can be in the form of audio, video, image, tabular form or any other formats. The developed algorithms are complex enough and GPUs have progressed sufficiently to provide unbelievable results, for these reasons the demand for data has also increased. The power hungriness of data has given rise to data paradox. The accuracy and performance of an algorithm is directly linked with the amount and quality of data and for that data analysts and specialized data collectors and data annotators are trained and hired to deal with data, clean it and ready it for the algorithm in use.

In Machine Learning and Data Analysis there is a term quoting:

"*Garbage in, Garbage out*"

It suggest that if the input is flawed or has issues in it then the model will be faulty and not perform well. It also suggest that the data is more important than the algorithm itself in few cases. It's because the wrong choice of algorithm may decrease the accuracy or increase the time delay in detection but the wrong data or complex may make the model underfit and not converge at all. While if the data is too simple the model will overfit meaning it will perform too well on seen data but when it will be checked at unseen data, the performance parameters will decline drastically. Data Analysis is very important before choosing and selecting appropriate model for a ML problem.

2.1.2 Data Analysis and Cleaning

Data Analysis has emerged as a separate and important field that sets the pace and even direction of our outcome. Before feeding the data to the machine learning algorithm, it is necessary to observe data for expected errors and omissions. Data cleaning is the process of finding incomplete, missing, irregular or irrelevant data and delete, modify, replace or update it according to the well suited conditions. The data went through a comprehensive check by professionals to verify its suitability for its use in the problem at hand.

Data is the most important asset in Machine Learning and even a little faulty data can stall or even fail the whole result which can cause loss of efforts, time and money. Data can be biased and that biasness can appear due the origin of data for example the face detection model trained on only Chinese faces may perform quite poorly when implemented in Africa no matter how good results it showcased in China; it doesn't mean that the algorithm was faulty in any way but it mean that the algorithm was not prepared well for the input that it may have to account for thus it performed poorly when unexpected input went through its layers. To remove such biasness, we need data that is huge and contains examples with people from all over the world in ample quantity so the machine learning model doesn't picks up biasness (i.e. face color) as a feature and is able to perform equally well in all the conditions. The following line graph (Fig. 4) showcases how unclean or corrupted data drastically affect our results.

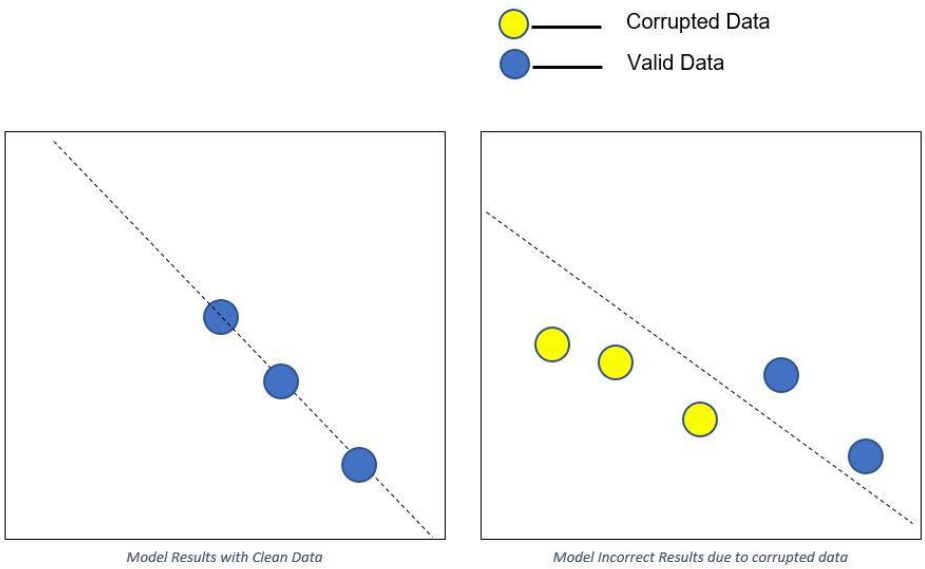


Figure 4 Impact of Data on Machine Learning Models

Cleaner the data, better and faster will be the model with lower convergence with and better Intersection over Union (IOU) score and accuracy. Images, videos and audio files also go through this process where data experts analyze and pre-process them. In this research, the data consisted of images. In this process, every individual image was analyzed multiple times at multiple steps. Unsuitable images consisting of irrelevant or bad data were deleted, many images exposure was fixed and all the images were compressed to make the dataset lighter (For details see Section 3.1).

2.1.3 Overfitting & Underfitting of Data

After preparing the data for the Deep Learning model, selecting appropriate model for data is also crucial. If the model is too simple for the data the accuracy will not be good and the model may converge at a local minima, being unable to improve accuracy any further. This phenomenon is called under fitting. Underfitting created high bias and low variance in the model. The model is weak to handle the data so it is not able to extract any useful patterns. Such a model is useless for the researchers as it shows poor performance but it is not discussed quite often because it is very easy to detect. It causes financial loss and computation loss and the computation time used to train such model is time wasted as such model is unable to capture important underlying features of data. To reduce underfitting different techniques can be used. Usually the complexity of model is increased, noise is removed from the data and training time is increased to tackle underfitting [27].

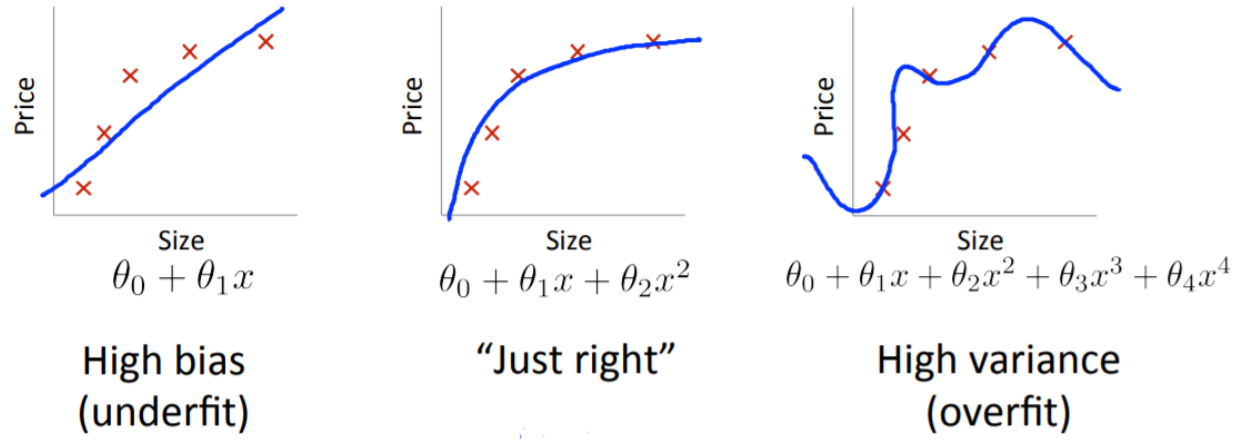


Figure 5 The Price vs Size Model showcasing cases of Overfitting, Underfitting and perfect case (courtesy to Andrew Ng)

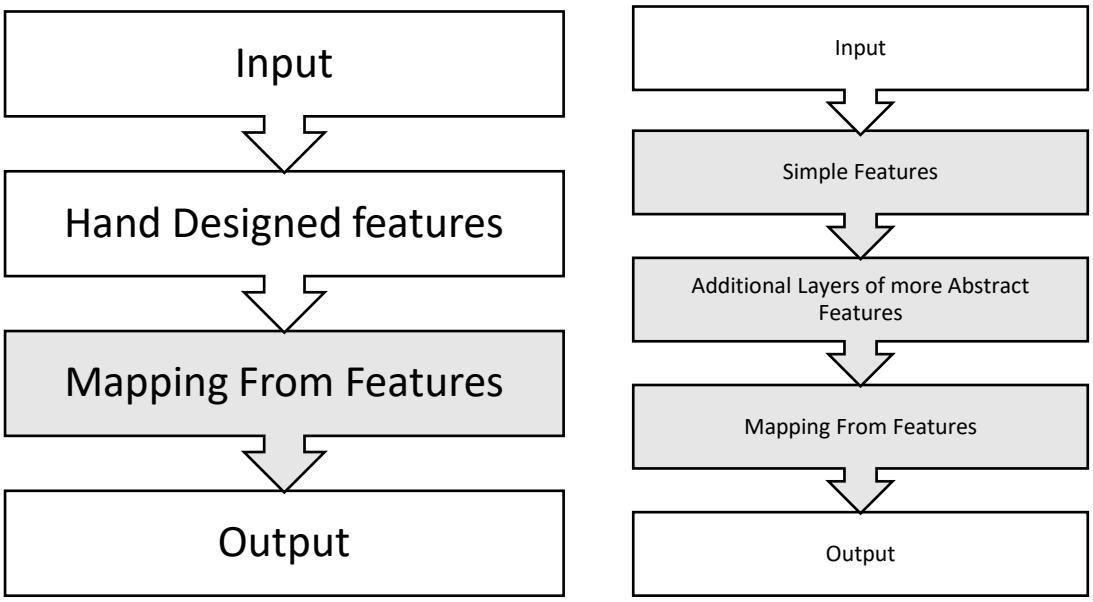
Let's look at an example of house Price and Size relation. If the predicted price has less accuracy and high bias then its underfitting, if it has error but it's quite low and has least bias and variance then it's perfect while if it is not showing any error or error is too low and the model performs quite poorly at the unseen data (as shown in Fig. 5) meaning it has high variance then the model is overfitting. Bias consists of the assumptions made by the model in order to make the patterns easier to learn. Variance is the change in error amount when we change the seen data and feed unseen data to our model.

