



**Duarte Dias Bragadesto**

Master of Science

**Nutritional Value extraction of food exploiting  
computer vision and near infrared Spectrometry**

Dissertation submitted in partial fulfillment  
of the requirements for the degree of

Master of Science in  
**Electrical and Computer Engineering**

Adviser: Professor, José António Barata de Oliveira,  
NOVA University of Lisbon

Co-adviser: Researcher Engineer, Francisco Antero Cardoso Marques,  
UNINOVA University of Lisbon

Examination Committee

Chairperson: Professor Dr. José Manuel Matos Ribeiro da Fonseca  
Member: Researcher Dr. Sanaz Nikghadam Hojjat



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA

September, 2019



## **Nutritional Value extraction of food exploiting computer vision and near infrared Spectrometry**

Copyright © Duarte Dias Bragadesto, Faculty of Sciences and Technology, NOVA University Lisbon.

The Faculty of Sciences and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.



*To my family and friends*



## ACKNOWLEDGEMENTS

I would like to express my gratitude to my adviser, José António Barata de Oliveira and co-adviser, Francisco Antero Cardoso Marques, for all the support, patience, and help demonstrated during the investigation.

An acknowledgement also to Faculdade de Ciências e Tecnologia of Universidade Nova de Lisboa (FCT-UNL) for providing me the tools necessary to complete this dissertation and for the entire master's degree.

I would like to give a special acknowledgement to my colleagues who accompanied me throughout all these years of college, especially to Nuno Muchaxo, Ricardo Mota, Tiago Figueiredo, Pedro Rodrigues, Afonso Campos, Ricardo Martins, and Thomas Noronha. Thanks to them was possible to overcome all the obstacles that arose and made me the person I am today.

A special acknowledgement to my good friend Alexandre Guerreiro for showing me the work world and teaching me to live life and be passionate about what you do.

It is impossible not to mention the people who are part of who I am. My family. My infinite gratitude to my parents, Ana Bragadesto and João Dias, for providing me with this opportunity and the unconditional love throughout this years. Without them, I would not be the man I am today. To my grandparents, Maria de Lurdes and Francisco Bragadesto, I cannot put into words everything you did for me, from welcoming me with open arms in your house to supporting me unconditionally every day. A special acknowledgement to my aunt, Rita Bragadesto, for taking me to FCT-UNL and for being the coolest aunt ever.

This acknowledgement is for my little cousins, Margarida Pina and Francisca Pina, that lit up my world in the most challenging days.

A deep acknowledgement for my younger brother, Francisco Bragadesto, for being who he is. No matter how far and how long it passes, he is always there for me, and I will always be here for him.

Finally, a special thanks to my partner, Dumitrita Angheluta, who helped me achieve this goal, sometimes far, sometimes close, but always there for me.





## ABSTRACT

---

The population growth in the last few decades has led to the development of urban areas, which induced an increased difficulty in finding quality food. The difficulty in finding quality nourishment and a growing offer in the fast-food industry due to the fast pace at which life is lived in big cities has caused increasing obesity and sedentary lifestyle. In 2016 more than 1.9 billion adults aged 18 years and older were overweight[1].

However, this tendency has started to reverse, and with the increasing concern for diseases such as obesity and diabetes, people started return to shopping in farmers markets and choosing wisely the locals where they eat, which led to the development of more healthy fast food chains. This new tendency has made new technologies appear that were created to help improve customer choices and facilitate choosing the best food items that have the best quality.

This dissertation will analyse the different devices and solutions in the market, such as near-infrared sensors and computer vision. The objective of this dissertation is to build a system that can detect which type of food item we choose and obtain nutritional information.

The development begins with researching the different options of small devices that already exist in the market and with which a person can take shopping and assist them by obtaining the nutritional information, such as SCIO or Tellspec. This device cannot detect which type of food is being analysed, so human interaction it is still needed to obtain the best results possible. However, it can return the nutritional information necessary for the first part of this dissertation's development. Besides being small (palm-handed), these sensors are also cheap and faster compared to equivalent laboratory equipment.

The second objective of this dissertation was developed to solve the lack of detection of which type of food is present in the module. To solve this problem and taking into account the objective, it was decided to use computer vision and, more specifically, image recognition and deep machine learning applied in food databases.

This dissertation's main objective is to create a module that can classify and obtain the nutritional information of different types of food. It also serves as a helping hand in the kitchen to control the quality and quantity of the food that the user ingests daily.

There will be an exhaustive testing session for the near-infrared sensors using different

---

types of fruits to prove the concept. For the computer vision, it will be applied a deep learning algorithm with supervised training to obtain a high accuracy result.

**Keywords:** Near-infrared sensor, Detection, Classification, Module, Food, Nutrition

---

## RESUMO

---

O crescimento populacional nas últimas décadas tem levado ao desenvolvimento de áreas urbanas, o que, por sua vez, levou a um aumento da dificuldade em encontrar alimentos de qualidade. Devido ao ritmo acelerado de vida nas grandes cidades, juntamente com a dificuldade em obter alimentos frescos, como vegetais e fruta, e à expansão da indústria de fast-food, levou a um aumento da obesidade e do sedentarismo. Em 2016, mais de 1,9 bilhões de adultos com 18 anos ou mais sofrem de excesso de peso[1].

Esta tendência começou a reverter-se e, com a crescente preocupação com doenças como obesidade e diabetes, as pessoas recomeçaram a comprar nos mercados de produtores e a ter mais atenção aos locais que escolhem para comer, o que levou ao desenvolvimento de redes de fast-food saudáveis. Esta tendência fez surgir novas tecnologias que foram criadas para ajudar a melhorar as escolhas dos clientes e facilitar a compra dos alimentos com a melhor qualidade.

Esta dissertação irá analisar os diferentes dispositivos e soluções existentes no mercado, como sensores de *near-infrared* e visão computacional. O objetivo desta dissertação é construir um sistema que possa detectar qualquer tipo de alimento e obter informações nutricionais.

O desenvolvimento começa com a pesquisa das diferentes opções de pequenos aparelhos já existentes no mercado e com os quais uma pessoa pode fazer compras e auxiliá-los na obtenção de informações nutricionais, como o SCIO ou o Tellspec. Este dispositivo não consegue detectar que tipo de alimento está a ser analisado, portanto a interação humana ainda é necessária para obter os melhores resultados possíveis. No entanto, pode retornar as informações nutricionais necessárias para a primeira parte do desenvolvimento desta dissertação. Além de pequenos (do tamanho da palma da mão), estes sensores também são baratos e mais rápidos quando comparados a equipamentos equivalentes de laboratório.

O segundo objetivo desta dissertação foi desenvolvido para solucionar a falta de detecção do tipo de alimento que está presente no módulo. Para resolver este problema e tendo em conta o objetivo, optou-se por utilizar a visão computacional e, mais especificamente, o reconhecimento de imagem e *machine-learning* aplicado a bases de dados alimentares.

O objetivo principal desta dissertação é criar um módulo que possa classificar e obter a informação nutricional de diferentes tipos de alimentos. Também serve como auxiliar

---

na cozinha para controlar a qualidade e a quantidade dos alimentos que o usuário ingere diariamente.

Haverá uma exaustiva sessão de testes para os sensores de *near-infrared* usando diferentes tipos de frutas para provar o conceito. Para a visão computacional, será aplicado um algoritmo de *machine-learning* com treino supervisionado para a obtenção de um resultado de alta precisão.

**Palavras-chave:** Sensores Infravermelhos, Detecção, Classificação, Módulo, Comida, Nutrição

---

# CONTENTS

<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xix</b>
<b>Listings</b>	<b>xxi</b>
<b>Acronyms</b>	<b>xxiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Framework and Motivation . . . . .	1
1.2 Research Problem . . . . .	2
1.3 Dissertation Structure . . . . .	2
<b>2 Supporting Concepts</b>	<b>5</b>
2.1 Near-Infrared Spectrography . . . . .	5
2.2 Computer Vision . . . . .	7
2.2.1 Image Classification . . . . .	11
<b>3 State of The Art</b>	<b>19</b>
3.1 Near-Infrared Spectrometers . . . . .	19
3.2 Electronic noses and tongues . . . . .	22
3.3 Computer Vision Applications . . . . .	24
3.3.1 Image Recognition and Food Classification Applications . . . . .	25
<b>4 Nutritional Value Extractor Proposed Model</b>	<b>29</b>
4.1 General Overview . . . . .	29
4.2 Module . . . . .	30
4.2.1 Devices . . . . .	31
4.3 Vision . . . . .	31
4.4 Near-Infrared Sensing . . . . .	32
4.5 Controller . . . . .	34
<b>5 Experimental Implementation and Setup</b>	<b>35</b>
5.1 Implementation . . . . .	35

## CONTENTS

---

5.1.1	Food Image Classification . . . . .	35
5.1.2	Near-Infrared Sensing . . . . .	43
5.1.3	Processing Unit . . . . .	47
5.2	Experimental Setup . . . . .	50
<b>6</b>	<b>Experimental Results</b>	<b>53</b>
6.1	Image Recognition Algorithm . . . . .	53
6.1.1	Training . . . . .	53
6.1.2	Detection and Model Metrics Evaluation . . . . .	54
6.2	Near-Infrared Sensor . . . . .	57
6.2.1	Repeatability Tests . . . . .	58
6.2.2	Accuracy Test Results . . . . .	62
6.2.3	Spectral Images Comparison . . . . .	67
<b>7</b>	<b>Conclusion and Future Work</b>	<b>71</b>
7.1	Conclusion . . . . .	71
7.2	Future Work . . . . .	72
	<b>Bibliography</b>	<b>75</b>

## LIST OF FIGURES

2.1	Workflow overview of a Fourier transform spectrometer [7] . . . . .	6
2.2	Example of computer vision application running in a camera inside a car that is analysing the road information like other cars, traffic signals and lights. [9]	7
2.3	CIE colour space example diagram. [17] . . . . .	9
2.4	Hough Transform application. [20] . . . . .	10
2.5	Overview of an artificial neural network. [26] . . . . .	11
2.6	Model of an artificial neuron. [29] . . . . .	13
2.7	Architecture of a CNN. [30] . . . . .	16
2.8	Demonstration of a convolution in a CNN. [31] . . . . .	16
3.1	Light dispersion of a prism. [34] . . . . .	20
3.2	Scio device. [35] . . . . .	21
3.3	Tellspec device. [36] . . . . .	22
3.4	Electronic nose device, Cyranose 320. [42] . . . . .	22
3.5	Electronic tongue device, ASTREE.[45] . . . . .	24
3.6	Example of computer vision image processing. [53] . . . . .	25
3.7	Calorie mama mobile app. [55] . . . . .	26
4.1	Messages schematics - This diagram represents the communication sequence that happens after the user presses the button. The green arrows represent the requests made, and the blue arrows represent the requests' results. After each step, the information obtained is processed in the controller and shown to the user. . . . .	30
4.2	Architecture of the module. The blue colour represents the beginning and the end of the process. The green colour represents the devices used and processes made by these devices. Device 1 is responsible for obtaining the RGB image necessary for the process run in Device 2. The image classification process results in Device 2 are then sent to Device 3, in which the nutritional value is extracted. After every process is complete, the information is sent to the controller to be stored and presented to the user in the lcd. . . . .	30

---

4.3	Vision process explained in more detail. Device 1 takes a picture that is stored in the database. The database will store every photo taken since it can be used to improve the image recognition algorithm. Each step of the image recognition algorithm is run in the controller. The first step consists of resizing the image to the standard in which the image recognition was trained. The next procedure consists of extracting the image features; this means running different layers with different activation functions that will subtract the necessary features. Having the features, the next step is to classify the object based on those exact features. The results of this classification are also stored in the database to be easily accessed by the user. In the end, the results are presented to the user on the LCD. . . . .	32
4.4	Near-Infrared Sensing, the process explained in detail. Device 2 needs to be manually pressed against the object to achieve better results. After that, the device starts its process that consists of striking the object with a light beam and observing how the molecules react. At the end of this process, the information about the molecules, namely its transmittance and absorbance, is stored in the device and processed in an algorithm that returns the object's chemical composition. The final results are then presented to the user. . . . .	33
4.5	Controller process architecture. The controller is used throughout the different stages of the process. At the beginning of the process, the controller is responsible for running the image recognition algorithm process represented in 4.3. After this first process, the controller needs to send the information to the user and the near-infrared sensor for the next step. In the next step, the controller will not have part of the near-infrared processing since the sensor has its own processor but will communicate with the sensor to obtain the nutritional information results and present them to the user. The controller is also present in the database where the results from each step will be stored as well as all the pictures taken. . . . .	34
5.1	Example of images from the dataset created for this dissertation. The example pictures are separated in different rows by the different types of fruit, apples, lemons, pears, and tomatoes, by this order. . . . .	38
5.2	Label program and examples of the labelling process for different fruits. The different fruits are separated in different rows, apple, lemon, pear and tomato by this order. The labelling process consists in only selecting the important parts or features of each picture as shown. . . . .	39
5.3	(a) - SCIO[72] mobile application dashboard. This page is where the user needs to choose which type of category the food that is being analysed is part of. (b) - Scan screen from where it starts the whole process. . . . .	44
5.4	(a) - This is the screen shown after the button of the sensor is pressed. (b) - The final screen before the results are shown to the user. . . . .	45



5.5	Example of a pear BRIX results in the SCIO mobile application . . . . .	46
5.6	Example of a pear scan results in the SCIO mobile application . . . . .	47
5.7	Scheme of inputs and outputs of the SBC. . . . .	48
5.8	Image of the module created. . . . .	50
5.9	(a) - 3D model of the testing area which contained the web camera and the near-infrared sensor, SCIO. (b) - 3D model of the electronics case, which included the SBC and LCD Display. (c) - 3D Model of the base where the object that is under test is put. . . . .	51
6.1	Example of the six different types of fruits present in the test dataset. . . . .	54
6.2	In this matrix it is possible to observe the results of the test dataset using the InceptionV3 model. . . . .	55
6.3	In this matrix it is possible to observe the results of the test dataset using the ResNet50 model. . . . .	56
6.4	Apple Repeatability test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes. . . . .	59
6.5	Lemon Repeatability test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes. . . . .	60
6.6	Pear Repeatability test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes. . . . .	61
6.7	Tomato Repeatability test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes. . . . .	62
6.8	Apple Accuracy test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes in the ten different apples. . . . .	63
6.9	Lemon Accuracy test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes in the ten different lemons. . . . .	64
6.10	Pear Accuracy test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes in the ten different pears. . . . .	65
6.11	Tomato Accuracy test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes in the ten different tomatoes. . . . .	66
6.12	Apple repeatability test spectral image. . . . .	67
6.13	Lemon repeatability test spectral image. . . . .	67

LIST OF FIGURES

---

6.14 Pear repeatability test spectral image. . . . .	68
6.15 Tomato repeatability test spectral image. . . . .	68
6.16 Apple accuracy test spectral image. . . . .	69
6.17 Lemon accuracy test spectral image. . . . .	69
6.18 Pear accuracy test spectral image. . . . .	69
6.19 Tomato accuracy test spectral image. . . . .	70

