الجمهورية الجزَائِرية الديموقرَاطية الشعبية

People's Democratic Republic of Algeria

وزَارة التعليم العَالي و البحث العلمي

Ministry of Higher Education and Scientific Research

University of Amar Telidji - Laghouat

Faculty of Sciences

Department of Maths and Information Technology

جَامعة عمَار ثليجي الأغوَاط

كلية العلوم

قسم الريَاضيَات و الاعلَام الآلي

Master thesis

Domain : *Mathematics and Computer Science*

Sector : *Information technology*

Option : *Information and Decision Systems*

By:

Yacine Bensadok

Apply Deep Learning techniques on Date palm trees

Publicly sustained on 30-06-2022 before the jury composed of:

Mr Mohame El Habib Maicha	**MA(A)**	**President**
Mr Lakhdar Kamal Ouladdjedid	**MC(B)**	**Examiner**
Mr Mohammed Redha Bouzidi	**MC(B)**	**Tutor**
Mr Younes Guellouma	**MC(A)**	**Co-Tutor**

Academic Year 2021 - 2022

ACKNOWLEDGMENT

First and foremost, I thank Allah Who helped me accomplish this work, and Who has been with me at all times of my live.

I especially thank my mentors, Dr. Bouzidi Mohamed and Dr. Guellouma Younes, who gave me an excellent opportunity to carry out this work, which allowed me to acquire a lot of knowledge and experience in the field of deep learning and agriculture. Their patience, their motivation and their deep knowledge always guided me.

I also thank the members of the jury composed of Mr Mohame El Habib Maicha Mr Lakhdar Kamal Ouladdjedid.

I would also like to thank all our professors specially Dr. Mohamed El Habib MAICHA and our comrades at the University of Laghouat.

Last but not least, thank you to my family for always being with me every step of the way.

ABSTRACT

The objective of this project is to implement a deep learning model for observing the state of the date palm flower, hoping to improve the production of date fruit, to detect the state of the (spathe) in images using "Deep Learning" and "Transfer Learning". We gathered a dataset containing more than 19000 images for training. We then used Transfer Learning by modifying the weights of a pre-trained YOLO model. Satisfactory learning, testing, and validation results were obtained.

Keywords: Date palm, deep Learning, Transfer Learning, CNN, YOLO, YOLOv5, Date fruit, Pollination.

L'objectif de ce projet est de mettre en œuvre un modèle d'apprentissage profond pour observer l'état de la fleur de palmier dattier, dans l'espoir d'améliorer la production de fruits dattiers, pour détecter l'état de la (spathe) en images en utilisant "Deep Learning" et "Transfer Learning". Nous avons rassemblé un ensemble de données contenant plus de 19000 images pour la formation. Nous avons ensuite utilisé l'apprentissage par transfert en modifiant les poids d'un modèle YOLO pré-formé. Des résultats satisfaisants en matière d'apprentissage, de tests et de validation ont été obtenus.

Mots clés: Palmier dattier, apprentissage en profondeur, apprentissage par transfert, CNN, YOLO, YOLOv5, datte, pollinisation.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 General introduction

In the past decade, human existence has come to be more tied to digital technologies and this connection is growing daily, so why not? We hear about and see new technology every day. We barely ever see a home or a person without a tech device, digital technologies have become an integral part of everyday life. The development of computer vision, a field that has attracted intensive research and growing attention for decades, is crucial to the success of digital systems. The goal of computer vision is to teach a computer how to "understand" a scene or feature in an image. In the age of artificial intelligence, computer vision is regarded as a game-changer.

We may claim that artificial intelligence is the current leader in all major new technologies because it is influencing most of them. The future of computer vision appears more promising than ever thanks to the incredible breakthroughs made by AI using this technology. One of the most fascinating areas of artificial intelligence, known as deep learning, is responsible for the success of computer vision systems. Deep learning has taken control of real-time computer vision's new applications, which are using the YOLO (You Only Look Once) deep neural network, which has become the standard deep model for real-time computer vision due to its superfast in real-time image recognition capabilities.

The lack of information and interest in agriculture nowadays is the biggest issue, as the majority of farmers don't care less. Take, for instance, my project's usage of gathering images of a certain fruit, in this case the date fruit. Farmers rely on the external examination of fruit: colour, shape, and bruising in order to assess the quality of their fruit and vegetable crops. As a result, quality checks are typically done by hand. However, evaluating a fruit based on its exterior characteristics is an imprecise process, and a person's opinion may differ from another's. The margin of error can be significantly reduced by using a computer vision system that can automatically grade and extract important information with a level of sensitivity closer to that of a human.

In this project we focus on date palms female flower, from a little spathe to fully opend flower full of date fruits, using deep learning algorithms to make an accurate system from it the farmer can tell in which state the flower is in and what they can do to make the crop better, for that we propose a model that do all that hust by looking through an image

Besides a general introduction and a conclusion, our master thesis is organized into three chapters. In the first chapter, we present the palm tree and the date fruit reproduction. For the second chapter, we yield the concept of neural networks, deep learning, and their models and applications. Finally, the third chapter presents our take of making YOLO-based models, implementation and the discussion of the achieved results.

CHAPTER 2

DATE PALM TREES

2.1 Generality

Date palm or *Phoenix dactylifera* L, dactylifera is a Greek word that means a finger hence the looks of the fruit, and Phoenix is the mythical Egyptian bird that can re-grow from fire similar to date palms that can survive fire damage.

It has been cultivated for more than 4000 years, that be included 200 genera and more than 2500 species also date palm is grown extensively in the deserts and semiarid regions of the world like northern Africa, the Middle East, and South Asia. While its fruits are considered a source of food and income for local people, in these sectors they play an important role in the economy, society and environment[37].

Figure 2.1: Date palm tree

2.2 Botanical classification

The botanical classification of the date palm, given by *Djerbi in 1994* is as follows:

Group : $Spadiciflora$
Branch : $Angiosperms$
Class : $Monocotyledons$
Order : $Palmals$
Family : $Arecaceae$
Subfamily : $Coryphoidae$
Tribe : $Phoeniceae$
Kind : $phoenix$
Species : $dactyliferaL$

2.3 Date palm morphology

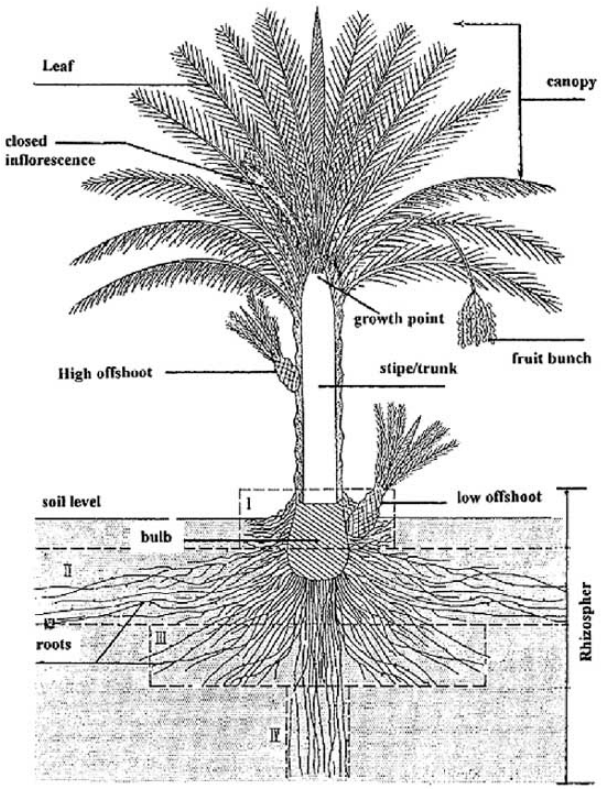


Figure 2.2: Diagrammatic construction of a date palm[43]

2.3.1 Root system

The date palm has no tap root, Its root system is of the fasciculate kind. The primary root, which grows directly from the seed, develops secondary roots. These secondary roots produce lateral roots of the same type (tertiary roots, etc.) with roughly the same diameter throughout their length.

They can be located as far as 25 meters from the palm and as deep as 6 meters distributed to 3 parts as explained in the table below , but in rich loamy soil, 85 % of the roots are distributed in a zone of 2 meters deep and 2 meters on both sides[15].

Roots order	Origin	Form	Average length (m)	Average diameter (mm)	Characteristics
Primary	Trunk base	Cylinder	4 (up to 10)	9.5 (7-12.5)	- vertical - adventitious - no root hair - conic tip called auxirhyzes and also main roots
Secondary	Primary roots	Similar to primary roots	0.20 - 0.25	3.5	- called mesorhyzes
Tertiary	Secondary roots	Similar to secondary roots but thin	0.02-0.1	0.3 - 1.5	- Low growth - short and abundant called brachyrhizes

Table 2.1: Date palm root morphology and distribution

2.3.2 Trunk

The trunk (stem) of the date palm varies extensively in dimensions and appearance among species. Generally, they are cylindrical and keep an average circumference of 1 to 1.10 m throughout their life.[25]

The stem is covered for several years with the bases of the old dry fronds, making it rough. Vertical growth of the date palm is ensured by its terminal bud, called phyllosphere, and its height can reach 20 meters.

2.3.3 Leaves

Depending on the date palm variety, age and environmental conditions, the leaves (also called "Djerida") of a date palm are 3 to 6 meters long, 0.5 meters wide with a life span ranging from 3 to 6 years.

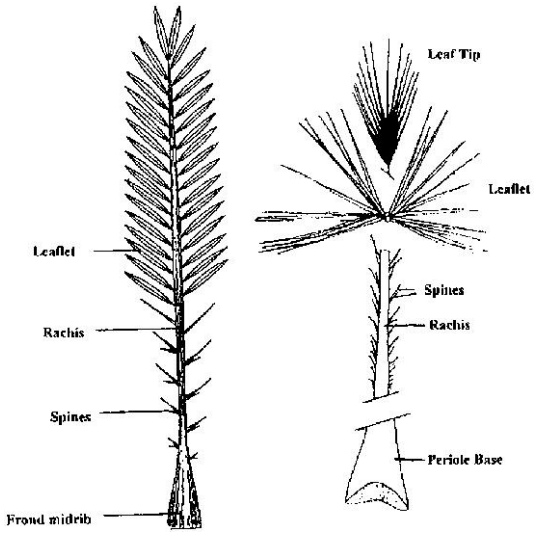


Figure 2.3: Date palm leaf characteristics[43]

The leaf compound of the frond midrib or petiole base is relatively v-shaped in cross-section with two lateral angles and one dorsal. The Rachis is bare of spines for a short distance but full of leaflets on both sides after.

2.3.4 Reproductive organs

The date palm is a dioecious species, producing male and female flowers. The flowers are pistillate (female) and staminate (male). They are born in a big cluster (inflorescence) called a spadix or spike, which consists of a central stem called rachis and several strands or spikelets (usually 50 – 150 lateral branches).

In its early stages, the inflorescence, also known as a flower cluster, is contained in a hard covering/envelope known as spathe, which breaks apart as the flowers mature, exposing the whole inflorescence for pollination. The spathe protects the flowers from the immense desert heat until they get mature and are ready to perform their function. The spathe is greenish at first, becoming brown as it gets closer to splitting (splitting is longitudinal). Male spathes are narrower and shorter than female spathes. Each spikelet has a great number of small flowers, ranging from 8,000 to 10,000 in female inflorescence to more than 10,000 in male inflorescence. A palm's typical number of spathes ranges from none to roughly 25 in females and much more in males[43].

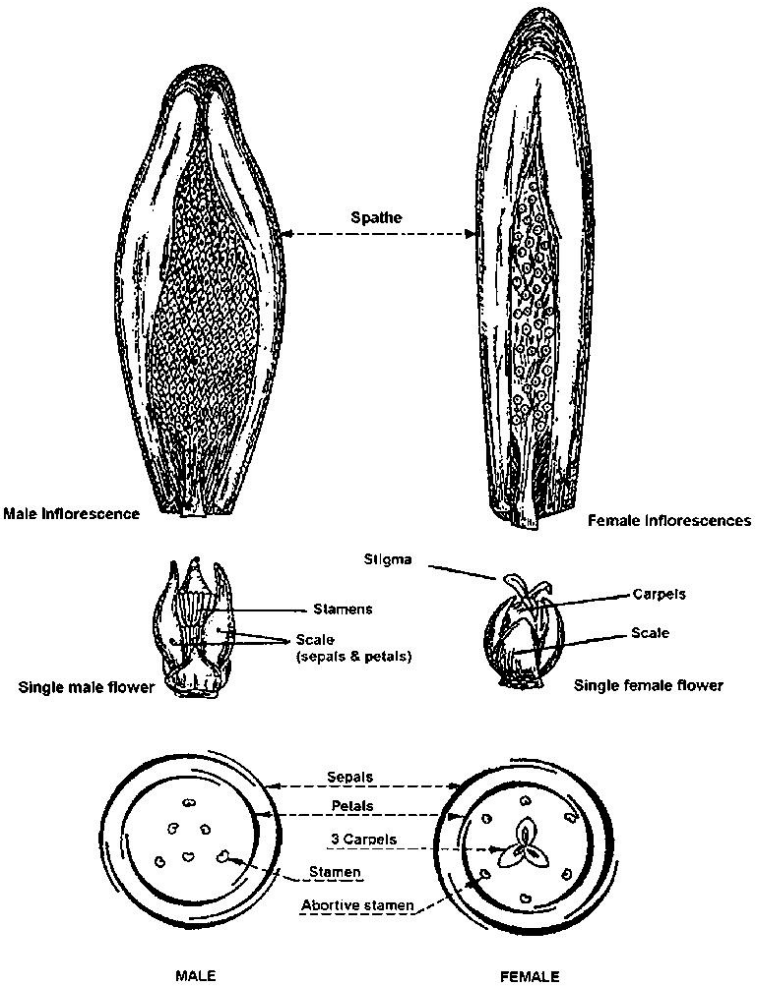
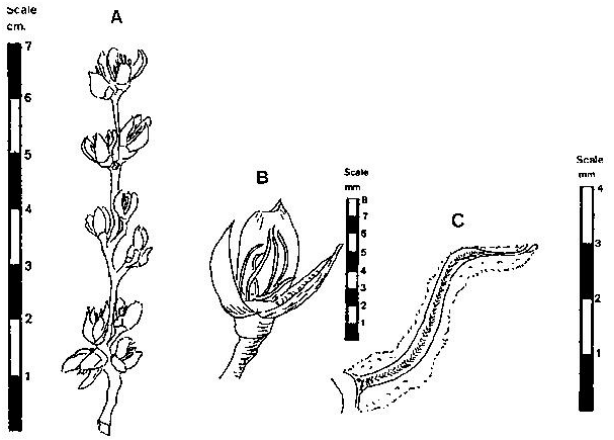


Figure 2.4: Date palm male and female inflorescences and flowers[43]

The male inflorescence

The male inflorescence is heavily crowded near the end of the rachis, whereas branches of the female cluster's inflorescence are less densely crowded. These features make it possible to determine the sex of an inflorescence before it matures.The male flower has six stamens and waxy scale-like petals and sepals, and is sweetly fragrant (3 each). Two little yellowish pollen sacs make up each stamen as shown in figure 2.5.

