People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Mohamed Khider – BISKRA
Faculty of Exact Sciences, Science of Nature and Life
Computer Science Department

Order number: SIOD28/M2/2021

THESIS

Submitted in fulfilment of the requirements for the Masters degree
in Computer Science
Option: Information Systems, Optimisation and Decision (SIOD)

Title

A CNN based architecture for multispectral image classification: Application on Dates Sorting

---

By:
Sarah SETTA

Defended on 28/06/2022 before a jury composed of:

Bouguetitche Amina	M.C.B	President
Djeffal Abdelhamid	Professor	Supervisor
Moussaoui Manel	M.A.A	Examiner

Academic year 2021-2022

Acknowledgements

In the name of ALLAH, the most gracious, the most merciful

I would like to express my heartfelt thanks to ALLAH the Almighty who gave me the will and patience, to have enlightened and helped us in the preparation and the realization of this thesis

My gratitude and thanks go to my thesis supervisors **Prof. Abdelhamid Djeffal vice-rector in charge of higher education for the first and second cycles of university of Biskra** for giving me the chance to be one of his students, for his encouraging and advising me during the realisation of this work, and for the time he devoted to me. Thanks to him I have experienced true research and my knowledge on the subject matter has been broadened

I would particularly like to thank the members of the Jury for having done us the honour of accepting to judge and evaluate this work.

I would like to thank my **Prof. Bachir Abdelmalik** for his generous advices and encouragement that pushed me to continue my master degree, to **Dr. Sahraoui Soumia** for being the amazing example of teacher during the last three years of university.

Many thanks to my teachers at the University of Mohamed Khider Biskra who left a good impact on students by listening, supporting and encouraging them.

Finally, my thanks go to who have contributed in accomplishing this work, either from near or far, may Allah grant them all the best

Dedication

This study is wholeheartedly dedicated to my beloved family.

To my mother "Fifi" who has been my source of inspiration and gave me strength when I thought of giving up, who continually provides her moral, spiritual and emotional support

To my siblings "Ibrahim , Djebril".

To my aunts and uncles.

To all my cousins "Darine, Wassel, Ritej, Assinet, Alaa, Assil , Ilef, Layane, Manessa, Adelaadim, Rimen"

To "Bella" who made my university years full of love and laughs, "Panda" who tough me the friendship meaning, "Jiji" my guardian angel I appreciate your help.

To "Batman" the most annoying person on earth, "Badou" the smartest person on earth

To all the members of "BCSC" and "meme" group

To the strongest person I know me

Abstract

Dates are a popular and abundant fruit in the Middle East and North Africa, with a growing international presence. There are numerous types of dates, each with unique characteristics. Sorting dates is an important process in the date industry, but it can be a time-consuming task when done manually.

The rapid advancement of AI technologies and their applications in agriculture opens up completely new avenues for intelligent systems to better forecast trends and assist farmers in making decisions. A variety of advanced computational techniques are used to recognize and identify the types and quality of various fruits using a variety of methods. Machine vision is used for grading and sorting in multispectral imaging, monochrome imaging, and color imaging.

Deep Neural Networks (DNN) are extremely effective at identifying and classifying fruit images. They outperform other machine learning algorithms in terms of accuracy. Convolutional Neural Networks are the most common type of Artificial Neural Network and are considered to be the most efficient Deep Learning Algorithms.

Convolutional Neural Networks (CNNs) are intended to map a single image data input. The majority of the literature focuses on improving the architecture itself, feeding only one fruit side as input, which may result in overfitting. Others employ multi-input CNNs on all fruit sides. However, given the complexity of CNN, this method is time-consuming because each side passes through the CNN. As a result, in such cases, image processing is the gold standard for overcoming this issue. but it must still be chosen as the best processing tool.

As a solution, we proposed a new simple yet effective processing method that explores all fruit faces, based on thermal image and other characteristics as the date weight, combines them as one image input using image processing steps, and applies two different CNN models to the final data result.

Keywords: Convolutional Neural Networks (CNN), Deep Learning, Keras, TensorFlow,Thermal image, quality control.

Résumé

Les dattes sont des fruits populaire et abondante au Moyen-Orient et en Afrique du Nord, avec une présence internationale croissante. Il existe de nombreux types de dattes, chacun ayant des caractéristiques uniques. Le tri des dattes est un processus important dans l'industrie de la datte, mais il peut être une tâche fastidieuse lorsqu'il est effectué manuellement.

L'évolution rapide des technologies de l'IA et de leurs applications dans l'agriculture ouvre de toutes nouvelles perspectives aux systèmes intelligents pour mieux prévoir les tendances et aider les agriculteurs à prendre des décisions. Diverses techniques informatiques avancées sont utilisées pour reconnaître et identifier les types et la qualité de divers fruits à l'aide d'une variété de méthodes. La vision industrielle est utilisée pour le classement et le tri en imagerie multispactrale, monochrome et en couleur.

Les réseaux neuronaux profonds (DNN) sont extrêmement efficaces pour identifier et classer les images de fruits. Ils surpassent les autres algorithmes d'apprentissage automatique en termes de précision. Les réseaux neuronaux convolutifs sont le type le plus courant de réseau neuronal artificiel et sont considérés comme les algorithmes d'apprentissage profond les plus efficaces.

Les réseaux neuronaux convolutifs (CNN) sont destinés à mettre en correspondance une seule entrée de données d'image. La majorité de la littérature se concentre sur l'amélioration de l'architecture elle-même, en n'alimentant qu'un seul côté du fruit en entrée, ce qui peut entraîner un surajustement. D'autres utilisent des CNN à entrées multiples sur tous les côtés du fruit. Cependant, étant donné la complexité des CNN, cette méthode prend beaucoup de temps car chaque côté passe par le CNN. Par conséquent, dans de tels cas, le traitement d'image est l'étalon-or pour surmonter ce problème. mais encore faut-il le choisir comme meilleur outil de traitement.

Comme solution, nous avons proposé une nouvelle méthode de traitement simple mais efficace qui explore toutes les faces du fruit en utilisant l'image thermique et d'autres caractéristiques comme le poids de la date, les combine comme une seule entrée d'image en utilisant des étapes de traitement d'image, et applique deux modèles

CNN différents au résultat final des données.

Mots-clés: Réseaux neuronaux convolutifs (CNN), Deep Learning, Keras, TensorFlow, Image thermique, contrôle de qualité.

Contents

List of Figures

List of Tables

General introduction

The date palm serves as the foundation of the oasis system. It is an important aspect of agricultural activity. Algeria classifies the date sector as a strategic sector, alongside red and white meat, milk, cereals, and potatoes. Given the socioeconomic importance of this sector, the Ministry of Agriculture and Rural Development, as well as other agricultural research centers and institutions, implement numerous research and development programs.

The economic importance of phoeniculture is demonstrated by its role in stabilizing the population in Saharan areas, the jobs it provides, the product it provides, the product that is sold on national and foreign markets, and the hard currency that is generated. It also provides employment, a product that is marketed on national and international markets, and the hard currency that is earned each year from the export of its product.

Algeria's orchards are concentrated in the country's south-eastern region, including Biskra, El Oued, Ouargla, and Ghardaya. Since independence, and especially since the agricultural development policy, the number of producers has increased significantly.

Algeria is a major country with significant agricultural potential. The Deglet Nour (fine date) variety is particularly well-known. This significant agricultural potential has always been an inexhaustible resource for populations over the centuries. Today, the sector makes a significant contribution to the national economy, but it has yet to reveal all of its performances in order to become a flagship product on the national market and abroad (Merzaia abroad).

The date adds value to the Algerian economy; in 2019, Algeria exported 45 million dollars worth of dates. Biskra is a leading date palm wilaya, accounting for 42 percent of production and 26 percent of Algerian land area.

The ripening process of a date has four distinct stages, which are traditionally described by variations in color, texture, and taste. Certain pests, insects, mites, and mechanical equipment damage some of the product during the growing, ripening, and harvesting processes. Certain pests, insects, mites, and mechanical equipment damage the product, causing significant economic losses in the storage and export of dates.

Dates are typically harvested at various stages of ripeness. are inspected first to determine their maturity levels The matured dates are graded and packaged based on their quality. It is not recommended to pack dates of varying maturity for two reasons. It is not recommended to pack dates with varying maturity levels in the same package for two reasons. For starters, they may have mutually destructive effects on each other, and secondly, in order to improve marketing conditions, Second, in order to improve marketing conditions, different customer tastes must be considered.

A large number of workers are employed in the manual grading and packing process for a traditional packing system and sorting according to different qualities. This manual visual inspection process is labor intensive, time consuming, and prone to inconsistencies and inaccuracies caused by human workers' inexperience or fatigue. With rising labor costs, manual grading is a significant expense for date producers. a significant cost for date producers Thus, automatic sorting systems should be developed to save energy, improve packaged product quality, and achieve customer satisfaction.

In this master project, we focus on the design and implementation of a classification application that allows to use LCD and thermal images and the weight of a date as an input for Deep Learning methods and to detect their quality. The aim of this work is to demonstrate the quality of these images and their potential to be a necessary approach for the classification task,

This dissertation is composed of four chapters, this introduction, and a general conclusion, it is organized as follows:

The first chapter is devoted to the basic conceptions of Deep learning where we try to explain this Field.

The second chapter a general view on computer vision and its relevance to the field of quality control, In addition,reviewing the latest yet the most effective works about the Dates fruits classification.

In the third chapter, describes the design of our contribution, which corresponds to a new deep neural network architecture for multispectral image classification application on dates sorting.

In the last chapter, we illustrate the experimental study by explaining the different stages allowing the implementation of our project, the tests carried out, and the obtained results.

The thesis ends with a general conclusion containing our contemplated prospects.

Chapter 01

Insights into deep learning

Chapter 1

Insights into deep learning

1.1 Introduction

Less than a decade after cracking the Nazi encryption machine "Enigma" and helping the allies win World War II, mathematician Alan Turing changed history for the second time with a simple question: "Can machines think?".

Turing's paper "Computing Machinery and Intelligence" (1950) introduced the Turing Test and established the fundamental goals of Artificial Intelligence.

Essentially, Artificial Intelligence is a branch of computer science that aims to answer Turing's question by replicating or simulating human intelligence in machines.

Deep learning is driving the advances in Artificial Intelligence that are changing our world. It is used in many fields such as computer vision, speech recognition, natural language processing and machine translation where it produced results comparable to and in some cases surpassing human expert performance.

Deep learning is a type of Machine Learning which is a subset of Artificial Intelligence. The figure 1-1 shows the relation between Artificial intelligence, Machine learning and Deep learning.

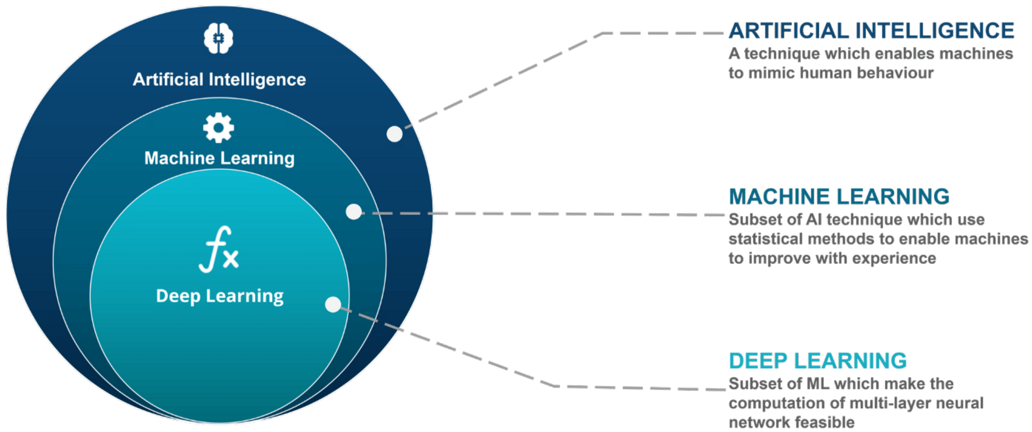

Figure 1-1: Artificial Intelligence vs Machine Learning vs Deep Learning

In this chapter, we will introduce Machine Learning and dig deeper into Deep Learning and its application. Lastly, we will understand what Neural Networks are and how they work.

1.2 Machine learning

Definition 1: Machine Learning (ML) is an approach to analyze data dealing with the construction and evaluation of algorithms. It is the science that gives computers and computing machines the ability to act without external control. ML has the ability to choose effective features for pattern recognition, classification, and prediction based on the models derived from existing data [35].

Definition 2: Machine learning is a branch of AI, according to Tom Mitchell definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

1.2.1 Types of Machine learning algorithms

Machine Learning algorithms can be divided into 3 types depending on the data nature [36]:

Supervised learning:

In supervised learning, algorithms are trained with labeled data to classify new data using a function that maps inputs to a specific output. Figure 1-2 shows an example of a supervised learning algorithm, where the model uses labeled data for training.

Once the model finishes the training phase. It will be able to classify new apple images as apples [2].

Examples of this type of algorithms : *SVM, Random decision forest, etc.*

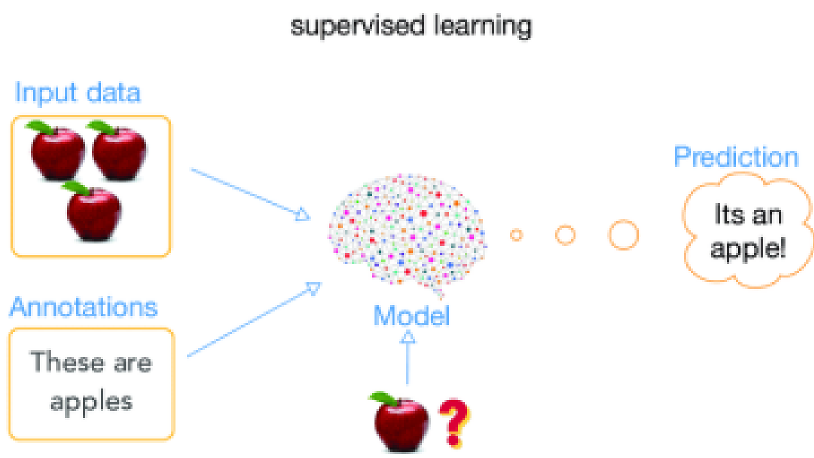

Figure 1-2: Supervised learning example [2]

Unsupervised Learning:

In unsupervised learning we do the opposite of supervised learning by feeding the algorithm with unlabeled data.

Following the training phase, the model will be able to recognize labels according to the similarities shared between samples. whereof, the ML algorithm will be able to classify new data to the new labels created [2].

Figure 1-3 shows an example of an unsupervised learning algorithm.

After the training, the model could make a difference between the 3 classes, and classifies the samples based on their similarities.

Examples of this type of algorithms : *KNN, K-means, etc.*

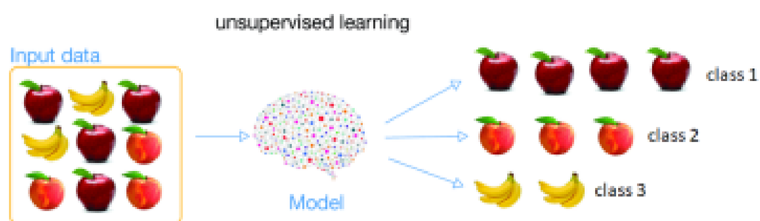

Figure 1-3: Unsupervised learning example [2]

The application of unsupervised machine learning is generally split into two classes:

- **Clustering**: is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorized data.

- **Association:** association rules allow you to establish associations amongst data objects inside large databases.

The most famous unsupervised machine learning algorithms: *K-Means Clustering, Principal Component Analysis*

Reinforcement Learning:

It is the ability of an agent to interact with the environment and make decisions based on which actions to take such that the outcome is more positive.

It follows the concept of hit and trial method. The agent is rewarded or penalized with a point for a correct or a wrong answer. On the basis of the positive reward points gained the model trains itself [2].

Once trained, it gets ready to predict the new data presented to it.

Figure 1-4, the dog is playing the role of the agent, when the dog does a desired action or a close one, the trainer who is playing the role of the environment will provide a reward for the dog otherwise no rewards or a negative reward [2].