When excessive data is fed to a model and it is also complex, it starts capturing many features. Some of those features are not supposed to be detected as those features exist due to irregularities and noise. This phenomenon created in the model creates high bias and low variance. This improves accuracy of the model when seen data is fed to it but such models perform poorly at the unseen data. To avoid overfitting we use different techniques like regularization, using more data and stopping the training earlier so bias remain low [27]. Ideally, it is desired that a model has low variance and low bias meaning that a spot between over fitting and underfitting is needed with least amount of error for the model under training.

2.2 Deep Learning and Artificial Neural Networks

Deep Learning is a branch of Machine Learning which completely deals with Artificial Neural Networks and use them to generate patterns from data and conclude meaningful results by building relations from those discovered patterns. The base unit of Neural Networks is called a Neuron. The neural networks are interlinked with each other layers after layers and transfer data to each other just like a human brain, thus the name 'Neural Network' is common for them. Deep Learning is not something new in scientific community. First time work was started

on artificial neural networks was by Igor Aizenberg and colleagues in 2000 but not much work was done in this field till much later on due to limited processing power, less amount of available data and non-availability of open source software and tools that are essential for Deep Learning.



Classic Machine Learning Model *Deep Learning Model*

Figure 6 Flowcharts showing the behavior of Machine Learning and Deep Learning. The shaded boxes represents processes which can learn from data.

The first question at hand is:

"Why we need to use or learn Deep Learning?"

To answer this one need to understand the difference between Machine Learning and Deep Learning. Machine Learning uses the features that are usually hand engineered and thus are not stable or scalable. On the other hand deep learning learn these features itself in a hierarchical manner starting from Low level features and ending at high level features after going through mid-level features [15]. The Figure 6 also contains the flowcharts showcasing the same concept where the highlighted blocks represents steps where the model learns from the data. In Machine learning chart, the model only learn from one block that is after the model has mapped from the features while in the deep learning case, the model learns from multiple steps. It start learning when model extract low level features, it improves the learning process till more features are extracted and at the end it perfects the learning while mapping the features [15].

2.2.1 Single Layered Neural Network

To understand deep neural networks (DNN), the most basic example is a single layered neural network. Single layered neural network consist of an input layer, an output layer and a single hidden layer where all the computation happen (as shown below in Fig. 7).

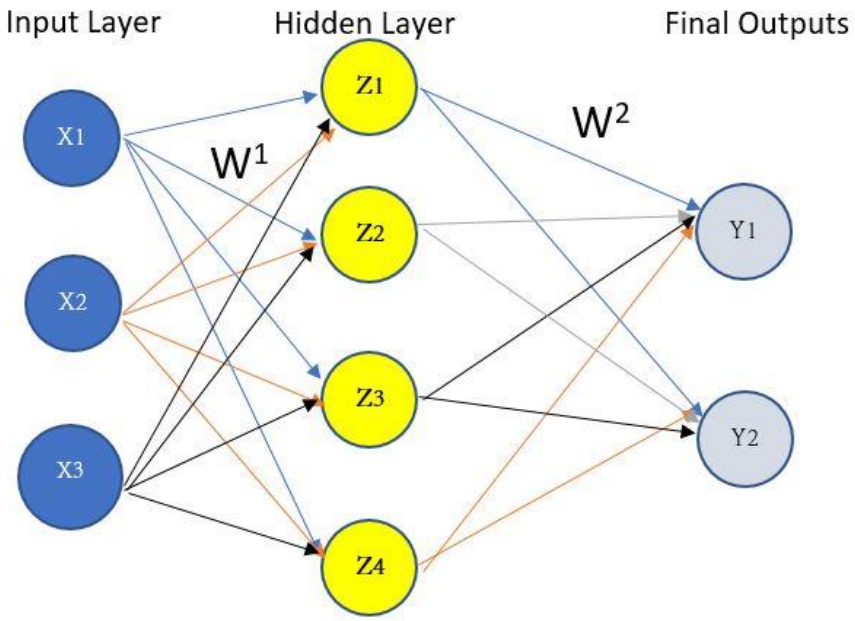


Figure 7 A Single layered Neural Network

A Single layered neural network is made up of neurons. In the Figure 7 we have three neurons in the input layer, four in hidden layer and only two neurons in the output layer. It shows us that this network represents a multi class classification problem and has two classes. The Input layer shows us the number of inputs we have to provide the model. Hidden, as the name suggests, are hidden layers and does not show up. There can be multiple sets of hidden layers that are interconnected and the number of hidden layers is directly proportional to the complexity of the model but inversely proportional to the training time. It means that if number of hidden layers is smaller than the model will converge faster but will be simpler and less complex; so, such models will be perfect for simpler data with simple tasks. Contrarily, if the hidden layers are more in numbers then the neural network will be able to learn even the minute features and will be more complex but this will require a larger data set as the complex models are more data hungry.

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks also resembles the ordinary neural networks that were discussed in previous section. They are made up of neurons which are used to learn features by

carrying specific weights. Every neuron receives an input from the previous layer and multiplies it with the weight assigned to it and forwards it. It has a single score function and also have a Loss Function just like a DNN. It usually use images as Input so while designing it, we use this assumption to set it up more efficiently.

There are also many differences between a simple DNN and a Convolutional Neural Network. CNNs also have the Input layer, hidden layers and an output layer. The neurons in each layers are interconnected with all the neurons in the next layers and previous layers and where the neurons are independent and are not connected with any other layer further then that layer is called output layer. When we are using regular neural networks, the images don't scale well with the network. For example the 32 x 32 RGB image will have 32 x 32 x 3 = 3032 weights in first layer of neural network. Similarly if we increase size of image to 200 x 200 then weights in first layer will be 120,000. This will create many such many such parameters due to which this full connectivity will be wasteful and will be cause of overfitting.

Unlike a regular Neural Network, a CNN is 3 dimensional which gives it stronger ability to deal complex data sets, an image which is also arranged in a 3-Dimensional structure as every image pixel contains three values (RGB) ranging between 0 to 255 (shown in Fig. 8), this benefits the ConvNet. The neurons of CNN are arranged in 3 dimensions namely the height, width and the depth. This help the CNN understand the image pixels better than most of the other available options. The CNN transforms the 3D input into a 3D output which has its unique features and parameters for respective classes.

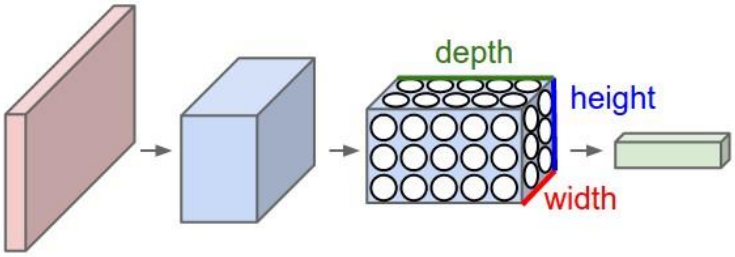


Figure 8 As the Images are arranged in 3-dimensions, A Convolutional Neural Network is perfect to deal with them owning to their 3d structure and ability to deal with data in 3 dimensions

In a CNN, different layers are combined for it to function effectively. Each layer has its unique functionality. There are three main type of layers that are used in CNN named Pooling layers, Convolutional layers and Fully Connected layers. These layers are stacked up together in order of convenience and efficiency to make a fully deployable Convolutional Neural Network. Then there is also a RELU layer that is basically an activation function that thresholds

at zero, it saves the weights from completely dying out without affecting the volume. The Pooling layer is the downsampling layer i.e. if the input is [32 x 32 x 10] then it will be downsampled to [16 x 16 x 10]. Convolutional layer will calculate the product of neurons and their respective weights. Lastly, the fully connected layers are the deciding layers that calculate the class scores in the end.

2.3 Object Detection

Object detection is a sub branch of Computer Vision that deals number of computer vision and deep learning tasks on an image that leads to detection of an object by creating a bounding box around it and correctly mentioning its class. It can be used for single class classification as well as multi class problems. It can often be confused with Image recognition and Image localization but it is more advanced than those occurrences. Image recognition only recognize object of interest presence in the image without pin pointing the coordinates or axis. Image localization is able to draw the bounding box around the object of interest. Lastly, the object detection not only draws the bounding box but is also able to classify i.e. associate characteristics with object of interest. So, object detection algorithm is more intelligent and innovative.

There are multiple algorithms that help performing object detection. Object detection is a very diverse field and it has tones of different applications in modern world. It is making roads more secure, cities safer, reforming the health care, increasing the agricultural output and helping to achieve automation in fields that were never thought before. The cameras, over the years have been transformed with the reform in field of semiconductors. From red rooms, the image development has changed to being fully digital and the reel have been phased out, the size too has reduced massively and now the available lenses are of more advanced technology. The fully digital cameras are now also available in form of small sensors of few inches with decent processing power and a wide compatibility. It has transformed the computer vision and thus the need to make the computer understand what it sees and make decisions based on that is even more. Thus, object detection is being used even more now-a-days for all such minute tasks at a massive scale. It is being used as a base algorithm for many connected and complex methodologies. It has mainly three approaches all falling under AI. First is purely computer vision based approach where we use image processing techniques only like histogram equalization, color matching etc. This approach is basic in nature and can’t perform well on a generalized dataset. Second approach is machine learning based object detection. Here, the algorithm needs data at a much larger scale. The data is annotated and labelled. It can have one

or more than one classes. When data has more than one class then it is called multi-class classification problem.