## LIST OF TABLES

5.1	Table containing some examples of the label's output inserted in a .csv file, and it is where the model is going to get the locations of the features and the corresponding labels. . . . .	40
6.1	Training parameters of each network. . . . .	54
6.2	Results of the classification metrics of the two models, ResNet50 and InceptionV3. . . . .	56
6.3	Both models detection time using the same prediction algorithm, and an apple as the test subject . . . . .	57
6.4	Results of the repeatability tests performed ten times in the same apple. . . .	58
6.5	Standard deviation and average for the apple test results presented in 6.4. . .	58
6.6	Results of the repeatability tests performed ten times in the same lemon. . . .	59
6.7	Standard deviation and average for the lemon test results presented in 6.6. . .	59
6.8	Results of the repeatability tests performed ten times in the same pear. . . .	60
6.9	Standard deviation and average for the pear test results presented in 6.8. . . .	60
6.10	Results of the repeatability tests performed ten times in the same tomato. . . .	61
6.11	Standard deviation and average for the tomato test results presented in 6.10. . .	61
6.12	Reference values for Apple[77], Lemon[78], Pear[79], Tomato[80]. . . . .	62
6.13	Results of the accuracy tests performed in ten different apples. . . . .	63
6.14	Standard deviation and average for the apple test results presented in 6.13. . .	63
6.15	Results of the accuracy tests performed in ten different lemons. . . . .	64
6.16	Standard deviation and average for the lemon test results presented in 6.15. . .	64
6.17	Results of the accuracy tests performed in ten different pears. . . . .	65
6.18	Standard deviation and average for the pear test results presented in 6.17. . . .	65
6.19	Results of the accuracy tests performed in ten different tomatoes. . . . .	66
6.20	Standard deviation and average for the tomato test results presented in 6.10. . .	66



## LISTINGS

5.1	Required dependencies to train the algorithm. . . . .	40
5.2	In this snippet of the code the objective is to mount the dataset folder. In line 6, the model used is defined, in this case, it is ResNet50. In line 7, it was defined the directory where the dataset is. . . . .	41
5.3	This line represents all the variables used in the training of both algorithms, InceptionV3 and ResNet50. . . . .	41
5.4	This listing represents the show_network_summary variable that was set to "True" in 5.3. This is a part of the ResNet structure that is shown in console in the beginning of the training process. . . . .	41
5.5	In this listing is possible to see the number of images as well as the number of different classes that exist in our dataset. After that, the JSON Mapping file is created and stored in the same folder of our dataset. In line 6 is the number of epochs defined in the last line of the training algorithm shown in 5.3. In line 8, is the first epoch with the 3 variables, the ETA, the value loss, and the accuracy of the CNN in the end of that moment of training. . . . .	42
5.6	Prediction code that was run to test the image recognition algorithm . . .	42
5.7	Example of a prediction result. . . . .	43
5.8	This snippet of code represents the import of all the libraries needed to execute the detection script. After that, is defined the GPIO pins that are utilised for the button. The last three lines represent the size of the picture that is being captured and the driver of the lcd. . . . .	48
5.9	This part is related to the picture capturing, and the lcd messages that are presented to the user. . . . .	49
5.10	This is the end part in the detection code. In this part, the objective is to run the model, and obtain the result of the fruit that is under analysis. . .	50



## ACRONYMS

- AI** Artificial Intelligence.
- ANN** Artificial Neural Networks.
- ARM** Advanced RISC Machines.
- CCD** Charge-Coupled Device.
- CMY** Cyan Magenta Yellow.
- CNN** Convolutional Neural Network.
- FDA** Functional Discriminant Analysis.
- FT** Fourier Transform.
- FT-IR** Fourier Transform Infrared.
- GPIO** General-Purpose Input/Output.
- GPU** Graphics Processing Unit.
- HCI** Hue, Chroma, and Intensity.
- HDMI** High-Definition Multimedia Interface.
- HSD** Hue, Saturation, and Darkness.
- HSI** Hue, Saturation, and Intensity.
- HSL** Hue, Saturation, and Lightness.
- HVC** Hue, Value, and Chroma.
- InGaAs** Indium Gallium Arsenide.
- LEDs** Light-Emitting Diodes.
- MOS** Metal-Oxide-Semiconductor.
- NIR** Near Infrared.

## ACRONYMS

---

**PbS** Lead Sulphide.

**PCA** Principal Component Analysis.

**PLS** Partial Least Squares.

**RAM** Random Access Memory.

**RGB** Red Green Blue.

**SVM** Support Vector Machine.

**TSS** Total Soluble Solids.

**USB** Universal Serial Bus.



## INTRODUCTION

### 1.1 Framework and Motivation

Since the beginning of the millennium that, the general concern for health care has increased exponentially, coupled with the rapid advancement of technology led to the appearance of many innovating and groundbreaking technologies.

These technologies are being used to help the human civilisation to eat healthily and adopt a hearty lifestyle. In this dissertation, the main focus will be on technologies that help us obtain nutritional information about a particular type of food and how these technologies can be optimised. However, since these technologies are based on other, that already exist but in a laboratory environment, it will then be necessary to cover various topics related to the technology in question and beyond.

Bringing these technologies to the general population faces many challenges. They need to be adapted and become more user-friendly, so they can be used by all, even the ones that are less technologically advanced. This means creating new devices that are simple to use as well as mobile applications or websites. Another challenge that many companies face by adapt these technologies is that they are generally large equipment in terms of size. They need to become smaller and more portable to be useful on a day-to-day basis.

The price of the equipment is also a breaking point in bringing these technologies to the general population. For example, a laboratory class NIR spectrometer can cost thousands of dollars. Equipment with this price is not in reach of the general population, and because of that, another challenge is creating affordable devices but maintaining reliability.

In conclusion, the keywords for these new devices are small, user-friendly, affordable, and reliable.

## 1.2 Research Problem

In the last decade, being healthy, and the general improvement of life quality have raised much more awareness between the general population worldwide, the governments, and tech companies. However, there is no efficient and reliable way to control the quality of the food ingested every day since the chemical composition of the food and ingredients that are bought is usually confusing and not clear enough for most of the population to understand. Due to rapid population growth, the general quality of the food ingested daily has decreased due to mass production. This means that trying to make the best decision regarding the food has become much more complicated than in the 19th century.

Summing up, the current problem that society faces is that although trying to be healthier, it is impossible to guarantee the quality of the food is eaten on a day-to-day basis. This leads to poor nutrition among the population of the big urban areas cities which can increase the possibility of diseases such as heart problems, diabetes, and, in general, decrease the life expectancy. The psychological effect that is caused because of medical conditions such as obesity and diabetes is also an important part of the problem, in which, these types of devices can also help.

This dissertation aims to solve this problem by creating a module that will be an add-on to the general population life, which is both intuitive, complete, and user-friendly. This module will analyse every type of food and return the complete nutritional information. In that way, it is possible to keep track of the food the user ingests and know precisely what type of chemical components it is eaten.

## 1.3 Dissertation Structure

This dissertation is divided into seven chapters:

- **Chapter 1** - Introduction to the dissertation theme, research problem, and dissertation structure;
- **Chapter 2** - Reviews the supporting concepts which are based on of the development and implementation of this dissertation. The supporting concepts are: Near-Infrared Spectrography and Computer Vision;
- **Chapter 3** - Presents the state of the art in research and commercial available products related to the theme and also the commercial products that were used in the implementation of the proposed model;
- **Chapter 4** - Describes the proposed model explaining step by step every process that is present in this model;
- **Chapter 5** - Details the implementation of the module proposed, including the technologies used and frameworks;

- **Chapter 6** - Review and discussion of the experimental results obtained using the developed module. The experimental results are divided into three main sections: the near-infrared sensor, the image recognition algorithm, and the controller;
- **Chapter 7** - Aggregates the conclusions, describes possible future developments and research topics on the subject, and references the main contributions of this dissertation;



## SUPPORTING CONCEPTS

This chapter will address some of the critical aspects used as the basis for developing this dissertation. This dissertation is divided into two primary key aspects: Near-Infrared Spectrography and Computer Vision. Near-Infrared Spectrography is viewed in section 3.1, explaining the essential topics around this technology. In section 3.2, the critical aspects of Computer Vision are overviewed.

### 2.1 Near-Infrared Spectrography

Near-Infrared Spectrography has been used for many years in the food industry. There are different types of spectroscopy being the the most commonly used Raman spectroscopy and standard near-infrared spectroscopy.

Raman spectroscopy [2] is named after Indian physicist Sir C. V. Raman. It uses inelastic scattering, also known as Raman scattering, which consists of the fact that an incident particle's kinetic energy is not conserved. This means that some of the incident particle's energy is lost or increased, providing a molecular fingerprint of the structure and chemical composition of the samples probed by the beam of monochromatic light from a laser in the near-infrared range. Surface-enhanced Raman spectroscopy is a technique used to improve standard Raman spectroscopy results since the Raman scattering gives inherently weak signals. The enhancement factor can be as much as  $10^{10}$  to  $10^{11}$ , which can obtain single-molecule results.

The advantage of Raman spectroscopy is that it does not require any sample preparation, which means the process becomes less time consuming and more efficient in terms of cost. Raman spectra can also be obtained without removing the product from the package in which it is contained. It means that it can be used in product quality control in the final product and not during the fabrication procedure. Another advantage of Raman

spectroscopy is that the analysis process is relatively fast, which allows quick feedback.

Standard near-infrared spectroscopy [3] is a method included in infrared spectroscopy but uses the near-infrared region, which goes from  $780\text{nm}$  to  $2500\text{nm}$  in the electromagnetic spectrum. This method is based on overtones and combination tones of molecular vibrations. It mainly detects the stretch vibrations involving hydrogen bonds, which are more intensive in the near infrared range.

This type of spectroscopy is used in many research and investigation fields, primarily in pharmaceuticals, to detect the components of a specific medicine. However, it can also be used in agricultural, dairy, and chemical industries.

These companies use this method to make sure the products they produce meet the client's requirements, increase profitability, and improve production. This means that NIR is used for quantitative analysis, determining substance concentrations, and qualitative analysis to identify raw material, intermediate and finished products.

Infrared spectroscopy is divided into three categories: non-dispersive, dispersive, and Fourier transform or multiplex. Non-dispersive infrared spectroscopy [4] is usually used as gas detectors for carbon monoxide and carbon dioxide. It functions like a standard infrared sensor in which a beam of light passes through the sampling chamber, and the components of the sample absorb some particular infrared frequency.

After this process, the sample is compared to the gas used as a reference, which allows the determination of the the gas component's concentration. In dispersive infrared spectrometers, each wavelength spectrum is calculated sequentially, and they are also called grating or scanning spectrometers because of that.

Fourier transform or multiplex infrared spectroscopy [5] replaced dispersive infrared as the most commonly used type of spectroscopy. An FT-IR spectrometer is a single-beam instrument [6] divided into two using a beam splitter (usually a thin film of germanium on a potassium bromide substrate). One beam is directed to a fixed mirror that reflects some of the energy to the source and some of the energy to the detector; the other beam is directed to a moving mirror. The result is the same, some of the energy to the source and some of the energy to the detector.

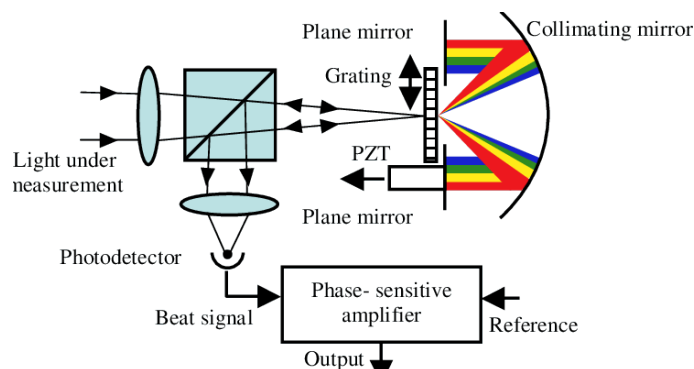


Figure 2.1: Workflow overview of a Fourier transform spectrometer [7]

The resulting spectrum is calculated based on the sum of the combined beams [6]

that reach the detector. It is called an interferogram, which will be subject to Fourier transform and result in the infrared spectrum.

The advantages of Fourier transform spectrometers [3] is that they are faster because each stroke of the moving mirror part results in a scan of the entire infrared spectrum. This allows many scans to be completed in a short period of time and compiled to obtain a better final result at the same time that a dispersive spectrometer completes one scan.

It is more reliable, considering there is only one moving part inside the FT spectrometer, the relation signal-to-noise is much lower than in a dispersive spectrometer, and because of that, better values of energy are obtained in the detector. The fact that FT spectrometers use a He-Ne laser as an internal standard is also an advantage because there is no need for external calibration as it is in a dispersive spectrometer.

## 2.2 Computer Vision

Computer vision definition is the act of giving the ability to see to a computer [8]. This means providing the computer with the necessary tools to see, such as cameras and hardware capable of interpreting the image signal.

The computer vision's objective is to mimic the human visual system and develop computational models that can be used in autonomous systems to perform tasks in which the human visual system is needed.

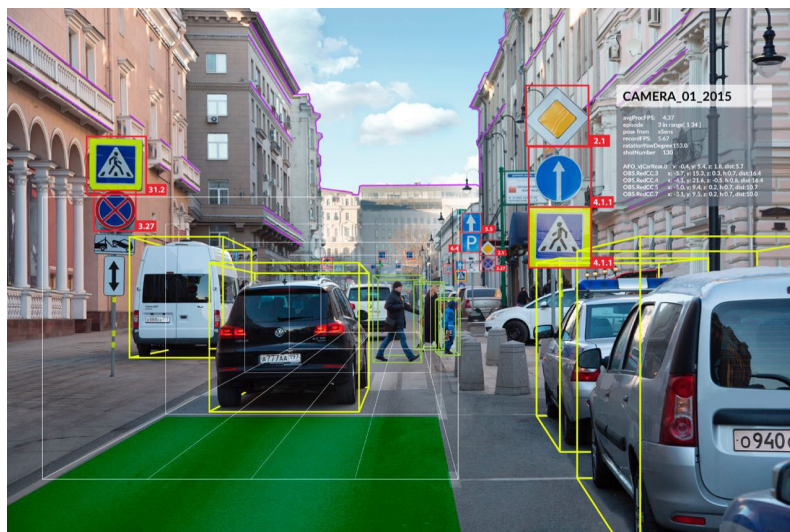


Figure 2.2: Example of computer vision application running in a camera inside a car that is analysing the road information like other cars, traffic signals and lights. [9]

The father of computer vision is Larry Roberts [10], whose Ph.D., written in 1960, discussed the different approaches of extracting 3D information from 2D drawings being this the first development related to computer vision. After this, Artificial Intelligence engineers and investigators started to use this discovery in their projects and discovered that it opens doors to numerous possibilities. In the early 80s, researchers began to

address the real world with computer vision. However, it was still necessary to investigate more in vision tasks such as segmentation and edge detection.

Object detection is an essential part of computer vision and one of its primary uses. This is due to computer vision being more efficient and adaptable than the traditional methods allowing better and faster detection and classification.