Examples of this type of algorithms: *Markov decision process, Approximate dynamic programming, etc.*

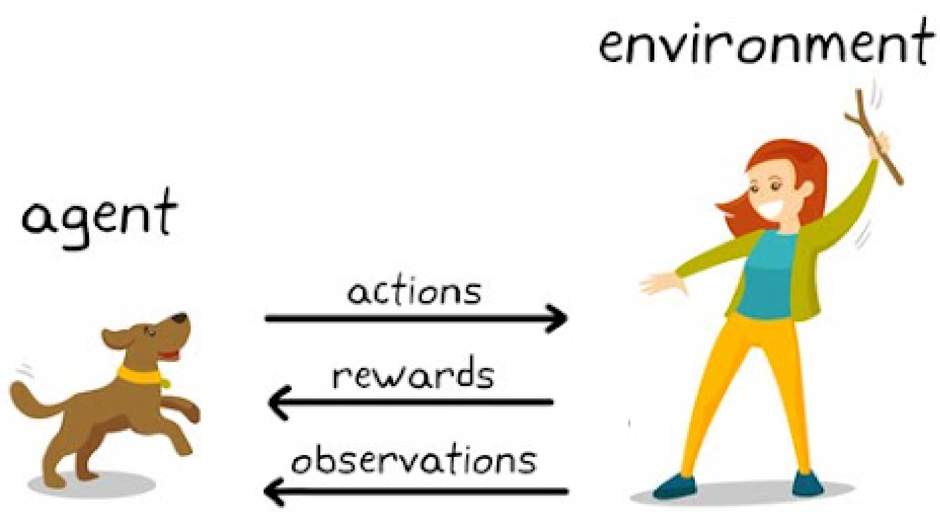

Figure 1-4: Reinforcement learning example [2]

1.3 Deep Learning

Definition: Deep learning is a subset of machine learning and based on the algorithms that are stimulated by the functioning of the brain and the way they are structured. Deep Learning educates the computer system to produce results by training them on the previously available examples.

The capability of deep learning has enabled it to accomplish results that were impossible once upon a time. They remain as the pivotal mechanics behind most of the applications that are found nowadays such as self-driving vehicles, voice control in consumer electronics such as the phone, Television, smart machines etc [17].

1.3.1 Data organization in Deep Learning algorithms

Classification is a two-step process: a learning step (training and validation) and a classification step (use).

In the learning step, the classifier (usually a function) is built by learning from a database of training examples (sample of data used to fit the model) with their respective classes, An example $X = (x1, x2, .., xm)$ is represented by a vector of attributes of dimension m. Each example is assumed to belong to a class predefined represented in a particular attribute of the database called attribute of class .

The built model in the first step is used to classify the new data. But before proceeding to use, the model must be tested to ensure its ability to generalize to data not used in the training phase. The model obtained can be tested on the training data themselves or by using a validation dataset that represents a sample of the data used, to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters, and this makes sense since this dataset helps during the development stage of the model.

Once the model is completely built and evaluated, it is ready to fit out on a use set illustrated as the gold standard used to provide an unbiased evaluation of a final model fit on the training dataset [24].

Dividing the dataset into two sets is a good idea, but not quite reasonable. You can greatly reduce your chances of overfitting by partitioning the data set into the three subsets. Use the validation set to evaluate results from the training set. Then, use the test set to double-check your evaluation after the model has "passed" the validation set. The figure shows 1-5 this workflow:

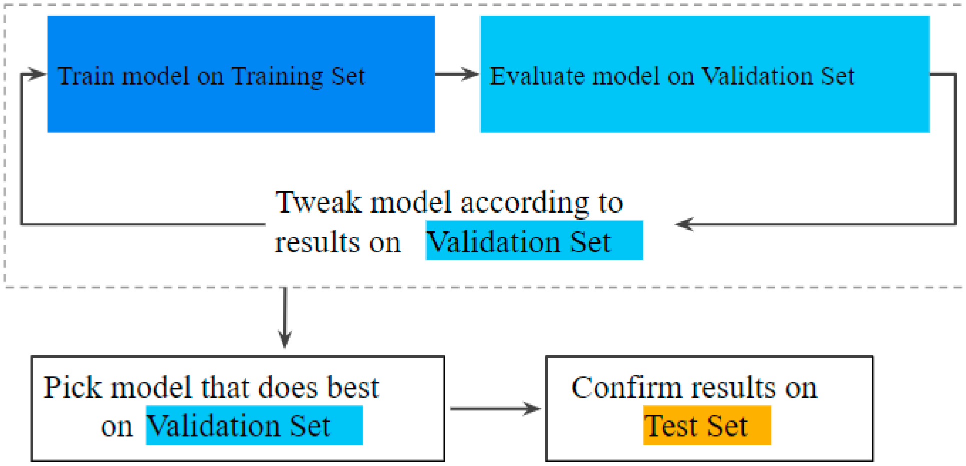

Figure 1-5: Dataset organization: a workflow [38]

1.3.2 Deep learning applications

The different application domains are Computer vision, prediction, semantic analysis, natural language processing and information retrieval[26].

- **Computer Vision:** Object recognition, object detection, and object processing, for example, we mention *facial recognition*: The eyes, the nose, and the mouth, are all characteristics that a Deep Learning algorithm will learn to detect in a photo. The first step is to give the algorithm a certain number of images, and then, with practice, the algorithm will be able to detect a face on an image

- **Prediction:** The various sub-domains here are classification, analysis, and recommendation. text classification, document classification, image analysis, medical diagnosis, prediction of network intrusion detection, and predicting denial of service attack have been successfully implemented using machine learning for example Deep learning is extensively utilized in cancer detection and diagnosis. This latter method is especially intriguing since it fits with a rising trend toward customized predictive therapy.

- **Semantic Analysis, Natural Language Processing and Information Retrieval:** Semantic analysis is the process of relating syntactic structures from paragraphs, sentences, and words to the level of writing as a whole.

  Natural language processing is how to program computers to correctly process natural language data.

  Information retrieval is the science of searching for information in a document, searching for documents, and searching for metadata that describes the data and for databases of sounds and images.

  These are three domains in which machine learning techniques have been explored in the past.

  We mention also Voice search and voice-activated assistants: A popular area of use for Deep Learning is voice search and voice-activated intelligent assistants. With the big tech giants already making significant investments in this area, voice-activated assistants can be found on almost every smartphone. Apple's Siri has been on the market since October 2011. Google today's voice-activated assistant for Android, was launched less than a year after Siri. The most recent voice-activated intelligent assistant is Microsoft Cortana

1.4 Deep learning algorithms:

1.4.1 Artificial Neural Networks (ANNs)

Definition [43]: All Deep Learning algorithms are, in practice, neural networks . Artificial neural networks, often known as ANNs, are information processing models that imitate the operation of the human nervous system. At the working level, it is analogous to how the brain manipulates information. All ANN are composed of linked neurons organized in layer [43]s.

The neuron: Neural networks are derived from the biological concept of neurons. A neuron is a cell-like structure in the brain. To understand the neural networks we need first to understand how the biological ones work. A neuron is made up of four major components. Dendrites, nucleus, soma, and axon are the four types [43].

The figures 1-6 and 1-7 show a representation of a real neuron and an artificial neuron:

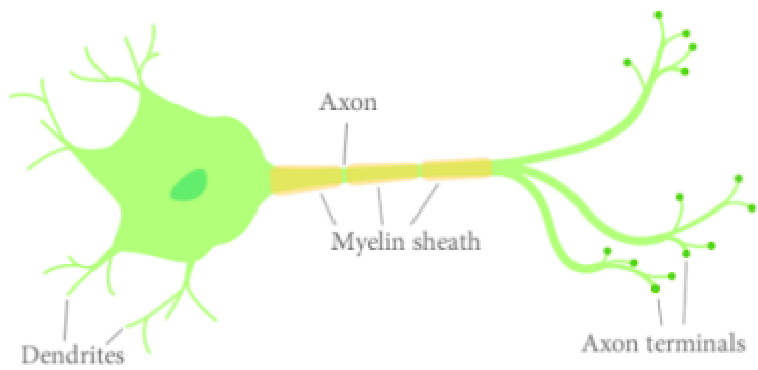

Figure 1-6: Real neuron [43]

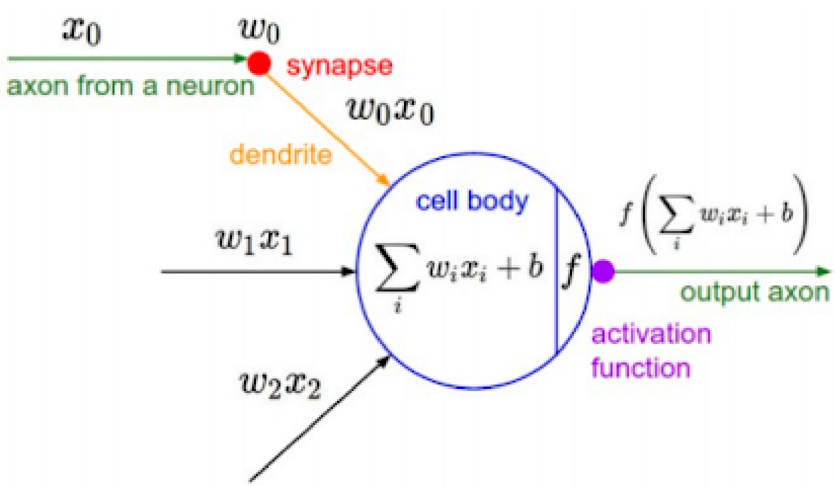

Figure 1-7: Artificial neuron [43]

1.4.1.1 Neural network Components

- **Hyperparameters:** are the variables which determines the network structure and the variables which determine how the network is trained. Hyperparameters are set before training (before optimizing the weights and bias) [20]. These are various Hyperparameters:
  - Number of Hidden Layers and units.
  - Dropout.
  - Network Weight Initialization.
  - Activation function.
- **Neurons:** are the basic units of a neural network. In an ANN, each neuron in a layer and is connected to each neuron in the next layer. When the inputs are transmitted between neurons, the weights are applied to the inputs along with the bias.
- **Weights:** control the signal (or the strength of the connection) between two neurons. In other words, a weight decides how much influence the input will have on the output [35].
- **Biases:** which are constant, are an additional input into the next layer that will always have the value of 1. Bias units are not influenced by the previous layer (they do not have any incoming connections) but they do have outgoing connections with their own weights. The bias unit guarantees that even when all the inputs are zeros there will still be an activation in the neuron.

$$y = \sum(weight * input) + Bias$$

- **Layers:** There are Neurons in each layer. In most cases, the amount is entirely up to the creator. Having too many layers for a simple task, on the other hand, can unnecessarily increase its complexity and, in most cases, decrease its accuracy. The Neural Network is made up of three types of layers:

- The input layer: contains the neural network's initial data.
- Hidden layers: an intermediate layer between the input and output layers and the location of all computation.
- Output layer: generates the output for the given inputs.

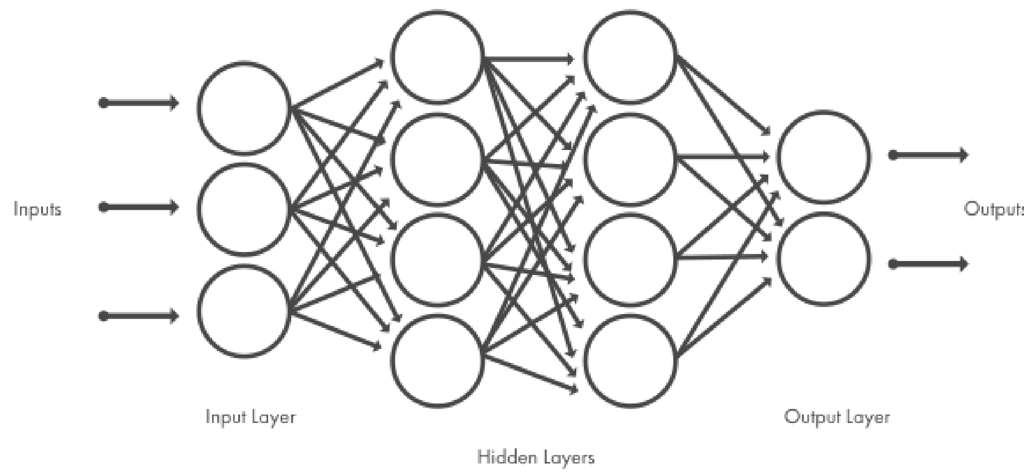

Figure 1-8: Neural network layers [34]

1.4.1.2 Activation functions

After the neuron has performed the product between its inputs and weights, it applies a nonlinearity to this result. also applies a non-linearity to this result. This nonlinear function is called the activation function.

The activation function is an essential component of the neural network. What this function decides is whether the neuron is activated or not. It calculates the weighted sum of the inputs and adds the bias. This is a transformation of the input value.

- **Sigmoid function:** This function is one of the most commonly used. It is bounded between 0 and 1, and can be interpreted stochastically as the probability of the neuron activating, and is usually called the logistic function or the logistic sigmoid.

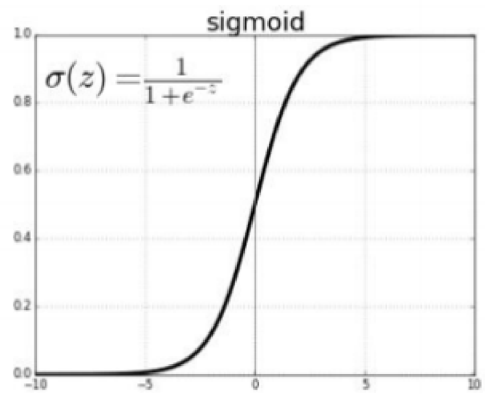

Figure 1-9: Sigmoid activation function [42]

- **ReLu function:** The ReLu function is probably the closest to its biological counterpart [42]. This function has recently become the choice for many tasks (especially in computer vision) [15]. As in the formula above, this function returns 0 if the input z is less than 0 and returns z itself if it is greater than 0.

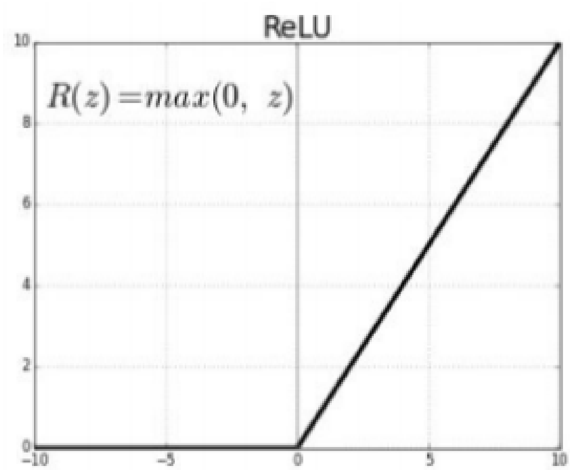

Figure 1-10: ReLu activation function [42]

- **Softmax function:** Softmax regression (synonyms: Multinomial logistic, Maximum entropy classifier, or simply Multiclass logistic regression) is a generalization of logistic regression that we can use for multi-class classification [15]. Unlike other types of functions, the output of a neuron in a layer using the softmax function depends on all other neurons in its layer. This is because it requires the sum of all outputs to be equal to 1.

1.4.2 Deep Neural Networks: Types and architectures

Deep learning is a collection of algorithms and architectures that may be applied on a wide range of problems to be solved.

In this section we cover the most commonly used deep learning architectures as recurrent neural network, deep neural networks, long short-term memory networks, convolutional neural network.

1.4.2.1 Recurrent Neural Networks (RNN)

A recurrent neural network (RNN) is a neural network with at least one cycle in its connection graph.

Recurrent neural networks function on the looping and chaining concept [41].

Recurrent Neural Networks (RNN) are a sort of Neural Network in which the previous step's output is given as input to the current phase. In typical neural networks, all inputs and outputs are independent of one another; however, when predicting the next word in a sentence, the prior words are necessary, and so the previous words must be remembered. As a result, RNN was created, which solved this problem with the aid of a Hidden Layer. The Hidden state, which remembers certain information about a sequence, is the core and most essential aspect of RNN. which can be demonstrated using the figure in the next page.

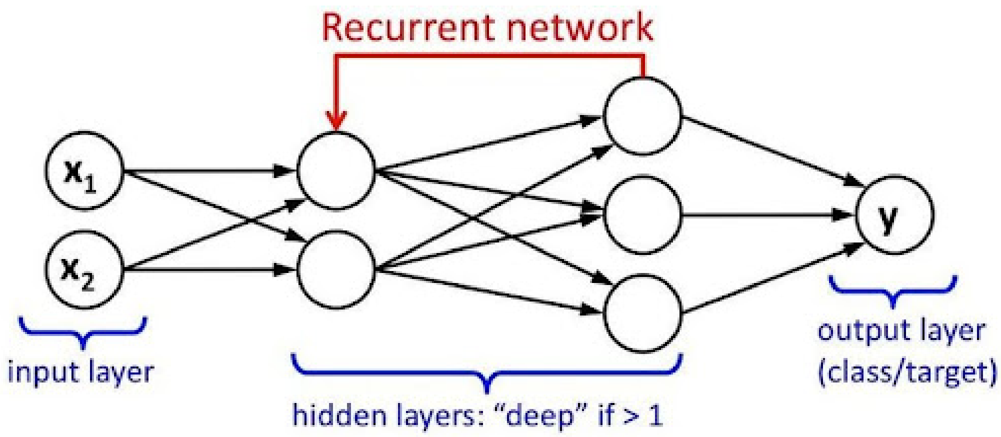

Figure 1-11: Recurrent Neural Network

1.4.2.2 Deep Neural Networks (DNN)

Deep ANNs are most widely referred to as deep learning (DL) or deep neural networks (DNNs). They are a relatively new area of ML research allowing computational models that are composed of multiple processing layers to learn complex data representations using multiple levels of abstraction [29].