((a)) Male flower



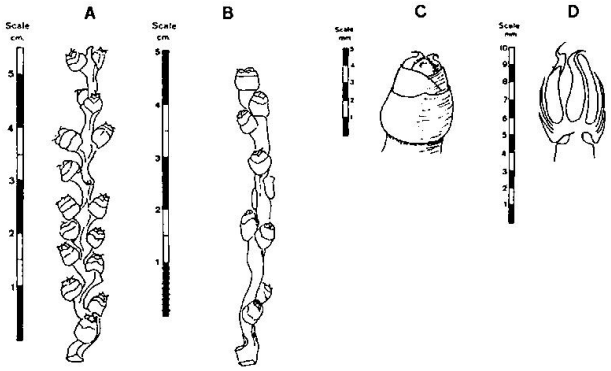
((b)) Sketch of the male flower[43]

Figure 2.5: Date palm male flower

The female inflorescence

The female flower has a size of 3 to 4 mm, rudimentary stamens, three carpels that are tightly packed together, and a superior ovary (hydrogenous). Only the tips of the three sepals and three petals diverge because they are joined together. Female flowers have a richer golden tint as they open, whilst male flowers have white-coloured dust that is formed by shaking. After the spathe bursts, the pollen sacs normally open within an hour or two. Only one ovule per flower is fertilized, resulting in the formation of one carpel, which produces a date-like fruit, while the remaining ovules are discarded. Aborted carpels appear in the calyx of mature fruits as two brown dots(figure 2.5).

((a)) Female flower

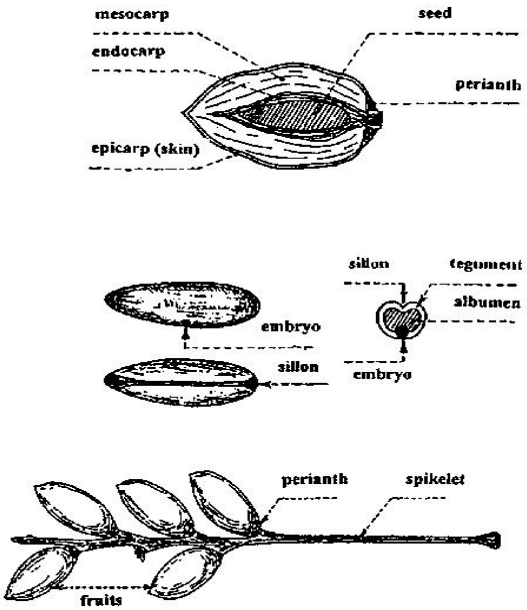


((b)) Sketch of the female flower[43]

Figure 2.6: Date palm female flower

2.3.5 Fruit

Date fruit features vary greatly depending on type, environmental circumstances, and technical care (fertilization, pollination, thinning, etc.). It is a single, oblong, terette, one-seeded berry weights around 2 to 60 grams with a terminal stigma, a fleshy/mealy pericarp, and a membrane endocarp that is botanically categorized as drupes[6].



((a)) Date fruit anatomy[43]

((b)) Date fruits

Figure 2.7: Date fruit

2.4 Date fruit types

Dates are designated into three main cultivar groups depending on moisture content at the time of ripeness or harvest, and they are[36][17]:

Soft dates (those with up to 30% moisture) they are based on inverted sugars (fructose, glucose) such as Ghars (Algeria), Ahmar (Mauritania),Kashram and Miskrani (Egypt, Saudi Arabia) ... etc.

Semi-soft dates (from 20 to 30% moisture) Deglet-Nour (Algeria) Mehjoul (Morocco and Mauritania), sifri and zahidi (Saudi Arabia).

Dried dates (less than 20% moisture content)Rich in sucrose, Degla Beida and Mech Degla (Tunisia and Algeria) and Amesrie (Mauritania).

2.5 Reproduction and pollination process

2.5.1 Gender identity

Date palm sexes are borne by distinct individuals because it is a dioecious species. Pistillate (female) and staminate (male) flowers are the two types of unisexual flowers. Pollen is produced by the male palm, whereas fruit is produced by the female palm. Flower stalks emerge from the axils of the leaves in similar locations to where offshoots emerge. The inflorescence is made up of a long, stout spathe that bursts to reveal several densely packed floral branchlets that are stout and short in males and long and thin in females. On average, one adult female palm produces 15 to 25 spathes with 150 to 200 spikelets apiece.The male blooms are waxy white and borne singly, while the female flowers are yellowish green and borne in clusters of three.

2.5.2 Pollen harvest

A male spathe that is ready to split is brown in colour and has a smooth feel. The male inflorescence matures rapidly after the spathe splits, and male flower clusters must be cut early in the morning after one or two days to prevent pollen loss either to wind or bees. They're normally collected and cured in a dry, shaded location. The strands are then separated and stored until pollination of female inflorescences is required[21].

Figure 2.8: Crop of male date palm flowers in drying chambre[2]

2.5.3 Pollination

Pollination of 60 to 80% of the female flowers is regarded as adequate and will usually result in a healthy fruit set. Pollination efficiency is influenced by a number of elements, and as a result, fruit set is highly dependent on these variables. The (usually around 09:00 to 12:00 in the morning), temperature(35C considered as optimum), the male palm's flowering period(the Northern Hemisphere's regular pollination season is March and April, while the Southern Hemisphere's is in July and August), the type of pollen, its viability and volume, and the female flowers' receptivity are all factors to consider[34].

Male and female palms should flower at the same time so that adequate pollen is available when the female is spathes open.Therefor, Within 4 or 6 days of the female spathe opening, satisfactory pollination outcomes are attained.Note that the opening of the spathes may be delayed or advanced depending on the variety and season.

2.5.4 Fruit thining

Fruit thinning is a standard procedure in most date-growing locations across the world to gain from, the benefits listed below:

- Avoiding alternancy and ensuring abundant flowering for the following season.
- Thinning will help the palm to produce consistently each year, rather than being damaged by a high crop and producing little, slender fruits the following year.
- Improving the fruit size and texture, which will have an impact on the pricing.
- Early thinning will provide more area for the fruit to develop, as well as less loss of nutrients that will need to be supplied by fertilisation.
- Fruit bunches will be lighter and more compact, which will assist harvesting and packing operations.

For that there is lot of different procedures most commun are:

Bunch thining The Deglet Nour variety's bunch thinning operation is very climate-dependent, and it aids in reducing humidity-related damage by increasing airflow around the fruits. Cutting away the centre strands will have to wait until the cluster has matured. To determine the appropriate number of flowers to remove from an

average-length strand, count the total number of flowers on the strand, which is equivalent to the strand's length. The earlier you begin thinning, the more effective it will be at increasing size. Fruit rot and souring will occur if large bunches are combined with wet conditions.

Bunch removal Each year, an adult date palm can yield 20 or more fruit bunches. An alternancy phenomena is established if next year's production is poor. At 10 to 15 years, full production with the maximum quantity and size of leaves is normally achieved. The maximum number of leaves and bunches per palm can be limited to around ten. Pruning an adult palm, as well as other long-strand variants, to 100-120 leaves (a ratio of eight to nine leaves per bunch).As a result, bunch removal is done immediately after fruit set on the following:

- bunches with poor fruit set.
- early and late bunches: small, poorly pollinated, and located at the lower and upper levels of the inflorescences production level.
- bunches that are high in number on one side of the palm (their removal will ensure palm equilibrium).
- bunches with snapped fruitstalks or broken strands.

2.6 Geographical distribution

Phoenix dactylifera is a widespread species that may be found in a broad range of geographic, soil, and climatic conditions. Although the crop has been established in California, Arizona, and Mexico in the Americas, it has also been planted in Australia, the great majority of trees are found in the Middle East and North Africa[14].

The high temperature (35 C) required for optimum pollen production and low relative humidity required for fruit setting and ripening are typical requirements in all date palm producing sites. Large amounts of water are required for desert-adapted trees, which must be extracted from deep into the soil via a well-established root system or via surface irrigation. Date palms flourish in practically rainless areas between 9 and 39 degrees north latitude.

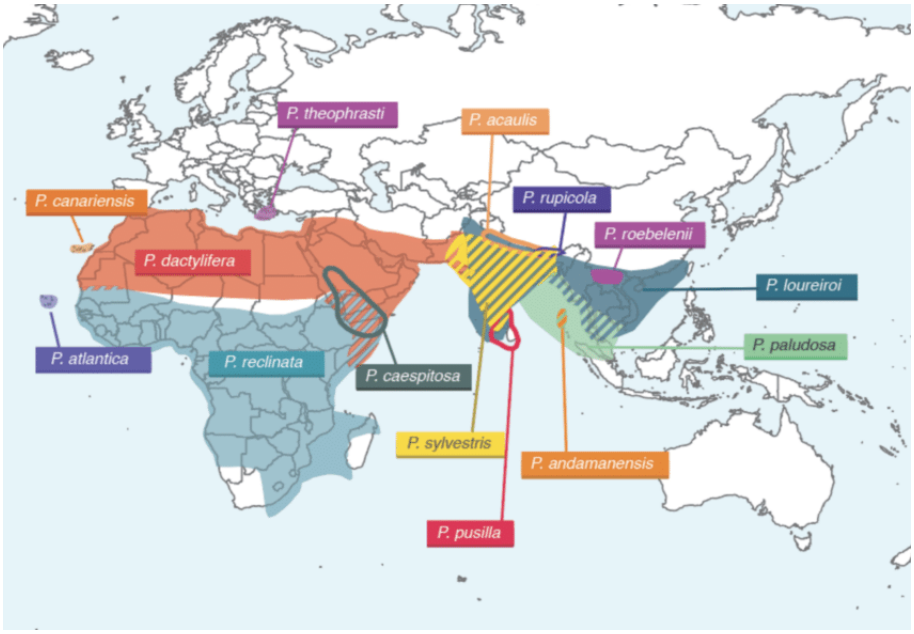


Figure 2.9: Distribution of the currently recognized Phoenix species[14]

2.7 production of dates

2.7.1 Worlwide

The first producers of dates have all experienced an increase in production. Egypt is the first producer, followed by Iran, Saudi Arabia (Table 1). According to data, Algeria is the world's fourth largest producer of dates. But from a qualitative point of view, it is considered among the pioneer countries in the production of dates which are highly appreciated worldwide (FAOSTAT, 2020).

Area	Unit	Value	Flag	Flag Description
Egypt	tonnes	1690959	Im	FAO data based on imputation methodology
Iran (Islamic Republic of)	tonnes	1283499	Im	FAO data based on imputation methodology
Algeria	tonnes	1151909	-	Official data
Saudi Arabia	tonnes	1541769	-	Official data
Iraq	tonnes	735353	-	Official data
Tunisia	tonnes	332000	-	Official data
United Arab Emirates	tonnes	328669	Im	FAO data based on imputation methodology
Morocco	tonnes	143160	-	Official data
United States of America	tonnes	56790	-	Official data

Table 2.2: World production of dates 2020 [12]

2.7.2 Algeria

WILAYA	Deglet nour		other varieties	
	Production	Rdt	Production	Rdt
	qx	kg/tree	qx	kg/tree
1 ADRAR	0	00	934562	331
3 LAGHOUAT	3 880	420	9327	810
5 BATNA	6 780	759	9963	1180
7 BISKRA	3 070 000	1,141	1653000	1996
8 BECHAR	0	00	398130	400
11 TAMANRASSET	0	00	105181	0
12 TEBESSA	9 550	441	10350	564
17 DJELFA	10 470	764	2740	1284
28 M'SILA		00	0	0
30 OUARGLA	938 022	745	712142	1237
32 EL-BAYADH	824	392	9331	469
33 ILLIZI	713	302	17479	540
37 TINDOUF	0	00	11330	273
39 EL-OUED	1 823 080	740	929020	1370
40 KHENCHELA	35 300	797	48570	1279
45 NAAMA	436	00	6069	297
47 GHARDAIA	240 000	508	364000	1094
TOTAL	**6 139 055**	**879**	**5221194**	**1138**

Table 2.3: Production of dates in Algeria 2019[1]

2.8 Importance of the date palm in Algeria

The date palm is the basis of oasis ecosystems in Algeria's Sahara, where it helps to reduce silting and provides protection for the underlying crops from harsh sun radiation (fruit trees, vegetable crops and cereals). Its existence in these arid locations allows for different types of animal and plant life that are necessary for population maintenance and survival. It also plays a significant socioeconomic role for the people of these areas, as it supplies, on the one hand, fruit, date, with unquestionable food characteristics and is a significant source of money for more than 100,000 households in the South. Furthermore, it costs 9% of agricultural exports, but a plethora of by-products (culinary, artisanal, and carpentry...).

2.9 Conclusion

In this chapter we learned about all different characteristics of the date palm especially how its reproduced from pollination to harvest. The next chapter we will define different technologies that will be used in our project.

CHAPTER 3

DEEP LEARNING

3.1 Introduction

As the concept of deep learning gains traction, academics and developers are proposing novel approaches of implementing it. Each implementation defines deep learning differently, but in general, DL refers to any form of computer system that can think logically and solve problems on its own. How can you keep track of all the different DL implementations when there are so many? What are some of the DL subfields that we'll be hearing more about in the near future?

This chapter will provide an overview and breakdown of some of the most common subfields within neural networks and deep learning, as well as crucial concepts that we need be familiar with in order to take part.

3.2 Neural Networks

The heart of deep learning algorithms is neural networks, which are one of the most used machine learning algorithms today. They are designed to transfer information across layers of linked neurons in the same way that the human brain does. These layers of neurons are known as artificial neural networks (ANNs)[27]. These networks are designed to learn to recognise patterns and make predictions based on them. An ANN must be trained with a large amount of data to achieve this. It can make predictions based on the data it has been given once it has been trained.

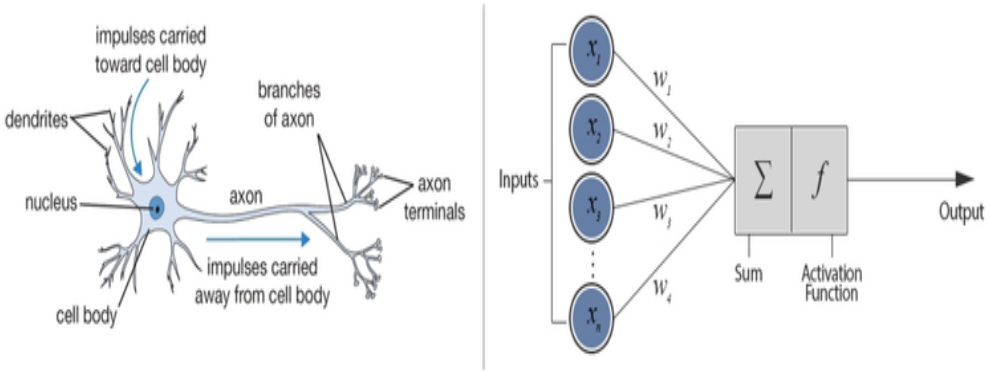


Figure 3.1: Scheme for human brain neuron and a neural network[40]

The three main components of a neural network are input neurons, hidden neurons, and output neurons. The data that has to be processed is taken in by the input neurons, and it is processed by the hidden neurons before being sent on to the output neurons. When the output neurons have finished processing the data, they interpret it and send a response back.These networks are made up of layers of "neurons" that are connected to each other. Each neuron is connected to a number of other neurons in different layers, allowing information to be passed from one layer to the next[27].