In object classification, there are two main approaches to detection: colour-based and shaped-based [11].

Colour-based is, as the name says, based on the colour information of the object. This is the most relevant feature in food products for classification, as seen in meat and fruit. For the colour-based detection to be useful, the image captured needs to be segmented in different regions, being that the level of subdivision depends on the problem that is being solved. This segmentation can be achieved in different approaches: colour threshold, colour indexing, and colour space transformation.

Colour threshold[12] consists of analysing an image pixel by pixel after setting a colour value range, by comparing pixel if its colour value is among the reference range [11] then is because it belongs to the object, becoming a required object pixel as it is usually called. This detection method is compelling and simple to use since it only consists of comparisons between pixels and reference values. However, the image that is being analysed needs to have a substantial amount of contrast between the required object and the background. Without this contrast, the reference margin will be very slim, and it probably will not be able to distinguish the required object.

Colour indexing[13] is based on the colour histograms of the images that are being analysed. A colour histogram consists of obtaining the number of times that each discrete colour appears in the image array after discretising the image colours. Colour indexing through histograms is a very robust approach to perform object detection since it is not affected by moving variations[13] such as translation and rotation on the axis and are only slightly affected by the change of the view angle. However, depending on the algorithm used, colour indexing could be very sensitive to incidental lighting on the object. The main problem of colour indexing[14] is usually the complex backgrounds of the images that require much more computational processing. In the paper's objective, this problem does not apply because the image background will be a simple one-colour background removing this variable from the equation [11].

Colour space transformation is based on the sensations that the brain gets from the analysed image. These sensations are brightness, hue, colourfulness, lightness, chroma, and saturation. There are different ways to define colour, and humans and machines define it differently.

Colour space is a method from where it is possible to specify and visualise colour. This means that different colour spaces are used depending on the objective, being the most common ones:

- RGB (Red Green Blue) - This system is straightforward to implement and, for that,



is used worldwide in almost every visual electronics equipment. It is based on trichromatic theory.

- HSL (Hue Saturation and Lightness)[15] - Similar colour spaces are HSI (Intensity) [11], HCI (chroma/colourfulness), HVC, HSD (hue saturation and darkness), and all are linear transforms from RGB, so they maintain the same technical features. The main advantages of all these colour spaces are how they specify colour that is very intuitive, simple to understand, and easy to apply in the classification part. The fact they separate luminance from chrominance is also favourable for image processing.
- CMY(K) (Cyan Magenta Yellow (Black))[16] - This colour space is mostly used in 2D printing, being that CMY is the primary colours and that Black is added to give density and improve the colour gamut.
- CIE - CIE system characterises colours by a luminance parameter Y and two colour coordinates, X and Y, that specify the chromaticity diagram's point. There are two main colour spaces CIELuv and CIELab, and to help in some visual tasks were defined two analogous spaces CIELhs and CIELhc.

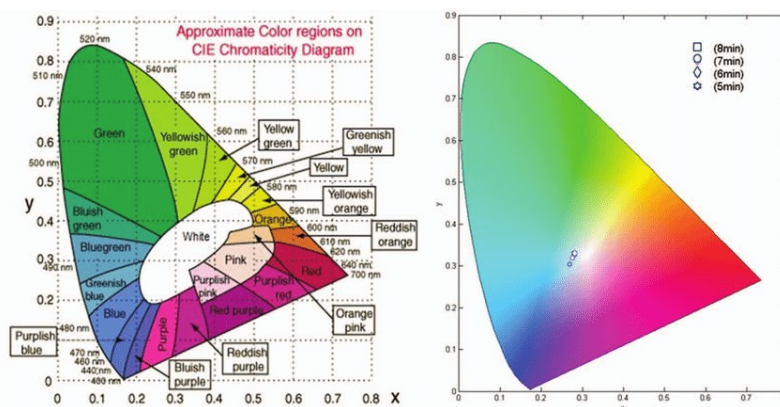


Figure 2.3: CIE colour space example diagram. [17]

The other approach for object detection is shape-based. The shape of the object is defined by the set of contours that describe his boundary. This approach is divided into two main categories: boundary-based and region-based. There many techniques to be used in shape-based detection [11], but most of them require a lot of computation power, so it is going to be focused on only the three more straightforward to implement: Fourier descriptor, Hough transform, and Moment.

Fourier descriptor[18] is used to represent 2D closed shapes, it is machine-independent and is not affected by moving variables such as rotation and translation. Fourier descriptor consists of mathematic expressions, assuming  $x[m]$  and  $y[m]$  the coordinates for the boundary pixel of the image that contains  $N$  pixels, then its obtained a complex number

$$z[m] = x[m] + jy[m] \tag{2.1}$$

and from there, obtain the DFT of  $z[m]$  that corresponds to the Fourier descriptor formula

$$Z[k] = DFT[z[m]] = \frac{1}{N} \sum_{m=0}^{N-1} z[m]e^{j2\pi mk/N} \quad (k = 0, \dots, N-1) \tag{2.2}$$

Hough transform method[19] was initially designed to detect curves where the boundary location cannot be detected, but its shape can be described as a parametric curve. Hough transform is applied point to point wherein one point the transform finds all the possibilities for continuing lines and next finds another point, repeating the process until the end of the line. This process is prolonged[19], especially in confusing backgrounds with much information. Once again, in the context of this dissertation, this does not apply because of its flat background. However, there is a probabilistic Hough transform where the algorithm, when it is travelling point to point, saves the possibilities that were previously chosen and tests the next point possibilities based on the ones that have a higher probability of being correct.

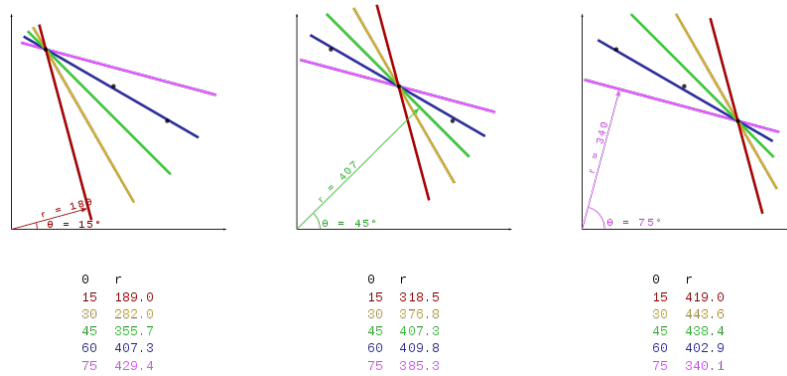


Figure 2.4: Hough Transform application. [20]

Moment vectors[21] are an essential part of 2D image processing due to its relative stability to moving actions such as translation, scale, and rotation. This method needs pre-processing of the image to pass it to grayscale and to normalise the image as a probability of a bi-variate random variable[22], being that this random variable is described by moment.

Combining all the methods referred in the paragraphs above, numerous approaches have been obtained to have a reliable, and fast object detection method. In the food industry, there are many uses for object classification but not for object detection considering that in quality control and pathogen agents detection, it is more related to the classification of a food item that is known a priori what it is, and not so much about finding what type food item it is. However, in image recognition and classification, there are many databases related to detecting what type of food is on the plate in the restaurant, what it contains, and is where it is used all the technologies mentioned above.

### 2.2.1 Image Classification

Image classification is one of the main fields of research and development of computer vision. It is widely used throughout the different areas that were specified in the last chapter. Image classification can be defined as sorting the images according to the information extracted, and grouping them in different categories according to the result of the classification. There are three main methods of classification: Template Matching, Support Vector Machine, and Neural Networks.

Template matching[23] is the most straightforward and adaptable method since it consists of finding, between a prearranged set of images, the most similar image pattern between the images and group them according to that same pattern. Again, this method's main problem is if the picture's geometry is very complex (confusing backgrounds or a picture with many different elements in it), this process can become very time-consuming. However, by adding a genetic algorithm[24] to this method, this problem can be solved.

Support vector machine[25] is based on Statistical Learning Theory and is very accurate and useful even if using a small training set of images. SVM is non-linear, and their nature is intrinsically binary classifiers, but there is some evidence in remote sensing studies where they become multi-class tasks. There are two main support vector machine strategies: one-against-one (1A1) and one-against-all (1AA).

In food recognition and image classification the method that is mostly used is Neural Networks.

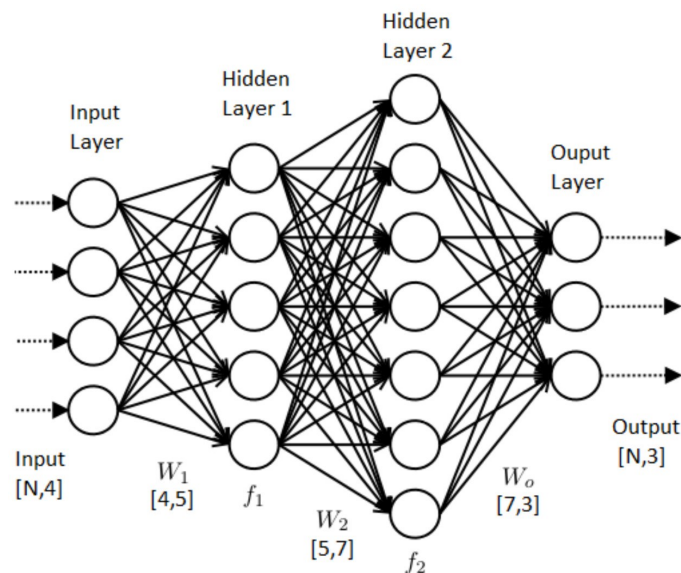


Figure 2.5: Overview of an artificial neural network. [26]

All the different types of architectures for neural networks are part of machine learning that in turn is a field of computer science that originated from the analytical study of patterns and the extensive development of the field of artificial intelligence.

Machine learning consists of creating and building algorithms that can learn from

datasets. These datasets can be pictures, numbers, or videos. If there is sufficient data about a particular subject, it is possible to create a dataset about it. The algorithms' objective is to make predictions based on the dataset that was supplied to the machine.

Machine learning is used in a variety of different situations. When the task at hand does not require human logic or common sense, it is solely based on past experiences, from where a mathematical algorithm can be developed and applied using machine learning. Also, when the data is not possible to process using human capabilities due to the amount of information present in the problem or if it is too complicated. Another example is when the data is increasing proportionally with the robot's use; then, if machine learning is used, the robot can become more efficient over time, for example, speech recognition.

Regardless of the end-use of the mathematical algorithm that is going to be used, all algorithms need training. Two types of training [27] are used:

- **Supervised Training** - In this case, the user defines both input and outputs of the sensor array obtained, so the network can process the inputs, obtain outputs, and compare them to the desired outputs. After comparing both outputs, the differences between them are placed on a scale of error and are sent back to the beginning of the process so the network can learn from their errors. The process of supervised training elapses until the margin of error is minimal. Many problems can result in the fact that the neural network never learns, which can be caused by using the wrong algorithms or wrong/insufficient input and output data.
- **Adaptive or Unsupervised Training** - In this type of training, the user defines only the input data leaving the artificial neural network to organise the data itself and find patterns between them. This method is only in its early studies, and it is not understood in the scientific world.

Deep learning is a type of machine learning. It is the form of machine learning that has seen the most advances recently especially in computer vision (e.g., image recognition and classification) and natural language processing (e.g., speech recognition).

One of the most significant and most recent achievements is the AlphaGo program. This program consists of a deep learning neural network that was trained to learn how to play the game of go, and when confronted with the world champion in 2016, was able to achieve victory.

### 2.2.1.1 Neural Networks

Artificial Neural Network[28], or most commonly known as ANN, was invented in 1943 by Warren McCulloch and Walter Pitts. The engineering approach is that this type of neural network consists of a non-linear application with numerous parameters where the ultimate goal is to associate and enter the output. The basis of an artificial neural network is an artificial neuron, which consists of an approach of the biological neuron.

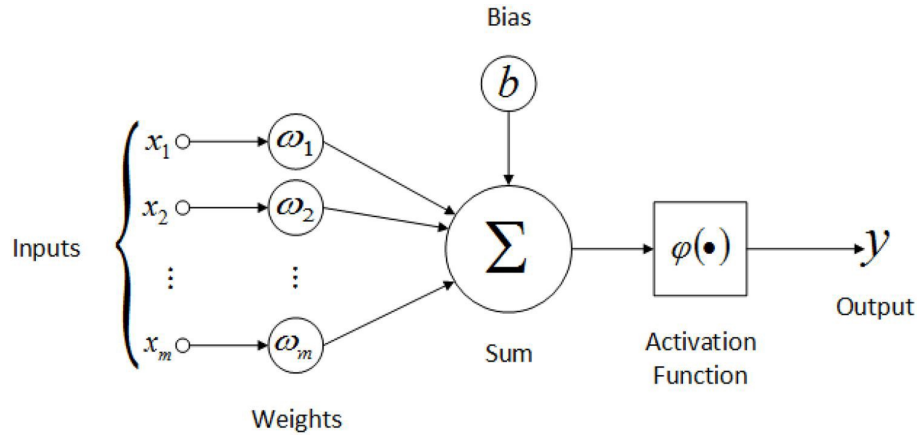


Figure 2.6: Model of an artificial neuron. [29]

In 2.6 is described an artificial neuron. An artificial neuron receives inputs,  $x[1] \dots x[m]$  that are associated to weights  $w[1] \dots w[m]$ . The strength of the signal is obtained by multiplication the weights with the inputs. The neuron is then constituted by a sum of all the inputs multiplied by their respective weights and an activation function. It is also added a bias to the sum. Bias is a constant that allows to shift the activation function.

An activation function can be linear or non-linear. It depends on the desired output seeing that the activation function controls the output of a neural network. There are many different types of activation functions:

1. Sigmoid Function - This function is non-linear and is mostly used in feedforward neural networks. Feedforward means that the connections between the nodes of the neural networks do not form a cycle. The following equation represents the sigmoid function:

$$f(x) = \frac{1}{1 + \exp^{-x}} \quad (2.3)$$

The main advantage of using the sigmoid activation function is because it is simple to understand and can achieve good results in many different areas such as binary classification problems and modeling logistic regressions.

There are three main variations of the sigmoid function: Hard Sigmoid Function, Sigmoid-Weighted Linear Units, and Derivative of Sigmoid-Weighted Linear Units.

2. Hyperbolic Tangent Function - This activation function is more commonly known as Tanh, and is a zero-centred function which makes the training process simpler in multi-layer neural networks. This function ranges from -1 to 1.

$$f(x) = \frac{e^x - e^{-x}}{e^x + \exp^{-x}} \quad (2.4)$$

The most common variation of the Tanh is the Hard Hyperbolic Function or Hard-tanh.

3. Softmax Function - Used mostly in multi-class models since it returns probabilities of each class. The output of this function is normalised between 0 and 1, with the sum of all probabilities equal to 1.

$$f(x) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (2.5)$$

4. Softsign - This is another example of a non-linear activation function that produces an output between  $[-1, 1]$  like the tanh function. The difference between them is that softsign converges in polynomial form and tanh converges exponentially.