One of the main advantages of DL is that in some cases, the step of feature extraction is performed by the model itself. DL models have dramatically improved the state-of-the-art in many different sectors and industries, including agriculture.

DNN's are simply an ANN with multiple hidden layers between the input and output layers and can be either supervised, partially supervised, or even unsupervised.

1.4.2.3 Long Short-Term Memory networks (LSTM)

Long Short Term Memory Networks (LSTM) usually referred to simply as "LSTMs" are an extension for recurrent neural networks, which extends their memory, mainly introduced to handle situations where RNNs fail [19]. Therefore, it is well suitable for learning important experiments that have very long delays between them. It's been designed in such a way that the vanishing gradient problem is nearly entirely eliminated, but the training model remains unchanged. Long-time delays are bridged in some cases utilizing LSTMs, which also deal with noise, distributed representations, and continuous data.

There are three doors in an LSTM: enter, forget, and exit. These gates determine whether to allow a new entry (entry gate), discard information that is no longer relevant (forget gate), or allow it to impact the output at the current time step (output gate). figure 1-12 represent an RNN with its three gates [19].

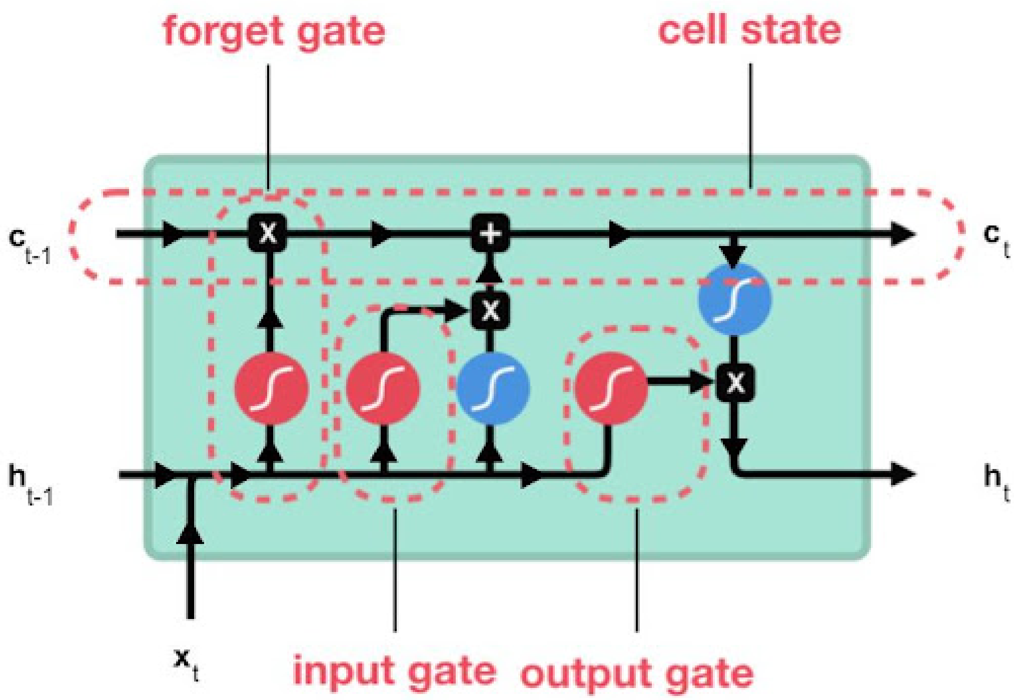

Figure 1-12: Long Short-Term Memory networks [19]

1.4.2.4 Convolutional neural network (CNN)

Convolutional neural networks are one of the most often utilized models nowadays. This neural network computer simulation comprises a multilayer perceptron version and includes one or more convolution layers that can be linked or pooled [19].

In this context, it is essential to introduce the CNN, which is fundamental to understanding fruit classification based on deep learning.

In the following, we present an overview of CNN.

1.4.3 Colvolutional neural network (CNN)

The CNN architecture is made up of numerous layers that work together to extract features and then classify them . The input picture is separated into receptive fields, which feed a convolutional layer that extracts features from it. Pooling is the following stage, which decreases the dimensionality of the retrieved features (by downsampling) while keeping the most critical information (usually via maximum maximum pooling). The data is subsequently sent into a fully linked multi-layer perceptron after another convolution and pooling process [19].

This network's final output layer is a collection of nodes that recognize picture characteristics (in this case, one node per identified number). Backpropagation is used to build the network [19].

1.4.3.1 Convolutional neural network layers

There are several layers in CNN as shown in Figure 1-13 [34]:

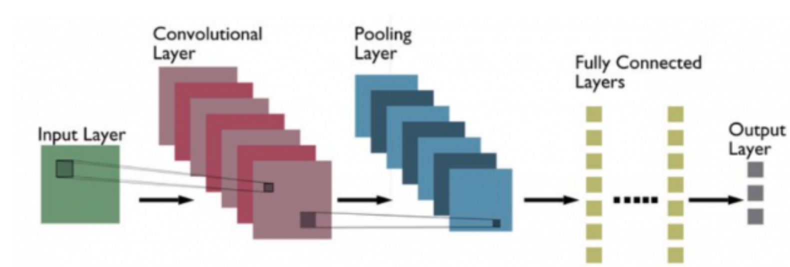

Figure 1-13: Convolutional neural network layers [34]

- **Input layer:** The CNN input layer will include data that describes the picture. A three-dimensional matrix represents the picture data, which must be molded into a single column in most cases (vector representation).

- **Convolution layer:** The convolution layer is also known as the feature extraction layer since it is where the image's features are extracted [34].

  First, as shown in Figure 1-14, a section of the picture is linked to the Convolution layer, which performs a convolution operation and calculates the scalar product between the receiver field (a local region of the input image with the same size as the filter) and the filter. The operation yields a single volume integer as a result. The filter is then dragged by a stride onto the next receiver field of the same input picture, and the process is repeated. The same procedure is used to repeat this operation until the full process is completed [34].

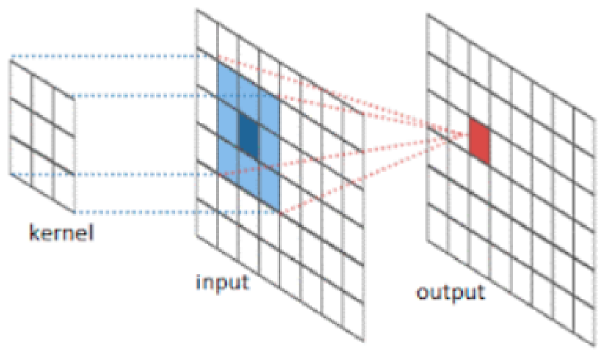

Figure 1-14: Convolution layer [34]

- **Pooling layer:** The pooling layer is used to reduce the spatial volume of the input image after convolution. It is used between two convolution layers. If we apply fc (Fully Connected) after the convolution layer without applying pooling or maximum pooling, the calculation will be expensive. Therefore, maximum pooling is the only way to reduce the spatial volume of the input image by encoding information [44].

- **Fully Connected Layer** A fully connected layer involves weights, biases and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images into different categories by training [34].

- **Logistic layer or Softmax** Softmax or the logistics layer is the last layer of CNN. It resides at the end of the FC layer. Logistics is used for binary classification and Softmax is for multi-classification [34].

- **Output layer** The output layer contains the label which is in coded form.

1.4.3.2 Transfer learning

Transfer Learning is a deep learning technique that allows you to do Deep Learning without having to spend a lot of time doing calculations. The idea is to apply the knowledge gained by a neural network while solving one problem to another that is more or less similar. A knowledge transfer is accomplished in this manner [18].

1.4.3.3 Convolution Neural Networks architectures

The following is a list of different types of CNN architectures [28]:

- **LeNet:** LeNet was the first and most successful CNN architecture. It was created to help with handwritten digit recognition problems. It is made up of several convolutional and pooling layers, which are followed by a fully-connected layer. Five convolution layers are followed by two fully connected layers in the model.

  Despite his success, LeNet was unable to train well due to the vanishing gradients problem. To address this issue, a shortcut connection layer known as max-pooling is used between convolutional layers to reduce image spatial size, thereby preventing overfitting and allowing CNNs to train more effectively. The architecture of LeNet-5 is depicted in the figure 1-15.

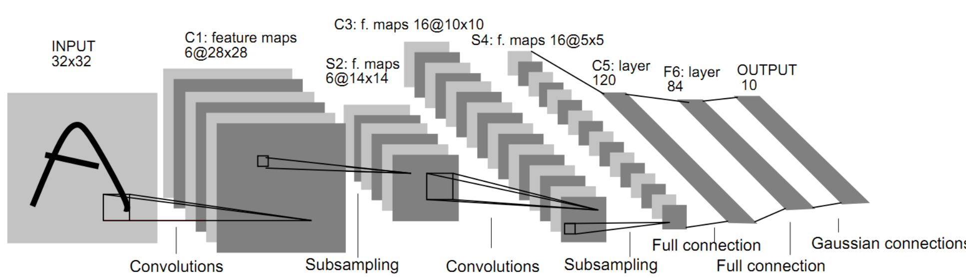

Figure 1-15: LeNet-5 architecture [28]

- **AlexNet:** AlexNet is the deep learning architecture that made CNN popular. Its architecture was very similar to that of LeNet, but it was deeper, larger, and featured Convolutional Layers stacked on top of each other.

  The AlexNet architecture was created to be used with large-scale image datasets; it is made up of 5 convolutional layers, 3 fully connected layers, and 2 dropout layers.

  The activation function used in all layers is Relu, while Softmax is used in the output layer. This architecture contains approximately 60 million parameters. The architecture of AlexNet is represented in the figure 1-16.

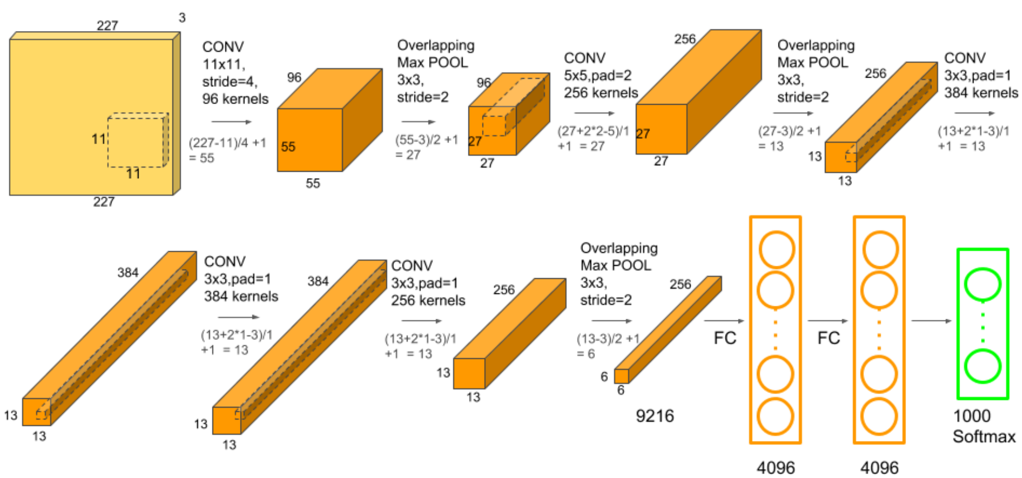

Figure 1-16: Alexnet architecture [28]

- **ZF Net:** ZFnet employs a hybrid of fully-connected layers and CNNs. It improved on AlexNet by adjusting the architecture hyperparameters, specifically by increasing the size of the middle convolutional layers and decreasing the stride and filter size on the first layer.

  The ZF Net CNN architecture has seven layers: a convolutional layer, a max-pooling layer (downscaling), a concatenation layer, a convolutional layer with a linear activation function, and stride one, a dropout for regularization before the fully connected output.

By introducing an approximate inference stage through deconvolutional layers in the middle of CNNs, this CNN model is computationally more efficient than AlexNet.

- **GoogLeNet:** It achieves deeper architecture through the use of a variety of techniques, including 11 convolution and global average pooling.

  The GoogleNet CNN architecture is computationally costly. It employs heavy unpooling layers on top of CNNs to remove spatial redundancy during training, as well as shortcut connections between the first two convolutional layers before adding new filters in later CNN layers to reduce the number of parameters that must be learned.

  Street View House Number (SVHN) digit recognition task, which is frequently used as a proxy for roadside object detection, is one of the real-world applications of GoogLeNet CNN architecture.

  The figure 1-17 is a simplified block diagram of the GoogLeNet CNN architecture.

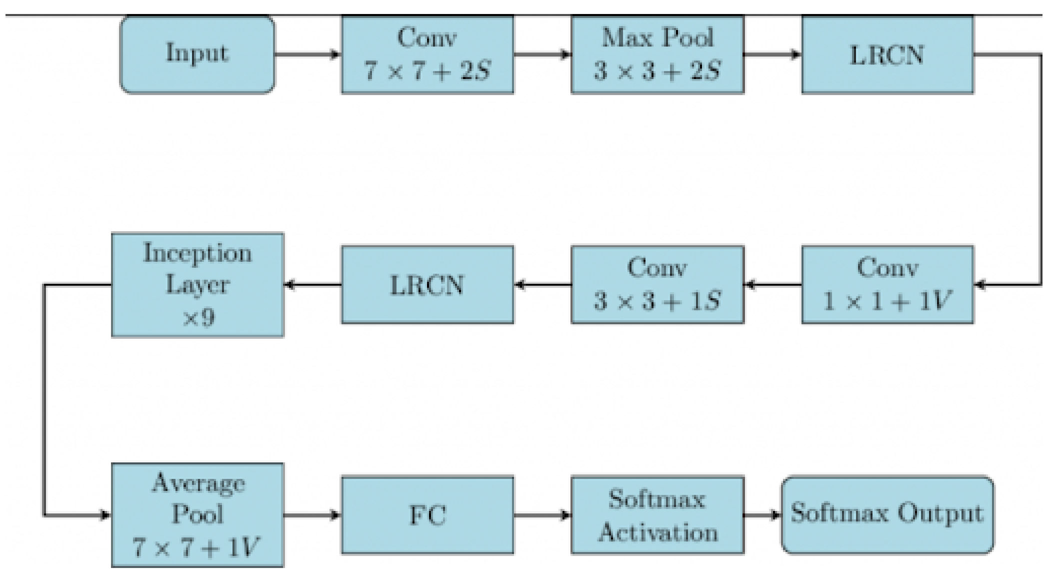

Figure 1-17: GoogleNet architecture [28]

- **VGGNet:** VGGNet is a 16-layer CNN that has been trained on over one billion images and has up to 95 million parameters (1000 classes). It can handle large input images of 224 x 224 pixels and has 4096 convolutional features.

  CNNs with such large filters are expensive to train and require a large amount of data, which is why CNN architectures like GoogLeNet (AlexNet architecture) outperform VGGNet for most image classification tasks with input images ranging in size from 100 x 100 pixels to 350 x 350 pixels.

  Because of its applicability for a variety of tasks, including object detection, the VGG CNN model is computationally efficient and serves as a strong baseline for many applications in computer vision. The figure 1-18 below represents the standard VGG16 network architecture diagram:

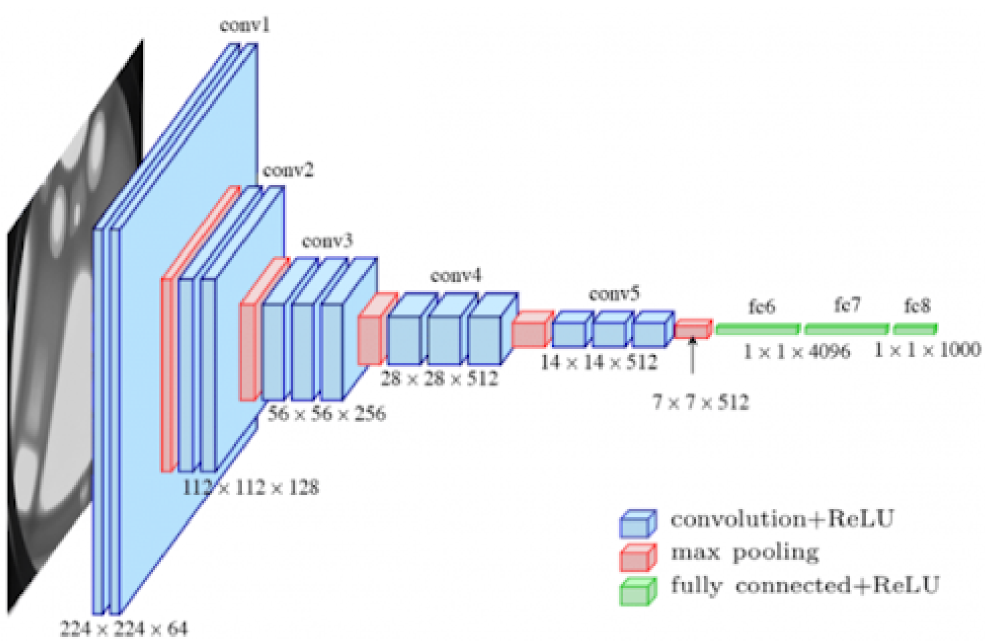

Figure 1-18: VGG16 architecture [28]

- **ResNet:** The network has 152 layers and over one million parameters. CNNs are typically used for image classification tasks with 1000 classes, but ResNet demonstrates that CNNs can also be successfully used to solve natural language processing problems such as sentence completion and machine comprehension.