Each neuron also contains a set of weights and thresholds that may be changed to alter its behaviour and improve its precision. When a neuron receives an input, it compares it to its threshold value and, depending on whether it exceeds the threshold value, fires or remains inactive.The weights of a neuron are modified as more inputs are received, based on how closest each input was to its threshold value.

3.2.1 History

1943:A logical calculus of the ideas immanent in nerve activity was published by Warren S. McCulloch and Walter Pitts. This study aimed to comprehend how interconnected brain cells, or neurons, in the human brain could generate complex patterns. The comparison of neurons with a binary threshold to Boolean logic (i.e., 0/1 or true/false assertions) was one of the key concepts that emerged from this work[20].

1958:The perceptron was created by Frank Rosenblatt and is described in his study "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." By include weights in the equation, he advances the work of McCulloch and Pitt. Rosenblatt was able to teach a computer to differentiate between cards labelled on the left and right using an IBM 704[33].

1974:Paul Werbos was the first person in the US to mention backpropagation's use in neural networks in his PhD thesis, despite the fact that many researchers contributed to the notion.

1989:In a study, Yann LeCun demonstrates how the inclusion of restrictions into backpropagation and neural network architecture may be utilised to train algorithms. This study utilised a neural network to effectively identify hand-written zip code digits supplied by the U.S. Postal Service.

3.2.2 Neural Network key components

Weight weights are numerical values that are multiplied with inputs.. They are adjusted in backpropagation to decrease the loss. Depending on the differential between projected output and training input, they self-adjust. To put it in other words, a weight determines how big of an impact the input has on the output. A negative word, for example, would have a greater impact on the text analysis model's choice than a pair of neutral words. The signal (or the frequency of the connection) between two neurons can also be influenced by a weight[26].

Bias Biases are an added input through to the next layer that always has the value of one, means that they are constant. The prior layer has no effect on the bias units because they have no incoming connections, but they do have outgoing connections with their own weights. The bias unit ensures that even if all of the inputs are zeros, the neuron will still be activated. The purpose of bias is to change the value that the activation function produces. It serves the same purpose as a constant in a linear function[27].

Transfer function The transfer function's job is to combine input values into a single output value that can then be used by the activation function. A simple summing of all the inputs to the transfer function performs this[27].

Activation Function is a mathematical formula which decides weather the neuron switch ON/OFF by calculating the sum of weights and further adding bias, to introduce non-linearity into the output of the neuron[38].
The most common functions are:

-Sigmoid: also known as the logistic function, it's widely employed in models where the output is a probability prediction.Because we need to forecast the probability as an output, and since the value of anything only exists between 0 and 1, the sigmoid is the best choice due to its range.The function is linearly separable and provides a smooth gradient, avoiding output value jumps.Indicating that The sigmoid activation function has an S-shape.Mathematically it can be represented as:

$$S(x) = \frac{1}{1 + e^{-x}}$$

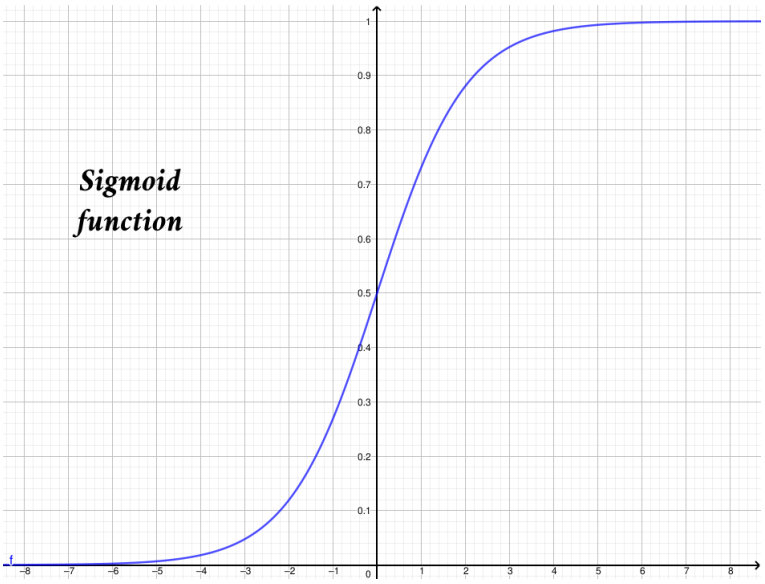


Figure 3.2: Sigmoid function graph

-Hyperbolic tangent(tanh): With a variation in output range of -1 to 1.0, the tanh function is pretty similar to the sigmoid function, and it has the same S-shape. The bigger the input (more positive), the closest the output value will be to -1.0, but the smaller the input, the more negative the output value will be. Because its values range from -1 to 1, it's commonly utilised in neural network hidden layers. As a result, the hidden layer's mean is either 0 or very near to it.Mathematically it can be represented as:

$$T(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

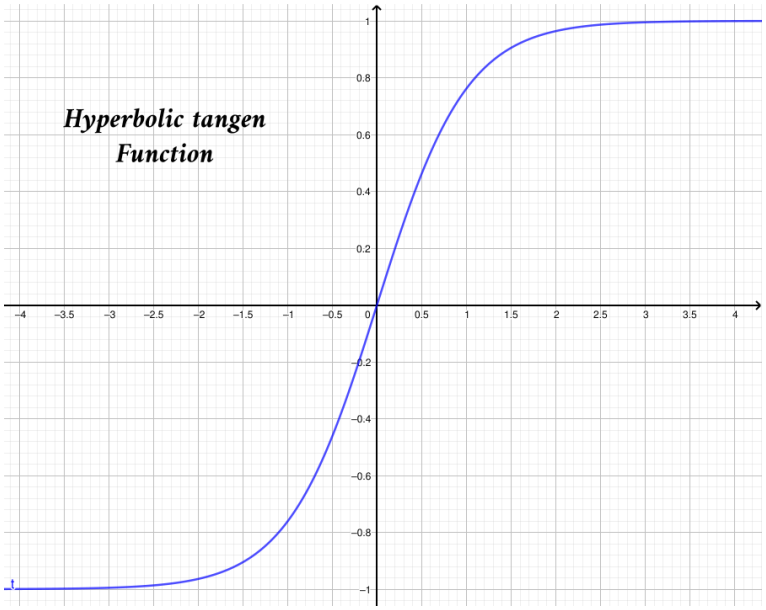


Figure 3.3: Hyperbolic tangent function graph

-The rectified linear unit (ReLU):ReLU has a derivative function that enables backpropagation while also being computationally efficient. Due to its linear, non-saturating property, ReLU accelerates the convergence of linear regression towards the global minimum of the loss function. The ReLU function does not simultaneously activate all of the neurons. Only if the output of the linear transformation is less than 0 will the neurons be deactivated. When compared to the sigmoid and tanh functions, the function is significantly more efficient.Mathematically it can be represented as:

$$R(x) = max(0, x)$$

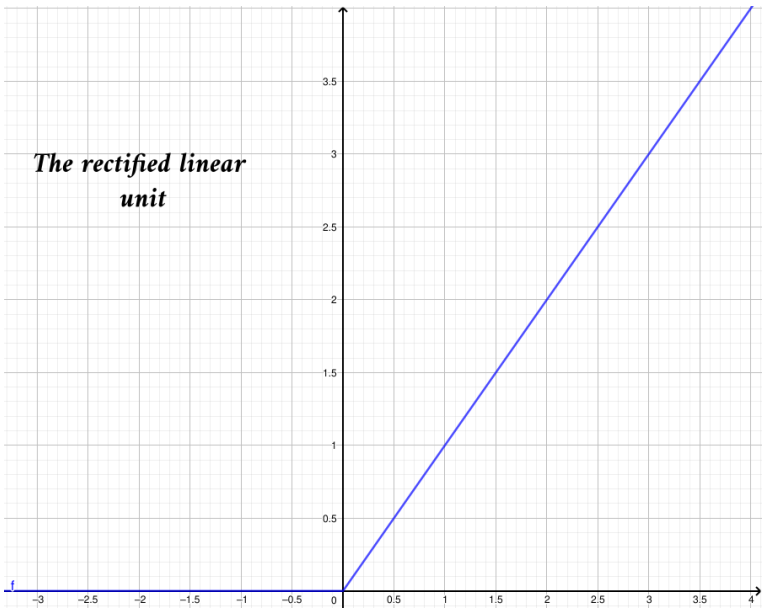


Figure 3.4: The rectified linear unit graph

-Softmax:The softmax function, like a sigmoid function, squashes the outputs of each unit to be between 0 and 1. It does, however, split each result so that the entire sum of the outputs equals one. As a result, Softmax's output is a probability distribution, which indicates the probability any of the classes is true.Mathematically it can be represented below where z is a vector of the inputs to the output layer:

$$S(\vec{z})_i = \frac{e^{z_i}}{\sum_{k=1}^{k} e^{z_k}}$$

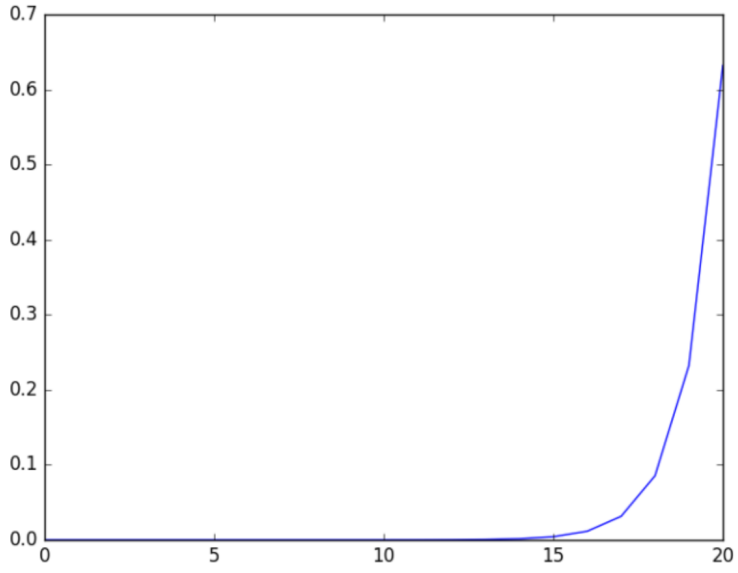


Figure 3.5: SoftMax function graph

Loss Function In a neural network, the loss function measures the difference between the expected outcome and the model's output. We may obtain the gradients that are utilised to update the weights from the loss function. The network's cost is calculated as the average of all losses. The mean squared error (MSE) is another term for this function and the most used one[26], is represented in the following equation:

$$MSE = \frac{1}{2m} \sum_{i=-1}^{m} (\hat{y} - y)^2$$

The result is simply "loss," and the value computed by the "loss function" is referred to as the "cost function" or "loss function."

Gradient Descent Is an optimization approach for tweaking machine learning model parameters or training a neural network. Its purpose is to use optimization to achieve the least or lowest degree of error. It's usually used to update the model's parameters. In this situation, parameters refer to regression coefficients and neural network weights. With each iteration(number of iterations defined by the learning rate in the hyperparameters) of parameter updates, the cost function in gradient descent works as a barometer, measuring its accuracy.
To minimise the cost function, or the difference between expected and actual y, the

model will continue to tweak its parameters until the function is close to or equal to zero[22].

$$w = w - \alpha \frac{\partial L}{\partial w}$$

w:learnable parameter $\alpha : learning\,rate$ $L : Loss\,function$

-Batch gradient descent:This is a form of gradient descent in which each iteration of gradient descent processes all of the training instances. Batch gradient descent, on the other hand, is computationally highly expensive when the number of training samples is big. As a result, batch gradient descent is not recommended if the number of training examples is big. Batch gradient descent has the advantages of being computationally efficient and producing a stable error gradient and convergence.Batch gradient descent adds the errors for each point in a training set before updating the model after all training instances have been reviewed. While batching improves computing efficiency, it can still take a long time to analyse big training datasets because all data must be stored in memory.
-Stochastic gradient descent:This is a type of gradient descent where each iteration only processes one training example. As a result, even after one iteration in which just one sample has been evaluated, the parameters are updated. Stochastic Gradient Descent adjusts the parameters based on the error's gradient in relation to a single training example. Batch Gradient Descent, on the other hand, adjusts the parameters after all of the training samples have been reviewed. The frequency of updates can also generate noisy gradients and cause the error rate to fluctuation instead of gradually decreasing, but this can be useful in avoiding the local minimum and discovering the global one.
-Mini-batch gradient descent:Because it combines Stochastic Gradient Descent with Batch Gradient Descent, Mini Batch Gradient Descent is a common approach. It simply divides the training set into small batches and updates each of them separately. As a result, it works for larger training instances and with fewer iterations. This technique gives a balance between batch gradient descent's computing efficiency and stochastic gradient descent's speed.

3.2.3 How it works

Weights are assigned once an input layer has been defined. These weights, which are assigned at random at start, assist to define the importance of any given variable, with bigger ones contributing more substantially to the output than smaller ones. As shown in the formula below, all inputs are multiplied by their respective weights, then summed and bias added to it:

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + \cdots + bias$$

The output is then run through an activation function to determine the output as shown in the formula below. If the output reaches a certain threshold, the node "fires" (or activates), sending data to the network's next layer. The output of one node becomes the input of the following node as a result of this. This neural network is referred to as a feed-forward network since data is passed from one layer to the next.

$$output = \begin{cases} 1 \text{ if } \sum w_i x_i + bias 0 \\ 0 \text{ if } \sum w_i x_i + bias < 0 \end{cases}$$

We'll use a cost (or loss) function to evaluate the model's accuracy as we train it. The mean squared error (MSE) is another term for this. The ultimate goal is to reduce our cost function in order to ensure that each given observation is correctly suited. The cost function is used by the model to obtain the point of convergence, or the local minimum, as it adjusts its weights and bias.Gradient descent is the method by which the algorithm modifies its weights, allowing the model to discover the best approach to minimise errors (or minimise the cost function). The model's parameters adjust with each training case, gradually converging at the minimum.

3.2.4 Learning process types

3.2.4.1 Supervised learning

Supervised learning is where we provide a pre-labelled data set with the correct answer beforehand to learn from, in other words the input and output value is already known. After that, the network is given a new set of examples to produce a correct outcome by processing the input data and adjusting the weights accordingly to get the desired output.

Take, for example, a set of images of various shapes from rectangles, circles and so on. From the provided labels with the data set, the model learns to predict the category of unseen image data and apply the knowledge to the test data[10].

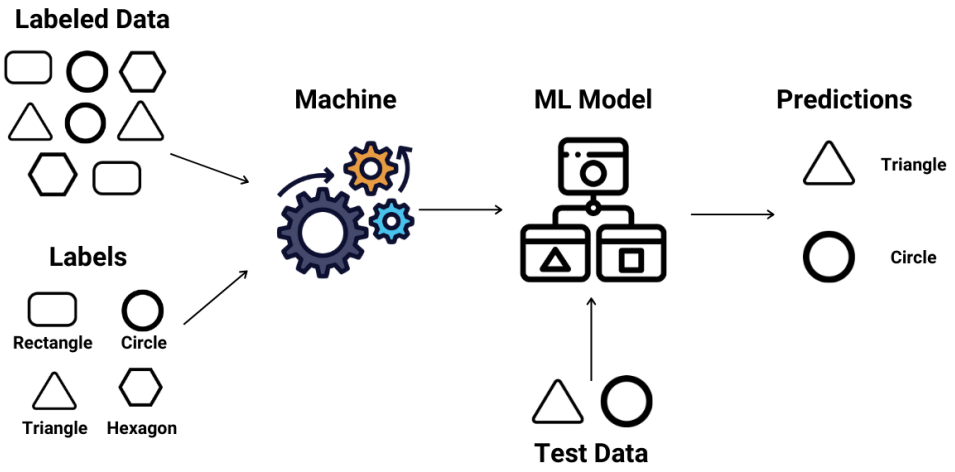


Figure 3.6: Scheme example on supervised learning[10]

Supervised learning is split into two categories of algorithms:

Classification where the output variable is a category, like 'Red' or 'blue', 'human' or 'animal'.