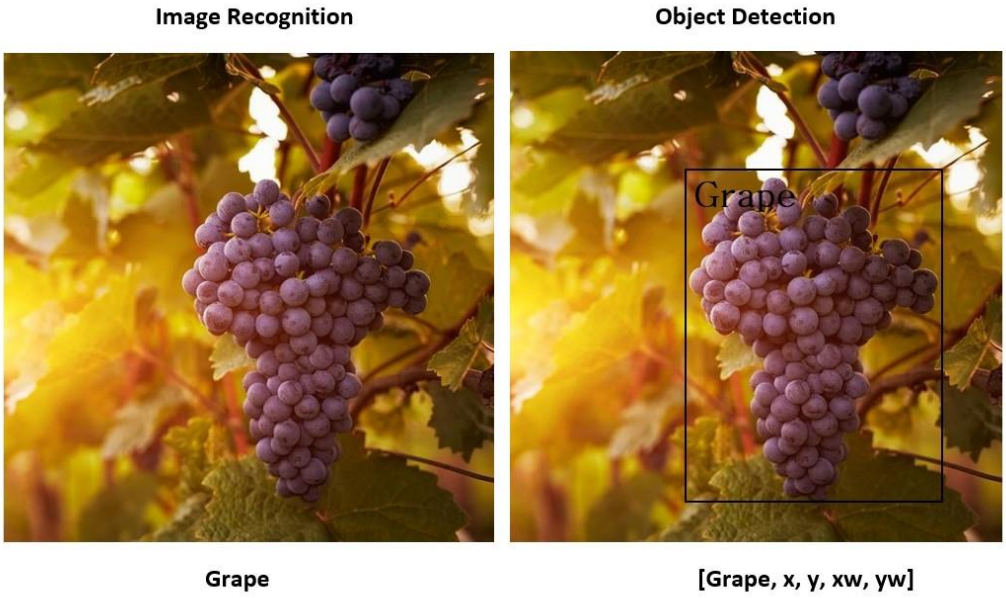


Figure 9 Image Recognition and Image localization are simpler tasks than the Object Detection

2.4 YOLO Framework

YOLO is a popular framework implemented on Darknet. Darknet is written using the CUDA technology and C language. It is available freely as it is open source software that uses neural network. CUDA is a NVIDIA driver mainly developed for Machine learning and deep learning that enables the usage of NVIDIA GPUs for training purpose. For real-time computation, it is essential for a model to be able to utilize GPU to perform computations and Darknet allows its users to do so, which makes it fast and preferable over other YOLO implementations. Darknet is also used for many other frameworks such as Nightmare, ImageNet Classification, Tiny Darknet, Dark Go, RNN and Image Classification on CIFAR – 10 etc. The model of Darknet is shown in Figure 10.

YOLO was developed by Joseph Redmone [26]. It is one of the fastest model available out there. It used to be most accurate but better models (e.g. RetinaNet, SSD etc.) took this title away from YOLO [28][29], still its best in term of speed especially when the applications are in real – time. YOLO is a developing model with many versions and still work is being done on its more versions. YOLO was first launched on 2016, YOLO v2 and v3 were released in 2017 and 2018[26] [30] respectively. After YOLO v3 the original author, Joseph Redmone, stopped working on YOLO stating privacy concern and military implications as the reason

[30]. After him other authors released YOLO v4, v5 and other versions but unlike the releases before, these versions weren't the sequential releases but had goals that suited the authors releasing those respective models.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 10 Darknet-53 architecture is shown in this Figure [30]

Some of YOLO versions are:

- YOLO v1
- YOLO v2
- YOLO v3
- Tiny YOLO
- YOLO v4
- YOLO v5

Here the last sequential release of YOLO will be discussed, that'll be YOLO v3. YOLO detect and recognize objects by deploying convolutional neural networks. It is a Fully Convolutional Network (FCN) based model as it make use of multiple layers of convolutional layers. It has 53 layers trained on ImageNet; for object detection, it has additional 53 layers

bringing up the total number of layers to 106 [30]. Its convolutional layers are 75 in total with unsampling layers and also skip connections, it also has a downsampling layer with the stride value of 2; this helps preventing loss of low features [30]. Leaky RELU is used as the activation function in this model. YOLO has many implementations for example Darknet implementation, PyTorch implementation and tensorflow implementation etc.

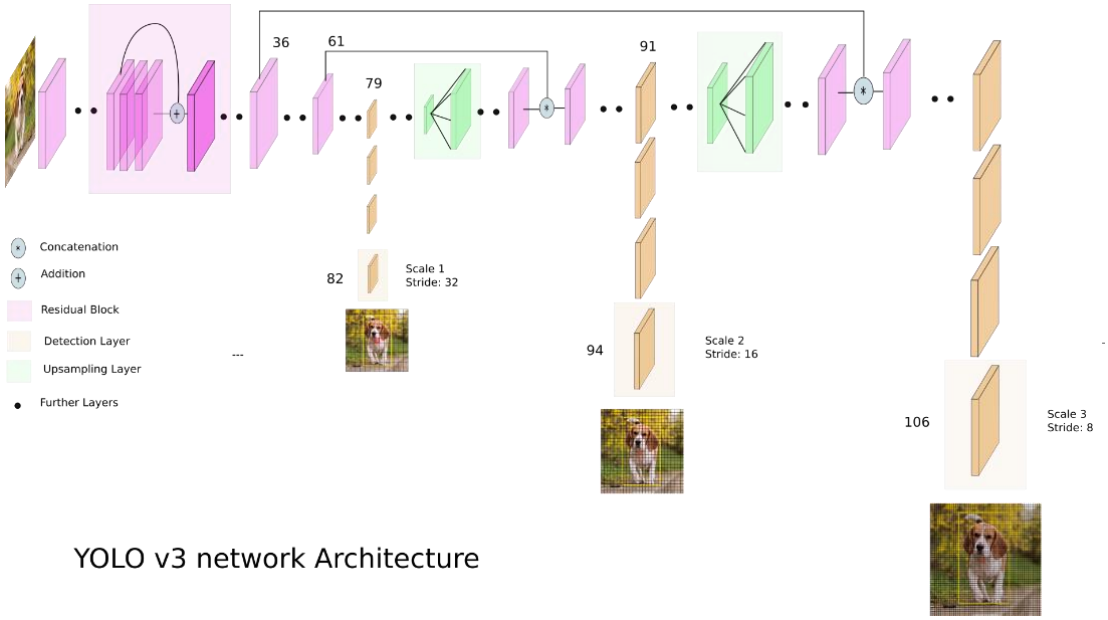


Figure 11 Multi-scale detector feature of YOLO

The newer and advanced architecture of YOLO v3 is able to make the decisions at three different scales (as shown in the Fig. 11), it is known as a multi–scale detection. Multi-scale detection is performed by a 1 x 1 kernels on the feature maps at three different stages.

Formula for shape of kernel:

$$1 \times 1 \times (B \times (5 \times C))$$

Here B is the no of Bounding Boxes; 5 is for 4 bounding boxes attributes and confidence while C represents the no. of classes.

For instance a 416 x 416 image is fed to the model (in Fig. 11) then the layers where the process of object detection will be performed will be 82nd, 94th and 106th layers. Firstly, the 81st layer, which is a downsampling layer, will do the downsampling and feed the downsampled data to 82nd layer where the kernel will be applied to the small image and results will be determined in form of 3d matrix. Secondly, the 79th layer onward are convolutional layers followed by upsampling layers. The feature map is concatenated with the link from the 61st layer and the 94th layer applies the kernel to determine result for the second time at a larger

image. Lastly, a similar step of upsampling and concatenating from previous layers is followed and third result is calculated at the 106th layer. This method gives YOLO v3 the capability to detect small object which was not better in earlier versions. The output from previous results concatenate with layers later and make the detection of all, small, medium and large objects, more probable.

2.4.1 Setting Up YOLO on a PC

Setting up YOLO on PC is a time consuming but a necessary process. Going into details of the process will not be possible but the summary of the process will be discussed here. For it, we need to install all the dependencies and pre-requisites. We need to install all the pre-requisites from scratch in order to successfully compile YOLO.

The pre-requisites include:

- CMake
- CUDA
- cuDNN
- Visual Studio
- Open CV

After this, the Darknet is cloned in the work directory. Visual Studio and CMake are used to compile the Darknet which creates an executable file in working directory. Lastly, the pre-trained weights will be downloaded and configuration file will be set up as per the requirements. Configuration file holds all the important information that is necessary for training i.e. batch size, sub-divisions, filters, classes etc. After that, the dataset will be annotated and the model will be ready to either be trained or tested.

2.5 Single Shot Multibox Detector (SSD)

Single Shot Multibox Detector (SSD) is a well-used and highly implemented model that's being used to detect images and classify them using the deep learning. SSD is fast, highly accurate and beats even the R-CNN and matches accuracy of Faster R-CNN and was introduced in November 2016 [28]. The writers introduced an innovative method where the regional proposed network was eliminated and instead of it and rather sum up all the steps in single forward process. In this process, default boxes are introduced in the image with different aspect ratios and scales. The default boxes are then adjusted after comparing the location of

boxes with objects to try bring up the scores. The single forward pass of the layers make this process faster and increase speed.

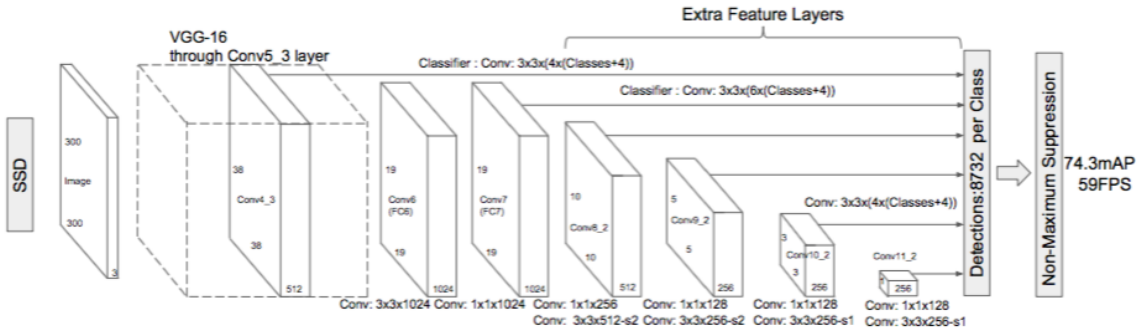


Figure 12 Single Shot Multibox Detector model

The SSD model uses VGG-16 as apparent model but without the fully connected layers. Both YOLO and SSD detects the objects in single pass but SSD came after YOLO and with a few improvements [28]. It consists of two main parts namely backbone layer and the SSD head. Backbone is the image classification network that has been pre-trained SSD head is made by stacking up the convolution layers over backbone layers [28]. Those convolutional layers are responsible for the output and detects the bounding boxes.