Microsoft's machine comprehension system is an example of ResNet CNN architecture in action.

ResNet is a computationally efficient CNN architecture that can be scaled up or down to match the computational power of GPUs.

- **MobileNets:** MobileNets are CNNs that can be fitted onto a mobile device and used to classify images or detect objects with minimal latency.

  They are typically very small CNN architectures, making them easy to run in real-time. The architecture is also flexible, as it has been tested on CNNs with 100-300 layers and outperforms other architectures such as VGGNet.

  CNNs built into Android phones to run Google's Mobile Vision API, which can automatically identify labels of popular objects in images, are real-world examples of MobileNets CNN architecture.

- **GoogLeNet DeepDream:** this inception network is used in CNN architecture to generate images based on CNN features. This architecture is frequently used in conjunction with the ImageNet dataset to generate psychedelic images or abstract artworks using human imagination.

1.5 Conclusion

Neural networks provide a versatile and reliable framework whose advantages have been proved in numerous experiments. They have enabled significant advancements in machine learning, as well as image processing and computer vision.

In this chapter we have covered some main aspects which are the basic of our work. We also detailed the basics of Deep learning and a general view of what are the datasets and the applications of deep learning, then we discussed neural networks and their different types.

In the next chapter we will focus on quality control and image-based quality assurance.

Chapter 02

Dates Fruits Quality Control: Methods Overview and Review

Chapter 2

Dates Fruits Quality Control: Methods Overview and Review

2.1 Introduction

Quality control is a critical process that ensures only products that meet specified quality requirements reach the final consumer or the next stage in the manufacturing chain; it secures the origin, traceability, and control of ingredients.

Many products are still assessed by human operators (manual operation), and while the human ability to visually inspect various objects is very high, subjectivity and fatigue caused by repetitive tasks may result in human errors.

However, the advantages of using AI rather than human inspectors outweigh the effort required to create a complex algorithm and train a model to make these decisions.

The primary applications of computer vision technology in the food industry are the quality evaluation of food grains and the detection of flaws in fruits and vegetables.

2.2 Quality Control

Quality control (QC) is a set of procedures used to ensure that the quality of manufactured products is maintained or improved. To accomplish this, companies should first create an environment in which management and employees strive for perfection by developing quality benchmarks and product testing to validate statistically significant improvements.

The creation of properly specified controls is a key part of quality control. These controls aid in the standardisation of manufacturing and the reaction to quality issues [27].

2.2.1 The importance of quality assurance

Quality control is critical in ensuring that products meet the criteria established by the government for the manufacturing process.

Knowledge about the origins of the things generates a favorable image, which leads to improved sales, and it also assists manufacturers/contractors in establishing the responsibilities of workers in the production process [7].

2.2.2 Benefits of quality control [1]

1. **Higher Production Performance:** The good condition of elementary products leads to Production Efficiency. This includes high-quality materials, a lack of imperfections, operational machinery, a functional warehouse, and qualified staff; eventually, fewer resources and time to complete the task.

2. **Improving Cost Efficiency:** The quality assurance prevents quality issues from the start, avoiding faults, returns, repairs, and other costly problems, by the end quality assurance, will result in cost savings.

3. **Retaining Customer Trust:** The level of customer trust in a brand, product, and company is determined by product quality. Performing quality assurance can help to improve the quality of the manufacturing output. It will elicit a positive response from customers and increase their trust in the company.

4. **Improving the working environment:** Quality assurance contributes to a more pleasant working environment. It reduces unsafe practices and hazards,

ensuring that both workers and the environment are safe.

2.2.3 Characteristics of a control [40]

- **Control frequency:** systematic, to ensure the effectiveness of quality control from the start, several critical points that offer it must be available we mention through taking samples of the product to be evaluated.

- **Knowing the type of inspection:** Non-destructive vs. destructive inspection.

- **The method used in inspection:** measurement, comparison and grading.

- **The tools will be used:** measurement equipment, reference system.

- **Who will carry out the inspection:** specialised staff or machines.

2.2.4 Quality types

1. **Destructive testing:** In certain cases, testing a trait requires damaging the product being tested.

   The test is complemented, and sometimes even replaced, by a test of the production factors (temperature, pressure, electrical intensity, and so on) that have an impact on the accomplishment of the feature and can only be detected destructively [40].

2. **Non-destructive testing (NDT):** Is a collection of procedures for characterising the status of the integrity of items without causing damage to them during manufacture or maintenance. As a result, it is vital to define what amount of fault is acceptable and then be able to identify them without breaking the item and, if required, replace it [40].

2.2.5 Relevance of control by an artificial vision:

It is difficult for a human operator to describe defects in a simple way, distinguish characteristics, or to make simple classifications as these require a high degree of intelligence and understanding.

The goal of a machine vision system is to enhance efficiency and quality of production by eliminating the tiresome work of manual control and allowing for speedier completion of repeated activities with the goal of "zero defect."

2.3 Deployment [32]

2.3.1 Machine Vision

Machine Vision is an applied artificial vision field that employs digital image processing to automate operations.

It uses cameras as sensors to detect defects on objects, inspect parts during manufacturing, count them, sorting, classifying, measuring, and so on based on their visual appearance to provide machines and robots with sensory capacities similar to human vision. However, we should be cautious not to imagine too deep an analogy between the performance of human vision, capable of exploring the spatial structure of its environment and recognizing complex and varied objects, known or unknown [39].

2.3.2 Image recognition

Image recognition is a computer vision task that aims to identify and categorise various features in images and videos.

Image recognition models are taught to take an input image and generate one or more labels that describe the image. The collection of possible output labels is referred to as the target classes. Picture recognition models can provide a confidence score relating to the model's assurance that an image belongs to a class in addition to a projected class [32].

2.3.3 Image acquisition

Image acquisition is an essential link in any image design and production chain.

In order to manipulate an image on a computer system, it is first essential to change it to make it readable and manipulable by the system. This is achieved by transforming optical pictures (real-world data) into usable digital data matrices. This operation is carried out by input systems, which are divided into two categories: Scanners and digital cameras [32].

2.3.4 Image processing

Image processing is the study of digital images and their transformations with the goal of making this operation possible, simpler, more efficient, and pleasant, improving the visual aspect of the image, and extracting information deemed relevant.

The provided image is converted into an electrical signal, that is necessary to extract the desired information about the scene from which the image was captured from this signal. The first foundations of image processing are directly derived from signal processing, which is a common occurrence because any image, whether continuous or digital, can be thought of as a 2-dimensional signal.

Vision is a difficult domain to master because locating a simple object in an image necessitates numerous operations. Processing, also known as pre-processing, relates to all of the techniques used to improve the quality of an image. Therefore, the starting data is the initial image and the result is also an image.

The following are the steps in a typical processing system:

1. **Pre-processing:** Image manipulation operations to correct acquisition defects, normalise images, and improve image quality.

2. **Image segmentation:** Image segmentation is the process of identifying structures of interest in an image. Segmentation methods are classified into two types:

- The contour method isolates the contour of the object of interest. The result is typically presented as a set of pixel strings, and further processing is usually required to associate the contour with the object of interest.
- The other method looks for homogeneous pixel areas in an image. The standard of uniformity can be intensity, colour, or even local texture. The end result is either a binary image or a labeled image, with each label representing a region.

3. **Analysis:** The series of operations used to extract information from an image.
4. **Interpretation:** For some goals, the shift from structural to semantic description, as image processing is part of a larger processing chain that includes collection development upstream and analysis and statistics on the results obtained downstream.

2.3.5 Images classification

The systematic classification of images into classifications is known as image classification, The goal of it is to create a system capable of automatically classifying pictures.

Automatic image categorization has a wide range of uses, from document analysis to medicine and the military, etc. The connecting thread between these applications is that they all involve the establishment of a processing chain based on accessible pictures that are formed of numerous steps in order to deliver an output decision.

For best effectiveness, each stage requires the use of a suitable approach. The extraction step involves the extraction of relevant information that has been transformed into digital vectors, allowing you to operate in a digital space. During the learning phase, a decision function is created from this initial data to determine if a new data item corresponds to one of the classes present.

2.4 The image

2.4.1 Definition

The image is a physical depiction of an item, live thing, or concept. It is an organized set of data in the form of a value matrix that represents a point-to-point scene for the human eye.

Each member of this matrix (pixel) represents a region of the original picture, and the value of each pixel corresponds to the average value of the received signal, which can be expressed using various codes to depict binary, greyscale, color, and even multi-spectral images.

The digital pictures contain the raster data that represents the scene They do, however, carry a little bit of information data known as metadata, which provides information on the picture capture settings. Metadatas are frequently required to comprehend the image itself, such as what the values recorded in a matrix indicate or when the image was captured[27].

2.4.2 Features of a digital image

The image is a structured set of information characterised by the following parameters [21]:

1. **Pixel:** The pixel is the basic unit of an image.

   The final image is constituted in an array of two dimensions which contains those pixels that provide all the image component information.

   Each pixel is located by its A pixel is usually rectangular or almost square and has a size between 0.18 mm and 0.66 mm on each side. size between 0.18 mm and 0.66 mm on each side.

2. **Resolution:** The number of pixels in an image per unit length is defined as its resolution. The resolution of an image is most commonly expressed in PPP

(Dots Per Inch) or DPI (Dots Per Inch). The higher the resolution, the more precise the image is in detail.

3. **Dimension:** The size of the image is represented as a digital matrix. The number of pixels in the image is calculated by multiplying the number of columns by the number of rows in the matrix .

4. **Noise:** Noise in an image is a random interference phenomenon (following a known or unknown probability distribution), and it corresponds to disturbances in either the acquisition device or the observed scene.

   There are numerous and diverse sources of noise in an image:

   - Noise from the shooting conditions (camera shake, scene lighting).
   - Noise associated with sampling.
   - Noise related to the scene's nature (dust, scratches)

5. **Neighbourhood:** A pixel p with coordinates (x, y) has four horizontal and vertical neighbours whose coordinates are:

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$$

   this set represents the fourth-order neighbourhood of p, the four diagonal neighbours of p have coordinates:

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

   The meetings of these two sets constitute the set of order 8.

6. **Contrast:** It is the distinct contrast between two regions of an image, more specifically the dark and light regions of this image . a high-contrast image will have almost only white and black, while a low-contrast image will contain a lot of intermediate values but not necessarily true black and white.

If $L1$ and $L2$ are the degrees of brightness, respectively, of two neighbouring areas $A1$ and $A2$ of an image, the contrast C is defined by the ratio :

$$C = L1 - L2/L1 + L2$$

7. **Luminance:** is the degree of brightness of the points in the image. It is also defined as the quotient of the light intensity of a surface by the apparent area of that surface.

   The unit of luminance is the candela per square metre, symbol $cd/m2$

8. **Contour:** A line drawn between two (or more) pixels that have a significant difference in grey level (colour). A contour is an intensity variation between the properties of two sets of points.

9. **Histogram:** The histogram of the grey levels or colors of an image is a function that gives the frequency of appearance of each grey level (colour) in the image. It allows to give a great deal of information on the distribution of the grey levels (colour) and to see between which bounds the majority of the grey levels (colour) is distributed in the case of an image is too light or too dark.

   It can be used to improve the quality of an image (Image Enhancement) by introducing some modifications, to be able to extract useful information from it. To reduce the quantization error, to compare two images obtained under different lighting, or to measure certain properties of an image, the corresponding histogram is often modified to the corresponding histogram.

10. **Greyscale images:** The greyscale is the value of the light intensity at a point.

    To represent greyscale images, each pixel in the image can be assigned a value corresponding to the amount of light reflected. This value can be between 0 and 255.

    The hardware used to display the image must be capable of producing various corresponding grey levels.

The number of grey levels depends on the number of bits used to describe the 'colour' of each pixel in the image. The larger this number, the more levels are possible.

11. **Colour images:** Multimedia applications most often use colour images. The representation of colours is carried out in the same way as for monochrome images, but with some particularities.

    First of all, a representation model must be chosen; by using their primary components, Light-emitting systems (computer screens, etc.) are based on the principle of additive synthesis. The colours are composed of a mixture of red, green and blue (RGB model).

2.4.3 Types of images

There are two types of images [25]:

2.4.3.1 Vector

A geometric description is included in an image's vector description. Each elementary form is an object with a set of attributes such as colour, transparency, and so on.

One of the benefits of vector images is that they are especially well suited to schematic and stylized representations consisting of geometric shapes uniformly filled with flat patterns.

However, it cannot encode an analog image, such as a photographic image, and each vector file format has its own attributes, making format compatibility difficult.

2.4.3.2 Bitmap

A bitmap image is computed in point mode, the most universal coding system consisting in decomposing the graphic representation, the image has a certain number of elementary points defined by their spatial coordinates and color. As a result, it is a graphic representation defined by the set of points that comprise it.

It allows anti-aliasing, contour enhancement, and local image modifications; it can be manipulated and treated by technical operations for a graphic designer who finds tools and manipulation very similar to those that characterize his trade and professional practice of analogical type.

However, bitmap images have a fixed resolution and are heavy and difficult to transport over a network.

2.5 Feature extraction

There are several image features, or descriptors, that can be used to measure image similarity or classification of images or parts of images. In general, we can discriminate between global and local characteristics [16].

2.5.1 Local features

The detection of points of interest on parts of the image using segmentation is all that the local features concern.

Local characteristics are calculated around the detected points of interest. In fact, the analysis based on local features risks losing the image's overall meaning by drowning it in insignificant details [16].

2.5.2 Global features

Global features are concerned with tracking global characteristics throughout an image, such as [16]:

- **Color:** which is defined by a specific color space or model and can be extracted from images or regions.

- **Texture:** LBP, HOG and HTD are the best-performing feature descriptors for a wide range of images.

- **Shape:** is thought to be an important cue for humans in identifying and recognizing real-world objects.

  Global features are unable to produce the desired results. It is easier to use because it considers the entire image. As a result, the descriptor is less sensitive to distortions and perturbations between images.

2.6 Thermal imaging (TI) as a technique for process analysis in the food industry [40]

Thermal imaging is recognized as a key technology in medical and military applications for detection, monitoring, and diagnosis.

However, it has recently become a powerful technology in other industries, such as agriculture and food processing, where it is applied to unchanging food and biological systems.

Infrared (IR) imaging is a two-dimensional, non-contact diagnostic technique for measuring the surface temperature of materials and observing temperature gradient variations over time. It is frequently regarded as a critical success factor in plant maintenance, as well as in the monitoring of energy efficiency and productivity.

Thermal imaging is a new tool for feature extraction that can be used for quality assessment, detecting bruises and foreign bodies, and assessing grain quality.

These infrared imaging methods have several advantages, including the fact that they are non-invasive, allow for rapid measurement at a distance from the subject, and provide the surface temperature of objects rather than the ambient air temperature.

2.7 Related works

This section is devoted to the analysis of the results of different papers that focus on dates fruits classification.

- **Dates Fruits Classification Using SVM [14]:** The main goal of this paper

[14]is to automate the process of recognizing fruit images. SVM is used to classify different types of dates based on their images. Dates have a variety of interesting characteristics that can help distinguish and determine a specific date type. Shape, texture, and color are examples of these characteristics. The model achieved 100 percent accuracy using 120 samples and was designed to classify dates that can and cannot be eaten.

- **Image-based deep learning for automatic date fruit sorting [33]:** The authors conducted this study [33]in 2019 with the goal of presenting a novel and accurate method for distinguishing healthy date fruits from defective ones. On an image dataset with four classes, namely Khalal, Rutab, Tamar, and defective date, a CNN model was trained and tested. The CNN model had an overall classification accuracy of 96.98 percent.

- **Date Fruit Recognition using Feature Extraction Techniques and Deep Convolution Neural network [30]:** This paper[30] presents a framework for date recognition using Deep Learning techniques based on color, shape, and size feature extraction methods. For the experiments, three types of dates were chosen: Aseel, Karbalain, and Kupor. A total of 500 date fruit samples were collected, with 350 used for training and 150 used for testing. During the experiments, the best cumulative accuracy of 97.2 percent is achieved.