However, this means that softsign function is smoother than tanh resulting in a faster and better learning model.

$$f(x) = \frac{x}{|x| + 1} \quad (2.6)$$

5. Rectified Linear Unit Function - As well known as ReLu, this activation function is one of the most used in deep learning applications. It has proven to give the best performance in training and the best results in deep learning models compared to sigmoid or tanh.

This function is linear for all positive values and zeroes for all negative values. This means that it converges faster and that it does not saturate when the data input gets large.

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & x_i < 0. \end{cases} \quad (2.7)$$

Like tanh, ReLu also has many variations, such as Leaky ReLu, Parametric Rectified Linear Units (PReLU), Randomized Leaky ReLu (RReLU), and S-shaped ReLU (SReLU).

6. Softplus Function - The main difference of this function is that it produces outputs that range from  $(0, +\infty)$ . This means that this function grows exponentially and is represented by the primitive of the sigmoid function.

$$f(x) = \log(1 + \exp^x) \quad (2.8)$$

7. Exponential Linear Units - This type of activation function is known because it has negative values, which reduces the training time by reducing the computational complexity.

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(\exp(x) - 1), & x \leq 0. \end{cases} \quad (2.9)$$

Some known variations of ELUs are: Parametric Exponential Linear Unit, and Scaled Exponential Linear Units.

8. Maxout Function - This function is the generalisation of the ReLu and the leaky ReLu functions. The artificial neuron present in the function has the same properties as the ones present in the ReLu and the leaky ReLu. This means that there are no dying neurons or saturation.

$$f(x) = \max(w_1^T x + b_1, w_2^T x + b_2) \quad (2.10)$$

However, this function's training time is much more extensive as ReLu or leaky ReLu since it has double the parameters.

9. Swish Function - Google Brain Team proposed this function as one of the first compound functions. This function is composed of the combination of the sigmoid function and the input function. This compound produces better optimisation when training deeper models.

$$f(x) = x \cdot \text{sigmoid}(x) \quad (2.11)$$

10. Exponential linear Squashing - This function is another example of a compound function by combining ELU and Sigmoid functions.

$$f(x) = \begin{cases} \frac{1}{1+\exp^{-x}}, & \text{if } x \geq 0 \\ \frac{e^x - 1}{1+\exp^{-x}}, & \text{if } x < 0. \end{cases} \quad (2.12)$$

### 2.2.1.2 Convolutional Neural Networks

Among deep neural networks, convolutional neural networks or CNN have demonstrated to achieve the best results, especially in image classification, pattern recognition, and machine learning problems. A CNN is analogous to an ANN meaning that both of them are composed of artificial neurons that can be optimised through learning.

The input of a CNN usually consists of an order three tensor, which in the case of an image recognition algorithm is an image with H rows, W columns, and three channels (R, G, B colour channels), this translates in width, height, and depth of the picture. If needed, a CNN can take a higher-order tensor; however, the time of both training and running the algorithm will be much higher than with an order three tensor.

Like an ANN, a CNN is divided into layers or processing units in which the input goes through one another. With each layer, the CNN will modify the input tensor according to the parameters presented in each layer.

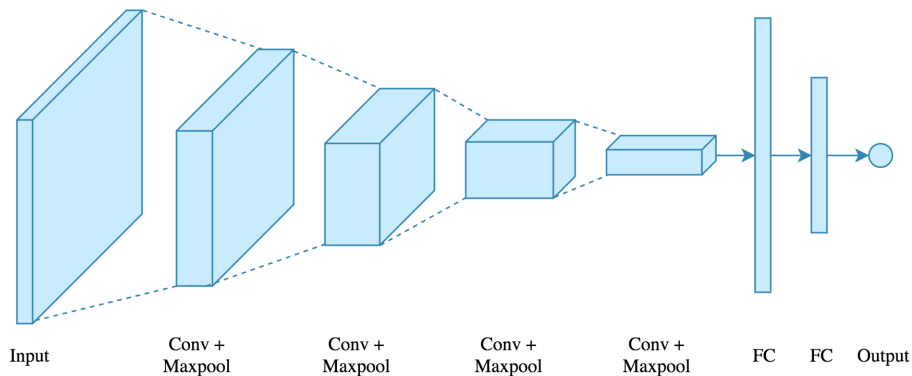


Figure 2.7: Architecture of a CNN. [30]

In the next paragraphs, it is going to be described the details of the different types of layers that are used in a CNN.

- Convolutional Layer - This layer always comes up first in a CNN. This layer's name is a derivation of the term convolution, which means the combination of two functions to originate the third function.

The two functions that are going to be combined are the input of the layer, and a filter that runs through the input. At each position, it performs the mathematical calculations originating a feature map, the third function. The filter that is used depends on the weights and parameters that are given to that convolutional layer.

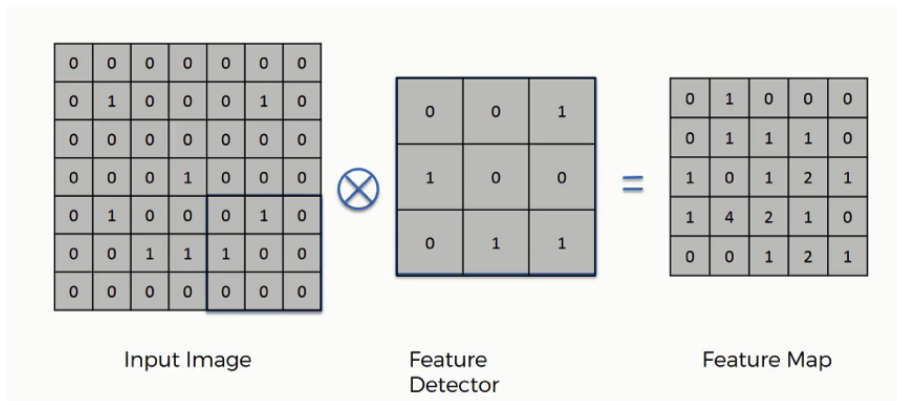


Figure 2.8: Demonstration of a convolution in a CNN. [31]

- Activation Layer - In this layer is where it is applied the different activation functions that were mentioned in this chapter. Given an input or set of inputs, this layer, using activation functions, shapes the output according to the parameters and the activation function used.
- Pooling Layer - This layer is generally used after a convolutional layer. It aims to summarise the presence of features in the feature map by removing the fine detail that is not so important to the model that is being trained.



This means that after applying this layer to the input, it would create a lower resolution version that focuses on the input's main structural elements. In most cases, the pooling layer is applied to a 2x2 matrix of the feature map. There are two main types of pooling layers:

Average Pooling - This type of pooling layer calculates the average value of the 2x2 matrix to where it was applied.

Max Pooling - In this case, the matrix 2x2 takes the feature with the most presence in the matrix, returning that same value.

- Fully Connected Layer - Also known as the Flatten layer. It is responsible for preparing the input of the next stage of the CNN by turning the previous layer's output into a single vector. The last fully-connected layer of the CNN gives each label's final probabilities; this means that this layer is responsible for classifying the output based on the feature extraction performed previously.

In conclusion, the significant difference between an ANN and a CNN is that in the last one, only the last layer of the network is fully connected versus the ANN, where every input is connected to every output.



## STATE OF THE ART

Due to the increasing concern for healthcare in the last decade, combining with the rapid advance of technology, many ideas and companies started combining the most recent technology with already existent laboratory types of equipment to help the general population increase their healthcare. Gadgets like smartwatches or smart bands, that control your daily physical activity, started being commonly used. The modern kitchen has also begun to be equipped with new devices like food processors, food dehydrators, smart scales, food apps, and other gadgets to eat more healthily and efficiently. This chapter aims to introduce some of these gadgets and devices as well as the scientific methods, and instruments in which some of them were based.

### 3.1 Near-Infrared Spectrometers

A near-infrared spectrometer is an analytical instrument used to realise near-infrared spectroscopy from where it can obtain the chemical composition of ingredients, intermediates, and final products.

A near-infrared spectrometer is constituted by a source, a detector, and a dispersive element that allows the recording of different wavelengths intensities. The spectrum can be measured in either reflection or transmission. For the source, it is typically used quartz halogen lights or LEDs (Light-emitting diodes). A wavelength-scanned laser is also a powerful source for high precision spectroscopy, and its what is used in the spectrometer that is going to be analysed.

There are many types of detectors:

- Charge-coupled device - is most commonly named CCD and consists of an integrated circuit that is light-sensitive. The objective of this integrated circuit is to capture, store, and display image information in the form of localized electric charge

that varies according to different light intensity. This type of detector is more effective in the shorter end of the NIR range.

- Indium gallium arsenide - InGaAs[32] is an alloy of gallium arsenide and indium arsenide, mainly used as a photodetector for optical fiber telecommunications due to its high-speed and high-sensitivity.
- Lead sulphide - PbS is an inorganic compound, semiconducting material and the main compound of lead. It is used as a photodetector for near-infrared sensors because it is sensitive to radiation at wavelengths between  $1\mu\text{m}$  and  $2.5\mu\text{m}$ .

Considering the information above, it is possible to conclude that the detector used in the near-infrared spectrometer mainly depends on the range of wavelengths to be measured.

The most common dispersive elements used in NIR spectrometers are prisms and diffraction gratings [33]. A prism is a transparent element bounded in part by two non-parallel plane polished surfaces that refract light. A prism works by having an angle in two of the nonparallel planes, causing the speed, and the angle of the light to be different from one medium to another, generating refraction. A phenomenon known as dispersion, states that the phase velocity of a wave depends on its frequency, which causes the light of different colours to be refracted differently and leave the prism at different angles.

A diffraction grating is also called a "super prism" , and its constituted by many closely spaced, and parallel slits that when a beam of light strikes the object, it diffracts into several beams travelling into different directions.

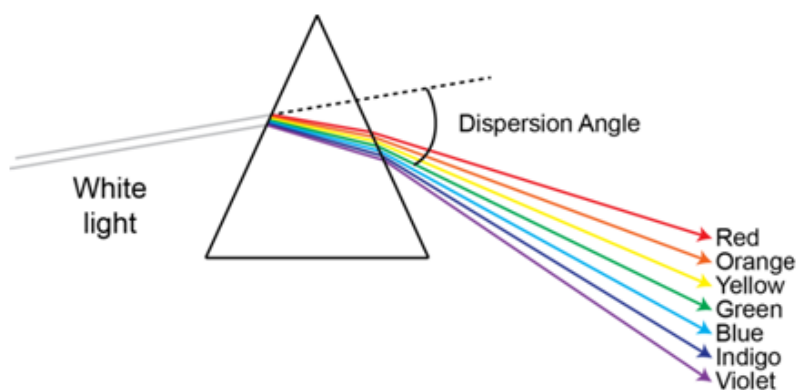


Figure 3.1: Light dispersion of a prism. [34]

Due to the increasing concern for well-being, several technologies have begun to facilitate various day-to-day challenges. This is the case for near-infrared technology, which is being used to help in many aspects like diet, allergies, and healthy eating. Many companies were created with innovative ideas like Scio.

Scio is a pocket-size near-infrared spectrometer used to analyse food and return information such as protein, fat, and sugar content. It was designed by Consumer Physics, a

San Francisco and Israel-based company that started a Kickstarter campaign which raised over 2.7 million dollars. After many delays with the devices' delivery dates, the first ones began being shipped around late 2016.

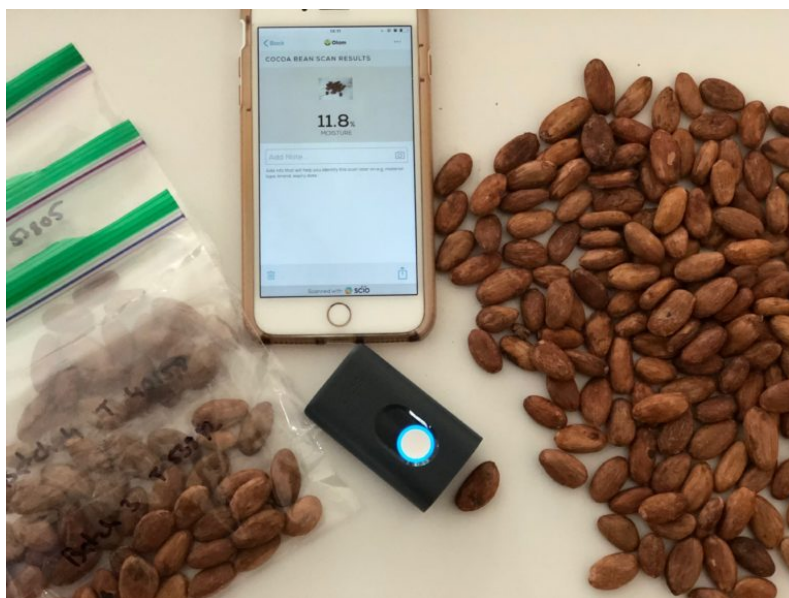


Figure 3.2: Scio device. [35]

The gadget itself uses standard infrared spectroscopy, which means it measures molecule's interaction with infrared light. It is classified in three ways: light absorption, emission, and reflection. Scio operates in the near-infrared wavelength that goes from  $0.7\text{nm}$  to  $1\text{mm}$  at the visible spectrum's nominal red edge.

Using a food database, it compares the atoms' vibration from the scan of the device, the food spectral fingerprint, with the food in the database and returns information to the mobile phone app. Scio can determine the sweetness of fruit (Brix measurement), the amount of cocoa content in a bar of chocolate, alcohol in drinks, protein, fat, and calories in the different types of food. Every time someone scans any kind of food, that increases the database leading to better results in the future.

Tellspec is another company that emerged in the food analysis market. It is a portable spectrometer scanner that, along with a custom algorithm, can determine the allergens, chemicals, nutrients, calories, and ingredients in the food by analysing the spectrum obtained through the amount of the photons that are reflected. Tellspec has not released any product to the public, so there is not much proves that it works besides the company word, but they have released many prototypes that did not match with what the company announced on the Indiegogo campaign. TellSpec works in Raman Spectroscopy.



Figure 3.3: Tellspec device. [36]

## 3.2 Electronic noses and tongues

This section aims to analyse other solutions that can obtain nutritional information from different types of food. Hereupon this part will focus on electronic noses and tongues. Electronic noses and tongues are solutions that are already implemented in the market and are used for many different purposes[37] such as medicine[38], environmental monitoring[39] [40], bomb detection, and food analysis[41] being the biggest market of all, the food industry.



Figure 3.4: Electronic nose device, Cyranose 320. [42]

The objective of an electronic nose[41] is to mimic the olfactory system of a human but obtaining the chemical results from the tested object.

Electronic noses are composed of two main components, sensing system and pattern recognition system. The sensing system that is usually used is a sensor array. The sensor

array consists of a group of sensors, usually chemical sensors, where each chemical sensor measures a different component of the object that is being tested. In the end, combining all the other chemical fingerprints is obtained the general chemical fingerprint of the item. The sensing system can also be a single device like a spectrometer that returns an array of the object's different chemical aspects. In more recent electronic noses, both sensing systems are used to produce more reliable results.