SSD adds several layers at the end of base model for detection which give it an edge over YOLO. Also the no. of detections produced by SSD supersedes the detections made by YOLO by 8634 detections as YOLO can make only 98 detections per class while SSD make 8732 detections. If we compare YOLO v3 with SSD then YOLO v3 is still better than the SSD as its detection accuracy is more than both SSD and Retina-Net [30]. Moreover, Yolo v3 detects 51 images per second while SSD detects 32. Both of these algorithms give value above 30 FPS which make them suitable for real-time detection.

2.6 Confusion Matrix

Normally researchers prefer the accuracy value and take it as the sole performance parameter but it is not a good practice. A model can have higher accuracy still it might be too basic to be implemented, it is also known as accuracy paradox. To deal with this issue, many other parameters are now being widely accepted and implemented. The matrix containing the summary of all such parametric values is known as confusion matrix [31]. It contains the details of the total number of predictions a model make, how many are correct, how many are incorrect and it also give us in depth details about the type of error a model is facing. It is one of the most carefully tailored matrix and it is made sure that the matrix results are experimentally

acceptable and valid as often the success or failure of a model can be figured out by looking at this matrix of that specific model.

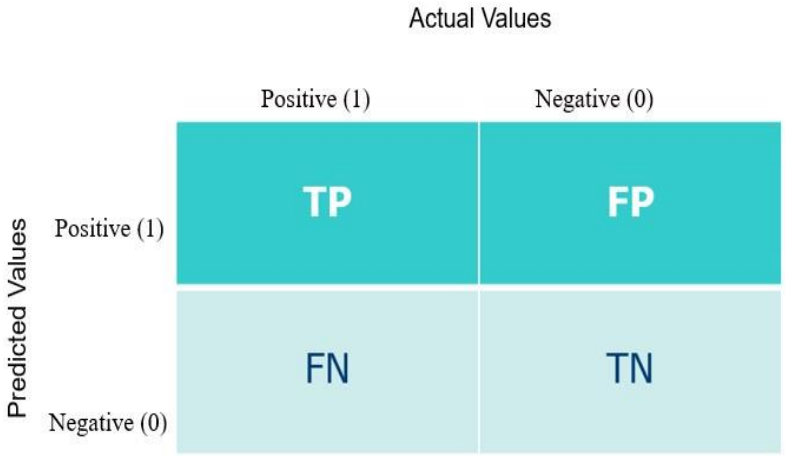


Table 1 Confusion Matrix for single classification problem

Confusion Matrix (shown in **Table 1**) holds four values. For single classification problem, there will be only one confusion matrix while for multi classification problem the confusion matrices will be '***C x 1'*** where C are the number of classes. One matrix can also be created for multi-class classification, the size of matrix for that case will be C x C (i.e. for 3 classes, the confusion matrix will be 3x3) [15]. Confusion Matric consist of following four values:

1. **True Positives:** When the value is detected positive by the model and it is true (i.e. rightly detected) then it is called true positive. It is the ideal case.
2. **False Positives:** When the value is detected positive by the model but it is not true (i.e. it is wrongly detected as positive) then this scenario will be termed as a False Positive.
3. **False Negatives:** When the value is detected negative by the model and it is false (i.e. a positive value is wrongly detected negative) then this will be termed as a False Negative.
4. **True Negatives:** When the value is detected Negative and truly detected so then it is termed as True Negative.

The False Positives are termed as Type 1 error and False Negatives are called Type II error. The columns represents the actual values and the rows shows us the predicted values. Column 1 contains the positives and column 2 contains the negatives. Row 1 shows us the positive predictions and row 2 shows us the negative predictions made by the model. In the

ideal case the Type 1 error and Type 2 error are negligible while the confusion matrix mainly has True Positives and True Negatives [31].

Following are the few performance parameters that are mainly calculated from the confusion matrix:

2.6.1 Accuracy

The most commonly used matrix in Machine learning and deep-learning is accuracy of the model. It is the ratio of correctly classified instances to the total number of instances. Accuracy is the most widely used and most widely accepted parameter. It is so overly used that its over use, itself became a problem. Issue with accuracy as a parameter is that if the data is balanced then it will give quite good results but if it is not the case then the accuracy will be giving out misleadingly decent results.

$$\text{Accuracy: } \frac{TP+FP}{TP+FP+TN+FN}$$

2.6.2 Precision

Precision is the ratio of positive instances predicted correctly over the total number of positive instances predicted overall. Its formula is True Positives divided by the sum of True Positives and False Positives. It gives us the rate that when an instance is predicted correct by a model then how often is it correct? Precision is one of the most important performance parameter of a Machine Learning model.

$$\text{Precision: } \frac{TP}{TP+FP}$$

2.6.3 Recall

Recall is the ratio of positive instances predicted correctly over the total positive instances. It also includes the False Negative which contains the missed positive instances. Its formula is True Positives divided by the sum of True Positives and False Negatives.

$$\text{Recall: } \frac{TP}{TP+FN}$$

2.6.4 F 1 Score

F1 Score contains the average of both Precision and Recall. It is generally considered a better parameter than accuracy as it proved to be more effective and easier to be understood. It

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

ranges from 0 to 1 with 1 or closer to 1 meaning that the model is successful and 0 or closer to 0 meaning that the model is a failure [31].

2.6.5 Intersection over Union (IOU)

Intersection over Union is one of the most important performance parameters specifically in object detection. It compares the bounding boxes and gives a value between one and zero where one means that the IOU is very good and zero means that the result makes no contact to the object of interest [31]. Normally, value over 0.75 is considered a good IOU measure and value over 0.0 is considered excellent (shown in Fig. 15). The basic objective of this parameter is to check the performance of the model being trained to see how accurate it is and keep improving till the results are somewhat satisfactory for the model to be implemented.

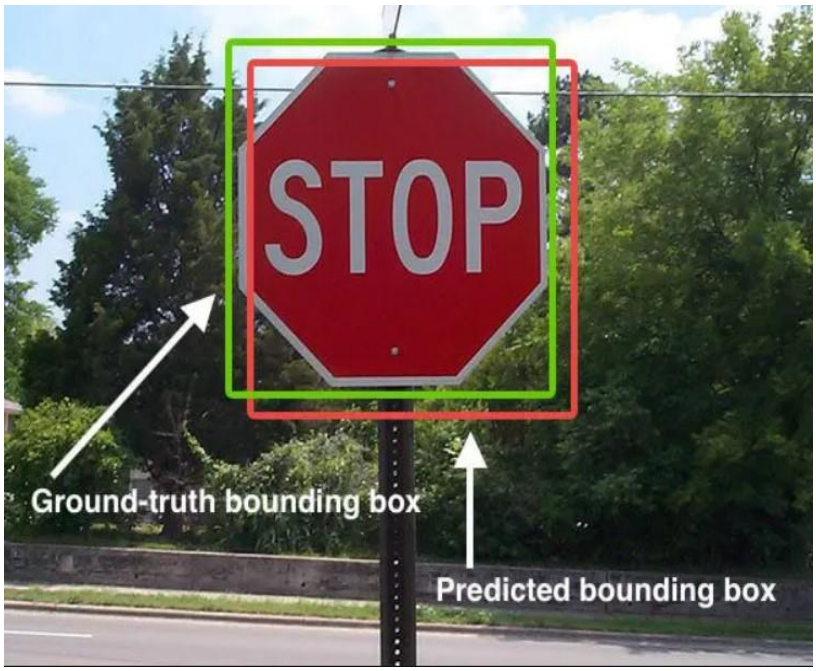


Figure 13 Stop sign with Ground Truth box and prediction box

The stop sign in Figure 13 showcase both the Ground-truth, represented by the red box, which is the annotated box and the prediction by the model colored in green. The boxes are overlapping by more than 50% which is a good sign. The stop sign is being detected by the model, though it is still not perfect and requires further training.

To calculate the IOU value, the intersection of both boxes will be divided by the union. The formula of IOU is showcased below:

$$IOU = \frac{Area\ of\ Intersection\ of\ two\ boxes}{Area\ of\ Union\ of\ two\ boxes}$$

Figure 14 Formula to calculate IOU

If the boxes doesn't overlap, the value of IOU will be zero as the Area of Intersection as well as the area of union of boxes will be zero and putting these values in formula shown in Figure 14 will give zero as the result. If both boxes overlap completely will give IOU value as zero.

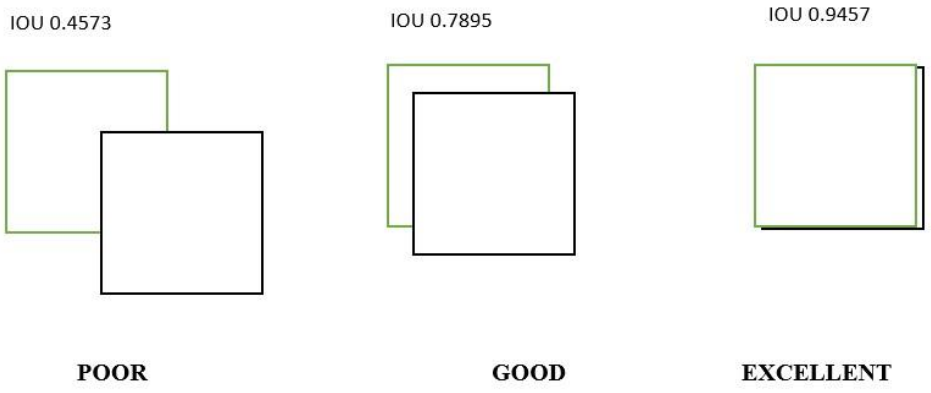


Figure 15 showcasing various IOU cases

2.7 Relation between Precision & Recall

Precision and recall are two of the most understood terms in the field of machine learning despite being the most crucial parameters. Once understood, it is easier to explain and use these parameters in our models to improve the results marginally.