Regression where the output variable is a real value, like 'dollars' or 'grams'.

Like any learning process, supervised learning has both advantages and it Disadvantages:

- Data collection and data output from previous experiences are made possible through supervised learning.
- With the help of experience, it is possible to optimize performance criteria.
- It is difficult to classify huge data.
- Training for supervised learning needs a lot of computation time. So, it requires a lot of time.

3.2.4.2 Unsupervised learning

Unsupervised learning is the process of teaching a machine to operate on data that hasn't been classed or labelled and letting the algorithm act on it without supervision. Without any prior data training, the machine's task is to sort unsorted data into categories based on similarities, patterns, and differences.

In such learning, no teacher is present, which implies the machine will not be trained. As a result, the machine is limited in its ability to discover hidden structures in unlabeled data on its own. Hence they are called Unsupervised learning[10].

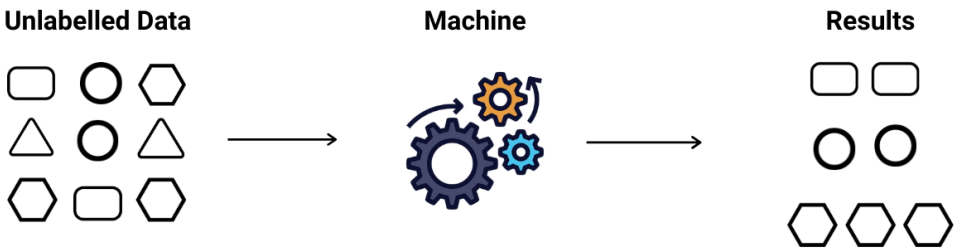


Figure 3.7: Scheme example on unsupervised learning[10]

Supervised learning is split into two categories of algorithms:

Clustering A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

Association An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Unsupervised learning is used more commonly with huge data sets due to labelling difficulty it is considered computationally complex and less accurate.

3.2.4.3 Semi-supervised learning

Unsupervised and supervised learning are combined in semi-supervised learning. It utilizes a small quantity of labelled data and a large amount of unlabeled data, combining the advantages of both unsupervised and supervised learning while avoiding the difficulties associated with locating huge amounts of labelled data. As a result, you may train a model to classify data without using as much labelled training data.

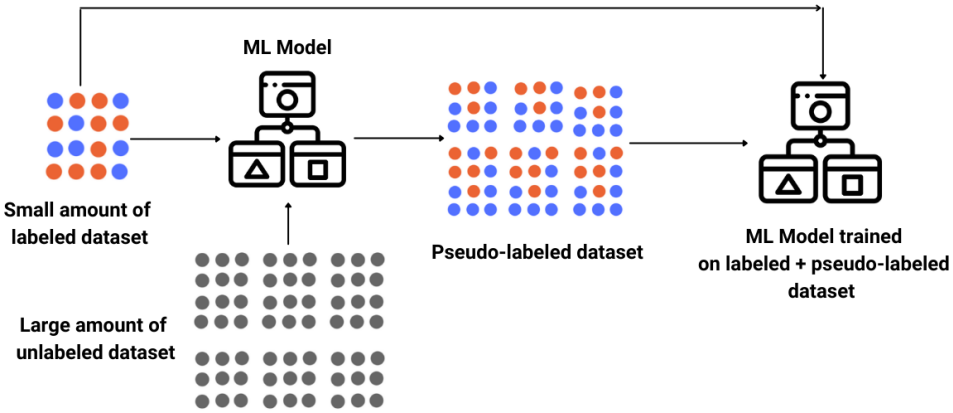


Figure 3.8: Scheme example on semi-supervised learning[10]

A text document classifier is the best example of a semi-supervised learning application. It would be nearly impossible to find a big number of tagged text documents, in this case, therefore semi-supervised learning is suitable. This is due to the fact that having someone read through full-text documents merely to assign a simple classification is inefficient.
As a result, semi-supervised learning enables the algorithm to learn from a small number of labelled text documents while classifying a large number of unlabeled text documents in the training set.

3.2.4.4 Reinforcement learning

Machine learning includes reinforcement learning. Its algorithms behave as agents in the environment, making decisions about what actions to take. The agent chooses the best action from all the possibilities available in that environmental condition and is rewarded or receives risks accordingly. The algorithms learn over time by maximizing the reward and minimizing the risk[9].

Reinforcement learning differs from supervised learning in that supervised learning includes the solution key, allowing the model to be trained with the right answer, whereas reinforcement learning does not. Instead, the reinforcement agent selects what to do to complete the job. It is obligated to learn from its experience in the absence of a training dataset.

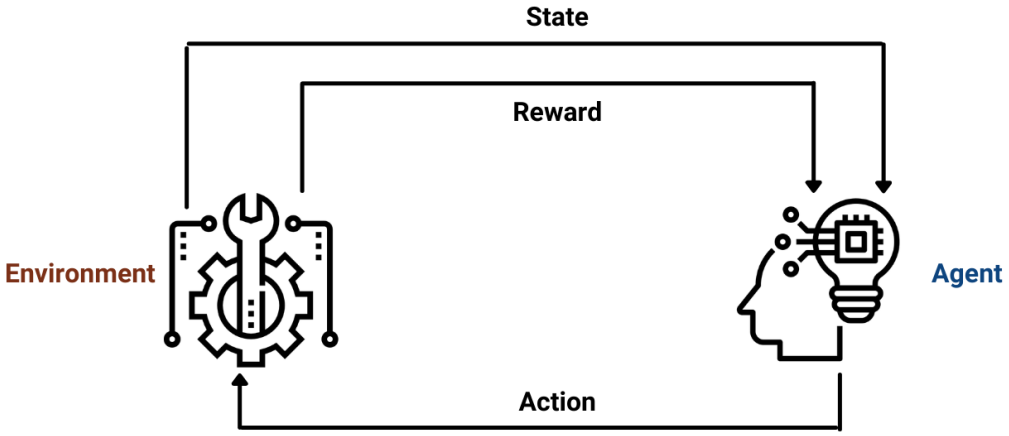


Figure 3.9: Scheme example on Reinforcement learning[10]

There are two types of Reinforcement learning:

Positive When an event occurs as a result of a certain behavior, the strength and frequency of the behavior increases. In other words, it influences behavior positively. Its main advantages:
- Maximizes Performance.
- Too much reinforcement might lead to an overflow of states, lowering the efficacy of results.
- Change can be sustained for a long time.

Negative is defined as the strengthening of behavior as a result of the elimination or avoidance of a negative situation. Its main advantages:
- Increases Behavior.
- It Only provides enough to meet up the minimum behavior.

3.2.5 Types of neural networks

Different architectures exist for neural networks. The direction of information flow is determined by the connections between their neurons. They are classified as Feed-Forward or Recurrent(back-propagation) based on network connections.

3.2.5.1 Feed-forward neural networks

In these neural networks information flows only in one direction, from the input layer to the output layer. The weights are usually not modified once they have been determined. Weights can be determined either explicitly or by the use of functions such as the Radial Basis Function. The nodes in this system work without knowing if the results they create are accurate or not (i.e., they don't re-adjust based on the results they produce). There is no feedback from the layers ahead[35].

3.2.5.2 Recurrent neural networks

To produce a result in back-propagation, information is passed from the input layer to the output layer. If there is a problem with the outcome, it will now be sent back to the preceding layers. Nodes learn how much they contributed to the incorrect solution. Then the weights will be re-adjusted. As a result t The neural network will be enhanced and improved. Information is exchanged in both directions. This essentially implements both feed-forward and back-propagation algorithms[24].

3.3 Deep Learning

3.3.1 Definition

Deep learning is a subset of machine learning where neural networks(algorithms inspired by the human brain) learn from large amounts of data. Deep learning algorithms perform a task repeatedly and gradually improve the outcome through deep layers that enable progressive learning. It's part of a broader family of machine learning methods based on neural networks[11].

3.3.2 Deep Learning models

3.3.2.1 Convolutional neural networks (CNNs)

The Convolutional Neural Network (CNN) is a type of neural network CNN, also known as a ConvNet, is a feed-forward deep neural network model for data processing that is commonly used to analyse visual images by processing them with a grid-like topology. It is designed to learn spatial hierarchies of features automatically and adaptively, from low- to high-level patterns[4].
In 1988, Yann LeCun created the first CNN, which he called LeNet. It could recognise characters such as ZIP codes and numerals.

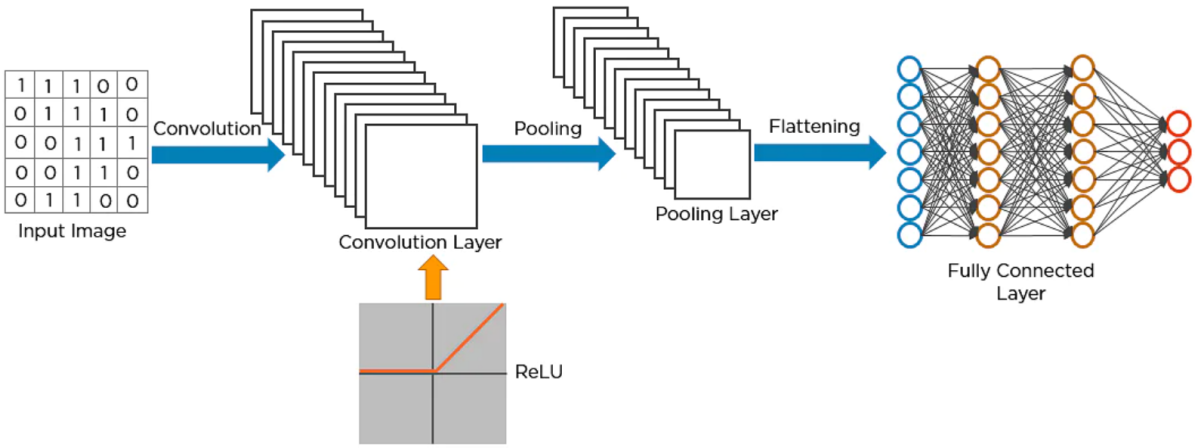


Figure 3.10: Scheme example of a CNN [4]

Convolutional neural networks (CNNs) are made up generally of four main types of layers: convolution, ReLU, pooling, and fully-connected layers:

Convolutional Layer This is the initial layer that extracts the different features from the input image. The convolution mathematical process is done between the input image and a filter of a specific size in this layer. The idea is to "drag" a frame that represents a feature on the picture and calculate the convolution product between the feature and each part of the scanned image. As a result, the convolutional layer takes multiple images as input and calculates the convolution of each with each filter. The Feature map is the output, and it contains information about the image such as edges, colour, gradient direction, and so on. This feature map is then supplied to other layers to learn more about the input image's features.

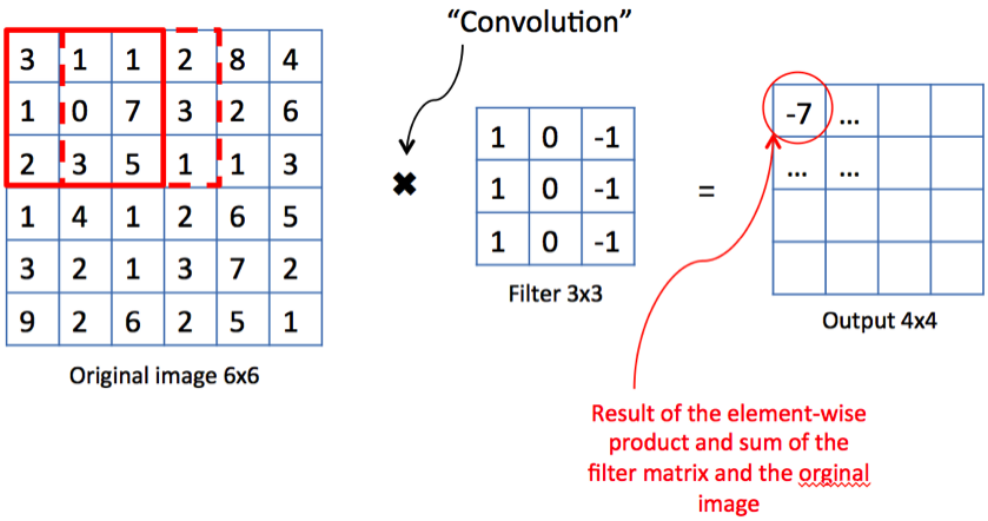


Figure 3.11: Example of a convolutional layer process[4]

ReLU Layer The ReLU activation function is utilised in this layer, and every negative value in the convolution layer's output volume is replaced with zero. This is done to avoid the values summing up to zero.

Pooling Layer This layer is frequently placed between convolution layers, it receives various feature maps and performs the pooling operation to each of them. To reduce computing resources, the pooling operation decreases the size of the photos while keeping their important characteristics. This is achieved by minimizing the connections between layers and operating independently on each feature map. There are numerous sorts of Pooling operations, depending on the method adopted:
-Max Pooling:is the most common type of pooling operation, which extracts patches from input feature maps, outputs the maximum value in each patch, and discards the remaining values. It removes all noisy activations and performs denoising and dimensionality reduction at the same time.
-Average Pooling:is used to compute The average of the elements present in the region of the feature map covered by the filter. It just takes the features from the feature map and averages them. As a noise suppressing mechanism, it simply performs dimensionality reduction.

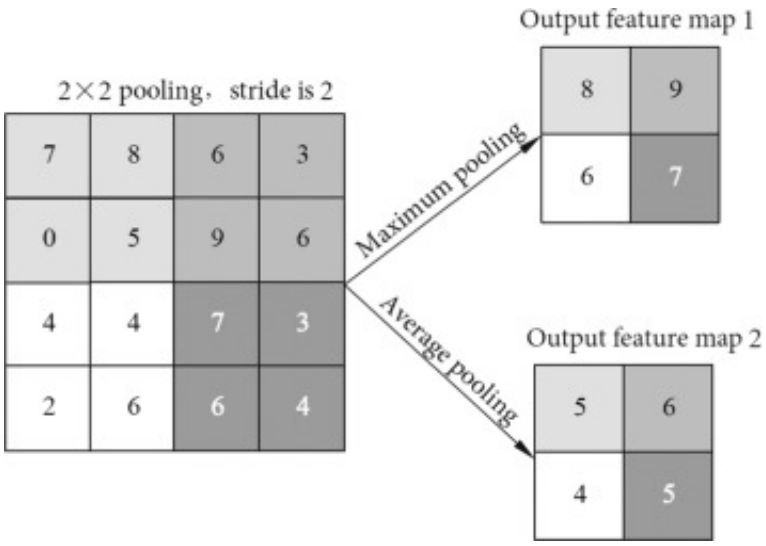


Figure 3.12: Maximum pooling and average pooling[19]

Fully-Connected layer The weights and biases, as well as the neurons, make up the Fully-connected layer, which is used to connect the neurons between two layers. The last several layers of a CNN Architecture are usually positioned before the output layer.