Electronic noses can use different types of sensor[40] for the sensing system according to the different types of gas detection principle that are being used. The three most used are:

- Metal-oxide-semiconductor - is usually called MOS[41], which consists of a layered semiconductor. Starting with a layer of the silicon substrate, above this first layer is a layer of silicon dioxide (SiO<sub>2</sub>), and finishing with a layer of polycrystalline silicon.
- Conducting Polymer [43]- Polymer or plastic is most commonly known as a good insulator. However, in 1977 Hideki Shirakawa, Alan Heeger, and Alan MacDiarmid discovered the first conducting polymer after noticing that halogen vapour produces a very high conductive polyacetylene film. This started numerous investigators and companies to invest in this area, and from that moment, many more conducting polymers were discovered.
- Piezoelectric Crystal Sensors[44]- The properties and functionalities of piezoelectric material were found by two French scientists in the early 1800s, Pierre and Jacques Curie. Piezoelectric crystals work by a change in pressure, creating an artificial charge in the crystal material. The most commonly used are barium titanate, and zirconate titanate. Mechanical sensors are typically used to determine pressure.

For the sensing system, it is imperative that is reliable and achieves good results because those same results are used to calibrate the other component of electronic noses that is the pattern recognition system. This means that with better results, the better will be the final pattern recognition system.

There are different types of pattern recognition systems such as principal component analysis (PCA), functional discriminant analysis (FDA), partial least squares (PLS), cluster analysis, fuzzy logic, or Artificial Neural Networks, and this last one is the most used nowadays, and is showing the best results. In the case of electronic noses, the artificial neural network[27] is trained to recognise the different sensor arrays obtained by the sensing system, namely the chemical identifiers of a specific object.

Electronic noses'[27] main advantage compared to traditional chemistry equipment is that electronic noses produce faster results. Since it is a real-time analysis, it is less time consuming; they are also cheaper and can have a qualitative output. This type of output is important in the environmental classification of pollution in the air or in grading a specific food or ingredient.

Electronic tongues were created to mimic the human body's taste system, and their constitution and functions are relatively similar to an electronic nose. They are also constituted by a sensing system and a pattern recognition system.

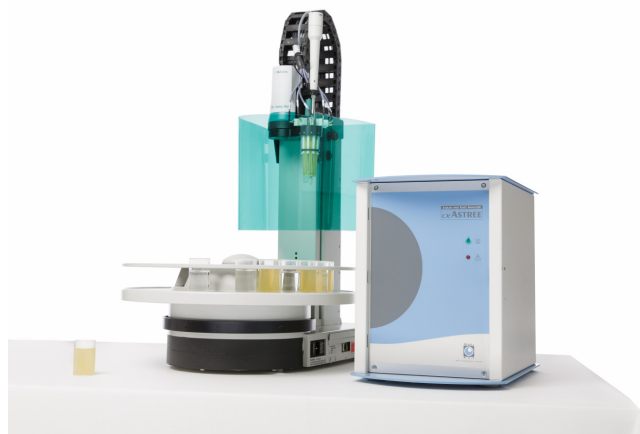


Figure 3.5: Electronic tongue device, ASTREE.[45]

The ideal sensing system of an electronic tongue[46] is generally composed of both selective and cross-sensitive sensors such as electrochemical sensors (potentiometric and voltammetric) and optical sensors that measure absorption, reflection, and luminescence. Combining the two different types of sensors will result in obtaining a better fingerprint of a given substance, which will lead to a better pattern recognition system with better classification and identification of many different objects.

The pattern recognition system[47] that is typically used is Principal Component Analysis (PCA). It is a mathematical technique that reduces many possibly correlated variables into a smaller number of variables called principal components. PCA applies a vector space transform using mathematical projection. The original data set that is composed of all the different chemical identifiers obtained by the sensing system is reduced to a smaller group of chemical identifiers to approximate the results of an electronic tongue to the real taste system that detects five significant components in every food or object, sourness, saltiness, umami, bitterness, and sweetness.

### 3.3 Computer Vision Applications

Computer vision can be applied in very different areas being the most common ones:

1. Quality control[48][49] - Visual inspection is an extensive and expensive process if it is human-made. It requires much testing in different machines and a great loss of time in that same tests. It is also less reliable than computer vision testing. Computer vision is used in the entire supply chain, from production, to storage, and to the consumer in the most varied products from meat[50] to dairy. In this case, the computer vision detection system is mostly based on colour to assess the product's



quality[51]; for example, in pork, it is the lightness of the meat that determines its quality[50].

2. Retail and Retail Security - Computer vision started being used by many retail store companies to improve customer experience. Amazon Go project[52] is a retail store where there is not any cashiers or self-service cashier stand, you need to install an app, go inside the store, grab the items you like to purchase, and walk out. Then Amazon debits your credit card for the things you bought, and you receive a receipt in your email. This is possible thanks to computer vision and artificial intelligence, where the cameras track every customer, and save every item you pick up from the shelf, meaning that when you exit the store, you are charged correctly. This is possible by using facial recognition, and tracking. It is also used in retail to reduce the amount of robbery, and errors made by the cashiers by detecting fake barcode scans or missed items in the shopping cart.
3. Automotive - Besides being used in the production line by the robots that assemble the automobile to confirm the positions of the different parts that come in the car, the new and main application which is being targeted of many studies and researches, are the self-driving vehicles that are being developed by Google, Tesla or even Uber. In this aspect, computer vision is the most significant component of development because the car computer and AI needs to make decisions based on what it "sees".



Figure 3.6: Example of computer vision image processing. [53]

### 3.3.1 Image Recognition and Food Classification Applications

Image classification and food recognition[54] are related directly. In the last few years, with the increasing concern around diseases like obesity and diabetes, many applications are related to diet and to eat healthily has surged in the tech business.

Most of them are related to the need for manual input of fat, sugar, and general nutritional values of the food that is consumed by the user daily. Image classification with food recognition began to be introduced in these same apps to help the process be faster, more comfortable, and effective. This means there is no need to insert the values yourself but instead take a picture of the food you are eating to obtain all the information.

The most common application for food image classifiers is by incorporating them into a mobile application. There are already a few image recognitions apps available in the app stores like Calorie Mama, Foodai, and Foodtracker.

Calorie Mama is a mobile app that is available for iOS and Android and can recognise plates of food by taking a picture of the food using the camera in your smartphone, and when you run that picture in the app, it returns the type of food as the example in the image below. The app is paid monthly and has different plans with a minimum of 100\$ and a maximum of 2.000\$. The Calorie Mama app is a machine learning API that does not recognise every dish but has trained models capable of differentiating between many preparation methods and presentation styles.

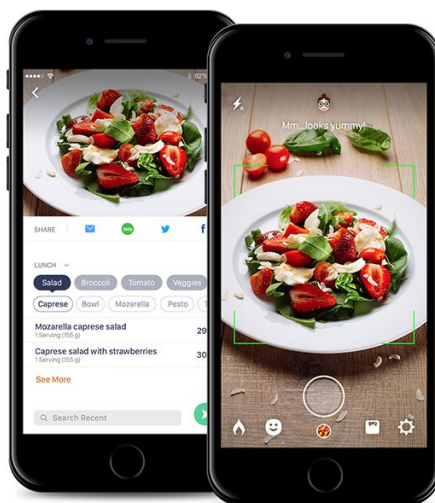


Figure 3.7: Calorie mama mobile app. [55]

Foodai is a web-based food recognition API trained with over 400,000 food images from 756 different Singapore local cuisine classes. It is possible to test the algorithm on the company website. According to the development team, the objective of creating this food API was to help introduce a healthy lifestyle to Singapore residents by allowing them to know what they are eating on a day-to-day basis.

Foodtracker is a real-time food classifier mobile app. It was built using a combination between a deep convolutional neural network and YOLO. YOLO is a real-time object detection system. Using this mobile app, there is no requirement to introduce any data or take any picture manually. It is as simple as opening the app and point the mobile phone camera to the food plate, and the app shows the results in real-time and gives access to a page where you can find the nutritional information of the food results. This nutritional

### 3.3. COMPUTER VISION APPLICATIONS

---

information is obtained by using a food information database, so it is not specific for the food that is on the plate.



## NUTRITIONAL VALUE EXTRACTOR PROPOSED MODEL

This dissertation's proposed model is designed to create new functionality that can help today's society knowing what they are eating, and assisting them in becoming more healthy by eating well. Besides being a new functionality or gadget, depending on its application, this model's purpose is to be as complete as possible and gather as many functionalities as possible. In this chapter it is going to be stated the necessary approaches for the different components in this dissertation in order to achieve the objective of creating a nutritional value extractor module.

### 4.1 General Overview

This dissertation's objective is to classify an RGB image to be able to extract the nutritional value of a particular food item. That said, this module's purpose is to have a way of capturing a RGB image, a place to run it, and obtain its rank. It is also necessary to have a way to get the nutritional value of that particular food item.

One of this model's main objective is to be user-friendly. This means that besides the module being reliable and accurate in the information that it gathers, it also needs to present the nutritional information in a organised and easy to decipher, so that the general population can interpret the data. Several food platforms were analysed to achieve this objective, and conclusions were drawn on how these platforms present information to the user. These conclusions were then applied to our module.

This model is composed of two main elements. These two main elements communicate in different message types, so the controller must understand both elements to achieve the desired result. However, for the process to be completed successfully, the

information must arrive in sequential order because the whole process is sequential, and each stage needs the data from the previous one.

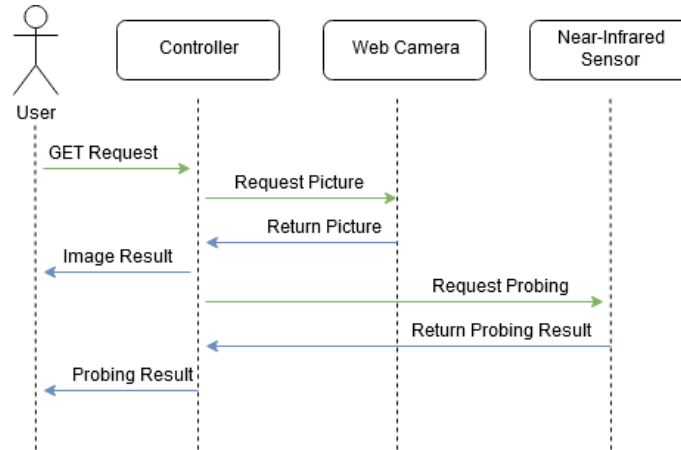


Figure 4.1: Messages schematics - This diagram represents the communication sequence that happens after the user presses the button. The green arrows represent the requests made, and the blue arrows represent the requests' results. After each step, the information obtained is processed in the controller and shown to the user.

## 4.2 Module

The module will be composed of two data-collecting devices. One is responsible for collecting an RGB image that is going to be processed by an image recognition algorithm, while the other device is responsible for collecting nutritional information. Each device is activated in a different step of the process.

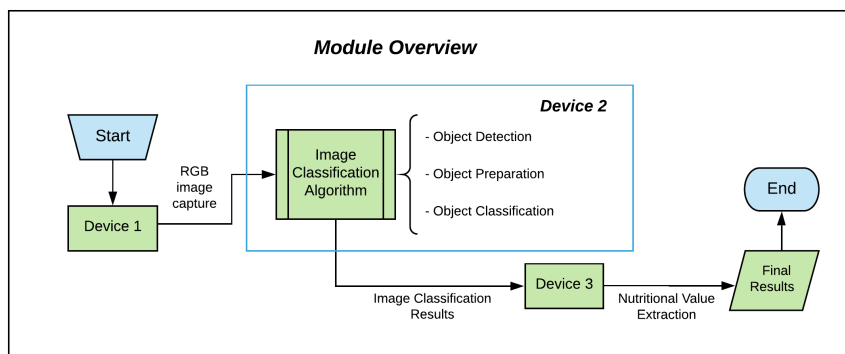


Figure 4.2: Architecture of the module. The blue colour represents the beginning and the end of the process. The green colour represents the devices used and processes made by these devices. Device 1 is responsible for obtaining the RGB image necessary for the process run in Device 2. The image classification process results in Device 2 are then sent to Device 3, in which the nutritional value is extracted. After every process is complete, the information is sent to the controller to be stored and presented to the user in the lcd.

### 4.2.1 Devices

This module will need three different types of device to work as intended since the aim is to collect different kinds of information.

- Camera - The purpose of this component is to obtain an RGB image. The image obtained must have quality enough to then be processed by the image recognition and classification algorithm.
- Near Infrared Sensor - This component is responsible for obtaining the nutritional information.
- Processing Unit - This device is responsible for receiving the information from both sensors. It then processes it in order to obtain the results that will be presented to the user.

Based on the device list above, it is possible to divide this module into three main systems and topics of interest. That said, these topics are Vision, Near-Infrared Sensing, and Controller.

## 4.3 Vision

This sub-system is the first stage of the process associated with the model. It consists mainly of image classification algorithms. The main goal is to detect what type of food is being analysed by the module. This part of the process is crucial since the near-infrared sensor cannot detect what kind of food is under his probe.

In this sub-system, there are two main parts:

- Object Detection - Detects the object that is present in the image subtracting the background.
- Object Identifier - After subtracting this object, it is necessary to identify that same object and classify it.

An RGB image is required to run the image classification algorithm, and that is where the camera comes into place.

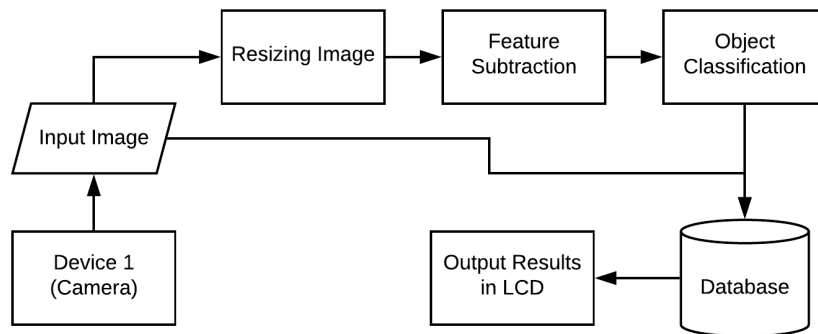


Figure 4.3: Vision process explained in more detail. Device 1 takes a picture that is stored in the database. The database will store every photo taken since it can be used to improve the image recognition algorithm. Each step of the image recognition algorithm is run in the controller. The first step consists of resizing the image to the standard in which the image recognition was trained. The next procedure consists of extracting the image features; this means running different layers with different activation functions that will subtract the necessary features. Having the features, the next step is to classify the object based on those exact features. The results of this classification are also stored in the database to be easily accessed by the user. In the end, the results are presented to the user on the LCD.

#### 4.4 Near-Infrared Sensing

The second stage of the process is associated with the model. In this sub-system, the objective is to obtain the nutritional information of the food. To achieve this goal, it is required to use a device capable of interpreting the food components that are being analysed, and in 3.1 were presented several devices capable of achieving this goal.