Whenever a model is being fine-tuned, it is made sure that the model is neither overfitted nor underfitted. It is tried that the bias and variance are both kept low and a manageable value can be achieved somewhere in-between the bias and variance. While working on the bias - variance tradeoff, the precision – recall tradeoff is often forgotten. Whenever a specific case of classification is under consideration, the precision – recall matrix is often more useful as it has to take care of specific conditions. In those conditions, the precision- recall tradeoff is needed to be minimized and more attention is needed to be given to these metrics. It somewhat represent accuracy with more focus on Type I and Type II errors. Precision is dependent on Type I error, It will be smaller with increase in Type I error. Similarly, Type II error influences the Recall's value by decreasing it with increase in the Type II error i.e. False Negative.

Some models are sensitive enough to not let a single mistake filter out as a False Positive while other models are not sensitive enough and have more flexibility. To take care of these specific scenarios, values of precision and recall are changed manually; this phenomena is known as precision – recall trade off. Let's take an example of a model predicting a cancer to be benign or malignant. In this case, we can afford low accuracy and lower recall but the model must be precise and should make minimum mistakes, the model can detect a benign tumor as malignant but the model must avoid detecting a malignant tumor as benign for it to be considered successful and implementable. In the Figure 16, two scenarios are being showcased. In a, the decision boundary has all of the benign tumors but also has a malignant tumor so its TP are 5, FP is 1 and TN is 0 as decision boundary added one malignant tumor and labelled it as a benign tumor falsely. Hence in case a, the precision value is 80% and Recall is 100%. Similarly in case b, the model failed to identify one benign tumor but did not falsely identified any malignant tumor as a benign tumor thus the TP are 4, FP is 0 and TN is 1. Thus here the Precision is 100% though the recall is 80%. So when such cases are under consideration, the case b will always be preferred over a as it is safer to mislabel a benign as a malignant for the model but under no circumstance a model should label a malignant tumor as benign.

In other cases we need more Recall than Precision for instance let's take an example of harvesting system that harvests a specific crop. It harvests the only crop making sure that the leaves doesn't mix up, or the amount of foreign objects that get mixed up in the crop remain low.

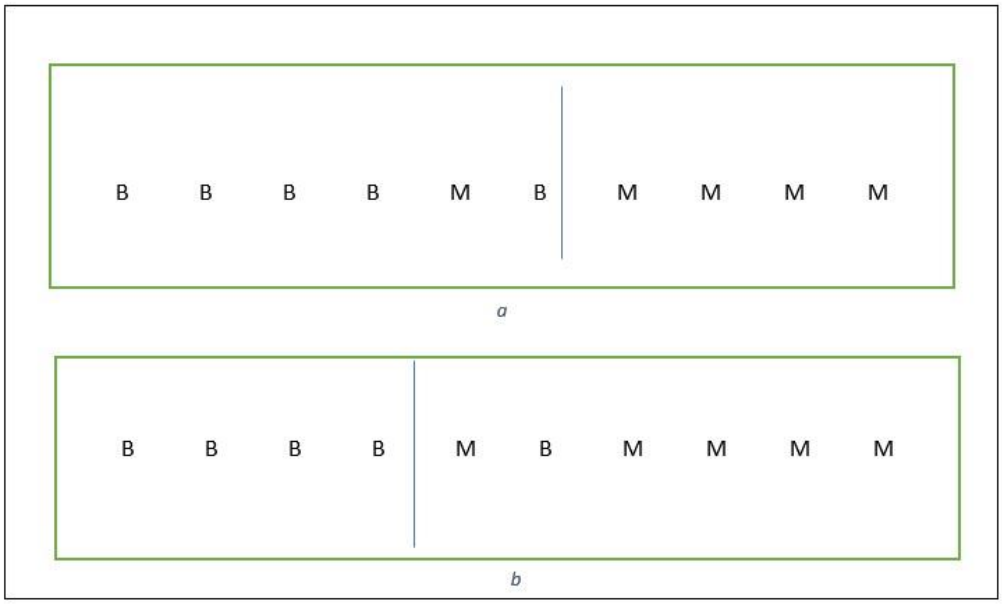


Figure 16 Moving the Decision Boundary directly affect the Precision and Recall

If we use high precision option then the crop will go with the waste with the foreign objects but if we use the low precision, high recall option then least crop will be wasted still keeping the amount of foreign objects low.

2.8 Data Annotation

In object detection, data annotation is of utmost importance and much attention is given to this task. It is the process in which the data (images, videos or text) is labelled in different formats using different tools. In this process the object of interest is annotated and labelled using a suitable tool. Multiple tools are available for this task and newer and better tools are being made available with each passing day.

Main types of Data annotation are:

- Images Annotation
- Video Annotation
- Text Annotation

In this research, the dataset consists of Images so text annotation will be used. There are many different types of image annotation like semantic segmentation, bounding box, 3D cuboid, Polygons, key points and key lines etc. A few major types of Image Segmentation will be discussed.

Figure 17 Examples of Image Labelling (A) shows Key Points labeling, (B) represents semantic Segmentation and (C) represents the bounding box labelling

- Semantic Segmentation: In this method, a mask is drawn over the object of interest and the mask can be of any shape.
- Bounding Box: In this method, a bounding box is drawn around the object of interest and model is able to localize and label multiple classes.
- Key Points: In this method, key points are used over objects of interest. It is particularly useful when working with hand gestures or facial landmarks detection.

CHAPTER 3

PROPOSED METHODOLOGY

In this research we propose to develop a dataset consisting of images taken in multiple conditions in a vineyard and based on that dataset, we'll develop a model that is able to perfectly detect the grape clusters in a vineyard environment.

The major objectives of this research are:

- To create a large dataset
- To create a scalable model
- To achieve high accuracy and F1 score

It is desired that the dataset is collected and customized from scratch. The dataset is desired to be large enough for the model to avoid underfitting and be able to extract the necessary information required of it. The larger and well-collected dataset will help the model to be able to perform better universally and give better results on unseen data. Here, well-collected means that we need our data to have multiple examples of different types in different conditions. It will stop the model from overfitting and converging at local minima but will also make it more scalable. The model will be ready to be implemented on different types of vision sensors and in different conditions. The model is deemed to be able to perform well in different light conditions, weather conditions, with different hardware and under unobserved scenarios. If the model satisfies all the above mentioned recommendations than it will perform better than the counterparts. When its parameters are calculated, it will show superior results like better F1 score and accuracy.

The task is divided into following sub categories:

3.1 Data Collection

The first and most challenging part of this research is data collection. One or more vision sensors will be needed for this to be performed under different conditions. It is desired from the data to not be taken from a single type of lens but different sensors and lenses are preferable. Similarly the dataset should be taken during different times of the days. Data should be taken using both the auto mode and manual mode of the camera using different ISO values, color and manual focus.

3.2 Data Handling

After the data collection, all the images will be needed to be examined individually to check out their suitability as the test set. The data coming from all the multiple sources will be collected at the same place and the bad data will be discarded. Bad data consist of images that carries faulty or wrong examples that may create biasness in the model or affect the accuracy of the dataset if not filtered out. After the deletion of data, the remaining data will be readied to be fed to the model and will carry actual number of examples for our dataset.

An online cloud storage will be used to upload the data to be accessed online and be used so that while training, the model is able to access the data faster online and doesn't need to be dependent on the personal machine being used. It will be beneficial as by using this approach, the model can be trained anywhere and data can be accessed from any other place.

3.3 Preprocessing

In this process the data went through different processes before feeding it to the model. In this research, the data consists of high quality images. The images size, contrast, brightness and sharpness etc. will be changed in this process and images will be cropped where necessary. It will be made sure to manipulate the data to enhance the performance of model. Data will be split between test and training set where the training set will carry the major chunk of data and will be used for training of data and test set which carries minor chunk of data will be used to test the performance of the dataset. Generally train-test split common percentages are 70%-30%, 80%-20% and 67%-33% etc.

3.4 Data Labelling

After the split, the training dataset need to be annotated. A suitable annotation tool will be selected to annotate the data and label it in the selected format. Every image will be manually labelled, selecting the object of interest exactly inside the bounding box and then successfully saving the label. The label will be saved in the form of a .txt file holding numeric values that contain class name and bounding boxes specifications. It will have five values where the first value represents the x pixel of the center of the rectangle and second value represents the y pixel. Third value represents the length of x axis and fourth represents the length of y axis. Fifth value represents the class to which the object inside that rectangle belongs to. The .txt file will produce for every Image file with the mirroring name, showcasing that it contains annotation data of that specific image. For SSD model, data will be labelled in PascalVOC

format [29]. PascalVOC is a standard for many more state of the art algorithms such as R-FCN [32] and ResNet [33].

3.5 Creating the Training Model

Though the importance of data can't be denied but creating a training model is the most crucial step in this process. A suitable object detection method will be chosen based on the data type and data annotation format. Its parameters will be set up and experimented to attain suitable result. Data will be divided into different batches and GPU will be used to train the data. The training model needs to be reliable, fast and accurate which is easy to run and also scalable so our model can be implemented from different computers to even mobile phones and smaller micro controllers i.e. Raspberry Pi 3 and NVIDIA Jetson Nano etc. By scalability, the model will have enough scope to be used in a handheld device that can be used in the agricultural environment.