The prior layers' input image is flattened to a one-dimensional array of numbers or vector, which is then fed to a neural network with back-propagation applied to each training iteration. The model can distinguish between dominating and certain low-level features in images over a series of epochs and classify them using the last layer activation function.

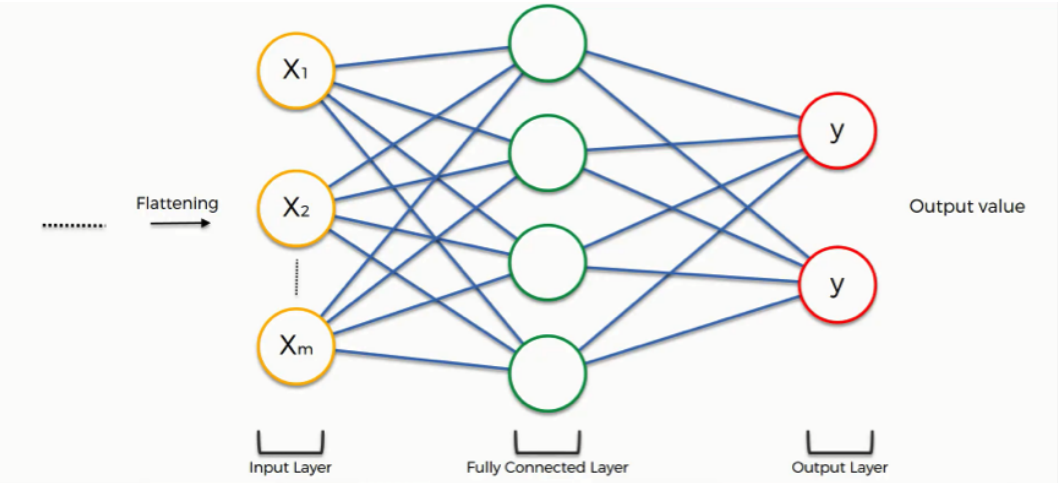


Figure 3.13: Fully-Connected layer diagram[39]

Last layer activation function The last fully connected layer's activation function is generally different from the others. Each task requires the selection of suitable activation function. The softmax function is an activation function applied to the multi-class classification task that normalises output real values from the last fully connected layer to target class probabilities, where each value ranges from 0 to 1 and all values add to 1. The table below summarises typical last layer activation function selections for various types of tasks[42].

Task	**Last layer activation function**
Binary classification	Sigmoid
Multiclass single-class classification	Softmax
Multiclass multiclass classification	Sigmoid
Regression to continuous values	Identity

Table 3.1: A list of commonly applied last layer activation functions[42]

3.3.2.2 Generative Adversarial Networks (GANs)

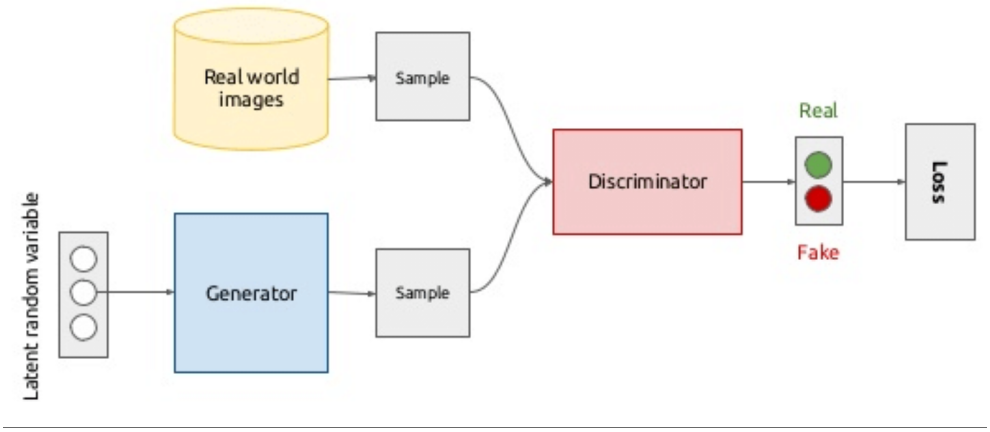


Figure 3.14: Architecture of GAN[41]

Ian J. Goodfellow and co-authors introduced Generative Adversarial Networks (GANs) in 2014.
GANs are a type of neural network that can be used for unsupervised learning. GANs, since they Learn-Generate-Improve, can create anything you feed them. It can be used to produce new samples that may or may not have come from the original dataset.
In machine learning, GANs execute unsupervised learning tasks. It is made up of two models that automatically discover and learn patterns in input data. These two models are called as Generator and Discriminator, and they compete with each other to examine, capture, and replicate variations within a dataset[22].

-Discriminator : is a neural network consisting of many hidden layers and one output layer that differentiates real data from the Generator's fake data,as shown in the below image.

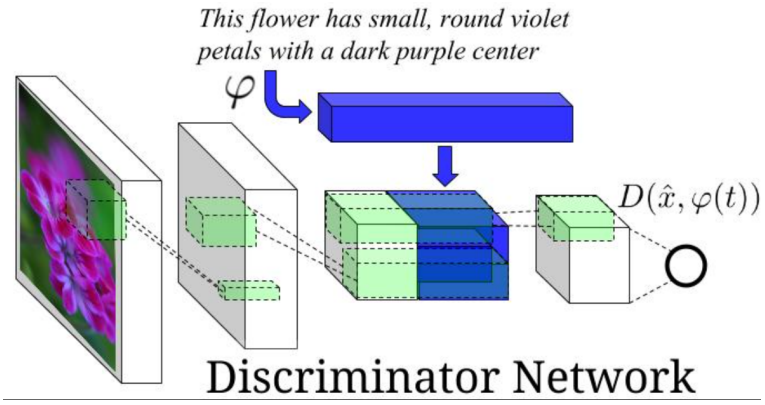


Figure 3.15: Example of a GAN discriminator[32]

The training data for the discriminator comes from two different sources:
- The real data instances, such as real photos used as positive samples by the Discriminator during training.
- During the training stage, the Generator's fake data instances are used as negative examples.

-Generator : is an Inverse Convolutional Neural Net that generates fake data for the discriminator to be trained on. It learns to use its imagination to create plausible data.

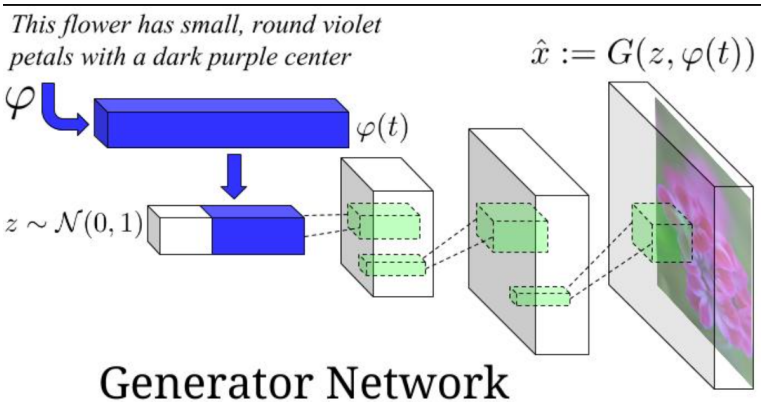


Figure 3.16: Example of a GAN generator[32]

A random value vector is given as input to Inverse-CNN, and after passing through the hidden layers and activation functions, an image is received as the output, as shown in the above image.
The discriminator uses the generated examples/instances as negative training examples. It generates a sample from a fixed-length random vector with noise.

The GANs are formulated as a minimax game, where the Discriminator is trying to minimize its reward V(D, G) and the Generator is trying to minimize the Discriminator's reward or in other words, maximize its loss. It can be mathematically described by the formula below:

$$min_G max_D V(D, G)$$

$$V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))]$$

where:
G = Generator
D = Discriminator
$p_{data}(x)$ = distribution of real data
$p(z)$ = distribution of generator
x = sample from $p_{data}(x)$
z = sample from $p(z)$
$D(x))$ = Discriminator network
$G(z)$ = Generator network

3.3.2.3 Autoencoders

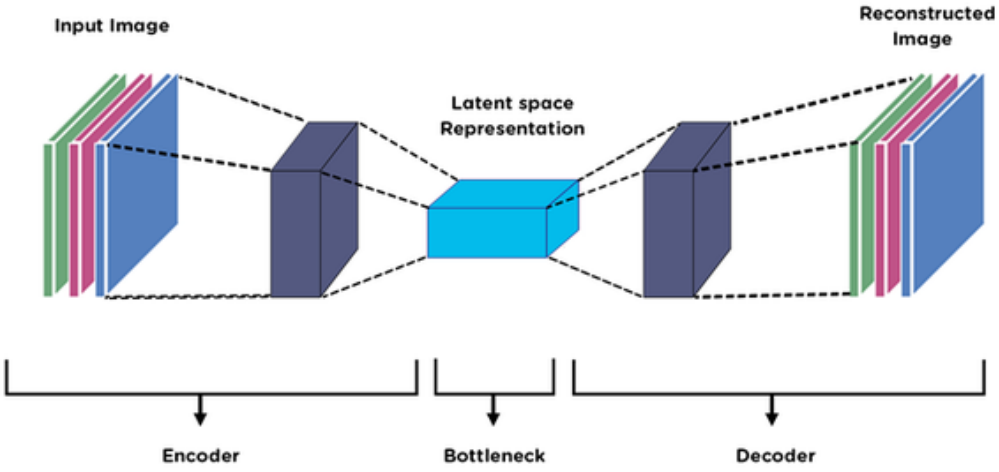


Figure 3.17: Autoencoder Architecture[10]

Autoencoders are a form of feedforward neural network where the input and output are both identical. In the 1980s, Geoffrey Hinton created autoencoders to resolve unsupervised learning problems. Its objective is to train the network to capture the most important parts of the input image in order to learn a lower-dimensional representation (encoding) for higher-dimensional data, generally for dimensionality reduction. They're utilised for things like drug development, popularity projection, and image processing[16].

As shown in the figure 3.17 autoencoders consist of three main parts:
-Encoder: is a module that compresses the input data into an encoded representation which is often several orders of magnitude less. As a result, this section is designed to focus on only the most significant and representative aspects of the data. The encoder is made up of a series of convolutional blocks that are followed by pooling modules that compress the model's input into a small section known as the bottleneck.
-Bottleneck: is a module that includes compressed knowledge representations and is therefore the most crucial component of the network. The bottleneck exists to limit the flow of data from the encoder to the decoder, enabling only the most crucial data to pass through.

The bottleneck contributes to the creation of a knowledge representation of the input. As a result, the encoder-decoder structure helps in extracting the most data from an image and establishing useful correlations between multiple inputs inside the network, as well as preventing the neural network from memorising the input and overfitting the data. As a general rule, the smaller the bottleneck, the lower the risk of overfitting.

-Decoder: is a combination of upsampling and convolutional blocks that rebuild the bottleneck's output by "decompressing" the knowledge representations and reconstructing the data from its encoded state. After that, the output is compared to the ground truth.

3.3.3 Problems encountered in deep learning

Overfitting and underfitting are two important characteristics in machine learning that are the most frequent cause of a deep learning model's poor performance.

3.3.3.1 Overfitting

Overfitting is a term used in data science to describe when a statistical model fits its training data perfectly. In other words, the model works well with training data but not so much for with test data.

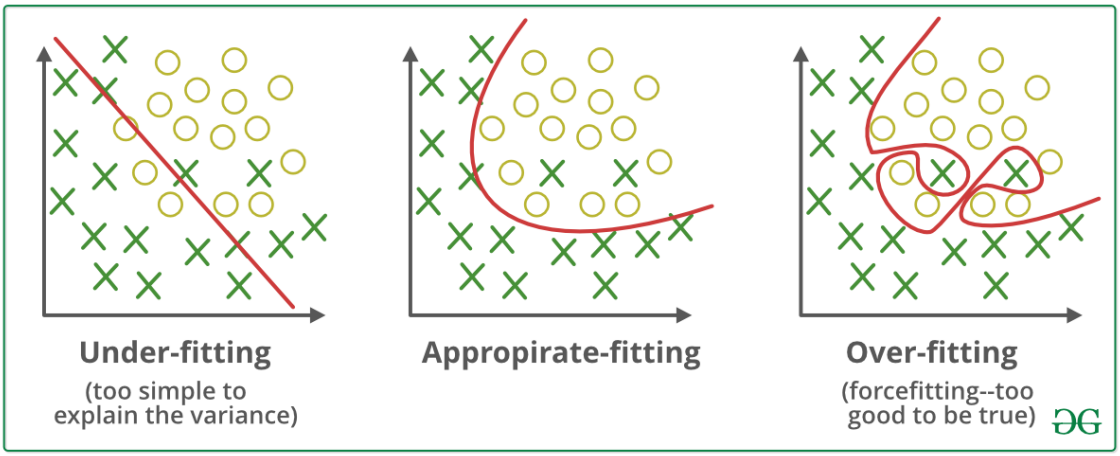


Figure 3.18: Graph example of overfitting & underfitting

When machine learning algorithms are constructed, a sample dataset is used to train the model. However, if the model is trained for too long on sample data or is too complex, it may begin to learn the dataset's "noise," or irrelevant information. The model gets "overfitted" when it memorises the noise and fits too closely to the training set as shown in figure 3.18 above, and it is unable to generalise adequately to new data[42].

Overfitting is characterised by low error rates and a high variation. To avoid this, a part of the training dataset is usually set aside as the "testing data," which is used to check for overfitting. Overfitting occurs when the training data has a low error rate and the test data has a high error rate.

How to avoid overfitting: Understanding how to detect overfitting is important, but so is understanding how to avoid overfitting entirely. A number of techniques for avoiding overfitting are listed below:

Early stopping This is an attempt to determine the "sweet spot" of a computer model's underfitting and overfitting. This strategy aims to put a stop to training before the model learns the noise. The goal is to prevent early terminating the training process, which would result in the opposite problem of underfitting.

Train with more data Extending the training set to include additional data can improve model accuracy by offering more potential to identify the dominant relationship between the input and output variables.

Data augmentation While it is ideal to integrate clean, relevant data into your training data, noisy input is sometimes used to stabilise a model. This approach, however, should be used with caution.

Feature selection Feature selection is the process of detecting the most important features in the training data and then removing the irrelevant or redundant ones. This is often confused by dimensionality reduction, although the two are not the same. Both methods assist in the simplification of the model and the identification of the data's dominant trend.

Regularization If a model is very complex, reducing the number of features makes sense. Regularization approaches can be useful if we don't know which features to delete. The parameters with larger coefficients receive a "penalty" as a result of regularisation. The amount of variance in the model is so limited.

3.3.3.2 Underfitting

Underfitting is a data science scenario in which a data model fails to accurately capture the relationship between input and output variables. It happens when a model is oversimplified, which can be caused by a lack of regularisation, longer training time, or more input features.

Underfitted models are frequently easier to recognize than overfitted models because this behaviour can be observed while utilising the training dataset. Underfitting is associated with high bias and low variance[13].