- **A Deep Learning-Based Model for Date Fruit Classification: [13]** A CNN-based model is proposed by the authors of this article [13], which is capable of classifying eight different popular date fruits in Saudi Arabia. the proposed model is trained on an in-house dataset that contains around 1750 images of eight different date fruits and achieved 99% accuracy.

2.8 Conclusion

This chapter identified the most important fundamental elements and concepts pertaining to quality control, as well as its methods and types. It has covered the various image processing techniques that aid in the assessment of food quality, as well as the presence of thermal imaging as a quality control technique in the food industries, providing a general view on our topic which is An application to date sorting.

Chapter 03

Conception of the architecture for the classification of dates fruits

Chapter 3

Conception of the architecture for the classification of dates fruits

3.1 Introduction

The rapid advancement of AI technologies and their applications in agriculture creates entirely new opportunities for intelligent systems to better forecast trends and assist farmers in making decisions. This chapter will provide a general overview of the classification system for the ten types of date fruits that we propose.

First, we present the general architecture of our CNN-based classification model. Then, in the following sections, we will go over the specifics of how our model works.

3.2 Problem statement

Convolutional Neural Networks (CNNs) are designed to map one image data as an input. Most of the literature works focus on enhancing the architecture itself, feeding only one fruit side as input which may lead to the overfitting phenomenon because the information are not completely provided. Others, use multi-input CNNs with all fruit sides. However, this method is time-consuming putting into consideration CNN's complexity since each side passes through the CNN (4 sides = 4 CNNs). Hence, in such cases image processing is a gold standard to overcome this issue, but still choosing the best processing tool is required (for example: concatenating all sides with a thermal image require a permutation task to not have a labeling issue).

as a solution, we suggested a new simple yet effective processing method that explores all fruit faces with the thermal image and other characteristics as the date weight. More details will be presented in the next sections.

3.3 General architecture

In general, our date quality assessment system will follow certain steps as shown in the Figure in the next page:

The system starts with the acquisition phase where we learn pictures for each date, each image with its class after which we apply a preprocessing on the images the result will be a database. In the third phase, we feed our system with the previous database to learn In the third phase, we feed our system with the previous database to learn and obtain a new model to predict the class of the new date images in the images of dates in the prediction phase.

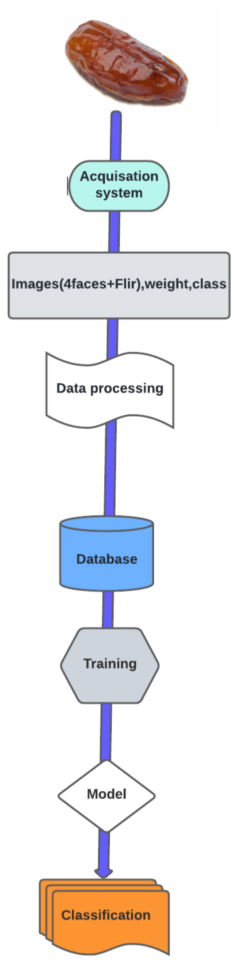

Figure 3-1: General Architecture

3.4 Detailed architecture

In order to build a deep learning model for CNN classification that can accurately classify, we must create a system architecture as shown in the figure on the next page.

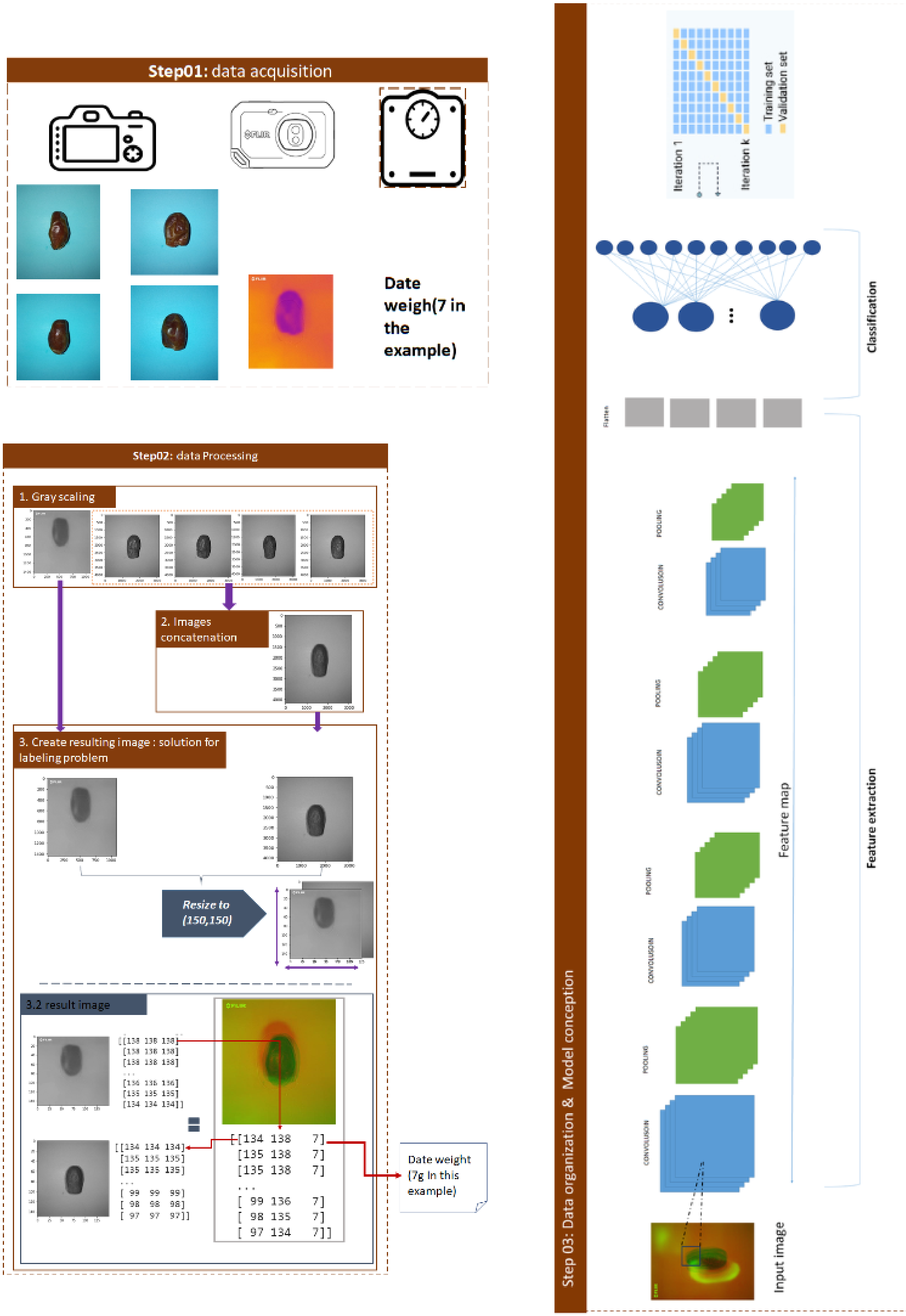

Figure 3-2: Detailed architecture

3.4.1 Step 01: data acquisition [40]

The first step in any image processing system is image acquisition. The overarching goal of any image acquisition is to convert an optical image (real-world data) into an array of numerical data that can then be manipulated on a computer. Special devices are used to capture images [40].

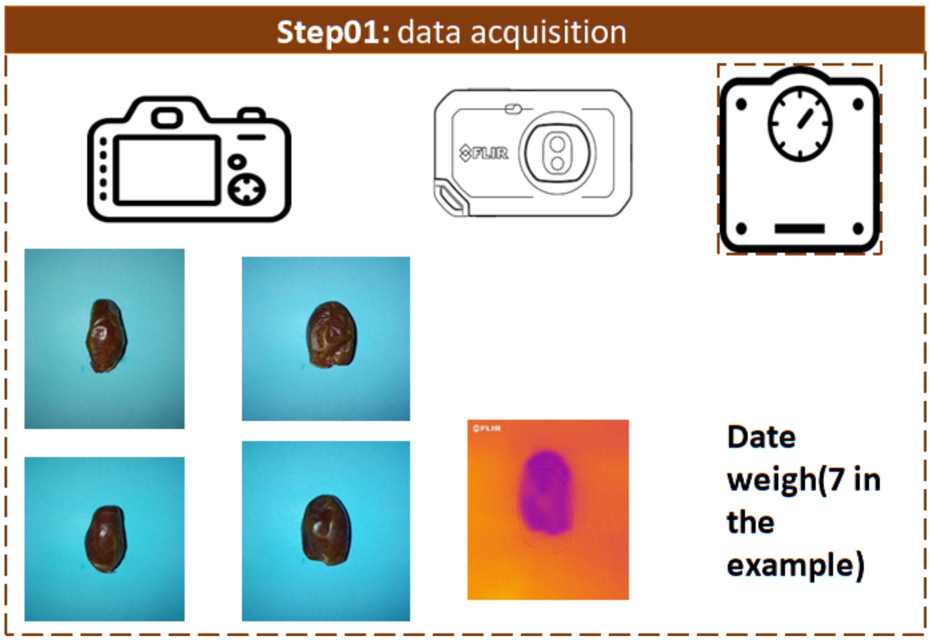

Figure 3-3: Data Acquisition

Thermal images, 4 color images (one for each side of the date), weight, and a class of each date are the data collected from each date.

- **Lighting:** is a critical prerequisite for image acquisition for food quality assessment. The lighting conditions can have a significant impact on the quality of the captured image.

  A high-quality image can help to reduce the time and complexity of subsequent image processing steps, lowering the overall cost of an image processing system. Four fluorescent lamps were used to illuminate the samples in our system.

- **Background:** The images were taken with a uniformly colored background (no texture), and a blue cast was selected for this step. This step requires a blue cast. There were no size restrictions on the image.

- **Weight:** The samples were weighed using a digital scale.

Figure 3-4: Digital scale

- **Thermal image:** The thermal images were captured using a "Flir" thermal camera in the "one" edition.

Figure 3-5: Flir thermal camera

The dataset collected included more than 5000 images of different quality dates. Each date was manually classified according to its quality by experts in one of the 10 classes.

These classes included respectively from 1 to 10: 109,105,69, 104,80, 63, 120, 203,140,110 of dates

3.4.2 Step 02: data processing

The pre-processing phase is a crucial phase in image classification tasks. images have to be looked at, shape, size, noise, and pixels are preliminary things, and for this we use image processing techniques to do all this work.

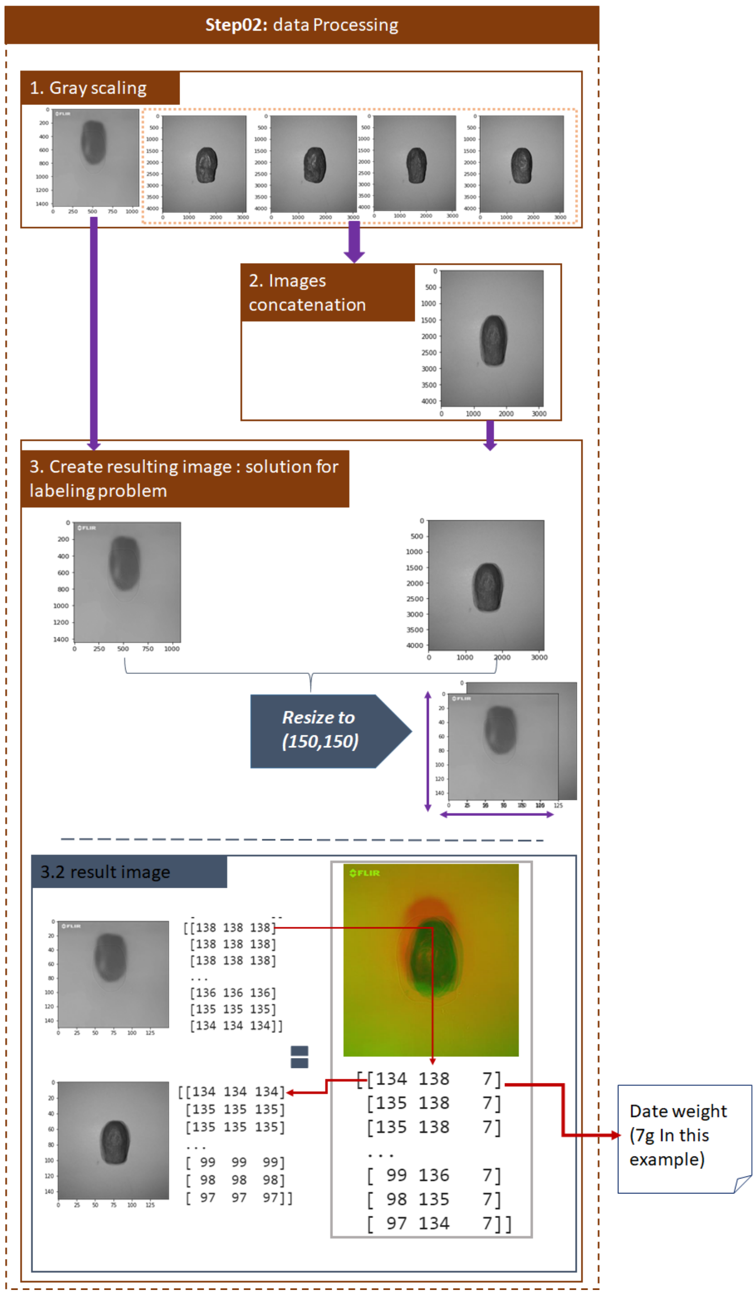

Figure 3-6: Data Processing

1. **Grayscalling:** Light is more sensitive to the human eye than chrominance, but luminance exists in all three RGB color components. As a result, the RGB system is not ideal for data analysis.

   For the five images, we use a grayscale transformation (the 4 faces and the thermal photo of the same date).

   The transformation is carried out using the following formula [23]:

$$grayscale = (0.299 * r + 0.587 * g + 0.114 * b)$$

2. **Image concatenation:** In this part, we concatenate the 4 images of the four sides of date fruit by calculating their average.

$$Image\,Result\,[i,j] = (image1[i,j] + image2[i,j] + image3[i,j] + image4[i,j])/4$$

   Or

$$Image\,Result\ = (\sum_{i=1}^{i=4}\sum_{j=1}^{j=4} image_k)/4 \qquad (k = 1..4)$$

3. **Result image creation:** This part concerns the creation of the final (result) image considered as the input of the CNN network.

   It is necessary to standardize the size of the FLIR image with the image resulting from the concatenation of the four faces, after that, a resulting image is created where each pixel of this one takes the following form:

   the red value takes the greyscale value of the averaged image, the green takes the grey scale value of the FLIR and the blue takes the date's weight.

   we repeat this collection for all pixels of the resulting image.

3.5 Database organization

Model training is the key step in machine and deep learning that results in a model that is ready to be validated, tested, and deployed. in this phase, we attempt to fit the best combination of weights and bias to a machine learning algorithm in order to minimize a loss function over the prediction range.

The model creation phase is divided into several steps. The first step is to split the used dataset into two subsets: the training set and the test set.

The most important step in the training phase occurs after the model is chosen. The trained model is then tested to see if it passes the evaluation, and if it does, we proceed to the stage of setting the hyperparameters.

3.5.1 Data split

The operation is to divide a dataset into two subsets. The training dataset is the first subset that is used to fit the model.

The second subset is not used to train the model, but the input from the dataset is fed into the model, and predictions are made and compared to expected values. This second data set is known as the validation data set.

We used the "Stratified K-Fold" cross-validation method to obtain those subsets.

3.5.2 Stratified k fold cross-validation

is an extension of the cross-validation technique used for classification problems. It maintains the same class ratio throughout the K folds as the ratio in the original dataset. So, for example, you are dealing with diabetes prediction in which you have the class ratio of 70/30; by using stratified K fold, the same class ratio is preserved throughout the K folds. In our case, k is chosen 10 [12].

3.6 Model conception

In this study we tried to implement two different CNN deep learning models, Google Inception and another one built from scratch.

3.6.1 Inception-based transfer learning[31]

Experts have created a variety of existing models that can be used for a variety of purposes. These models are designed for specific purposes; for example, some models are better suited to handling text, while others may be better suited to handling images.

The GoogLeNet architecture [31] increased the accuracy of state-of-the-art recognition by stacking Inception layers with variable receptive fields generated by different core sizes. Before computationally expensive layers, these kernels allowed for reduced dimensionality.