3.6 Reliable Testing Results

Another important requirement is to develop a suitable methodology to develop resulting parameters in such a way that they are acceptable and useful, the conditions are suitable whether it's the data, labeling or modelling and the results can be relied upon and taken as reference in other projects. The test dataset will be used to calculate the results on the model previously trained. The parameters observed and calculated have already been discussed in details in section 2.5. Accuracy, Precision, Recall and F1 score will be calculated on the test dataset and IOU value will also be compared and multiple models will be trained using different characteristics and dataset size and their results will be compared to devise a proper and good model for agricultural environment.

Test dataset will be smaller than or equal to the training dataset size and it will hold the unseen data which will prove to be helpful accessing and validating the performance of the model on unseen data. Test dataset may or may not include the .txt file that contains the annotation data because test dataset doesn't necessarily require training and annotation is strictly required only for training.

CHAPTER 4

MATERIALS & METHODS

This section holds all the information related to the extensive training process, the methods and tools that helped in completion in this training and the specifications of the required equipment. Complete details of the process; creation of the datasets, cleaning it, pre-processing, creating the model and training it, debugging issues and improving the process will be discussed in this section.

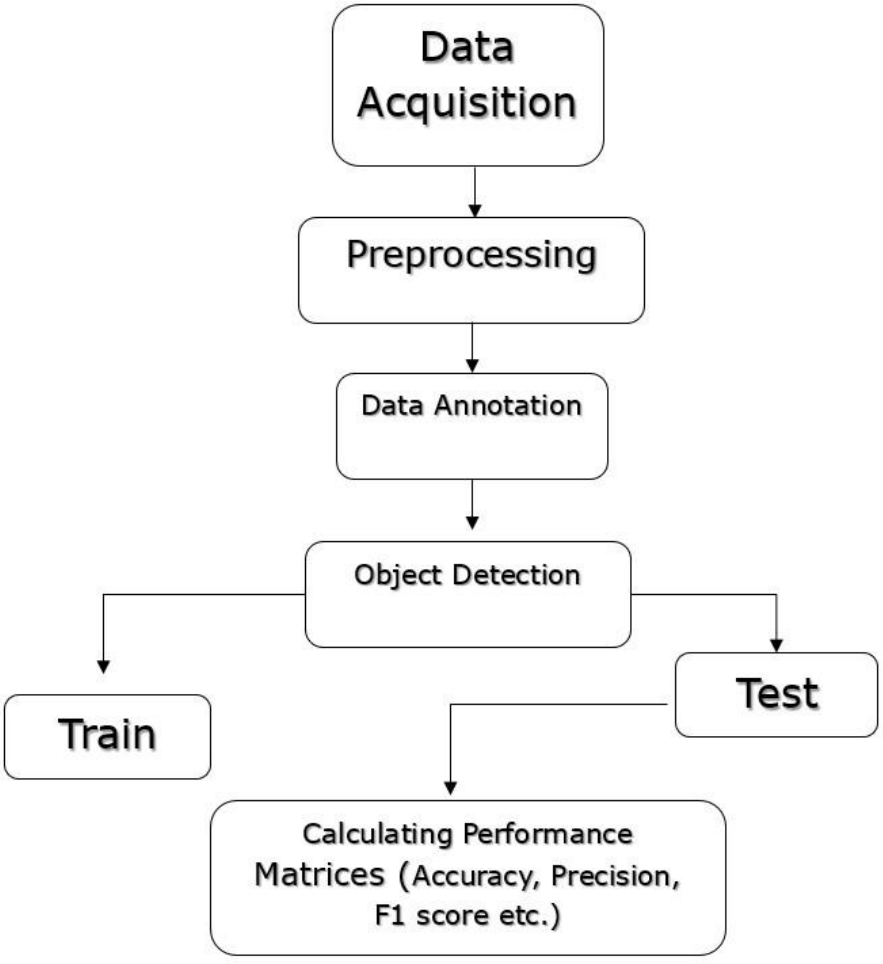


Figure 18 Block diagram showcasing major parts of research

The block diagram of this research is shown in form of a flowchart in Figure 18**.** It shows all the major parts of this research. It show that first a dataset of adequate size will be created that fulfills research requirements. The data will be pre-processed and labelled before

being fed to Object Detection algorithm. After being trained the algorithm will be tested on test dataset and the performance matrices will be calculated.

4.1 Dataset Creation

Dataset creation was one of the most laborious tasks throughout this process. It required physical presence in the agricultural land and manually clicking thousands of pictures. For this process, a grape farm near Chakri interchange Rawalpindi was selected (shown in Fig. 19). It is 40+ km away from the CE&ME campus. It is located in the plateau region of Potohar. This region is famous for corn and grape production and now olives are also being cultivated in this region.

Figure 19 An Image of Chakri Grape farm showcasing farm structure

To create the dataset, two camera lenses were used simultaneously in the grape farm. Multiple image examples were taken throughout in the farm in natural light. Time of the day was also an important factor in taking images and it was made sure to take pictures during different time of the day so images with diverse lightning conditions are part of the image dataset. Camera with auto-focus and manual focus modes was used with different ISO and color values. Images were taken at different distances and conditions like multiple cluster images, single cluster images etc. This all will make sure that our dataset has contrast images

of different types and cover many different details of the farm. It will make the dataset diverse and without any bias which will make the model converge faster, be more accurate and have higher F1 score.

Figure 20 A few of many grape Images taken at the grape farm

Figure 20 showcase few of the images from the dataset. The dataset consist of 1000+ images after data cleaning. Images were collected and then preprocessed. Normally the image size ranges from 2Mb to 7Mb which will unnecessarily balloon the dataset size so the images went through batch editing where their pixels were reduced by 50% and quality was reduced to 80% from 100%. This process reduced the images size by more than a quarter of the original in some cases. This reduction in size will reduce the training time with a minute difference in quality of the image. The model will converge faster and also save the processing time.

Some other datasets were observed and it was found that a specific pattern was followed in those datasets. A same type of lens was being used under similar lightning conditions. Moreover, the distance and angle of subject from the lens also doesn't change. This makes the data better in converging and give better accuracy due to similar kind of examples but the model being trained on such dataset has more biasness and is prone to giving poorer results if it sees unseen data. This research take care of all such issues and while collecting the dataset, it was made sure that this dataset doesn't follow the trends mentioned above.

Wine Grape Instance Segmentation Dataset (WGISD)

Collected Custom Dataset

Figure 21 A comparison of Collected Custom Dataset with WGISD

A comparison of the collected dataset has been made with the WGISD [34] (shown in Fig. 21) on which considerable research has been done. The comparison shows that the images taken currently in this research have more contrast (i.e. the images show more distinctive features). Three datasets were created for this research and their respective models were trained. The first dataset was a small dataset (220 images) which was simple and was fast to converge, giving accuracy more than 90%. Second and third datasets ranged from medium to large. Third dataset consist of 1172 images to be precise and the current model was trained on this dataset.

4.2 Annotating the Dataset

After making the dataset, the research will move to an equally important part that is annotating the data. As the data in this research consists of images, the type of annotation that was suitable was Image annotation. Bounding box type of annotation were used for the images. Many different tools and software are available for the task at hand that annotate and has the ability to save in multiple formats. Our requirement is a simple yet efficient software that give adequate situational awareness and help us manage the data as labelling a large dataset can be quite confusing elsewise and waste time. Annotation required more time and technical

knowledge than the creation of dataset. An open source software named “LableImg” was used to annotate the complete data.

Figure 22 LabelImg interface

4.2.1 LabelImg

LabelImg is a common but easy to use and open source software easily available online for image annotation. It has a simple interface that shows clusters of data easy to be understood. The dataset needs to be in one directory as the software go through all data in the working directory selected. So, if the data is at multiple, random places, it’ll create confusion for the annotator to deal with it also more care will be needed to select the directory storing the annotations. Hence to avoid the confusion, data is kept in one directory.

It is able to label and save annotations in two formats including YOLO format and Pascal VOC format that makes the annotations suitable to be used with YOLO algorithm and Pascal VOC algorithm respectively. Both of these algorithms are used for object detection. As YOLO is being used in this research, the annotations will be saved in the same format in a .txt file (as shown in Fig. 22).

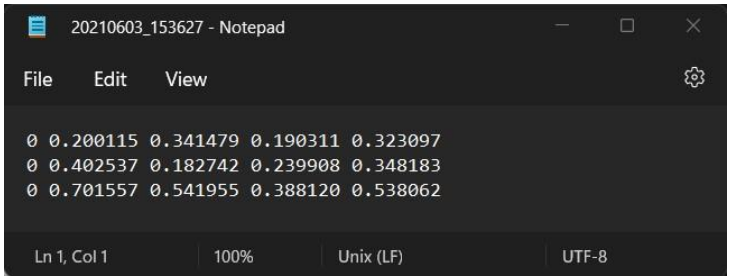


Figure 23 A file conaining annotation information of an image

The Figure 23 consist of the annotation of an image with three grape clusters thus it shows information of all those three. The first digit shows the class type which in this case is zero that is assigned to grapes and is the only class type. Second and third values represent the x and y pixel of the center of the bounding boxes of the respective clusters. Mouse will be used to draw every individual bounding box, covering exactly the area of interest. After covering all the clusters in the image, it will be saved and next imaged will be opened using the software to repeat the same process. This process will repeat for all the training images and the end of process will mark the data ready to be used for training of the model.

4.3 System Description

A core i7 9th Gen was used to handle this model. The system description is mentioned in the following table:

Model	MSI GF63 Thin 9RCX
Processor	Core i7 9th Gen
Memory	16 GB
SSD	256 GB
Graphics Card	NVidia 1080 Ti 4 GB

4.4 Training

After readying the data for training, the research will move to a more crucial process of actual model creation and training it. For this either the personal machine with enough computing power can be used or an online service like Amazon or Google Collab can be used. In this research, Google Collab was used because of limited free usage and superior computation power than the personal machine. The data was uploaded on google drive to make the model and training remote and independent of the personal machine being used.