How to avoid underfitting: We can avoid underfitting and produce more accurate predictions since we can detect it based on the training set. A few techniques for reducing underfitting are mentioned below:

Decrease regularization Regularization is a method used for reducing model variance by imposing a penalty on the input parameters with the largest coefficients. Underfitting occurs when the data features become too uniform, preventing the model from identifying the dominant trend. More complexity and variation are introduced into the model when the amount of regularisation is decreased, allowing the model to be effectively trained.

Increase the duration of training As explained previously, stopping training too soon can result in an underfit model, which can be prevented by increasing the duration of training. However, it is necessary to be aware of overtraining and, as a result, overfitting, and maintaining a balance between the two will be necessary.

Feature selection Specific elements of any model are utilised to determine a given outcome. If there aren't enough predictive features, more features, or features that

are more important, should be added. You could, for example, add more hidden neurons to a neural network. This approach will add complexity to the model, resulting in improved training results.

3.3.4 Deep learning applications

Deep learning applications in the real world are a part of our daily lives, but they are usually so effectively integrated into products or services that people are unaware of the complex data processing going on in the background. Some of these examples are as follows:

Biomedicine Deep Learning is useful for biological research and applications due to its capacity to integrate massive information, apply existing knowledge, and discover arbitrarily complicated correlations. Deep Learning can already anticipate changes in cellular processes caused by genetic diversity, determine whether radiographic images suggest disease, and find chemicals that influence the activity of therapeutically relevant proteins. However, extensive research in this sector is required to fully fulfil Deep Learning's potential.

Deep Learning applications in biomedicine include genomic sequence analysis, medical image categorization, protein structure classification and prediction, and so forth.

Big Data Deep Learning enables the analysis of massive unsupervised datasets, making it an important tool for Big Data Analytics. It can extract complex patterns from huge amounts of data, as well as perform data tagging, semantic indexing, quick data retrieval, and discriminative task acceleration. In real-time data processing, powerful Deep Learning-based algorithms are crucial for accuracy and efficiency.

Examples of deep learning in big data include automated complex data extraction, learning from vast volumes of unstructured data, simulations, social media, classification, prediction, and so on.

Investment Financial analysis Another field that has benefited from deep learning is investment analysis. Predicting the market requires the tracking and evaluation of thousands of data signals ranging from earnings call conversations to public events to stock pricing. Companies employ an adaptable deep learning technology to give real-time analysis on individual shares, earnings call material, and public corporate events to institutional investors. Through robo-advisors, deep learning technology provide sound advise on financial planning.

Agriculture A deep learning model trained on crop and soil ,soil condition data can be utilised to create a system that can successfully monitor crop and soil conditions in order to estimate yield. Also plant disease and pest detection, where a DL model classifies diseased plants from healthy plants. This type of device can assist farmers in providing correct plant care before they perish. Deep learning can be used to create robots that can classify and pick crops saving huge amount of time.

3.4 Conclusion

Although the subject of deep learning remains relatively broad and encompasses many different sectors, we have defined certain fundamental ideas as well as cutting-edge deep learning techniques like convolutional neural networks and GANs, which will be very useful in future agriculture.

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 Introduction

In our work, we focused on making a deep learning model that can recognize spathes conditions and others that get the quantity of date fruit in a newly opened spathe from images. To achieve this, we proposed two YOLO-based models, one of the successful deep learning algorithms in image recognition tasks.

In this chapter we are going to discover the process of how we prepare and pre-process our dataset for train and evaluate our models with the appropriate environments and tools and the achieved results.

4.2 YOLO

YOLO(or You Only Look Once) takes a new approach to object detection by applying an end-to-end neural network that predicts bounding boxes and class probabilities simultaneously. By a wide margin, YOLO outperforms other real-time object detection algorithms in achieving state-of-the-art results. The algorithm is significantly faster than its competitors, with a maximum frame rate of 45 FPS.
Additionally, YOLO have enhanced Intersection over Union in bounding boxes and increased prediction accuracy (compared to real-time object detectors).

4.2.1 How it works

The YOLO algorithm divides the image into N grids, each of which has an equal-sized SxS region. These N grids are each in charge of finding and locating the subject they contain. Accordingly, these grids predict the object label, the probability that the object will be present in the cell, and B bounding box coordinates relative to their cell coordinates.

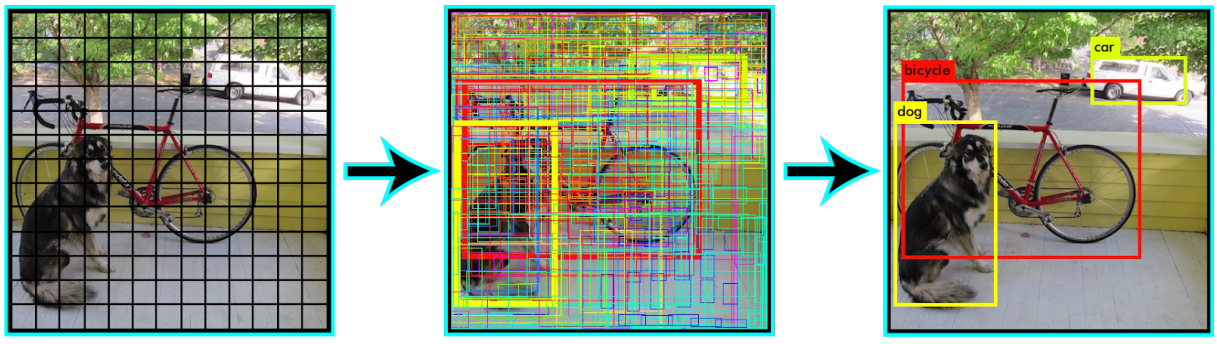


Figure 4.1: Example of how YOLO Operates

As cells from the image handle both detection and recognition, this technique significantly reduces computation, but Because of several cells predicting the same object with different bounding box predictions, it generates a lot of duplicate predictions.To solve this problem, YOLO uses Non Maximal Suppression. To do this, YOLO looks at the likelihood scores connected to each choice and selects the largest one. The bounding boxes with the largest Overlap over Union with the current high probability bounding box are then suppressed.

Up until the final bounding boxes are obtained, this phase is repeated.

4.2.2 Architecture

YOLO is a convolutional neural network, this network only uses standard layer types like convolution layer with 3x3 kernel and maxpooling with 2x2 kernel. There is no fully connected layer. The model used is composed of 9 convolution layers with permeable ReLU activations. After the maxpool6 layer the 416x416 input image becomes a 13x13 image, the latter is the Thousand of the grid that divides the image. The output of this model is a tensor lot size of 13x13x125. In this tensor, we therefore end up with 125 channels for each grid cell[29]. These 125 numbers contain data for bounding boxes and class predictions. The following information is encoded:

- Box definition composed of: (x, y, width, height, "is the object" trusted) x and y represent the center of the bounding box relative to the location of the grid cell.
- Class probabilities (only considered if "is the object" confidence is high)

YOLO is simple. It takes an input image (scaled to 416 x 416 pixels), loops it through the past convolutional network, and outputs the other end as a 13 x 13 x 125 tensor describing the bounding boxes for the grid cells. It calculates final scores for bounding boxes and discards those that score less than 30%.

The network architecture is shown in the table below:

Layer	**Kernel**	**Stride**	**output shape**
Input			(416,416,3)
Convolution	3x3	1	(416,416,16)
MaxPooling	2x2	2	(208,208,16)
Convolution	3x3	1	(208,208,32)
MaxPooling	2x2	2	(104,104,32)
Convolution	3x3	1	(104,104,64)
MaxPooling	2x2	2	(52,52,64)
Convolution	3x3	1	(52,52,128)
MaxPooling	2x2	2	(26,26,128)
Convolution	3x3	1	(26,26,256)
MaxPooling	2x2	2	(13,13,256)
Convolution	3x3	1	(13,13,512)
MaxPooling	2x2	1	(13,13,512)
Convolution	3x3	1	(13,13,1024)
Convolution	3x3	1	(13,13,1024)
Convolution	1x1	1	(13,13,125)

Table 4.1: YOLO archetecture

4.2.3 YOLOs different versions

4.2.3.1 YOLOv1

The first YOLO version was introduced in 2015 in the article "You Only Look Once: Unified, Real-Time Object Detection" by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi.
YOLO quickly took over the field of object recognition and became as the most popular algorithm because to its speed, accuracy, and ability for learning. The authors solved

the regression problem with different geographical bounding boxes and associated class probabilities using a single neural network, as compared to approaching object detection as a classification problem.
The YOLOv1 processed images in real-time at a speed of 45 fps, whereas Fast YOLO, a more smaller model, processed images at 155 fps while still achieving twice the mAP of other real-time detectors[29].

4.2.3.2 YOLOv2/YOLO9000

YOLOv2 (also known as YOLO9000) was released a year later, in 2016, by Joseph Redmon and Ali Farhadi in the article "YOLO9000: Better, Faster, Stronger"[30].

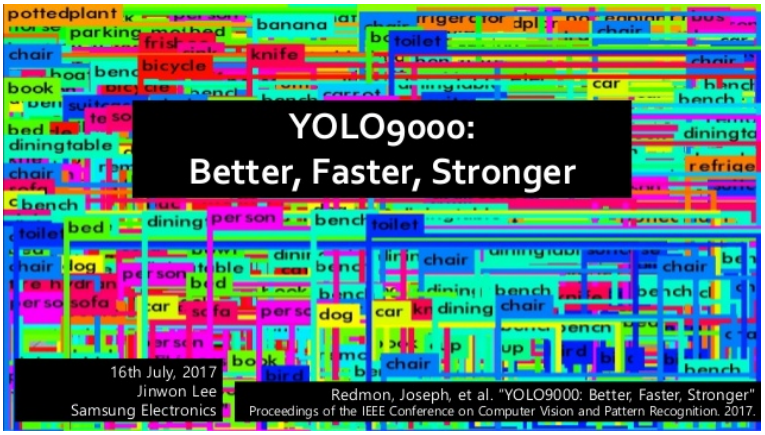


Figure 4.2: YOLO9000[30]

The model was given the number 9000 since it could predict even 9000 different item categories while still working in real-time. The new model version received Darknet-19 as the new baseline model in addition to being simultaneously trained on object detection and classification datasets.
Other engineers started experimenting with the algorithm and creating their own, distinctive YOLO versions as a result of YOLOv2's popularity and swift rise to the position of the next state-of-the-art object detection model. Some of them will be discussed bellow.

4.2.3.3 YOLOv3

In 2018, Joseph Redmon and Ali Farhadi released a new version of the algorithm in the publication "YOLOv3: An Incremental Improvement." It was constructed using Darknet-53 technology. The softmax activation technique in YOLOv3 was modified to independent logistic classifiers.
During training, the binary cross-entropy loss was used. With 53 convolutional layers, Darknet-19 has been enhanced and called Darknet-53. In addition, the predictions were performed on three different scales, which improved YOLOv3's ability to predict minute details[31].

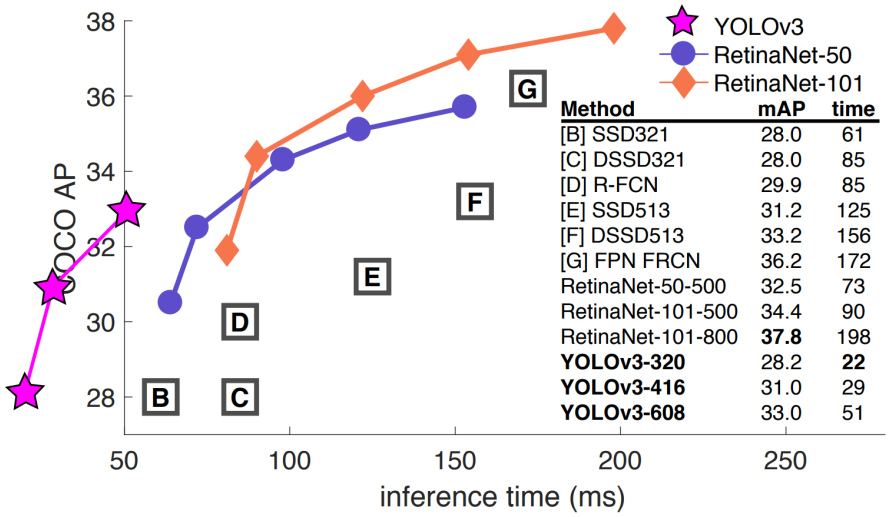


Figure 4.3: Comparison between YOLOv3 and other models[31]

Since Joseph Redmon decided not to continue working on YOLO improvements, YOLOv3 was the last version of the YOLO. These days, it mostly serves as a foundation for developing more object-detection architectures.

4.2.3.4 YOLOv4

As an improvement to YOLOv3, Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao published "YOLOv4: Optimal Speed and Accuracy of Object Detection" in April 2020. The algorithm produces results at an average precision of 43.5 percent while operating at 65 frames per second.
As part of the SPDarknet53 design, YOLOv4 proposes adding Weighted Residual Connections, Cross Mini Batch Normalization, Cross Stage Partial Connections, Self Adversarial Training, and Mish Activation[5].

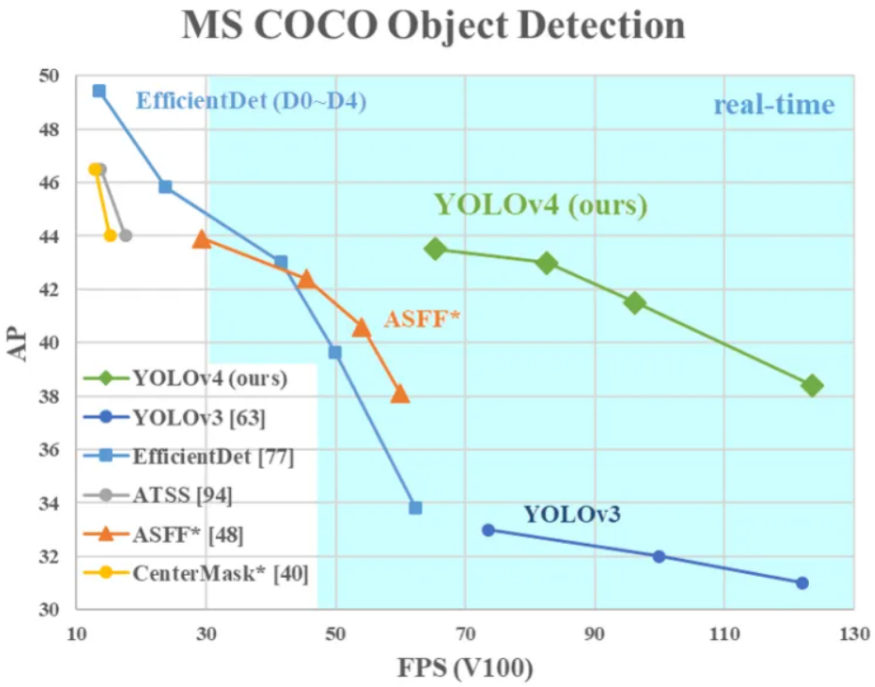


Figure 4.4: Comparison between YOLOv4 and other models[5]

Although it is considered member of the YOLO lineup, YOLOv4 was created by different researchers. Its architecture is composed of the SPDarknet53 backbone, spatial pyramid pooling, PANet path-aggregation as the neck, and YOLOv3 as the head.