The nutritional information of the food consists of the following data:

- Calories[56]- represents the amount of energy that is present in the food. It is also a unit of measurement.
- Fat[57] - there are many different fat types, some worse than others for human health. There are saturated fats present in animal-based products, and when ingested in large quantities, it can increase the risk of heart attack and blood-related problems. Unsaturated fats are the healthier type of fat and can help decrease the dangers of saturated fats' diseases. And in last, there are trans fats, which initially are unsaturated but are then processed and, in the end, behave like saturated fats.
- Protein[58] - This is a macronutrient essential for the proper functioning of the human body. Proteins are composed of hundreds/thousands of amino acids. Some examples of protein functions are antibody, enzyme, structural component, and messenger.



- Carbohydrates[59] - It is also a macronutrient and consists of a molecule that has carbon, hydrogen, and oxygen atoms.
- Water[60] - is a nutrient that is present in all the main food groups such as grains, meats, dairy products, fruits, and vegetables.

For the objective of this module, the chosen device was a near-infrared spectrometer.

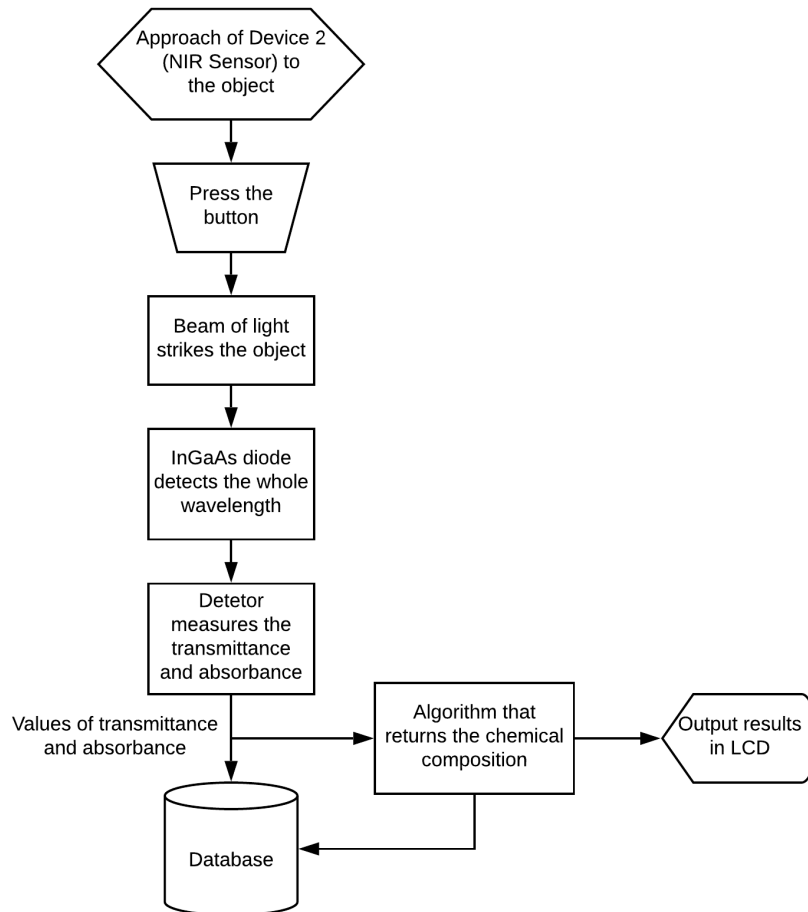


Figure 4.4: Near-Infrared Sensing, the process explained in detail. Device 2 needs to be manually pressed against the object to achieve better results. After that, the device starts its process that consists of striking the object with a light beam and observing how the molecules react. At the end of this process, the information about the molecules, namely its transmittance and absorbance, is stored in the device and processed in an algorithm that returns the object's chemical composition. The final results are then presented to the user.

## 4.5 Controller

In this model, to achieve the objective, it is crucial to have a unit responsible for processing the different input data types, organising them, and presenting the resulting information to the user in an lcd.

This unit needs to have the processing power capable of running an image recognition algorithm, be versatile enough to interpret different types of messages and input data, and display the final results and data to the user in an understandable way.

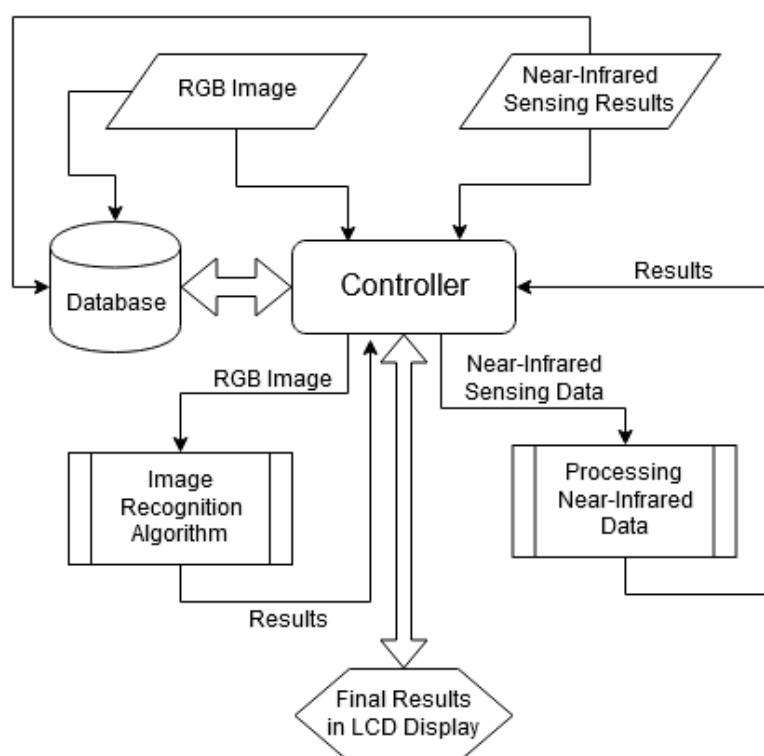


Figure 4.5: Controller process architecture. The controller is used throughout the different stages of the process. At the beginning of the process, the controller is responsible for running the image recognition algorithm process represented in 4.3. After this first process, the controller needs to send the information to the user and the near-infrared sensor for the next step. In the next step, the controller will not have part of the near-infrared processing since the sensor has its own processor but will communicate with the sensor to obtain the nutritional information results and present them to the user. The controller is also present in the database where the results from each step will be stored as well as all the pictures taken.

## EXPERIMENTAL IMPLEMENTATION AND SETUP

This chapter is the follow up of the concepts introduced in the last chapter. During this chapter it is going to be described the different methods using the respective components of this dissertation, Vision, Near-Infrared Sensing, and Controller. It is also going to be analysed how the author designed the final module to incorporate all these elements into one.

### 5.1 Implementation

This section will present a broad overview of the many different techniques of creating an image recognition algorithm, which was proposed in the model section 4.3. This section aims to describe precisely the technologies used in this dissertation food image classification algorithm as well as the reasons behind this choice.

#### 5.1.1 Food Image Classification

To create an independent module, which is this main dissertation objective, some key aspects need to be fulfilled:

- Versatility - The algorithm needs to be versatile enough to achieve the best results possible regardless of the module's position is or the food analysed.
- Reuse Code - This is an essential key aspect because the tools used to create the artificial neural networks in this dissertation are commonly used such as Keras, and Tensorflow. It is important that the code used can be easily adapted to the requirements in question.
- Image Dataset - Since the objective is to detect as many different types of food as possible, it is mandatory to have a large food image dataset so that the algorithm

training be the best possible and consequently reaching the best output results possible.

- Parsimonious - Since the module is independent, the processing power available is very low, so the algorithm is as light as possible, so the processing unit can run it and still achieves the necessary output results.
- Developer Friendly - Using techniques and technologies that are already well implemented in the developers' world is a point in favour because when questions and errors arise, there is a community behind to help.

Taking into account the different key aspects presented above, the following tools were used:

- Python[61] - This is a very important tool since this programming language is very developer-friendly, dynamic semantics, and most important is object-oriented, so it is ideal for training and testing the image recognition algorithm.
- Tensorflow[62] - This tool consists of an end-to-end open-source machine learning platform. It is used to train the different image recognition algorithms developed in this dissertation.
- Keras[63] - This is a high-level API integrated into Tensorflow 2.0 and was created to be user-friendly, modular, and work with Python.
- Google Colab[64] - Google Colaboratory, or Google Colab, allows you to use resources of Google without the need to configure drivers or OS. It also provides access to free GPU's to use as you wish. Google Colab was used to train the image recognition algorithm.
- ImageAI[65] - This is a open-source library that allows the user to train image recognition algorithms. Is possible to train using a custom dataset, and four different types of CNNs are available: SqueezeNet, ResNet50, InceptionV3 and DenseNet.

This section can be divided into two different subsections: Training and Detection.

#### 5.1.1.1 Training

This section describes how CNN is trained and how the dataset is created and prepared. The dataset used in training was created used a desktop computer including the labelling. The image recognition algorithm training was made using Google Colaboratory[64]. For this dissertation's objective and to make a hypothesis that could be proven and possible to analyse the results, it was decided that the dataset will only include fruits that are used in modern cuisine since they are easy to obtain and cheap.

The fruits chosen were: apples, pears, lemons, and tomatoes. There were two options available regarding the dataset: using an existing dataset or creating the dataset from scratch.

There are a few available free datasets to be used online, like Food-101[66], Fruits 360[67], and Fruit Recognition dataset[68]. There are advantages and disadvantages of using already existing datasets. The main advantage is that usually, these datasets are optimised for machine learning and image recognition algorithms. They have also significant data collections that require a substantial amount of time and manpower to be created and labelled.

However, since the objective is to train an image recognition algorithm for a specific set of fruits, the existing datasets can develop problems because they were not made in the conditions where this CNN will operate, leading to worst accuracy results. The conditions where this image recognition algorithm will operate are more controlled because it is inserted in a module, so there is no background changes removing the necessity of using a existing dataset in training.

Due to this dissertation's objective, the best way to obtain satisfactory results is to create our dataset. For that, it will be needed a camera and a labelling program to label all the collected images. For the camera, the chosen option was a web camera that is able to take FullHD photos, and for the labelling program was used labelImg[69]. This was the program chosen because it is easy to use and was recommended by most forums and websites that work with machine learning.

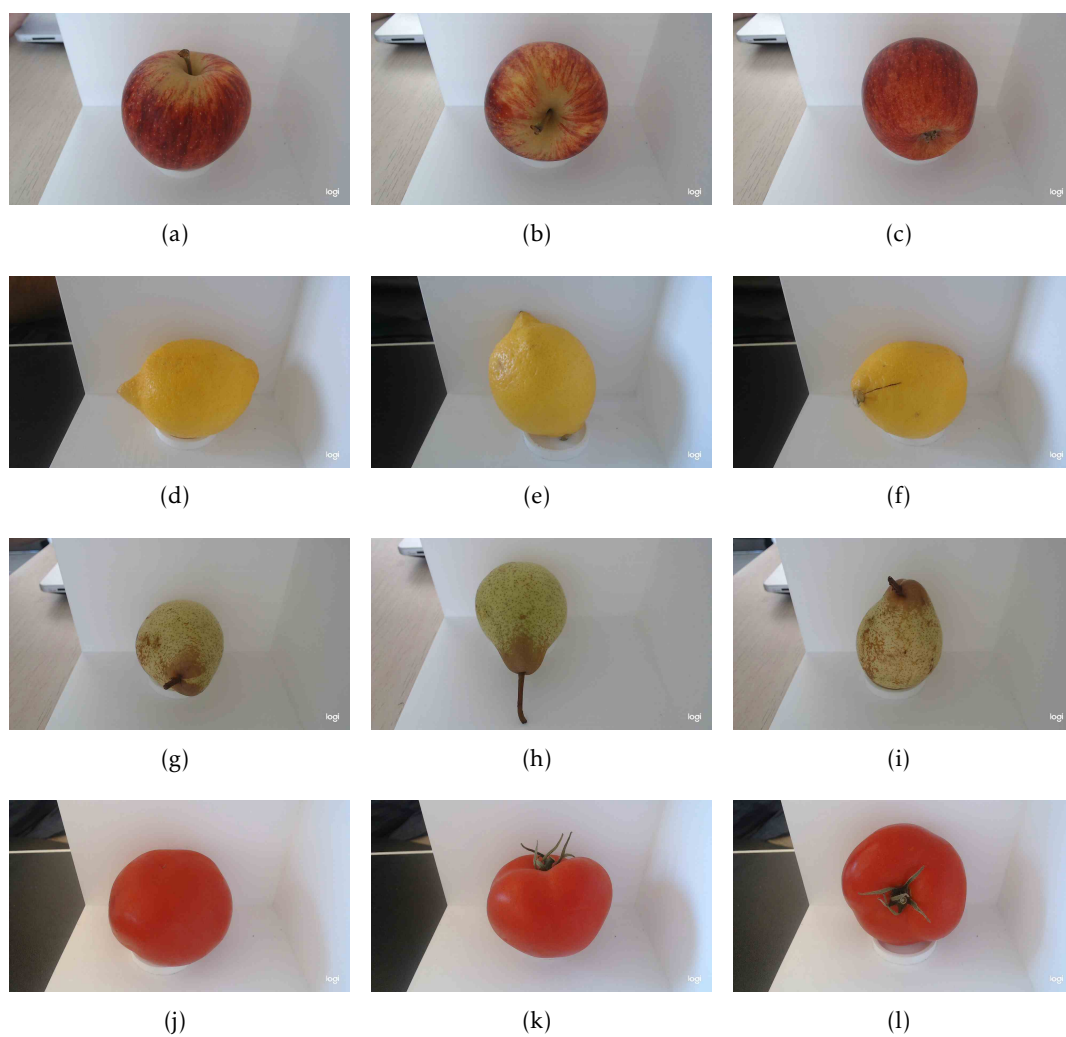


Figure 5.1: Example of images from the dataset created for this dissertation. The example pictures are separated in different rows by the different types of fruit, apples, lemons, pears, and tomatoes, by this order.