3.6.2 Deep learning Model [37]

The used CNN architecture contains 4 convolutional blocks, 2 fully-connected layers, and a Softmax layer as the output prediction (as shown in figure on the next page 3-7)

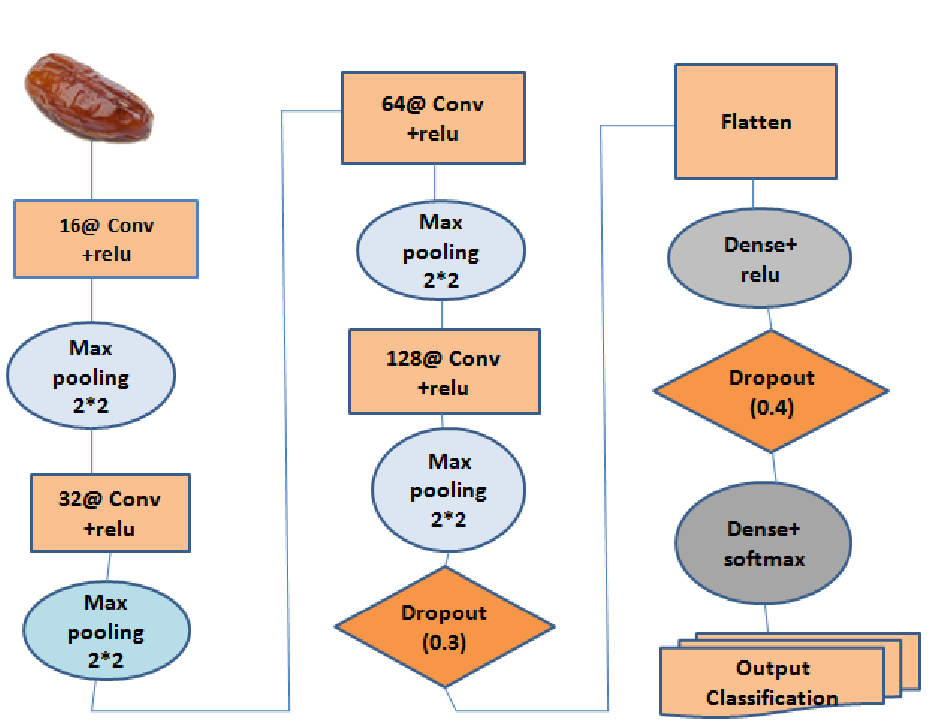

Figure 3-7: Basic Model [37]

3.7 Performance evaluation [22]

- **Classification Metrics:** t is very important to test the trained model in non-learned data. Many techniques and metrics are used to qualify and evaluate the obtained results by the trained model. These metrics are calculated either on the training examples themselves or on the reserved samples.

- **Precision:** The intuitive metric used is the precision of the model also called the recognition rate. It represents the ratio between the number of correctly classified data and the Number of data tested. The following equation is used to calculate the precision of the model.

- **Confusion matrix** The confusion matrix is a matrix that gathers in rows the observations, and in columns the predictions. The elements of the matrix represent the number of examples corresponding to each case.Here, we assume the class containing AF signals as the positive class, and therefore the sensitivity, specificity, F1 score are computed, respectively.

- **Sensitivity:** it represents the ratio of correctly predicted positive observations and the number of positive observations.

$$Sensitivity = TP/TP + FN$$

- **Specificity:** it represents the ratio of correctly predicted negative observations to the total number of negative observations.

$$Specificity = TN/TN + FP$$

- **F1 score (Fmeasure):** the F score is the harmonic mean of the precision and recall.

$$F_1 score = \frac{2 * S_v * S_p}{S_v + S_p}$$

3.8 Conclusion

We have provided a general description for designing a date classification model using Deep Learning with a Convolutional Neural Network (CNN) in this chapter, by presenting the two phases of operation, namely the training and classification phases.

The following chapter is dedicated to the implementation of the steps described in the conceptual chapter, as well as the evaluation of the proposed model and discussion of the results obtained.

Chapter 04

Experimental Study

Chapter 4

Experimental Study

4.1 Introduction

In this chapter, we will try to implement and validate our idea by presenting the working environment, programming languages, and tools we used to build our system.

Subsequently, we will explain all the experiments carried out on the proposed methods as well as the obtained results.

4.2 Development tools and programming languages

During the implementation phase of this project, it was necessary to use certain areas of programming and we had the opportunity to familiarize ourselves with various development software techniques and tools which are presented below:

4.2.1 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). This language can be used on a server to create web applications, connect to database systems. It can also read and modify files, handle big data and perform complex mathematics and so many other functionalities [11].

Figure 4-1: Python [11]

4.2.2 PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, and supports web development with Flask [10].

Figure 4-2: PyCharm [10]

4.2.3 Flask

Flask is an open-source Python web development framework. Its main goal is to be light, in order to keep the flexibility of Python programming, associated with a system of templates. Applications that use the Flask framework include Pinterest and LinkedIn. The distribution includes data-science packages suitable for Windows, Linux, and macOS [9].

Figure 4-3: Flask [9]

4.2.4 Google Colab

Training a deep learning model can require a large CPU/GPU workload, so we used the Google Colab cloud platform for this task.

Colaboratory is a Google research project created to help spread machine learning education and research. It is a Jupyter notebook environment that requires no configuration to use and runs entirely on the cloud [4].

Figure 4-4: Google Colab [4]

4.2.5 TensorFlow

TensorFlow is an open-source programming framework for open-source numerical computation published by Google in November 2015. Since its release, TensorFlow has continued to grow in popularity, quickly becoming one of the most widely used frameworks for Deep Learning and thus neural networks.

Its name is inspired by the fact that the current operations on neural networks are mainly done via multi-dimensional data tables, called of multi-dimensional data, called Tensors. A two-dimensional Tensor is equivalent to a matrix.

Today, Google's main products are based on TensorFlow: Gmail, Google Photos, and Voice Recognition [8].

Figure 4-5: TensorFlow [8]

4.2.6 Keras

Keras is a high-level neural network API, written in Python and capable of running on TensorFlow or Theano. It was developed with a focus on rapid experimentation. Being able to get from an idea to a result with as little delay as possible is key to doing good research. the key to doing good research. It was developed as part of the ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) research effort, and its main author and maintainer are François Chambert, a Google engineer.

In 2017, the TensorFlow team at Google decided to support Keras in the main TensorFlow library. Chollet explained that Keras was designed as an interface rather than an end-to-end learning framework. It presents a set of higher-level abstractions and more intuitive abstractions that make it easier to configure neural networks independently of the backend computer library. Microsoft is also working to add a CNTK backend to Keras as well [5].

Figure 4-6: Keras [5]

4.2.7 Numpy

NumPy is a Python library used to work with arrays. It functions for working with linear algebra, Fourier transforms, and matrices. transform and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely [6].

Figure 4-7: Numpy [6]

4.2.8 Flask-dropzone

Flask-Dropzone packages Dropzone.js into an extension to add file upload support for Flask. It can create links to serve Dropzone from a CDN and works with no JavaScript code in the application [3].

Figure 4-8: Flask-dropzone [3]

4.3 Database Description

The database provided consists of 5515 specimens, LCD images of the four faces with a blue background and a Flir one of the same date,(1103 Flir images, 4412 Lcd images). We also mention that each sample has been weighed, and classified into several categories (10 Qualities) according to some features depending on experts.

Dates were assigned to these classes in the following order: 109,105,69, 104,80, 63, 120, 203,140,110. The next figures describe how the database is displayed.

The figure below shows the 10 folders(10 qualities) where the samples are classified.

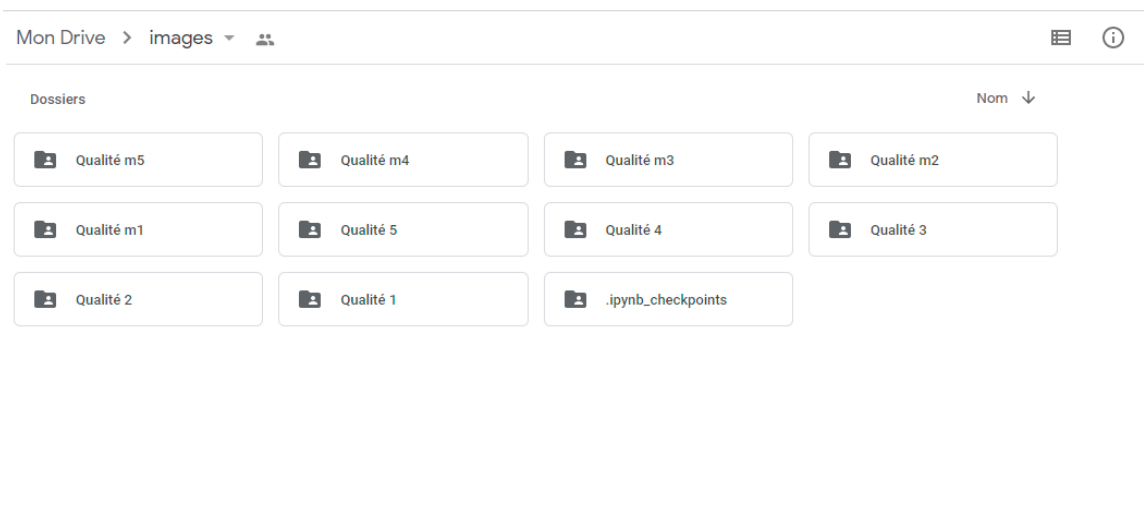

Figure 4-9: preview on database display

The next figure is an overview on "quality 1" presentation that shows sub-files on samples where each one is described with her number and it weight.

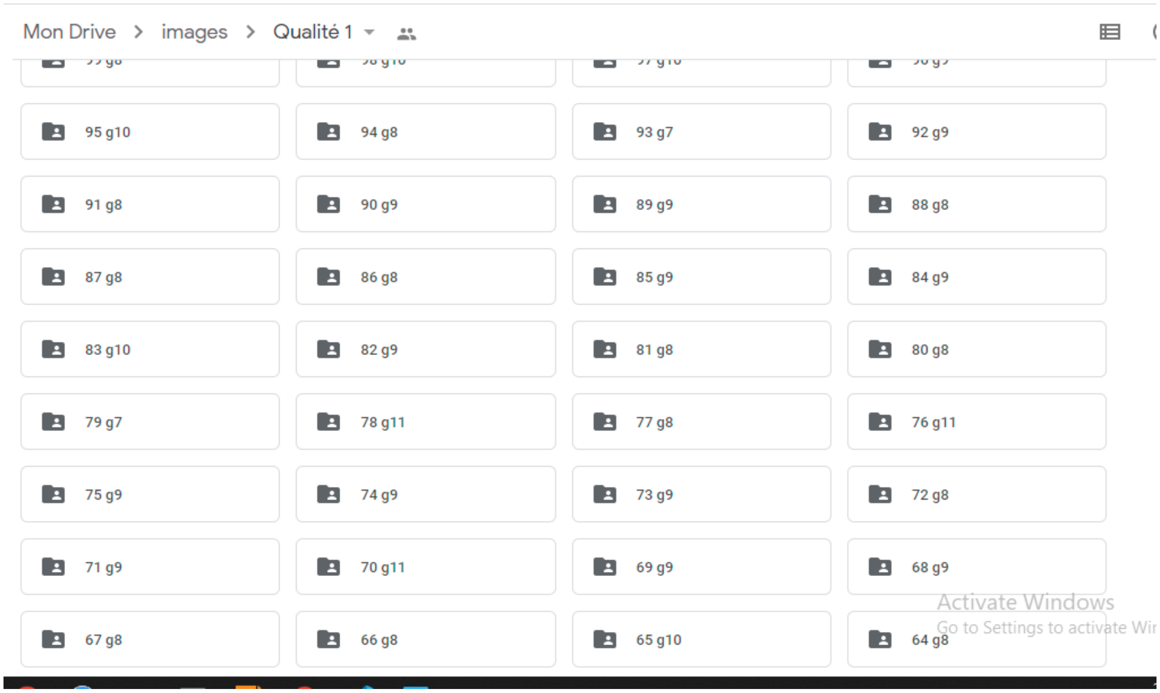

Figure 4-10: preview on Quality display

The figure 4-11 shows an insight on one sub-file where it contain 4 LCD images for the four sides and one Flir image.

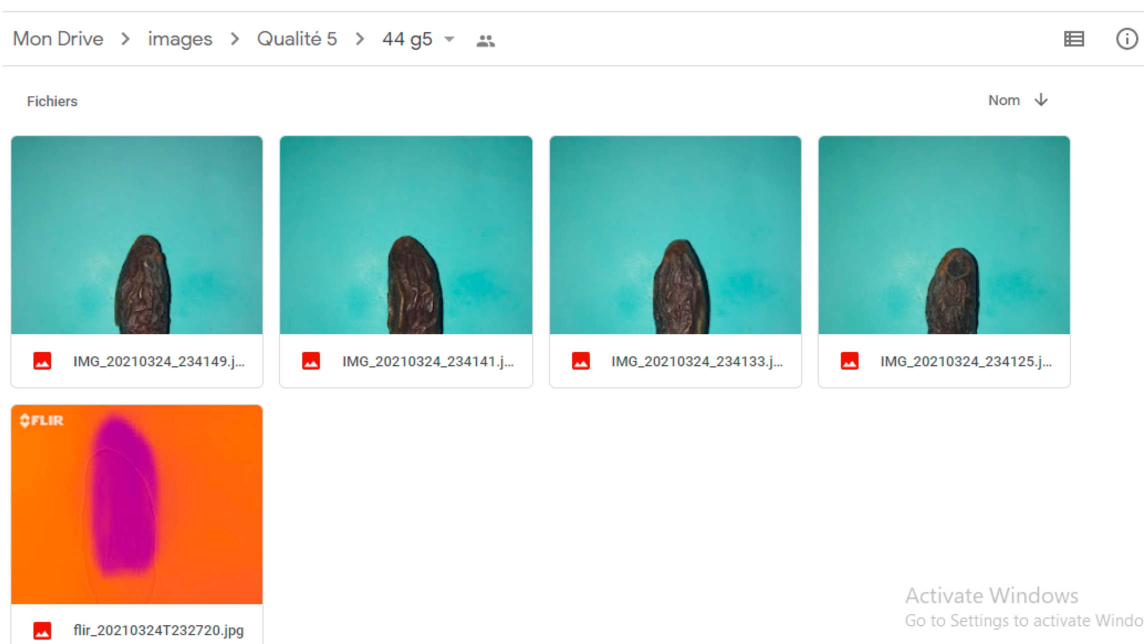

Figure 4-11: LCD + FLIR images: an example

4.4 Application

4.4.1 Data Preprocessing

1. **Gayscaling LCD images:** The following algorithm shows the function used for the grey transformation of images using luma conversion formula recommended by the supervisor.

$$grayscale = (0.299 * r + 0.587 * g + 0.114 * b)$$

```
#function to grayscale dates
#from the website of tutorialpoints + PR.djaffel thesis
def grayscale(picture):
    res = PIL.Image.new(picture.mode,picture.size)
    width, height = picture.size
    for i in range(0,width):
        for j in range(0,height):
            r, g, b = picture.getpixel((i,j))
            grayscale = (0.299*r + 0.587*g + 0.114*b)
            res.putpixel((i,j),(int(grayscale), int(grayscale), int(grayscale)))
    return res
```

Figure 4-12: grayscalling function

The next figure shows the result of images after using the grayscaling function.