4.2.3.5 YOLOv5

YOLOv5 is an open-source project that consists of a collection of object identification models and detection techniques based on the YOLO model pre-trained on the COCO dataset, with simple capabilities for TTA, model assembly, hyperparameter development, and export to ONNX, CoreML, as well as TFLite. It represents Ultralytics's open-source study into the future of Object Recognition works and is maintained by the organization[18].

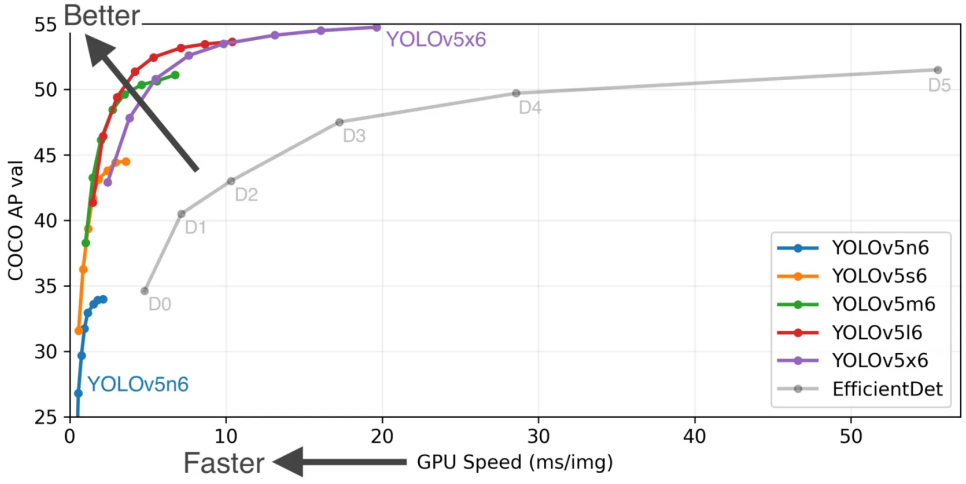


Figure 4.5: YOLOv5 models speed[18]

4.3 Tools and hardware used

4.3.1 Tools used

4.3.1.1 Python programming language

Python is a general-purpose, interpreted programming language. Python was developed by Guido van Rossum and originally made available in 1991. Its design philosophy places a strong emphasis on code readability and makes extensive use of whitespace. On both a local and a large scale, it offers constructs that enable clear programming[28].

Python has an intelligent memory management system and a dynamic type system. It features a sizable, thorough standard library and supports a variety of programming paradigms, including imperative, functional, object-oriented, and procedural.

4.3.1.2 Pytorch

Developed primarily by Meta AI, PyTorch is an open-source machine learning framework that is based on the Torch library and is used for tasks like computer vision and natural language processing. The Modified BSD licence regulates its distribution[7]. PyTorch features also a C++ interface, even though the Python interface is more refined and the main focus of development. PyTorch offers the following two top features:
Tensor computation (similar to NumPy) with strong GPU acceleration.
Deep neural networks based on a tape-based automatic differentiation systems.

4.3.1.3 Jupyter notebook

A server-client programme called the Jupyter Notebook App enables editing and running notebook papers from a web browser. The Jupyter Notebook App can be used locally on a desktop or remotely by installing it on a server.
An organised list of input/output cells in a Jupyter Notebook document that can also contain code, text written in Markdown, math, graphs, and rich media A notebook is a JSON document that follows a configurable format and typically ends in ".ipynb" extension[3].
The Jupyter Notebook App contains a "Dashboard" (Notebook Dashboard), a "control panel" exposing local files and letting to open notebook documents or shutting down their kernels, in addition to displaying, editing, and executing notebook documents.

4.3.1.4 Google Colab

Colab is a completely cloud-based Jupyter notebook environment that is free to use. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously modified by your team members - just the way you edit documents in Google Docs, not to mention it grants access to GPUs free of charge. Many well-known machine learning libraries are supported by Colab and are simple to load in your notebook[44].

4.3.1.5 LabelImg

LabelImg is an open source, free programme for graphically tagging images . It uses QT for its graphical user interface and is developed in Python. For an object identification project, it's a simple and cost-free technique to identify a few hundred images[8].
The PASCAL VOC format, which is the format used by ImageNet, is used to save annotations as XML files. Additionally, it supports the CreateML and YOLO formats.

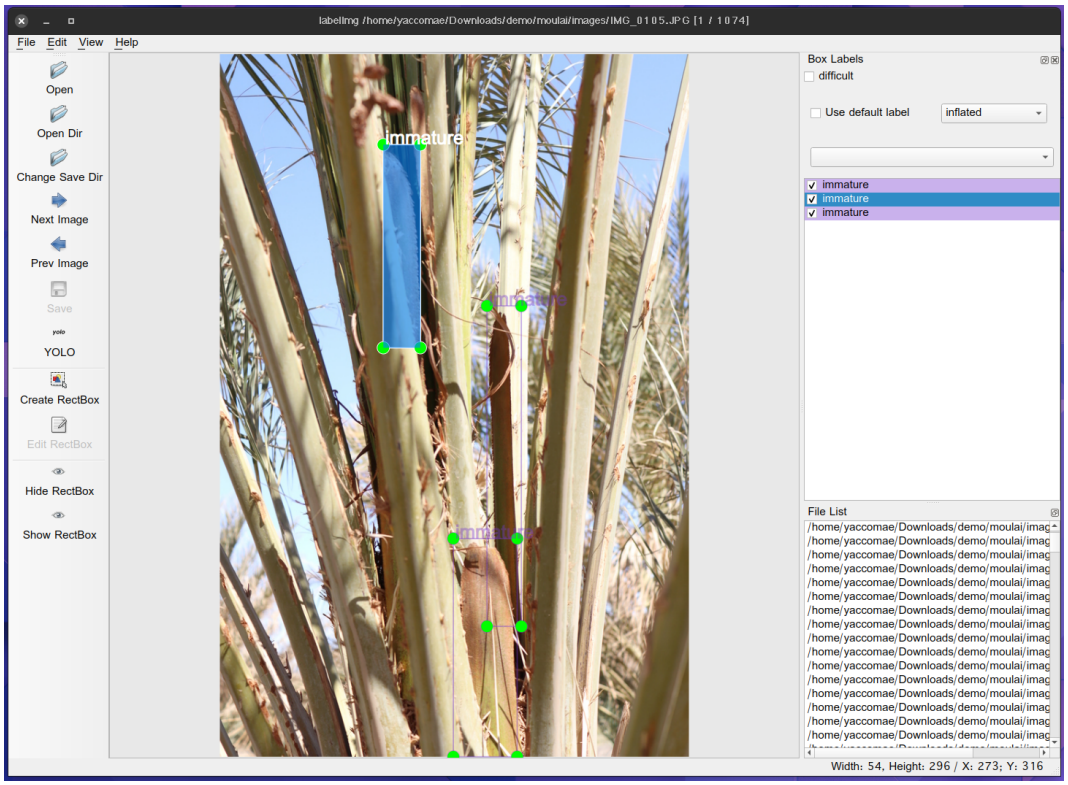


Figure 4.6: LabelImg tool UI

4.3.2 Hardware used

4.3.2.1 Workstation

For the workstation used in this project , it was provided by the university's laboratory with specs below:

- OS: Ubuntu 20.04.4 LTS (Focal Fossa)
- CPU: intel xeon
- 2xGPU: NVIDIA Quadro RTX 5000 GDDR6-16GB Clock-1815 MHz
- RAM: samsung 32GB-2400Mhz CL16
- Storage: Seagate 3.5 HDD 1000GB

4.3.2.2 DSLR camera

The camera used for this work was a Nikon DSLR D3300 camera[23] (Figure 4.7) provided by the university's laboratory with specs below:

- ISO: 100 - 12800 / Hi-1 (ISO 25,600)
- Image Area (pixels): up to 6000x4000 (RAW)

- Movie: up to Full HD 1,920x1,080 / 60 fps
- Storage: 64GB SansDisk SD card c10
- Lense: AF-P DX NIKKOR 18-55mm f/3.5-5.6G VR

Figure 4.7: D3300 Nikon camera[23]

4.3.2.3 Google colab hardware

- OS: Ubuntu 18.04.5 LTS x86_64
- CPU: Intel Xeon (2) @ 2.199GHz
- GPU: NVIDIA Tesla T4 GDDR6-16GB Clock-1590 MHz
- RAM: 12.986 GB
- Storage: google Drive 78.19 GB

4.4 Experimentation

4.4.1 Dataset gathering & preparation

Our work focuses on classifying the different states of the spathes and counting how many date fruits there are in each spathe after opening, a large dataset is a must.

At first we gathered a set of videos captured by a smartphone camera with a resolution of 720P of diffrent date palms in Mr bouzidis garden captured last year that has almost 28 date palms, this dataset had in it videos of each palm captured everyday for two months of mars and april.

We transferred the provided videos to a set of images for labeling. While doing so, we noticed that due to poor quality and also the loss of it while translating them into images, the set lacked features that help classify the spathe's status from color, size, girth..., As

a result, the training could not be resourceful with this kind of resolution, we concluded that a higher resolution is needed and abandon this set.

For the new set, we used the DSLR camera mentioned earlier in the hardware section to satisfy the problem of the first set. As a result, the camera captured 6000x4000px RAW images. We chose two date palm gardens to collect the data, Mr Moulai's garden containing more than 200 date palm trees and Mr Bouzidi's having more than 50 date palms, both located in Laghoaut province. The process is done in 2 to 3 sessions per week at a timetable of two months (Mars-April). Each date palm is captured with 360 degree style images.

The resulted outcome of the capturing sessions of this dataset is almost 19000 diffrent RAW images of more than 120 date palms with a resolution of 6000px by 4000px.

While this dataset satisfied all the features that lacked in the previous one ,due to the huge number of images and the short time, we couldn't use all the images so we chose about 2000 pictures where those pictured are more focused on the spathe rather than the whole date palm for labelling.

In the labelling part using labelimg, we started with 870 images. For the spathe's status, we chose three main labels for the classification models. They are "immature", "inflated", and "open". All labels describe the status of the spathes in their growing lifetime. These 870 images resulted in about 2501 different labels divided to: 451 for "open", "inflated" about 1291, and "immature" 759.

Data splitting is as important as choosing the right dataset, since the splitting ration effects the training result, in our training we used the 80/20 splitting ratio which also named the Pareto Principle. The overall point of this principle is that, in most circumstances, 20% of causes result in 80% of outcomes. Despite having a wealth distribution-based prologue, the concept statistically comes near to explaining a wide range of human, technological, and environmental events. For this reason, many people use an 80/20 split for training and validating.

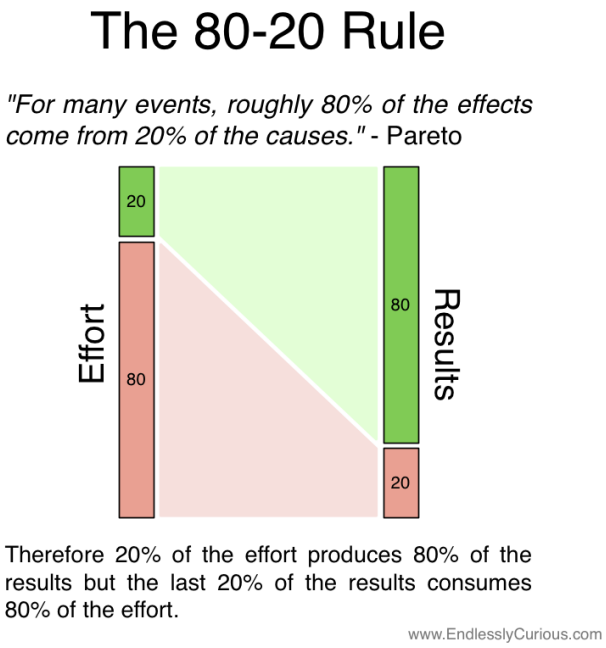


Figure 4.8: The Pareto Principle (80/20 rule)

While working with the model, we encountered another problem related to the resolution used in the dataset. The problem is that the YOLO model accepts a Max resolution of 1024px. Any image bigger than that. The model tries to reduce the resolution to its accepted resolution, resulting in the process losing all the features in the labels, also making the labels too small for the model to recognize.

The solution to this problem is compression by reducing the resolution and compressing the details in the new resolution. The software used for this process is ImageMagick. ImageMagick is a collection of command-line tools for editing and manipulating images. It can swiftly process a single image from a terminal, process lots of images in a batch, or be incorporated into a bash script. The script used in this process is:

bash:¿ $convert\ original.png\ -quality\ 95\ -resize\ 1024\ newRes.jpg$

We got a new dataset with a resolution of 1024px while the features reserved without the need of labeling again since YOLO labels dont use pixels as guide.

4.4.2 Model training

Before the training starts we must create dataset.yml file that have all information about the used dataset (files location both images and annotations, number an d names of used labels).

After that, we tinkered with both the hyperparameters and configuration of the YOLO model to reach the optimum results, the initial configurations we started with is as follows:

- **hyperparameters**:
  lr0: 0.01 lrf: 0.01 momentum: 0.937 weight_decay: 0.0005 warmup_epochs: 3.0 warmup_momentum: 0.8 warmup_bias_lr: 0.1 box: 0.05 cls: 0.5 cls_pw: 1.0 obj: 1.0 obj_pw: 1.0 iou_t: 0.2 anchor_t: 4.0 fl_gamma: 0.0 hsv_h: 0.015 hsv_s: 0.7 hsv_v: 0.4 degrees: 0.0 translate: 0.1 scale: 0.5 shear: 0.0 perspective: 0.0 flipud: 0.0 fliplr: 0.5 mosaic: 1.0 mixup: 0.0 copy_paste: 0.0

Future changes will be made in the training as we mention them in the next section below.

4.5 Results and discussion

4.5.1 Results of the date fruit counter model

Weight	Optimizer	Training time	Precision	maP .5
YOLOv5x	Adam	17min(colab)	22.1%	12.1%
YOLOv5x	Wadam	20min(colab)	38.7%	27.9%
YOLOv5x	sgd	34min(colab)	72.9%	70.1%
YOLOv5x	sgd	3.1hours(colab)	80.7%	80.5%

Training01 : We used YOLOv5x weight, 'Adam'as optimizer,100 epochs, learning rate of 0.01, 16 batch size we get in 17 minutes about 22.1% precision and 12.1% maP.5, which it is an improvement but not close to enough.

Training02 : We used YOLOv5x weight, and changed the optimiser to 'WAdam',100 epochs, learning rate of 0.01, 16 batch size we get in 20 minutes about 38.7% precision and 27.9% maP.5, which it is about 15% improvement, but not even close to enough to be considered a usable model.

Training03 : We use the same parameters only chnging both the optmizer to 'SGD', and increase the epoch to 1000. the model used only 374 epochs due to reaching the learning limits, we get in 34 minutes about 72.9% precision and 70.1% maP.5. This considered a huge leap from earlier tests, the model considered to be usable but not quite good.

Training04 : We used in this test the same configuration before change only the learning rate to 0.001. after 3.1 hours we got 80.7% precision and 80.5% maP.5, this is a remarkable improvement and considered a great model.