4.4.1 Google Drive

The data will be zipped and uploaded on the Google Drive. The dataset will hold 2 x + 1 files where x is the number of examples in the dataset. Every example will have its accompanying annotation file and there will be one extra file named “classes” that will hold the information of classes in the dataset. That dataset can then be accessed by the model created on Collab after verification of the google account. This saves time for research and provides the capability to work anywhere remotely without the need of carrying the primary machine. Moreover, by uploading the images remotely, the need of individually uploading dataset again

on to the Collab will no longer be needed. Considering the large size of dataset, it will save quite some time.

4.4.2 Google Collab

The dataset will be accessed from drive using Google Collab. Collab works in two modes; TPU and GPU mode. We'll choose the GPU which is much faster and more suitable for the task at hand. Then the drive will be mounted after authentication and the relevant folder will be selected. After setting up the model's parameters and compiling it the initial weights will be downloaded and zip file of dataset will be extracted. After that the training will start and this will go on for number of hours. Google Collab use NVidia Tesla K80 GPU which can be accessed for free for a period of 12 hours after which the training will be terminated; to avoid this pro version can be considered which comes with more allocated memory (details shown in Fig. 24).

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.44       Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P8    28W / 149W |      0MiB / 11441MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Figure 24 NVidia SMI showcasing details of allocated GPU at Google Collab

A total of four models were trained on google colab using different specifications. The first model was for test purpose and consist of only 200 images that's why it will not be discussed in details. Second model (referred as Model 1) was trained for 4000 epochs and it converged at avg. loss of 16.95 percent. This model was trained at one go with no breaks and it took 17 hours and 48 minutes for it to be trained. The third model (referred as Model 2), which is the last one, will be discussed in detail. Its dataset consist of 1172 images. It was trained for 4100+ epochs and it took approximately 28 hours for it to be trained. It was trained in multiple batches at different time. When the training was stopped, the weights were saved and later used as the pre-trained weight. It made the process of training quite smooth and easy.

It gave an avg. loss of 19.47 percent (shown in Fig. 25). Its convergence can be seen in Figure 27. As it was trained in batches, it took relatively longer time to converge.

```
 1699: 0.217218, 0.194757 avg loss, 0.001000 rate, 47.848423 seconds, 282034 images, 33.789757 hours left
Loaded: 0.000043 seconds
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.727865), count: 2, class_loss = 0.985813,
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.788136), count: 6, class_loss = 2.648218,
```

Figure 25 Google Collab Training parameters of Model 2

The fourth model (reffered as Model 3 in Sec 5) was trained using a different algorithm to strengthen the validation of results. Single Shot Multibox Detector (SSD) was used to train this model using tensorflow and google colab was used for training as well as testing. The model was trained for 26100 steps and achieved total loss of 9.66% (as shown in Fig. 26).

```
INFO:tensorflow:Step 26100 per-step time 1.783s
I0506 17:24:14.790477 140438674794368 model_lib_v2.py:707] Step 26100 per-step time 1.783s
INFO:tensorflow:{'Loss/classification_loss': 0.026158836,
 'Loss/localization_loss': 0.009515376,
 'Loss/regularization_loss': 0.060975943,
 'Loss/total_loss': 0.09665015,
 'learning_rate': 0.038461637}
```

Figure 26 Training Parameters of SSD

A few issues faced while using Collab are:

- Timeout Error: Happens in case of inactivity and terminates the training.
- Training Interruption: Happens in case of connection timeout.

To take care of the mentioned issues, one need a stable internet connection and constant presence throughout the process of training. Timeout error is rather a recent addition in Google Collab algorithm that is placed there to discourage the non-interactive programming.

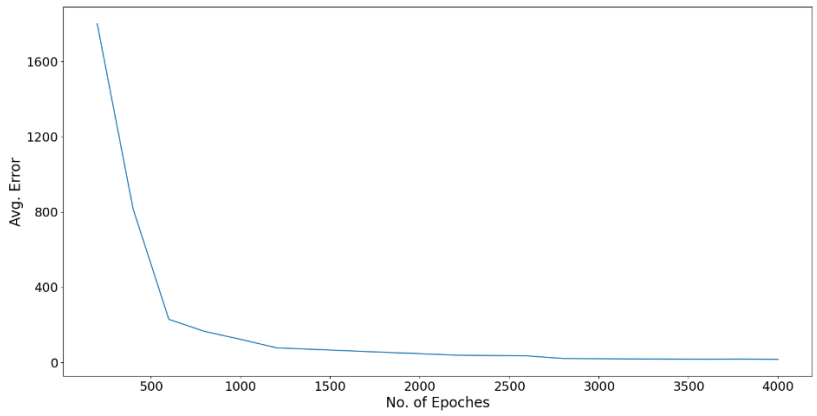


Figure 27 Graph showcasing the Convergance of model over no. of epochs

4.5 Testing and Compiling Results

After the training now the last but not least is the process of testing the model using test dataset and verify its performance. For this a python script was written and the downloaded weights from the model trained in this research was used. YOLO configurations were same here that were set-up to train the model (see Section 4.4 for details). Testing also holds much importance as it validates the models success or raises question over it.

The python script used different built-in and installed libraries like random, numpy and opencv. Numpy is a numeric computational library with exceptional capability to handle the 3D arrays; random library is used to fill in different kind of random variables in different shapes and ranges; opencv is a computer vision and image processing library with vast implications and foothold in ML, particularly in facial recognition and object detection. The weights, which have been downloaded earlier, will be inputted to the algorithm along with the path to the repository containing the test files. The algorithm will draw the box around the object of interest whenever the confidence is greater than 30% so the instances with lesser confidence will be discarded from the results.

CHAPTER 5

RESULTS & DISCUSSION

In this chapter the results of the models will be discussed in details and all the parameters that affected the results will be talked about. The performance of model will be validated over the test dataset and all notable characteristics of model will be talked about.

Table 2 represents the Confusion Matrix of Model 1 while Model 2 is represented by Table 3 below. The tables hold four values namely True Positive, False Positive, True Negative and False Negative. All the other parameters that are to be calculated by these values have been calculated and a comparative table has been compiled (refer to Table 4).

Table 2 Confusion Matrix of Model 1

True Positives	False Positives
136	11
03	N/A
False Negative	True Negative

The Model 1 consist of 150 instances in test set. Among those 150 instances 136 were detected as they were without any error, hence they were places in the category 'True Positive'. Eleven instances were either wrongly detected positive or two instances were detected as single instance, hence placed in 'False Positive' category. Only three are the instances that were positive but the model failed to detect so placed in 'False Negative'. This is a single-class classification hence no value is placed at True Negative (refer to Table 2 for details). Model 1 used transfer learning technique and trained of pre-trained weights.

Table 3 Confusion Matrix of Model 2

True Positives	False Positives
205	22
04	N/A
False Negative	True Negative

Model 2 was revised with improved labelling and more examples due to which it has bigger test dataset. We fine-tuned the Model 2 and improved its original specifications with better labelled data. It contains 205 True Positives, 22 False Positives and 4 False Negatives while the True Negatives are zero due to reasons discussed earlier. Its particulars are shown in Table 3.

Table 4 Confusion Matrix of Model 3

True Positives	False Positives
214	19
05	N/A
False Negative	True Negative

SSD was used to train Model 3 and its results on test dataset are being showcased in Table 4. It showed impressive results and converged at an excellent pace. Its True Positives are 214, 19 are the False Positives while the False Negatives are only 5. Just like Model 1, SSD also used transfer-learning technique to save time & processing power and gain better results in least amount of time.

Table 5 Performance Parameters of Model 1, Model 2 & Model 3

Model	**Accuracy**	**Recall**	**Precision**	**F1 Value**
Model 1	90.66	97.84	92.51	0.95
Model 2	88.74	98.02	90.3	0.94
Model 3	89.91	97.71	91.84	0.94

The Accuracy, Recall, Precision and F1 Value can be calculated using the values in confusion matrix (See section 2.5 for details). Table 5 shows the calculated values of different parameters that were taken from Table 2, 3 and 4. Model 1 has a high accuracy of 90.66%

while its Precision and Recall too are good giving values of 92.51% and 97.84% respectively. The success of any model depends on its F1 score. Higher F1 score means the model is successful and lower score means it failed to comply with real world situation. In this case, the model shows exceptional score of 0.95. Model 2 repeats the pattern followed by Model 1. Its accuracy is 88.74%, Precision is 90.3% and recall is 98.02% while the F1value is 0.94. It strengthens the claim of model's success. Model 2 was retrained on the better labelled data where the labelling errors were removed but the consistency in results proves the success of model. Figure 28 shows the results of the test images of Model 2. Model 1 and 2 have been trained on the same dataset, the only difference is the size of dataset. Model 2 was trained on dataset with more instances and an improved Model 2 was created by re-training Model 2 on improved data which consisted of better labelled dataset where the labelling mistakes due to human errors were removed, thus transfer learning was implemented which saved time and improved results drastically; Table 5 results showcase the same thing as the performance parameters of both models complement each other.

Figure 28 Results of Model 2

The result of SSD calculated from the confusion matrix in Table 4 have been showcased in row 3 of Table 5. The SSD is the Model 3 which was trained on tensorflow rather than Darknet. Its results have been showcased below in Figure 29. The results were calculated on

the trained model by passing the test images. The calculated accuracy of SSD is 89.91%, the recall is 97.71, value of the precision is 91.94 and lastly, the F1 score's value is 0.94. Model 3 is giving almost similar values as given by Model 2 and Model 1 despite the change in training algorithm. The reason for this is that the training data being used to train these object detection models is same and not being changed. This model validates the performance of the training data as well as the algorithm and strengthen the validity of the research being presented. Model 3 is based on SSD thus it will not be as accurate as YOLO v3 but its performance parameters are better than YOLO and is widely used and considered for real-time detection [28].