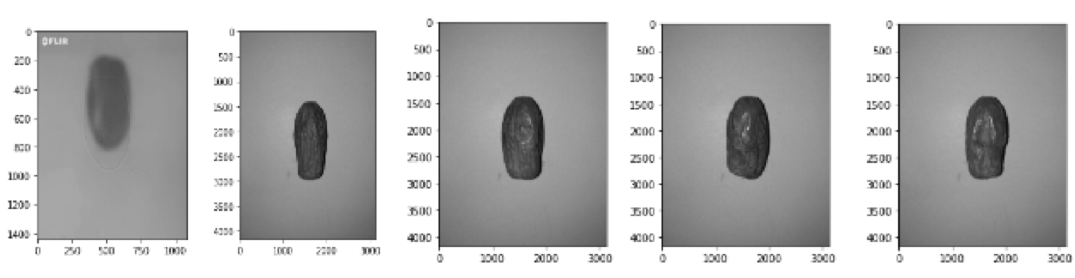

Figure 4-13: Resulting images of grayscalling function

2. **Images concatination:**

   The next algorithms shows the processus of images concatination by claculating their average, by charging the 4 images pixels to calculate their average and create a new result image. As shown in the next two figures.

```
#average of the 4 sides images
#Data standarization (flir size =! Lcd size) / minimize execution time
def image_concatiation(imgs):
    images = [Image.open(x) for x in imgs]
    images =[x.resize((150,150), Image.ANTIALIAS) for x in images]

    images = [grayscale(x) for x in images]



    pix1 = images[0].load()
    pix2 = images[1].load()
    pix3 = images[2].load()
    pix4 = images[3].load()


    result_img = PIL.Image.new(images[0].mode,(150,150))
    pixels = result_img.load()

    for i in range(result_img.size[0]):
        for j in range(result_img.size[1]):
          pix=(pix1[i, j],pix2[i, j] ,pix3[i, j] ,pix4[i, j])
          pixels[i, j] =tuple([int(sum(x) / len(x)) for x in zip(*pix)])
    return result_img
```

Figure 4-14: Concatenation of the four sides function

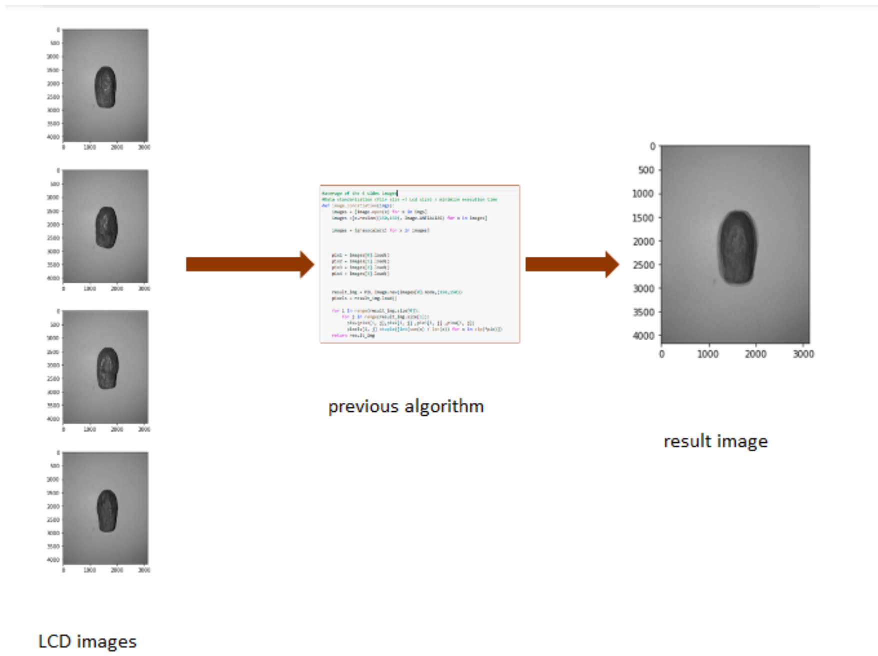

Figure 4-15: Preview on 4 images concatenation

3. **Creation result image:**

   The algorithm below explain the processus of final image creation, but first we need to resize the FLIR image to LCD ones to (150,150) then we load the pixels of the flir image and the image of the four sides. a resulting image is created where each pixel of this one takes the following form:

   the red value takes the greyscale value of the averaged image, the green takes the grey scale value of the FLIR and the blue takes the date's weight.

```
#Result image creation
def concatiation(concatinated,flir, poid):
    #concatinated = [img1,img2,img3,img4]
    img = image_concatiation(concatinated)
    img = img.resize((150,150), Image.ANTIALIAS)
    flir = Image.open(flir[0])
    flir = flir.resize((150,150), Image.ANTIALIAS)
    flir = grayscale(flir)

    pix1 = img.load()
    pix2 = flir.load()
    matrix= np.matrix(img.getdata())
    print(matrix)
    matrix= np.matrix(flir.getdata())
    print(matrix)



    result_img = PIL.Image.new(img.mode,(150,150))
    pixels = result_img.load()

    for i in range(result_img.size[0]):
        for j in range(result_img.size[1]):
          r, g, b = result_img.getpixel((i, j))
          r = (pix1[i, j][0] + pix1[i, j][1] + pix1[i, j][2])/3
          g= (pix2[i, j][0] + pix2[i, j][1] + pix2[i, j][2])/3 #[126,126,126]
          b=poid #9
          pixels[i,j]=tuple([int(r),int(g),int(b)])  #(128,145,9)
    return result_img
```

Figure 4-16: Preview on final result

4.4.2 Input data

When using TensorFlow as backend, Keras CNNs require a 4D array (which we'll also refer to as a 4D tensor) as input, with shape

$$(nb_samples, rows, columns, channels)$$

where nb_samples corresponds to the total number of images (or samples), and $rows$, $columns$, and $channels$ correspond to the number of rows, columns, and channels for each image, respectively.

The function below takes a string-valued file path to a color image as input and returns a 4D tensor suitable for supplying to a Keras CNN. The function first loads the image and resizes it to a square image that is 150x150 pixels. Next, the image is converted to an array, which is then resized to a 4D tensor. In this case, since we are working with color images, each image has three channels. Likewise, since we are processing a single image (or sample), the returned tensor will always have shape $(1, 150, 150, 3)$.

The function takes a numpy array of string-valued image paths as input and returns a 4D tensor with shape $(nb_samples, 150, 150, 3)$

Here, $nb_samples$ is the number of samples (1103 Image in our case), or number of images, in the supplied array of image paths. It is best to think of $nb_samples$ as the number of 3D tensors (where each 3D tensor corresponds to a different image) in your dataset.

The input data is normalized in the range of 0 and 1 which is useful in Deep Learning for identifying patterns between features. Quality label (output) is coded in a 1-dimensional array of 10 columns, representing the affiliation to one of the four classes as shown in the table below.

Class	Encoding	Output
Quality 1	[1,0,0,0,0,0,0,0,0,0]	0
Quality 2	[0,1,0,0,0,0,0,0,0,0]	1
Quality 3	[0,0,1,0,0,0,0,0,0,0]	2
Quality 4	[0,0,0,1,0,0,0,0,0,0]	3
Quality 5	[0,0,0,0,1,0,0,0,0,0]	4
Quality m1	[0,0,0,0,0,1,0,0,0,0]	5
Quality m2	[0,0,0,0,0,0,1,0,0,0]	6
Quality m3	[0,0,0,0,0,0,0,1,0,0]	7
Quality m4	[0,0,0,0,0,0,0,0,1,0]	8
Quality m5	[0,0,0,0,0,0,0,0,0,1]	9

Table 4.1: Data labeling

4.4.3 Design the CNN to classify Dates

4.4.3.1 Transfer learning-based architecture [31]

```
def create_model():
    pre_trained_model = InceptionV3(input_shape = (150,150, 3), # Shape of our images
                                    include_top = False, # Leave out the last fully connected layer
                                    weights = 'imagenet')
    for layer in pre_trained_model.layers:
        layer.trainable = False
            # Flatten the output layer to 1 dimension
    x = layers.Flatten()(pre_trained_model.output)
    # Add a fully connected layer with 1,024 hidden units and ReLU activation
    x = layers.Dense(1024, activation='relu')(x)
    # Add a dropout rate of 0.2
    x = layers.Dropout(0.2)(x)
    # Add a final softmax layer for classification
    x = layers.Dense  (10, activation='softmax')(x)

    model = Model( pre_trained_model.input, x)

    model.compile(optimizer = RMSprop(lr=0.0001),
                  loss = 'categorical_crossentropy',
                  metrics = ['acc'])
    return model
```

Figure 4-17: Google inception model [31]

The training process of the model was carried out in google colab. Training went for 50 epochs over 7 folds Of cross validation, for each epoch we save the accuracy, loss, validation accuracy and validation loss at the end, we print a global report about the training using $classification_report$ function from keras and the confusion matrix. We can see at the beginning of the training, accuracy was very low with a high loss value, as shown in the figure below.

```
  super(RMSprop, self).__init__(name, **kwargs)
50/50 - 7s - loss: 3.0912 - acc: 0.4074 - val_loss: 2.3261 - val_acc: 0.4304 - 7s/epoch - 137ms/step
Epoch 2/50
50/50 - 2s - loss: 1.4015 - acc: 0.5873 - val_loss: 1.7365 - val_acc: 0.5633 - 2s/epoch - 38ms/step
```

Figure 4-18: Colab result1 screenshot

After 50 epochs, at the end of the training, accuracy has a higher value (0.98.8), and a lower loss value (0.05) comparing to the beginning

```
Epoch 49/50
50/50 - 2s - loss: 0.0818 - acc: 0.9778 - val_loss: 0.9318 - val_acc: 0.8228 - 2s/epoch - 38ms/step
Epoch 50/50
50/50 - 2s - loss: 0.0509 - acc: 0.9884 - val_loss: 1.1143 - val_acc: 0.8038 - 2s/epoch - 36ms/step
```

Figure 4-19: Colab result2 screenshot

The improved accuracy over time, from 0.40 to 0.98 means that the Inception model was able to fit the data. Predicting correctly most of the time the dates quality, the validation accuracy is also considered high, it ranges between 0.80 and 0.95.

The trained model uses a different set of data for validation that was not included in the training set, thus it was able to predict correctly for completely different set of data.

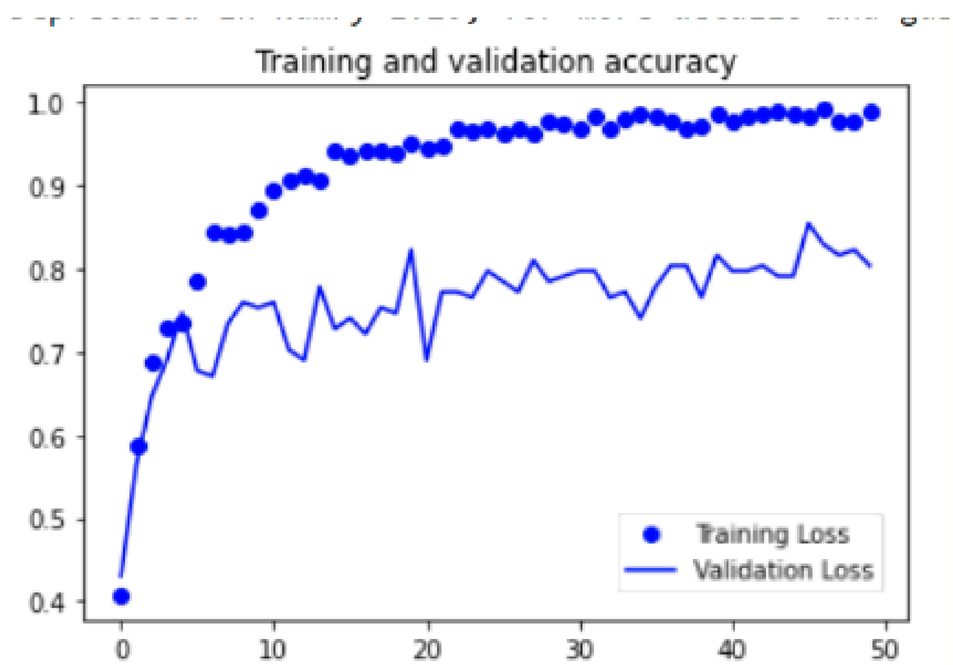

Figure 4-20: Inception accuracy: first fold

For calculating the loss, we used Categorical Cross Entropy function. It's the most common function when it comes to classification problems. Categorical Cross Entropy increases as the predicted probability diverges from the actual label. The model works on minimizing the loss function, which distills all aspects of our algorithm into a single numerical value that describes how efficient our model is.

At the earlier stages of the training, loss function had a high value (2.4), Model tries to learn form the data and thus minimizing the loss by using the back propagation as training goes. After few epochs the value drops significantly to 0.05, which indicates that the model succeeded at extracting features from the data set and identifying the pattern to correctly predict the Quality.

Figure 4-21: Inception loss: first fold

Classification report: The figure in the next page shows results of the precision, recall etc for the first folder on training and testing data.

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        93
           1       1.00      1.00      1.00        90
           2       1.00      1.00      1.00        59
           3       1.00      1.00      1.00        89
           4       1.00      1.00      1.00        69
           5       1.00      1.00      1.00        54
           6       1.00      1.00      1.00       103
           7       1.00      1.00      1.00       174
           8       1.00      1.00      1.00       120
           9       1.00      1.00      1.00        94

    accuracy                           1.00       945
   macro avg       1.00      1.00      1.00       945
weighted avg       1.00      1.00      1.00       945
 ['0', '1', '2', '3', '4', '0', '1', '2', '3', '4']
              precision    recall  f1-score   support

           0       1.00      0.81      0.90        16
           1       0.75      1.00      0.86        15
           2       0.88      0.70      0.78        10
           3       0.75      0.80      0.77        15
           4       0.64      0.64      0.64        11
           5       1.00      0.33      0.50         9
           6       0.73      0.65      0.69        17
           7       0.72      0.97      0.82        29
           8       0.94      0.75      0.83        20
           9       0.94      1.00      0.97        16

    accuracy                           0.80       158
   macro avg       0.83      0.76      0.78       158
weighted avg       0.82      0.80      0.80       158
```

Figure 4-22: Classification report on the training and testing data: first fold

The overall accuracy is taken by averaging the resulting accuracy of each fold:

Fold	1	2	3	4	5	6	7	Average
training accuracy (%)	100	94	100	92	99	99	100	97,71
Validation accuracy (%)	80	67	84	71	77	79	82	77,14

Table 4.2: Training and Validation accuracy of the first fold

class	Q1	Q2	Q3	Q4	Q5	Qm1	Qm2	Qm3	Qm4	Qm5
Quality 1	13	1	1	0	0	0	0	0	1	0
Quality 2	0	15	0	0	0	0	0	0	0	0
Quality 3	0	1	7	2	0	0	0	0	0	0
Quality 4	0	1	0	12	2	0	0	0	0	0
Quality 5	0	2	0	2	7	0	0	0	0	0
Quality m1	0	0	0	0	1	3	2	3	0	0
Quality m2	0	0	0	0	1	0	11	5	0	0
Quality m3	0	0	0	0	0	0	1	28	0	0
Quality m4	0	0	0	0	0	0	1	3	15	1
Quality m5	0	0	0	0	0	0	0	0	0	16

Table 4.3: Inception confusion Matrix

4.4.3.2 Proposal two : deep learning model architecture

```
def create_model():
    model = Sequential()
    model.add(Conv2D(filters=16, kernel_size=2, padding='same',activation='relu',input_shape=(150,150,3)))
    model.add(MaxPooling2D(pool_size=2))
    model.add(Conv2D(filters=32, kernel_size=2, padding='same',activation='relu'))
    model.add(MaxPooling2D(pool_size=2))
    model.add(Conv2D(filters=64, kernel_size=2, padding='same',activation='relu'))
    model.add(MaxPooling2D(pool_size=2))
    model.add(Conv2D(filters=128, kernel_size=2, padding='same',activation='relu'))
    model.add(MaxPooling2D(pool_size=2))
    model.add(Dropout(0.3))
    model.add(Flatten()) #1D array
    model.add(Dense(500,activation='relu'))
    model.add(Dropout(0.4))
    model.add(Dense(10,activation='softmax'))
    model.summary()
    model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

Figure 4-23: Proposed model

The training process of the model, was carried in google colab. Training went for 100 epochs Over 10 folds Of cross validation, a callbacks with an early stopping condition was added this time to prevent model overfitting [37].

We can see at the beginning of the training, accuracy was very low with a high stable loss value, as show in the figure shown on the next page.

```
Epoch 1: val_loss improved from inf to 1.56571, saving model to 2d_cnn_model.h5
100/100 - 13s - loss: 2.0348 - accuracy: 0.2893 - val_loss: 1.5657 - val_accuracy: 0.4685 - lr: 0.0010 - 13s/epoch - 135ms/step
Epoch 2/100

Epoch 2: val_loss improved from 1.56571 to 1.21346, saving model to 2d_cnn_model.h5
100/100 - 1s - loss: 1.3294 - accuracy: 0.5302 - val_loss: 1.2135 - val_accuracy: 0.5676 - lr: 0.0010 - 1s/epoch - 10ms/step
Epoch 3/100
```

Figure 4-24: Colab result3 screenshot

After 43 epochs, early stopping optimizer technique was used to reduce overfitting without compromising on model accuracy, because the val_loss value didn't improve (0,24782) The main idea behind early stopping is to stop training before a model starts to overfit.

accuracy has a higher value (0.9798), and a lower loss value (0.025) comparing to the beginning with a validation accuracy exceeded (92.79%) on the unseen data.

```
Epoch 42: val_loss did not improve from 0.24782
100/100 - 1s - loss: 0.0716 - accuracy: 0.9768 - val_loss: 0.2567 - val_accuracy: 0.9279 - lr: 1.0000e-06 - 942ms/epoch - 9ms/step
Epoch 43/100

Epoch 43: val_loss did not improve from 0.24782

Epoch 43: ReduceLROnPlateau reducing learning rate to 1.0000001111620805e-07.
100/100 - 1s - loss: 0.0617 - accuracy: 0.9798 - val_loss: 0.2563 - val_accuracy: 0.9279 - lr: 1.0000e-06 - 923ms/epoch - 9ms/step
Epoch 43: early stopping
```


Figure 4-25: Colab result4 screenshot

The improved accuracy over time, from 0.48 to 0.98 means that the inception model was able to fit the data predicting correctly most of the time the dates quality, the validation accuracy is also considered high, it ranges between 0.90 and 0.95.

The trained model uses a different set of data for validation that was not included in the training set, thus it was able to predict correctly for a completely different set of data.