4.5.2 Results of the spathe classifier model

			Precision			
Weight	**Optimizer**	**Training time**	**immature**	**Inflated**	**Open**	**Average**
YOLOv5s	SGD	1.58H (workstaion)	62.6%	77.7%	78.9%	73.1%
YOLOv5m	SGD	2.45H (workstaion)	63.1%	82.9%	86.3%	77.4%
YOLOv5L	SGD	3.03H (workstaion)	67.8%	86.2%	86.5%	80.2%
Transfer Learning	SGD	1.71H (colab)	85%	70.9%	100%	85.3%

Training01 We used YOLOv5s weight, image size 1024, 'SGD'as optimizer,1000 epochs, learning rate of 0.01, 8 batch size(due to bottleneck of the gpu memeory 8 was the maximum we could do) we got after 1.58 hours: only 302 epochs used we reach an average precision of 78.9% and yet the immature label precision still a bit low.

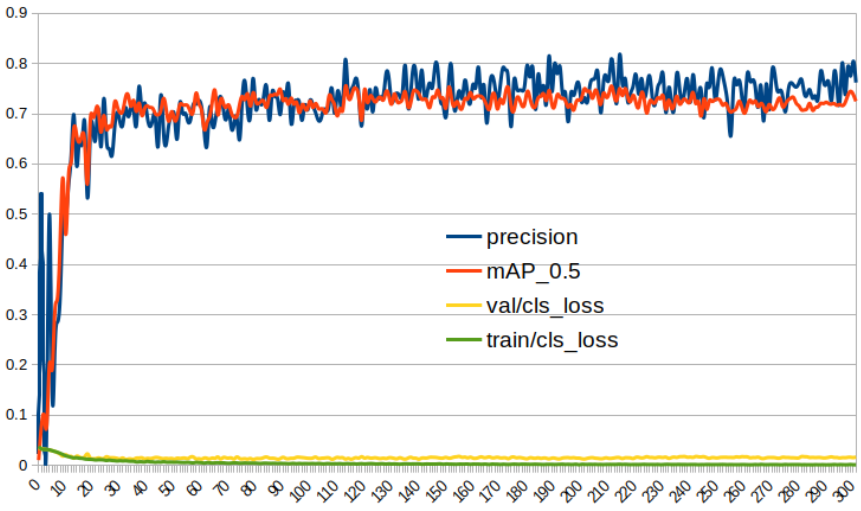


Figure 4.9: Graph result of the first training

Training02 We used YOLOv5m weight, where as for other parameters are the same as last training so after 2.45 hours of training: the model used 251 epochs while reaching an average precision of 77.4% a 5% improvement from last one yet still some lables still very low, immature only 67.8%.still needs improvements.

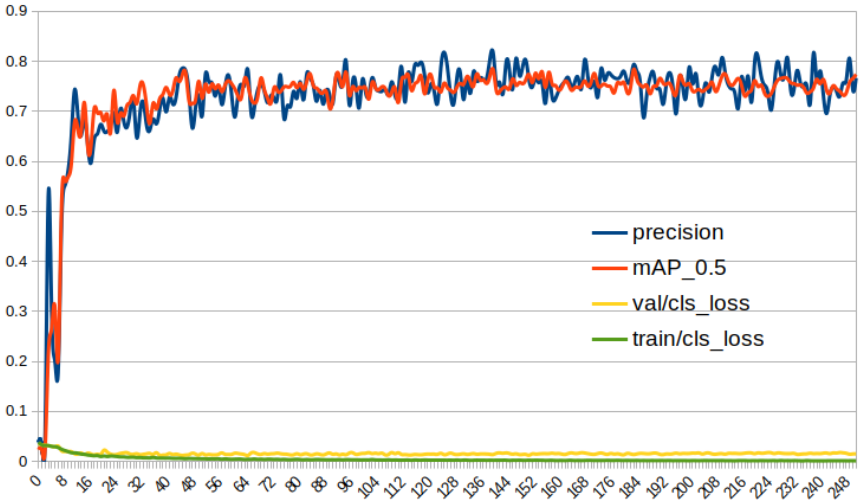


Figure 4.10: Graph result of the second training

Training03 We used YOLOv5m weight,same as before, nothing changed in the parameters they are the same as the previous training so after 3.03 hours of training: the model used 198 epochs while achieving an average precision of 80.2% not much increase from last training yet it can be considered as good model. Thus we used the model to label the Moulai's dataset, the model worked flawlessly, it got only mistaken on 32 labels out of 2500 labels detected

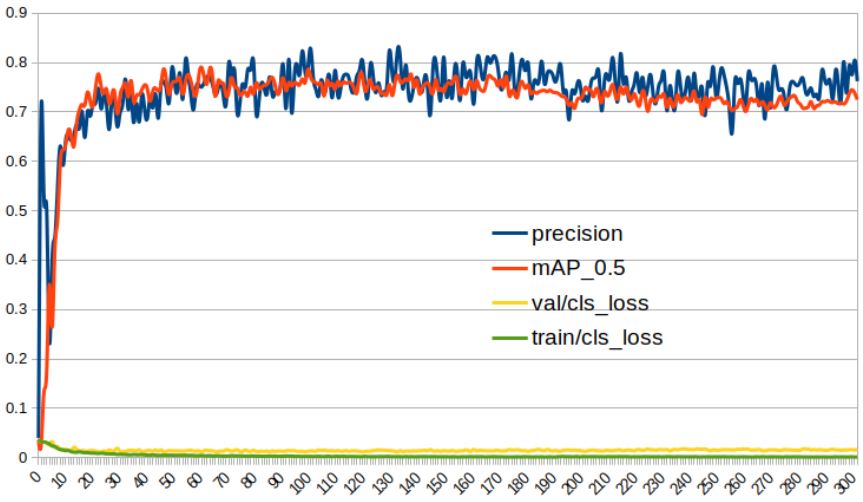


Figure 4.11: Graph result of the third training

Training04 For this training we couldn't use YOLOv5X model because we reached the limits of our gpu memory. For that, we got the idea of using transfer learning which is method where a pre-trained model acts like a starting point for an other model transferring its knowledge to it. So we choose the best performing model from

previous trainings(which is train3),we fed it portion of Moulai's dataset 315 image containing 372 labels,while also changing some parameters, learning rate 0.001, and batch 2, leaving other the same as before.
After 1.7 hours of training the model used 149 epochs while achieving an average precision of 85.3% and high precision percentage on all labels.

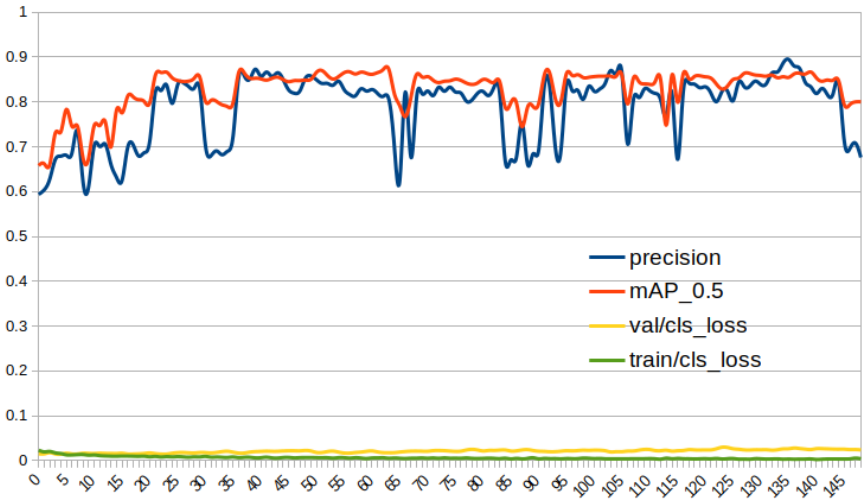


Figure 4.12: Graph result of the fourth training

4.6 Limitations

With the good results we had our two models yet still missing some points as the following:

- We have a quite high loss with 19.3% for the first model and 14.7% for the other, this leads to misslassify the labels specially for the second model since inmature and inflated have similiar futures.
- Both models won't recognize in eather small or far objects.

4.7 Future Works

In the future, from our perspective there's a possibility to develop a state of the art date palm pollination system by improving some point as mentioned below:

- Make our system recognise the spathes growing development more accurately.
- adding disease recognition to the system.
- train the system for very small objects.

4.8 Conclusion

In this chapter we presented the architecture of our system also the way how we prepared and gathered the dataset to train both models.From the results we got, we can tell that our two models can make a change in the date fruit production for the better.

CHAPTER 5

CONCLUSION

5.1 General Conclusion

Part of the work carried out within this thesis involves the automatic classification of date palm female flowers and counting the date palm fruits from the spathe.

To ensure knowledge about all-date palm pollination and growing production, this research looks for a necessary automation detection system to classify the spathes growing process. Due to this, we have used a supervised deep-learning method by proposing YOLO based model that can detect the date palms flower growing process from images.

The aim of our system is to make it a new or alternative solution for the current means in date fruit reproduction tasks anywhere and at anytime receive benefits from a deep learning model that helps to extract complex information from a spathes condition. Is it ready to be pollinated or not, how many dates have, and which fruit thinning method will be applied?

Such systems as an example can makes a big difference, it may prevent a lot of accidents caused from climbing the palm tree, making sure of quality of the date fruit by pollinating in the right time.

BIBLIOGRAPHY

[1] – MADR, 2019.

[2] A. Zaid and P.F. de Wet. Chapter I: Botanical and Systematic Description of the Date Palm. *Date Production Support Programme*, page 2, 2020.

[3] Antonino Ingargiola. jupyter_notebook_beginner_guide · GitHub, nov 2016.

[4] Avijeet Biswal. Convolutional Neural Network Tutorial, feb 2021.

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020.

[6] Timothy K. Broschat. Palm Morphology and Anatomy. *EDIS*, 2013(6):4, jul 2013.

[7] Anmol Chaudhary, Kuldeep Singh Chouhan, Jyoti Gajrani, and Bhavna Sharma. Deep learning with PyTorch. In *Machine Learning and Deep Learning in Real-Time Applications*, pages 61–95. 2020.

[8] Tzutalin darrenl. labelImg · PyPI, dec 2018.

[9] R O Dec. Reinforcement Learning. (c), jun 2022.

[10] Deepak Birla. Basics of Autoencoders. Autoencoders (AE) are type of... — by Deepak Birla — Medium, mar 2019.

[11] IBM Cloud Education. What is deep learning? — ibm, 5 2020.

[12] FAO. FAOSTAT Crops and livestock products, 2020.

[13] GeeksforGeeks. Underfitting and Overfitting -ML, jun 2022.

[14] Muriel Gros-Balthazard, William J. Baker, Ilia J. Leitch, Jaume Pellicer, Robyn F. Powell, and Sidonie Bellot. Systematics and Evolution of the Genus Phoenix: Towards Understanding Date Palm Origins. pages 29–54, 2021.

[15] Hannai Messaouda and Hammadi Akila. *Contribution à l'étude comparative des caractéristiques morpho-physiologiques de quatre variétés de dattes dans la région d'oued-souf et oued righ*. PhD thesis, ECHAHIDHAMMA LAKHDAR EL OUED FACULTE, 2020.

[16] Hmrishav Bandyopadhyay. Autoencoders in Deep Learning: Tutorial & Use Cases [2022], jun 2022.

[17] CHENCHOUNA Imene and TORCHI Fairouz. *Essai de lutte biologique contre la pyrale des dates (Ectomyelois Ceratoniae Zeller)*. PhD thesis, Université Mohamed Khider Biskra, 2021.

[18] Glenn; Jocher, Stoken Alex, Chaurasia Ayush, Borovec Jirka, NanoCode012, TaoXie, Kwon Yonghye, Michael; Kalen, Changyu Liu, Fang Jiacong, V Abhiram, Laughing;, Tkianai;, YxNONG;, Skalski Piotr, Hogan Adam, Nadar Jebastin, Imyhxy;, Mammana Lorenzo, and AlexWang1900;. YOLOv5, 2020.

[19] Xiaoyao Liang. Theoretical basis. *Ascend AI Processor Architecture and Programming*, pages 1–40, jan 2020.

[20] Warren S Mcculloch and Walter Pitts. A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY. *BULLETIN OF MATHEMATICAL BIOPHYSICS*, 5, 1943.

[21] Naked. Pollination of Date Fruit: The Process, mar 2017.

[22] Donges Niklas. Gradient Descent: A Quick, Simple Introduction — Built In, jul 2021.

[23] NIkon. Nikon D3300 — Read Reviews, Tech Specs, Price & More.

[24] Takayuki Okatani. On Deep Learning. *Journal of the Robotics Society of Japan*, 33(2):92–96, 2015.

[25] L A Plante, Hote Le, and Palmier Dattier. Règne Division Classe Ordre Famille Genre Espèce Plantae Magnoliophyta Liliopsida Arecales Arecaceae Phoenix 2-Caractéristiques Morphologiques du palmier dattier Le Palmier. pages 3–31, 2000.

[26] Austin Pollard. What are neural networks?, aug 1990.

[27] Pragati Baheti. The Essential Guide to Neural Network Architectures, jun 2022.

[28] Python Foundation. About Python™ — Python.org, 2016.

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 779–788, 2016.

[30] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 6517–6525, 2017.

[31] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. pages 1–6, 2018.

[32] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative Adversarial Text to Image Synthesis. *33rd International Conference on Machine Learning, ICML 2016*, 3:1681–1690, may 2016.

[33] F Rosenblatt. THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN 1. *Psychological Review*, 65(6):19–27.

[34] Marie Roué, Vincent Battesti, Nicolas Césard, and Romain Simenel. Ethnoecology of pollination and pollinators. *Revue d'ethnoécologie*, (7):1–27, jan 2015.

[35] Naassi Mohamed Salah. Weather Conditions Image Classification Using Deep Learning. Technical report, 2019.

[36] Amel Seifia and Ahlem Derfalou. *Caractéristiques physico-chimiques des noyaux et des l'huiles des noyaux de cinq variétés du palmier dattiers*. Mémoire de master, Université Mohamed Khider de Biskra, 2021.

[37] Besma Sghaier, Mouna Bahloul, Radhia Gargouri Bouzid, and Noureddine Drira. Development of zygotic and somatic embryos of Phoenix dactylifera L. cv. Deglet Nour: Comparative study. *Scientia Horticulturae*, 116(2):169–175, 2008.

[38] Sagar Sharma. Activation Functions in Neural Networks — by SAGAR SHARMA — Towards Data Science, sep 2017.

[39] SuperDataScience Team. Convolutional Neural Networks (CNN): Step 4 - Full Connection - Blogs - SuperDataScience — Machine Learning — AI — Data Science Career — Analytics — Success, aug 2018.

[40] Azati team. Disease Prediction And Classification With Artificial Neural Networks - Azati: Uniting experts to fulfil important projects, apr 2022.

[41] Zhixiao Wang, Mingyu Chen, Wenyao Yan, Wendong Wang, Ang Gao, Gaoyang Nie, Feng Wang, and Shaobo Yang. Revisiting Recent and Current Anomaly Detection based on Machine Learning in Ad-Hoc Networks. In *Journal of Physics: Conference Series*, volume 1288. Institute of Physics Publishing, aug 2019.

[42] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology, 2018.

[43] A Zaid and P F De Wet. Date Palm Cultivation. In *Date palm cultivation. FAO Plant Production and Protection*, chapter 1: BOTANIC, pages 1–28. FOOD AND AGRICULTURAL ORGANIZATION OF THE UNITED NATIONS, 1983.

[44] Zhe Ming Chng. Google Colab for Machine Learning Projects, apr 2017.