Figure 29 Results of Model 3 (SSD)

Performance parameter in Object detection doesn't end if Intersection over Union (IOU) is not discussed. IOU is the parameter that give us the relation between labels drawn and given in output by the model numerically. IOU of greater than 0.75 means the value is good, considering this the IOU value of model 2 is very good as it has an average IOU of 0.8652 (shown in Fig. 30).

Figure 30 The labelled and calculated IOU

CHAPTER 6

CONCLUSION

This research compares different state of the art object detection modules on a custom dataset with objective of developing a reliable and fast model with vast applications for grape farms in Pakistan. This research was an extensive amalgam of on-ground work, data handling & labelling. Our objective was to test our models using different sizes and types of datasets and training algorithm to compare the accuracy and performance on grapes from Pakistani grape farms.

We created different models of state of the art algorithms after making a viable and excellent custom dataset that is able to comprehend contrasted information and features of local grape farms. Datasets were trained using pre-trained weights and techniques including fine-tuning and transfer learning were used to build-up the model [15] [27]. This research proposed a model that features scalability, fast processing and high accuracy. The proposed model is trained on a large dataset that consisted of data taken at a grape farm using multiple camera sensors at different angels and distance. We built the models in such a way that the models complement each other's findings. Different models were trained using multiple state of the art algorithms including YOLO and SSD.

We experimentally validate that the trained dataset is working perfectly without the problems such as overfitting or underfitting. Our dataset's features and results were calculated using multiple parameters. Our models also achieved a balance between precision and recall which lead to high F1 score of 0.94-0.95 and accuracy between 88-90%. The IOU of the models also falls under 'good' criteria [31], its value is greater than 0.8 for all the models. We trained a total of 3 different models; two of the models were trained using YOLO while the last model was trained using SSD algorithm. One of the YOLO models used fine tuning while the second YOLO model and SSD model was trained using the transfer learning technique, which drastically reduced the training time [15]. The models were trained using google colab and produced superior detection ability under challenging situations.

It is believed that this model provides solution to the grape detection problem in regions like Pakistan reliably. This will open the doors to future agricultural products development and increase in agricultural research related to grapes and its production.

CHAPTER 7

FUTURE WORK

The future prospects of this research are several as it holds a gateway to multiple fields including robotics, farm management, plant health observation, yield estimation and creating a hand held object detection device for usage in grape farms. The model is scalable due to which it can easily be implemented using available microprocessors or general purpose computers like Raspberry Pi or Jetson Nano and can be coupled with a range of different useful sensors to extract useful information which will give a real-time awareness to farmers and agriculture engineers in the grape farm. It will create ease of usage and convenience as it can be a fast and relatively cheaper solution to many problems.

There is a lot of scope of future work and improvement in following fields:

- Dataset Creation: Training models on even bigger and more efficient datasets.
- Mobile Robot Platforms: Implementing model practically using mobile robots in farm environment to perform automated tasks.
- Improving Performance parameters: Find ways to further improve the performance parameters by using other models such as Feature pyramid [35], R-FCN [32] or Fast RNN [25] etc.

Moreover, as this research is country specific so it will accelerate research related to grapes specifically in Pakistan and in rest of the hot regions as most of the previous research has been carried out on grape farms and vineyards of Europe and South America.

REFERENCES

[1] O. Sambucci, Julian M. Alston, "Grape in the World Economy," in The Grape Genome, 2013.

[2] M. Imran, A. Rauf, A. Imran, M. Nadeem, Z. Ahmad, M. Atif, M. Awais, M. Sami, M. Imran, Z. Fatima, and A. B. Waqar, "Health Benefits of Grapes Polyphenols," *Journal of Environmental and Agricultural Sciences,* vol. 10, pp. 40-51, 2017.

[3] Y. Majeeda, M. Karkeea, and Q. Zhanga, "Estimating the trajectories of vine cordons in full foliage canopies for automated green shoot thinning in vineyards", *Computers and Electronics in Agriculture,* vol. 176, 2020.

[4] D. Trinklein, "Grapes: A Brief History," University of Missouri, Missouri, 2013.

[5] X. Liu, Steven W. Chen, C. Liu, Shreyas S. Shivakumar, J. Das, Camillo J. Taylor, J. Underwood and V. Kumar, "Monocular Camera Based Fruit Counting and Mapping with Semantic Data Association," *IEEE ROBOTICS AND AUTOMATION LETTERS,* vol. 4, no. 3, 2019.

[6] N. Clara Eli-Chukwu, "Applications of Artificial Intelligence in Agriculture: A Review," Engineering, Technology & Applied Science Research, vol. 9, no. 4, pp. 4377-4383, 2019.

[7] D. Eckstein, V. Künzel, L. Schäfer, "GLOBAL CLIMATE RISK INDEX 2021," by GERMANWATCH, 2021.

[8] O. B. Shahar, and A. Shapiro, Y.Edan, and Ron Berenstein, "Grape clusters and foliage detection algorithms," *Intel Serv Robotics,* no. 3, pp. 233-243, 2010.

[9] A. Kicherer, L. Klingbeil, A. Milioto, and Laura Zabawa, "Detection of Single Grapevine Berries in Images Using Fully Convolutional Neural Networks," in CVF Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2019.

[10] B. Millan, Maria-Paz Diago, J. Tradaguila, and Arturo Aquino, "Automated early yield prediction in vineyards from on-the-go image," Computers and Electronics in Agriculture, vol. 144, pp. 26-36, 2018.

[11] E. Kelman, and Raphael Linker, "Apple detection in nighttime tree images using the geometry of light," Computers and Electronics in Agriculture, vol. 114, pp. 154-162, 2015.

[12] Anjana K. Nellithimaru, and George A. Kantor, "ROLS : Robust Object-Level SLAM for Grape Counting," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.

[13] M. Tresanchez, D. Martinez, J. Moreno, E. Clotet and D. Font, "Vineyard Yield Estimation Based on the Analysis of High Resolution Images Obtained with Artificial Illumination at Night," Sensors, vol. 15, no. 4, pp. 8284-8301, 2015.

[14] Diego G. Aguilera, Pablo R. Gonzalvez, David H. Lopez, Mónica H. Huerta, "Vineyard yield estimation by automatic 3D bunch modelling in field conditions," Computers and Electronics in Agriculture, vol. 110, pp. 17-26, 2015.

[15] Y. Bengigo, A. Courville, and I. Goodfellow, Deep Learning, MIT Press, 2016.

[16] James P. Underwood, and S. Bargoti, "Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards," Journal of Field Robotics, 2016.

[17] T. Huang, Z. Li, S. Liu, T. Hong, and H. Huang, "Design of Citrus Fruit Detection System Based on Mobile Platform and Edge Computer Device," Sensors, 2021.

[18] H. S. Ahn, M. Nejati, J. Bell, H. Williams, B. A. MacDonald, and J. Y. Lim, "Deep Neural Network Based Real-time Kiwi Fruit Flower Detection in an Orchard Environment," in Australasian conference on robotics and automation, 2019.

[19] F. Borne, J. Sarron, E. Faye, and P. Borianne, "Deep Mangoes: from fruit detection to .cultivar identification in colour images of mango trees," in International Conference on .Digital Image and Signal Processing, 2019.

[20] A. Vardar, W. S. Lee, and F. Kurtulmus, "A. Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural .network," Precision Agri, vol. 15, pp. 57-59, 2014.

[21] Motohisa Fukuda, Takashi Okuno, and Shinya Yuki, "Central Object Segmentation by Deep Learning to Continuously Monitor Fruit Growth through RGB Images", *Sensors*, vol. 21, no. 21, 2021.

[22] Steven W. Chen, Guilherme V. Nardari, Elijah S. Lee, C. Qu, X. Liu, Roseli A. F. Romero, and V. Kumar, "SLOAM: Semantic Lidar Odometry and Mapping for Forest Inventory", *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 2019.

[23] Y. Miura, Y. Sawamura, Y. Shinomiya, and S. Yoshida, "Vegetable Mass Estimation based on Monocular Camera using Convolutional Neural Network", *in 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2019.

[24] B. Arshad, J. Barthelemy, E. Pilton, and P. Perez1, "Where is my Deer? - Wildlife Tracking And Counting via Edge Computing And Deep Learning", *in 2020 IEEE Sensors*, 2020.

[25] K. Hee, R. Girshick, J. Sun, and S. Ren, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.

[26] S. Divvala, R. Girshick, A. Farhadi, and J. Redmon, "You Only Look Once: Unified, Real-Time Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] F. Chollet, Deep learning with python, New York, Manning Publication, 2017.

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, Cheng Y. Fu, and Alexander C. Berg, "SSD: Single Shot MultiBox Detector," in Springer, 2016.

[29] Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.", *in IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4 (2017): 834-848.

[30] A. Farhadi, and J. Redmone, "YOLOv3: An Incremental Improvement," in *Computer Vision & Pattern Recognation (CVPR)*, 2018.

[31] Aurélien Géron, Hands on Machine Learning with Sci-kit Learn & TensorFlow, O'Reilly, 2016.

[32] Dai, Jifeng, Yi Li, Kaiming He, and Jian Sun. "R-FCN: Object detection via region-based fully convolutional networks." *in Advances in neural information processing systems 29*, 2016.

[33] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *in Computer Vision and Pattern Recognition (CVPR)*, 2016

[34] Thiago T. Santos, Leonardo L. de Souza, Andreza A. doc Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Computer and Electronics in Agriculture*, vol. 170, 2020.

[35] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." *in Computer Vision and Pattern Recognition*, 2017.

Completion Certificate

It is certified that the thesis titled **"Grape Cluster Detection in Grape Farm"** submitted by registration no. **00000273684**, NS **Muhammad Osama Shahzad** of **MS-18 Mechatronics Engineering** is completed in all respects as per the requirements of Main Office, NUST (Exam branch).

Supervisor: _____

Dr. Anas Bin Aqeel

Date: ____ July, 2022