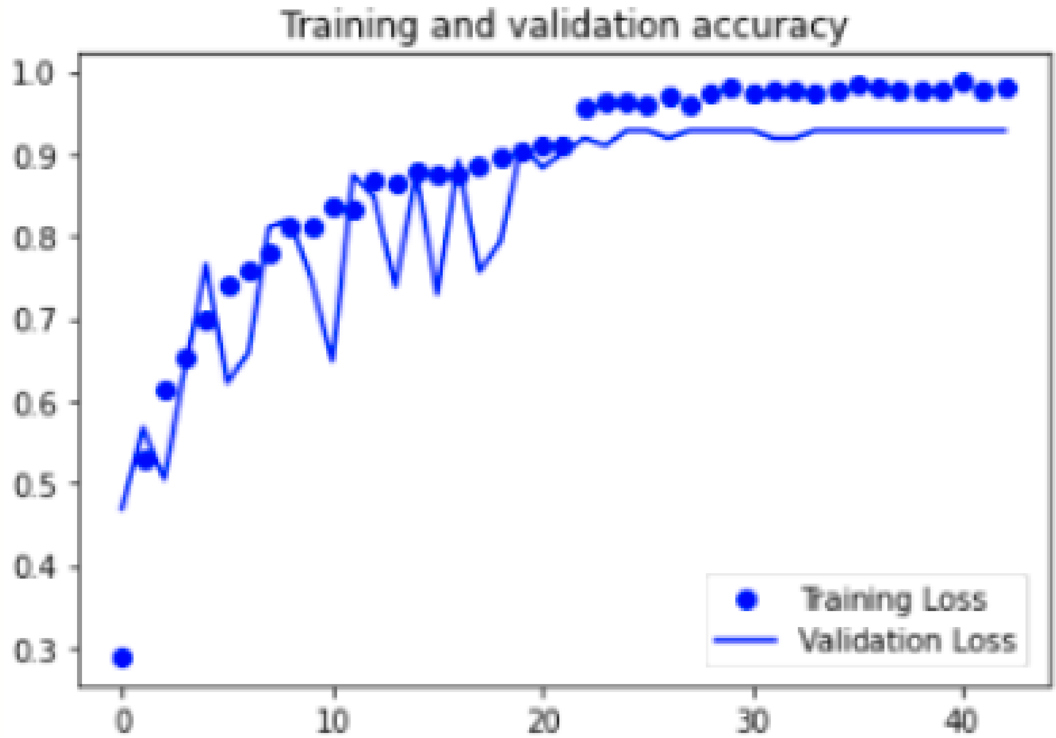

Figure 4-26: Model accuracy: first fold

For calculating the loss, we used "Categorical Cross Entropy" function. It's the most common function when it comes to classification problems. Categorical Cross Entropy increases as the predicted probability diverges from the actual label. The model works on minimizing the loss function, which distills all aspects of our algorithm into a single numerical value that describes how efficient our model is.

At the earlier stages of the training, loss function had a high value (2.1). The model tries to learn form the data and thus minimizing the loss by using the back propagation as training goes. After few epochs the value drops significantly to0.05, which indicates that the model succeeded at extracting features from the data set and identifying the pattern to correctly predict the quality.

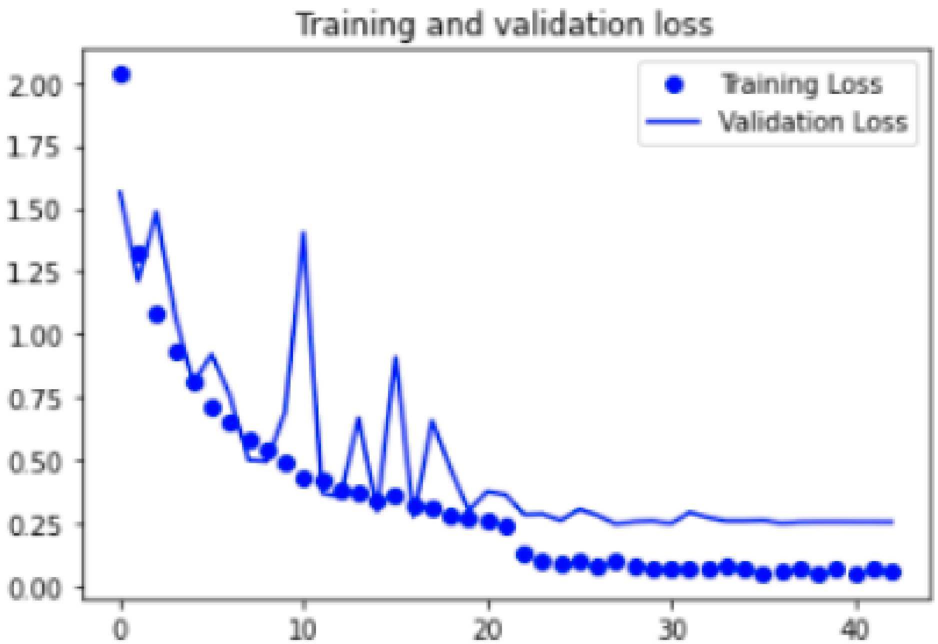

Figure 4-27: Model loss: first fold

The overall accuracy is taken by avereging the resulting accuracy of each fold :

Fold	1	2	3	4	5	6	7	8	9	10	Average
training accuracy (%)	99	100	99	99	100	97	100	98	98	100	99
Validation accuracy (%)	93	93	95	93	99	90	90	95	88	92	92.8

Table 4.4: Training and validation accuracy

4.4.4 Results Discussion and comparison

The first proposal a transfer learning method has been developed based on google inception network, we notice that after several epochs the distance between the training and validation results is getting higher producing a gap which means that our model is overffitted after (70% of accuracy, epoch 7 in figure 4-20). This issue may be resulted due to many reasons:

1. We didn't perform a grid search to choose best neurons parameters in the fully connected network.

2. Choosing k=7 may cause an issue, since usually models prefer k=10 or 5.

3. Insufficient epochs number (only 50); Learning on more number of epochs may enhance results.

4. Imbalanced distribution to some classes.

In order to solve some of these issues, we suggested using a built CNN model from scratch putting constraints on a number of folds =10, an early stopping method to not confront overfitting. The obtained model is more stable than the previous one.

We may conclude that the main advantage of CNN is that it automatically detects the important features without any human supervision. This is why CNN would be an ideal solution to computer vision and image classification problems, and choosing the right CNN architecture affects directly obtained results. Personally, I consider this a drawback since CNNs are still considered black boxes always a grid search must be performed to obtain desirable results, in contrast with SVMs that are mathematically proven and gives good results.

4.5 Model deployment, processing visualization

In order to use our model, we built an application dedicated to sorting dates according to their quality and offers a classification report that can be printed or shared as it is shown in the figures bellow.

Figure 4-28: Application Interface 01: The application accepts only one flir and four images

Figure 4-29: Application Interface 02: Images selection

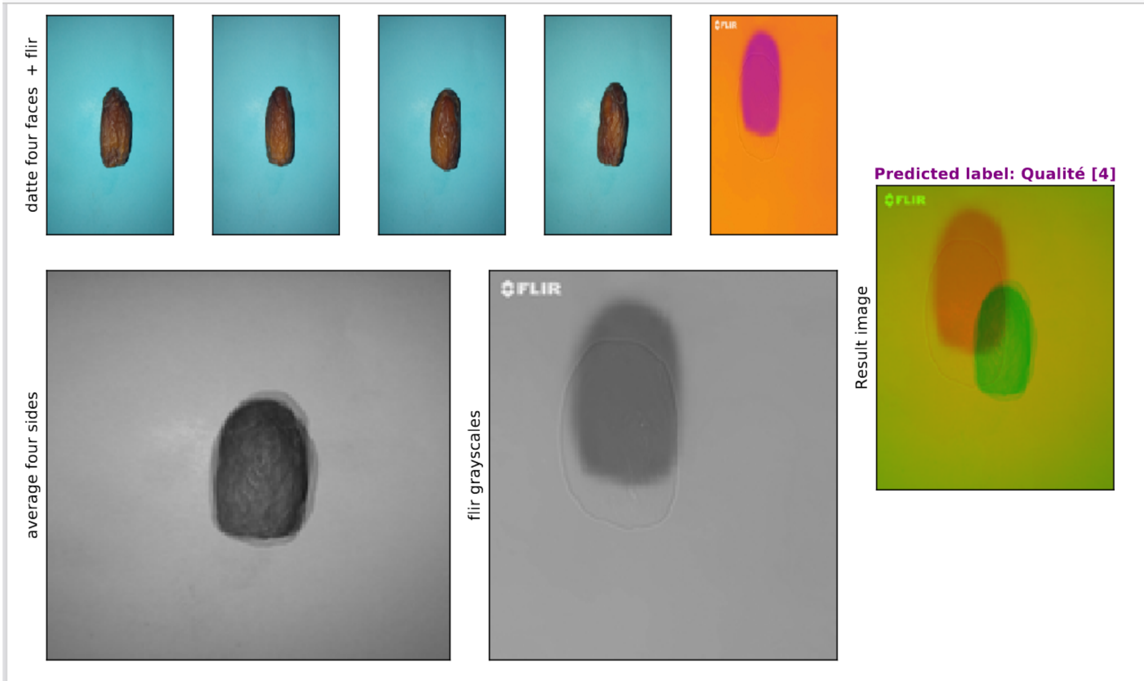

Figure 4-30: Interface showing application deployment

4.6 Conclusion

This last chapter has described the biggest steps we have done to implement our proposals. We have started with data; its acquisition, processing, and reshaping to the CNN input standards.

Then we have recited the implementation of our input data to the two CNN architectures Inception and a basic model where we achieved 97,71%, and 99% accuracy respectively as showing in pages 76 and 72.

General Conclusion

Date palms are not only an ancient fruit with important historical and symbolic significance, but are also a notable food component of several nations in the modern world, a material that is transformed into many others, and a product with a huge global market and high economic importance.

In this work, we investigated the use of deep neural networks, in particular Convolutional neural networks for the problem of multi spectral image classification. The CNN-based architecture for date sorting excels in learning relevant features directly from LCD images, thermal and weight values of a sample.

After successfully training the model, the test accuracy is calculated. For 5515 images belonging to 10 classes we achieved 77.14% "validation accuracy" and 97.71% "training accuracy" for google inception model, and 99% "validation accuracy", 92,8% "training accuracy" for the second model.

From this evaluation, we can conclude that as the number of samples increases, the performance of the CNN model increases significantly from inferred from the related works. Also, variation of the data set (use of more than one side of a date, different types of images, other characteristics) help to improve the CNN results.

Bibliography

[1] 12 importance or benefits of quality control — production management. https://www.yourarticlelibrary.com/production-management/12-importance-or-benefits-of-quality-control-production-management/26173.

[2] A beginner's guide to important topics in ai, machine learning, and deep learning. http: //www.cs.kumamoto-u.ac.jp/epslab/icinps/lecture-2.pdf.

[3] Flask-dropzone. https://flask-dropzone.readthedocs.io/en/latest/.

[4] Google colab. https://colab.research.google.com/?utm_source=scs-index.

[5] Keras. https://keras.io/.

[6] numpy. https://numpy.org/.

[7] Quality control (qc): Definition, importance and tools of quality control. https://www.yourarticlelibrary.com/production-management/quality-control-qc-definition-importance-and-tools-of-quality-control/41085.

[8] Tensorflow - learn ml. https://www.tensorflow.org/resources/learn-ml.

[9] Flask (web framework). https:/i/Flask_(web_framework), 2020.

[10] Pycharm. https:/wiki/PyCharm),2020), 2020.

[11] Python (programming language) — wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Python_(programming_language), 2020.

[12] Efosa G Adagbasa, Samuel A Adelabu, and Tom W Okello. Application of deep learning with stratified k-fold for vegetation species discrimation in a protected mountainous region using sentinel-2 image. *Geocarto International*, 37(1):142–162, 2022.

[13] Khalied Albarrak, Yonis Gulzar, Yasir Hamid, Abid Mehmood, and Arjumand Bano Soomro. A deep learning-based model for date fruit classification. *Sustainability*, 14(10):6339, 2022.

[14] Reem Alzu'bi, A Anushya, Ebtisam Hamed, Eng Abdelnour Al Sha'ar, and BS Angela Vincy. Dates fruits classification using svm. In *AIP Conference Proceedings*, volume 1952, page 020078. AIP Publishing LLC, 2018.

[15] Andreas Avgoustis, Themis Exarchos, Katia Lida Kermanidis, and Phivos Mylonas. Applied deep learning for categorizing dermoscopic images. In *2021 16th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP)*, pages 1–4. IEEE, 2021.

[16] WAFA AZZOUZ, IMTIAZ MEDELLEL. Tri automatique des pommes de terre par techniques d'apprentissage automatique - master m2-2019.pdf, university of hamma lakhdar el-oued.

[17] Abul Bashar et al. Survey on evolving deep learning neural network architectures. *Journal of Artificial Intelligence*, 1(02):73–82, 2019.

[18] Jason Brownlee. A gentle introduction to transfer learning for deep learning. `https://machinelearningmastery.com/transfer-learning-for-deep-learning/`, 2019.

[19] José Cabrero-Holgueras and Sergio Pastrana. Sok: Privacy-preserving computation techniques for deep learning. *Proceedings on Privacy Enhancing Technologies*, 2021(4):139–162, 2021.

[20] M Chambefort and J Messud. Building and understanding deep neural networks components for seismic processing: Lessons learned. In *82nd EAGE Annual Conference & Exhibition*, volume 2020, pages 1–5. European Association of Geoscientists & Engineers, 2021.

[21] Kaidi Dalia. *Classification non supervisée de pixels d'images couleur par analyse d'histogrammes tridimensionnels*. PhD thesis, Université Mouloud Mammeri, 2017.

[22] Abdelhamid Djeffal. Cours fouille de données avancée.

[23] Abdelhamid Djeffal. Doctorat en science, mai 2012, option: informatique.

[24] Abdelhamid Djeffal. Classification concepts. `http://www.abdelhamid-djeffal.net/web_documents/classificationconcepts2021.pdf`, 2021.

[25] Patrick Finot. La différence entre une image bitmap et une image vectorielle, 2018.

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[27] Xiaolei Huang, Linzi Xing, Franck Dernoncourt, and Michael J Paul. Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition. *arXiv preprint arXiv:2002.10361*, 2020.

[28] Ajitesh Kumar. Different types of cnn architectures explained: Examples. https://vitalflux.com/different-types-of-cnn-architectures-explained-examples, 2022.

[29] Hooi Ren Lim, Kuan Shiong Khoo, Wen Yi Chia, Kit Wayne Chew, Shih-Hsin Ho, and Pau Loke Show. Smart microalgae farming with internet-of-things for sustainable agriculture. *Biotechnology Advances*, page 107931, 2022.

[30] Aurangzeb Magsi, J Ahmed Mahar, Shahid Hussain Danwar, et al. Date fruit recognition using feature extraction techniques and deep convolutional neural network. *Indian Journal of Science and Technology*, 12(32):1–12, 2019.

[31] Romany F Mansour and Nojood O Aljehane. An optimal segmentation with deep learning based inception network model for intracranial hemorrhage diagnosis. *Neural Computing and Applications*, 33(20):13831–13843, 2021.

[32] Vikas Kumar Mishra, Shobhit Kumar, and Neeraj Shukla. Image acquisition and techniques to perform image acquisition. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 9(01):21–24, 2017.

[33] Amin Nasiri, Amin Taheri-Garavand, and Yu-Dong Zhang. Image-based deep learning automated sorting of date fruit. *Postharvest biology and technology*, 153:133–141, 2019.

[34] Yanwei Pang, Manli Sun, Xiaoheng Jiang, and Xuelong Li. Convolution in convolution for network in network. *IEEE transactions on neural networks and learning systems*, 29(5):1587–1597, 2017.

[35] DEHIMI Raouane. Qos-driven self-adaptive iot system/master thesis. 2021.

[36] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3):1–21, 2021.

[37] satih. https://github.com/satishf889/fruit-classifier-model.

[38] Tarang Shah. Train, validation and test sets. *Machine Learning. Obtenido el*, 14(03):2019, 2017.

[39] Linda G Shapiro, George C Stockman, et al. *Computer vision*, volume 3. Prentice Hall New Jersey, 2001.

[40] MOHAMED Aymene Slimane. Imagerie thermique pour le contrôle qualité,master thesis.

[41] Wei Song, Zijian Wen, Zhiyong Xiao, and Soon Cheol Park. Semantics perception and refinement network for aspect-based sentiment analysis. *Knowledge-Based Systems*, 214:106755, 2021.

[42] Tsvetan Tsvetkov. Application of neural network models for analysis of factors influencing patent activity. *NTUT Journal of Intellectual Property Law and Management*, page 19, 2021.

[43] Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, and Valentino Zocca. *Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*. Packt Publishing Ltd, 2019.

[44] Shuai Wang, Yexin Yang, Yanmin Qian, and Kai Yu. Revisiting the statistics pooling layer in deep speaker embedding learning. In *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5. IEEE, 2021.