## 5.1. IMPLEMENTATION

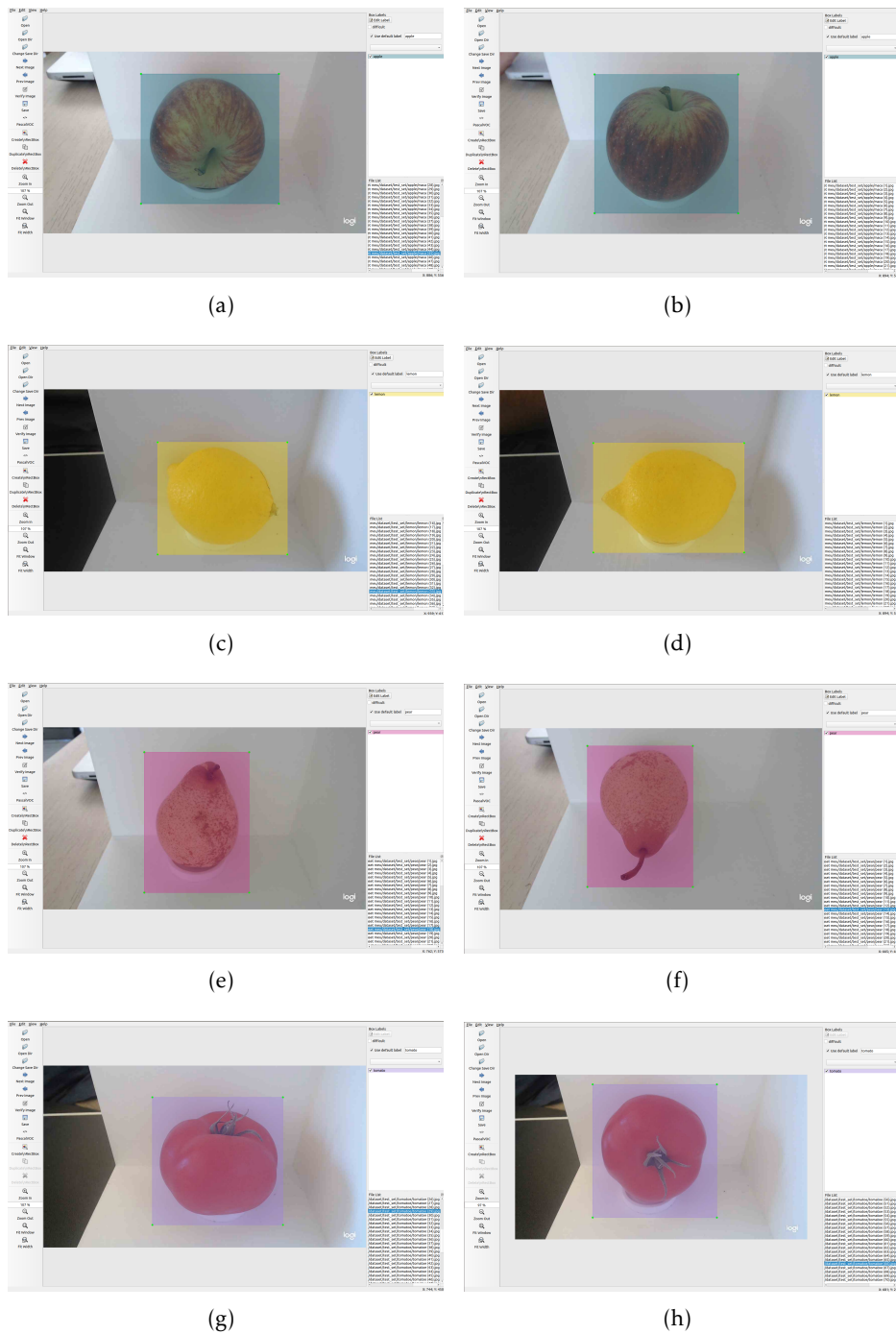


Figure 5.2: Label program and examples of the labelling process for different fruits. The different fruits are separated in different rows, apple, lemon, pear and tomato by this order. The labelling process consists in only selecting the important parts or features of each picture as shown.

filename	width	height	class	xmin	ymin	xmax	ymax
pear (35).jpg	1280	720	pear	405	67	791	593
lemon (42).jpg	1280	720	lemon	502	91	924	714
tomato (24).jpg	1280	720	tomato	443	113	1033	651
apple (50).jpg	1280	720	apple	407	65	1003	615

Table 5.1: Table containing some examples of the label's output inserted in a .csv file, and it is where the model is going to get the locations of the features and the corresponding labels.

After labelling all the images and using this dataset to train the image recognition algorithm, it was necessary to divide the same dataset into train and validation. Since the training part of the algorithm is the one who needs the biggest percentage of the dataset, the proportion defined was 65% for the training subset and 35% for validation.

The next step was to decide which type of neural network would be implemented in the image recognition algorithm. For this, it was agreed within this dissertation's theme that the best approach was to compare the performance between two existing model's, ResNet50 and InceptionV3. These CNN, already have a defined structure and the objective is to train them with our own dataset and see between both CNN's which is going to have better performance.

The performance is going to be analysed using two criteria, accuracy and prediction time. The two chosen CNNs used to train using our custom dataset where:

1. ResNet50[70] - Residual Neural Network, or ResNet, was originated due to the increasing challenges of machine learning. Residual Neural Networks are a type of ANN, and contain a "skip connection", which means, that it can bypass two or three layers of the network. This specific network is a 50 layer Residual Network.
2. InceptionV3[71] - InceptionV3 is a CNN, that was created by Google and is now in its third version. This network was created with the objective of burning less computational power. InceptionV3 has 48 layers which makes it possible to obtain a lower error rate.

Before starting the training of the algorithm, the installation of some dependencies is needed in order to make the open-source python library, imageai, work. The training of the algorithm was made in Google Colaboratory.

Listing 5.1: Required dependencies to train the algorithm.

```

1 !apt install python3.6
2 !pip install opencv-python
3 !pip install keras==2.2.4
4 !pip install numpy==1.16.1
5 !pip install tensorflow-gpu==1.15.2
6 !pip install imageai --upgrade

```



In order to use the dataset created in this dissertation, a tool bundled with Google Colab called drive is used to mount the drive folder and point the data directory to our dataset.

Listing 5.2: In this snippet of the code the objective is to mount the dataset folder. In line 6, the model used is defined, in this case, it is ResNet50. In line 7, it was defined the directory where the dataset is.

```

1 from imageai.Prediction.Custom import ModelTraining
2 from google.colab import drive
3 drive.mount('/content/gdrive')
4
5 model_trainer = ModelTraining()
6 model_trainer.setModelTypeAsResNet()
7 model_trainer.setDataDirectory("/content/gdrive/My Drive/Colab Notebooks /
8 dataset")

```

The next line in the training algorithm has all the different properties used in the training process:

1. num\_objects - this represents the different types of objects that exist in the dataset.
2. num\_experiments - or epochs, it is the number of times that the network will train using all the images in the train directory.
3. enhance\_data - this option is optional and is related to the possibility of the network creating different copies of the images used in training to achieve better performance.
4. batch\_size - this variable represents the number of images that are processed at once. They are usually in multiples of two.
5. show\_network\_summary - this variable is also optional and consists of showing the structure of the training network in the console before starting the training.

Listing 5.3: This line represents all the variables used in the training of both algorithms, InceptionV3 and ResNet50.

```

1 model_trainer.trainModel(num_objects=4, num_experiments=50, enhance_data=True,
2 batch_size=32, show_network_summary=True, save_full_model=True)

```

Listing 5.4: This listing represents the show\_network\_summary variable that was set to "True" in 5.3. This is a part of the ResNet structure that is shown in console in the beginning of the training process.

```

1 -----
2 Layer (type)                Output Shape          Param # Connected to
3 =====
4 input_1 (InputLayer)        [(None, 224, 224, 3) 0

```

```

5 |-----|
6 | conv2d (Conv2D)                (None, 112, 112, 64) 9472   input_1[0][0]
7 |-----|
8 | batch_normalization (BatchNorma (None, 112, 112, 64) 256   conv2d[0][0]
9 |-----|
10 |
11 | (...)
12 |
13 | activation (Activation)    (None, 112, 112, 64) 0   batch_normalization[0][0]
14 |-----|
15 | global_avg_pooling (GlobalAvera (None, 2048)          0   activation_48[0][0]
16 |-----|
17 | dense (Dense)              (None, 4)           8196  global_avg_pooling[0][0]
18 |-----|
19 | activation_49 (Activation)   (None, 4)           0     dense[0][0]
20 |=====|
21 | Total params: 23,595,908
22 | Trainable params: 23,542,788
23 | Non-trainable params: 53,120

```

Listing 5.5: In this listing is possible to see the number of images as well as the number of different classes that exist in our dataset. After that, the JSON Mapping file is created and stored in the same folder of our dataset. In line 6 is the number of epochs defined in the last line of the training algorithm shown in 5.3. In line 8, is the first epoch with the 3 variables, the ETA, the value loss, and the accuracy of the CNN in the end of that moment of training.

```

1 |
2 | Using Enhanced Data Generation
3 | Found 1841 images belonging to 4 classes.
4 | Found 980 images belonging to 4 classes.
5 | JSON Mapping for the model classes saved to /content/gdrive/My Drive/Colab
6 | Notebooks/dataset/json/model_class.json
7 | Number of experiments (Epochs) : 50
8 | Epoch 1/50
9 | 37/79 [=====>.....] -ETA: 11:31 - loss: 0.6289 - acc: 0.8573

```

### 5.1.1.2 Detection

This section will describe the methods used to achieve the detection of the fruit that is being analysed. The objective is to create a detection algorithm capable of using the two image recognition algorithms presented in 5.1.1.1. The prediction code used is the same for the two CNNs used in this dissertation.

Listing 5.6: Prediction code that was run to test the image recognition algorithm

```

1 | from imageai.Prediction.Custom import CustomImagePrediction
2 | import os
3 | from google.colab import drive

```

```

4 drive.mount('/content/gdrive')
5
6 execution_path = os.getcwd()
7
8 prediction = CustomImagePrediction()
9 prediction.setModelTypeAsResNet()
10 prediction.setModelPath("/content/gdrive/MyDrive/Colab_Notebooks/dataset/
11 models/InceptionV3_accu_1.00_epochs_50.h5")
12 prediction.setJsonPath("/content/gdrive/MyDrive/Colab_Notebooks/dataset/json
13 /model_class.json")
14 prediction.loadModel(num_objects=4)
15
16 predictions, probabilities = prediction.predictImage("/content/gdrive/MyDrive/
17 Colab_Notebooks/dataset/pear.jpg", result_count=3)
18
19 for eachPrediction, eachProbability in zip(predictions, probabilities):
20     print(eachPrediction, ":", eachProbability)

```

The listing above is the prediction code used. As it is possible to see in 5.2, it is essential to mount the folder where the model was saved. In line 9, it is defined the type of model that was used to train the dataset. In line 10 and 11, it is set the paths to the parameters and the model that was trained. In line 14 it is loaded the model and passed as argument the number of objects. In line 16, it is called the function, `predictImage`, that does the prediction and the argument passed is the picture in which it is needed to make the prediction and the number of results that is presented to the user.

Listing 5.7: Example of a prediction result.

```

1 pear : 99.9997615814209
2 tomato : 0.00022084332158556208
3 apple : 8.600648015999468e-06

```

### 5.1.2 Near-Infrared Sensing

In this section, it is going to be described the near-infrared sensing part of the module. This means defining which sensor is going to be used and how it is going to be implemented. After analysing all the sensors available in the market and reviews of each one, the student decided to use SCIO[72], described in 3.1.

After analysing and consulting with the team behind this sensor's development, it becomes clear that a full integration in the module was going to be impossible because of the secure, and closed software behind the sensor. This means that to achieve the goal of having near-infrared sensing, it is necessary to resort to the mobile application made available by Consumer Physics, the company responsible for SCIO. This mobile application is available for both Android and iOS.

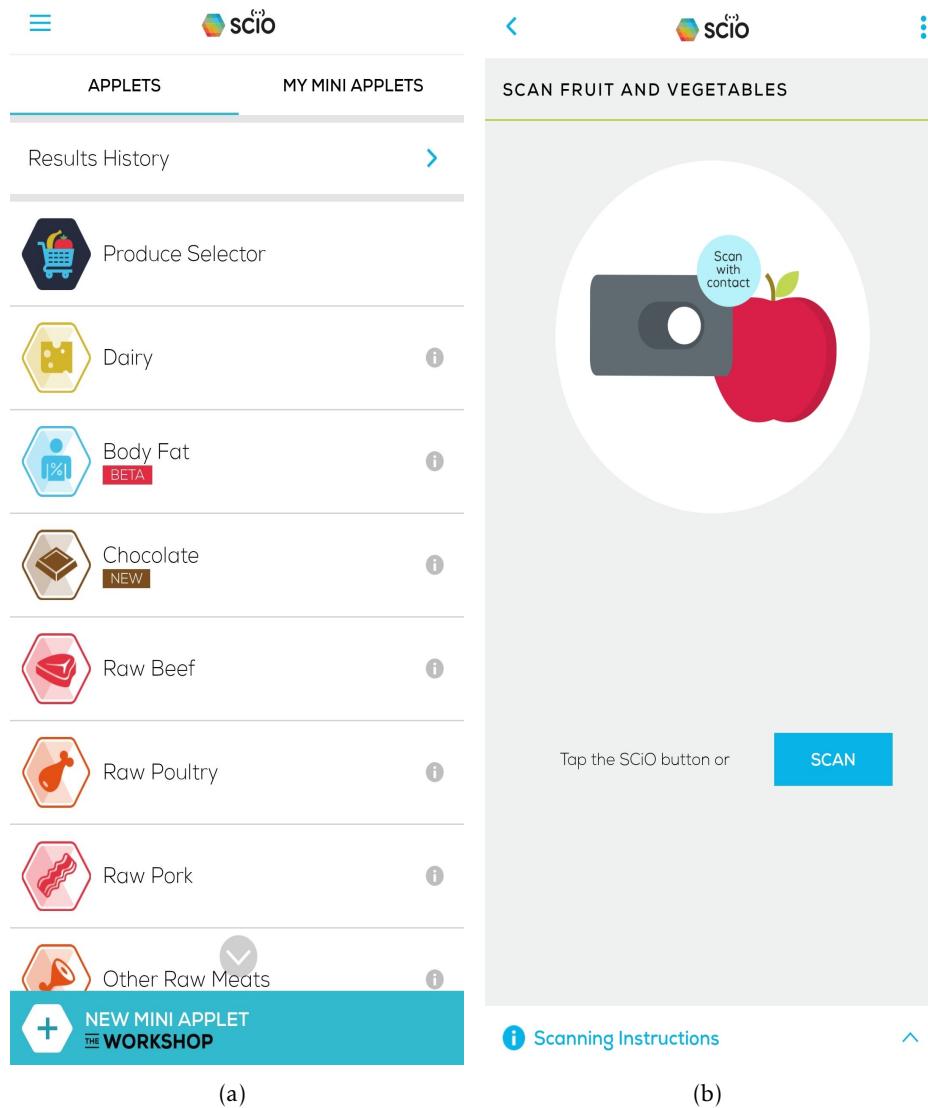


Figure 5.3: (a) - SCIO[72] mobile application dashboard. This page is where the user needs to choose which type of category the food that is being analysed is part of. (b) - Scan screen from where it starts the whole process.

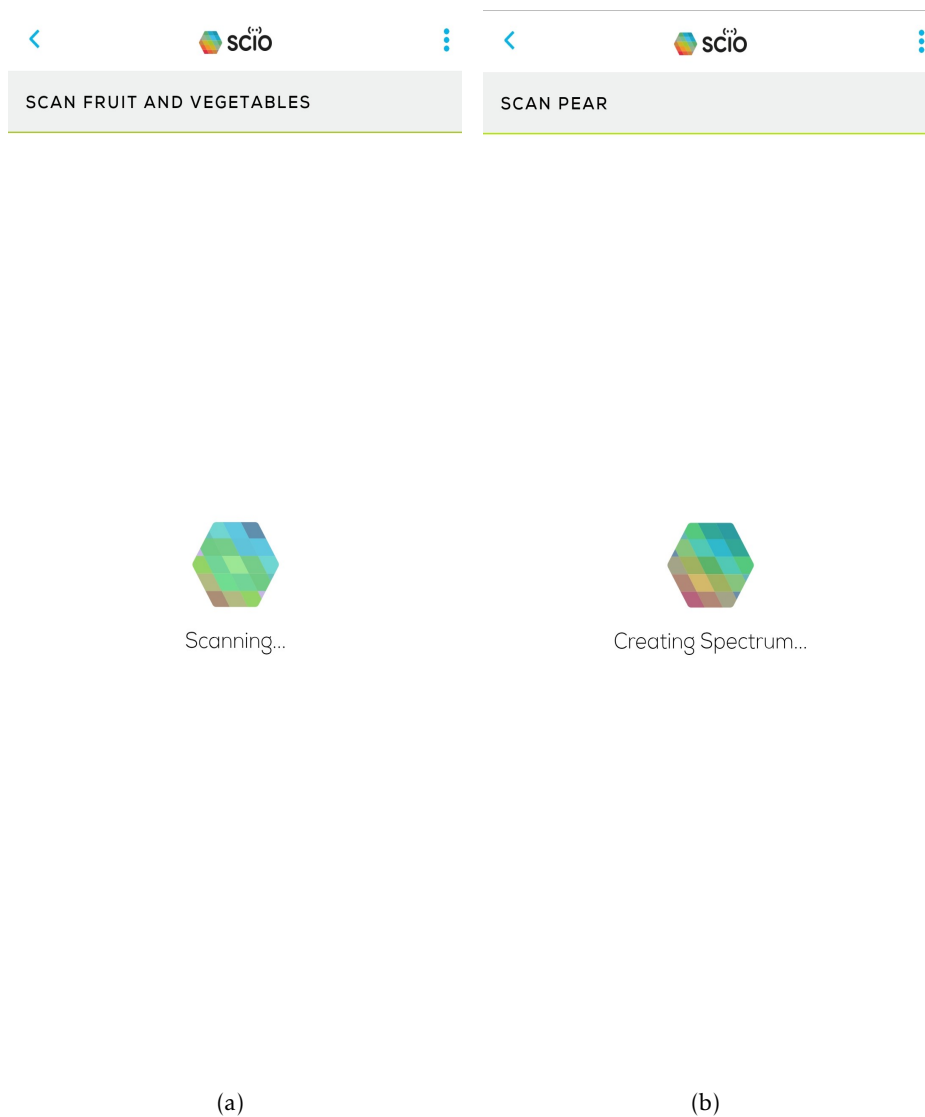


Figure 5.4: (a) - This is the screen shown after the button of the sensor is pressed. (b) - The final screen before the results are shown to the user.

The procedure of near-infrared sensing is described in image 5.3 and 5.4. The steps of the process are in order in the picture. The first step is identifying which category is the product you are scanning. The second step is to touch the sensor to the food being analysed and then press the button for one second or press the mobile application's scan button. After the scanning process, the results will appear in the mobile application.

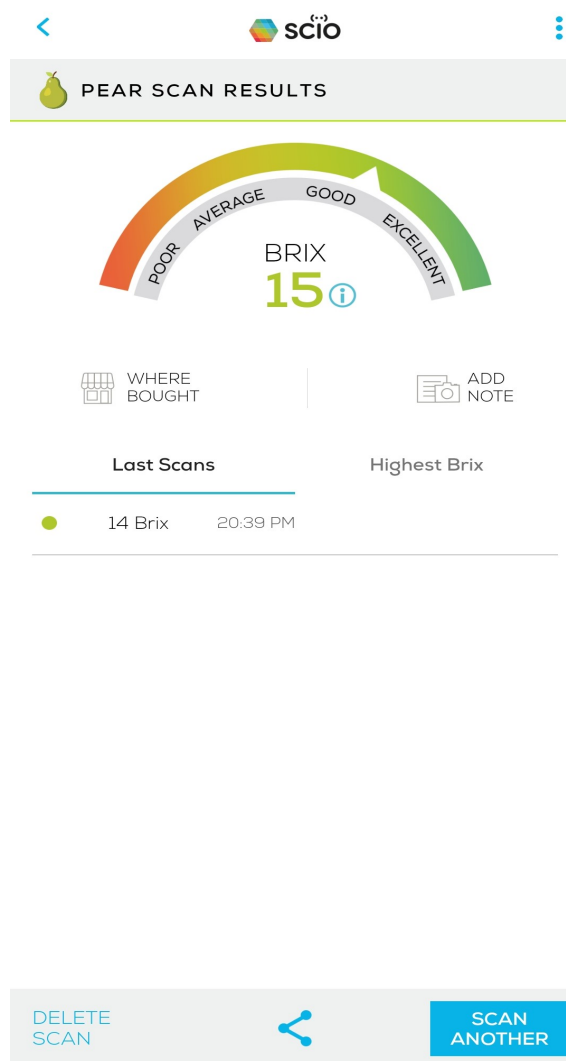


Figure 5.5: Example of a pear BRIX results in the SCIO mobile application

The SCIO app has two different types of measurements for fresh fruit and vegetables. The image 5.5 represents the measurement of BRIX[73]. BRIX or Degrees BRIX is a measure for the total soluble solids or TSS. BRIX cannot differentiate between different types of sugar but instead indicates the full content of all sugars. BRIX is used as an indicator of maturity, flavour, and level of sweetness in the case of fruits and vegetables.

Regarding the mobile application in the BRIX measurement screen, it is possible to see BRIX's scale that indicates what is an excellent value to have in a fruit. It also shows the latest BRIX results and the highest achieved.

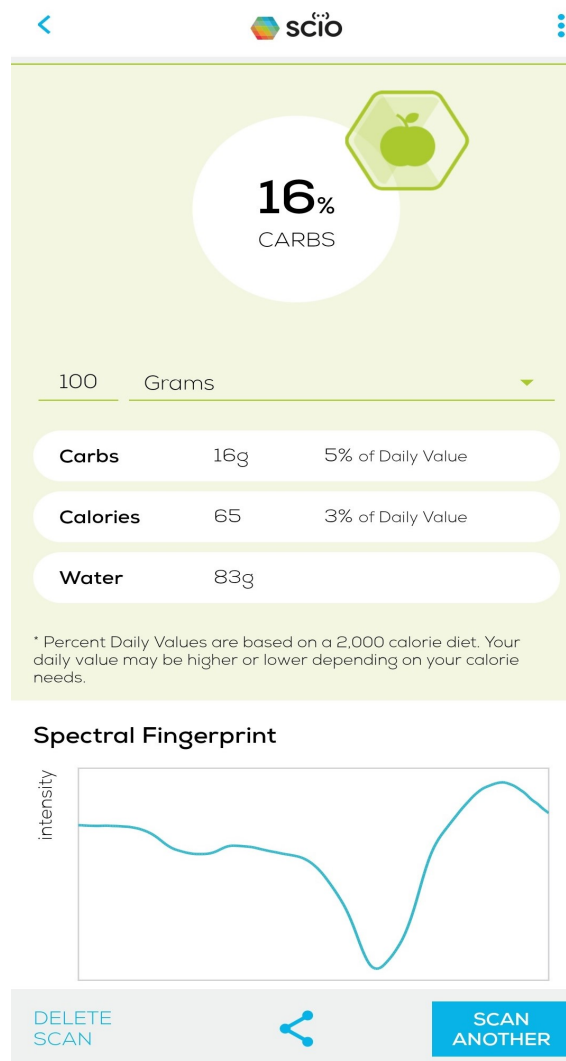


Figure 5.6: Example of a pear scan results in the SCIO mobile application

In image 5.6, it is possible to observe the screen where SCIO measures the nutritional information present in the fruit analysed, which was an apple in the example of 5.6. The nutritional information returned was Carbs, Calories, and Water. Combined with this information, SCIO also helps with healthy tips like matching each nutritional information's percentage to our daily intake needed, this is the amount that the average person should eat daily for each type of nutritional component for a healthy diet.

### 5.1.3 Processing Unit

This section will analyse the chosen processing unit and how it was structured to incorporate the technology described in 5.1.1 and 5.1.2. This processing unit needs to have the external ports required to connect the two devices used, an internet connection (preferably wifi), and an operating system capable of running the image recognition algorithms.

The processing unit chosen was a single-board computer, or SBC. A SBC has every component present in an ordinary computer, but it is built on a single circuit board. This

means that the microprocessor, memory module, graphics card, and input/output (I/O) are all on the same board.

Single-board computers were created to be used in development systems because they are cheaper and much smaller than an ordinary desktop computer. There are many available choices for SBCs in the market that are small, cheap, and meet this dissertation's requirements. The operating system chosen was Raspbian [74]. Raspbian is a free operating system that is based on Debian. This operating system has all the packages necessary, which are optimised to work in a low processing power unit.

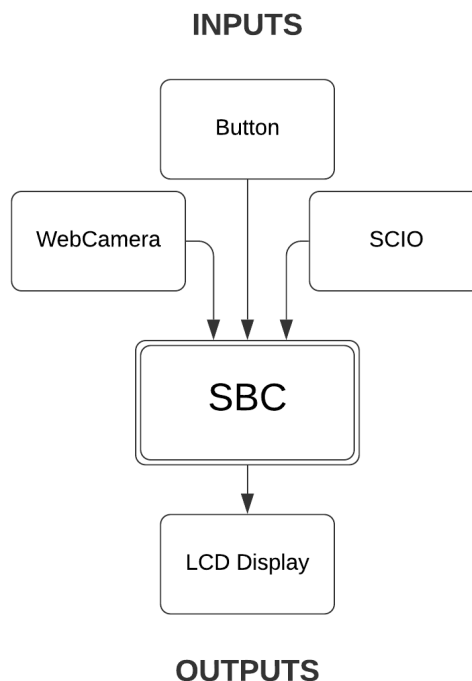


Figure 5.7: Scheme of inputs and outputs of the SBC.

As seen in image 5.7, the inputs of the SBC are the camera, the near-infrared sensor, and a button. The output is the lcd in which the information is displayed. The button is responsible for starting the whole process and is connected to the I/O pins. When the user presses the button, it runs a script that initialises the process and does performance checks after each stage.

Listing 5.8: This snippet of code represents the import of all the libraries needed to execute the detection script. After that, is defined the GPIO pins that are utilised for the button. The last three lines represent the size of the picture that is being captured and the driver of the lcd.

---



```

1 import RPi.GPIO as GPIO
2 import os
3 import pygame, sys
4 import lcd_driver
5 import time
6 from imageai.Prediction.Custom import CustomImagePrediction
7
8 from pygame.locals import *
9 import pygame.camera
10
11 GPIO.setwarnings(False)
12 GPIO.setmode(GPIO.BOARD)
13 GPIO.setup(10, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
14 width = 1920
15 height = 980
16 display = lcd_driver.lcd()

```

Listing 5.9: This part is related to the picture capturing, and the lcd messages that are presented to the user.

```

1 try:
2     display.lcd_display_string("Welcome to Duarte", 1)
3     display.lcd_display_string("Master Thesis", 2)
4     display.lcd_display_string("Press the button", 3)
5     display.lcd_display_string("whenever you want", 4)
6     GPIO.wait_for_edge(10, GPIO.FALLING)
7     pygame.init()
8     pygame.camera.init()
9     cam = pygame.camera.Camera("/dev/video0", (width, height))
10    cam.start()
11    #setup window
12    windowSurfaceObj = pygame.display.set_mode((width, height), 1, 16)
13    pygame.display.set_caption('Camera')
14    #take a picture
15    image = cam.get_image()
16    cam.stop()
17    #display the picture
18    catSurfaceObj = image
19    windowSurfaceObj.blit(catSurfaceObj, (0, 0))
20    pygame.display.update()
21    #save picture
22    pygame.image.save(windowSurfaceObj, 'picture.jpg')
23    display.lcd_clear();
24    display.lcd_display_string("Picture Taken", 1)
25    display.lcd_display_string("Running Recognition", 3)
26    display.lcd_display_string("Algorithm", 4)

```

Listing 5.10: This is the end part in the detection code. In this part, the objective is to run the model, and obtain the result of the fruit that is under analysis.

```

1   prediction = CustomImagePrediction()
2   prediction.setModelTypeAsResNet()
3   prediction.setModelPath("\home\pi\Desktop\resnet_model_ex-015_acc-0.967708
4   \model.h5")
5   prediction.setJsonPath("\home\pi\Desktop\model_class.json")
6   prediction.loadModel(num_objects=4)
7
8   predictions, probabilities = prediction.predictImage("\home\pi\Desktop\
9   \picture.jpg", result_count=1)
10  display.lcd_clear();
11  display.lcd_display_string("The_fruit_is", 1)
12  display.lcd_display_string(predictions, 2)
13
14  except:
15      pass
16  GPIO.cleanup()

```

## 5.2 Experimental Setup

This section will describe the construction of the testing module with all the components necessary to fulfil the battery of tests that analyse the feasibility and accuracy of the various components.

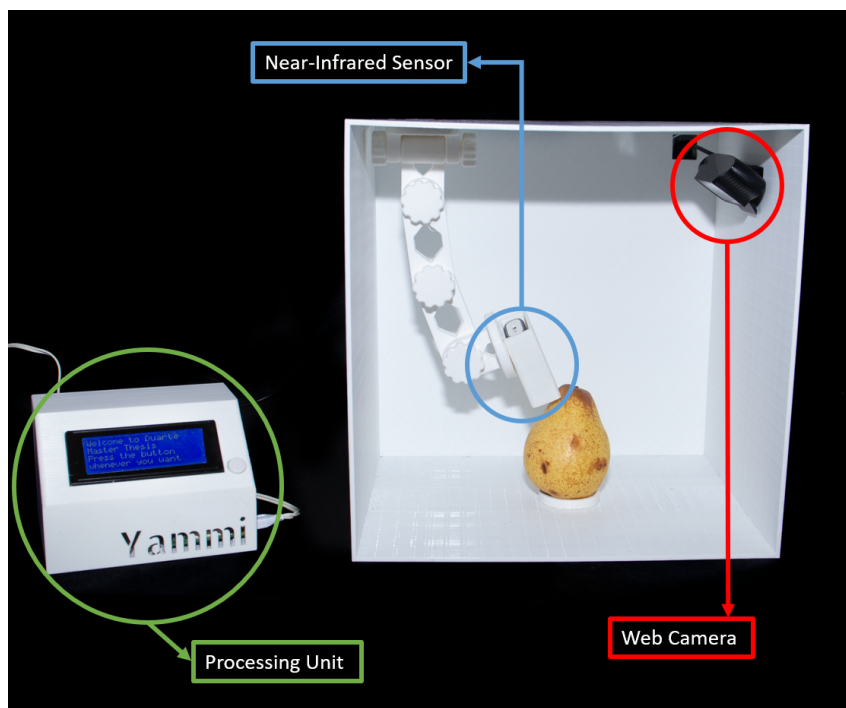


Figure 5.8: Image of the module created.

In 5.8 is represented the module, as well as, all the different components from which the model is composed, highlighted in different colours. In the colour green is the processing unit, the Raspberry Pi. In blue is the near-infrared sensor, SCIO. In red is the web camera, the Logitech C920. The module itself was designed using OnShape[75], a 3D modulation program that is license-free for academic students. The module was then 3D printed using a Blocks One MKII[76] with PLA filament.

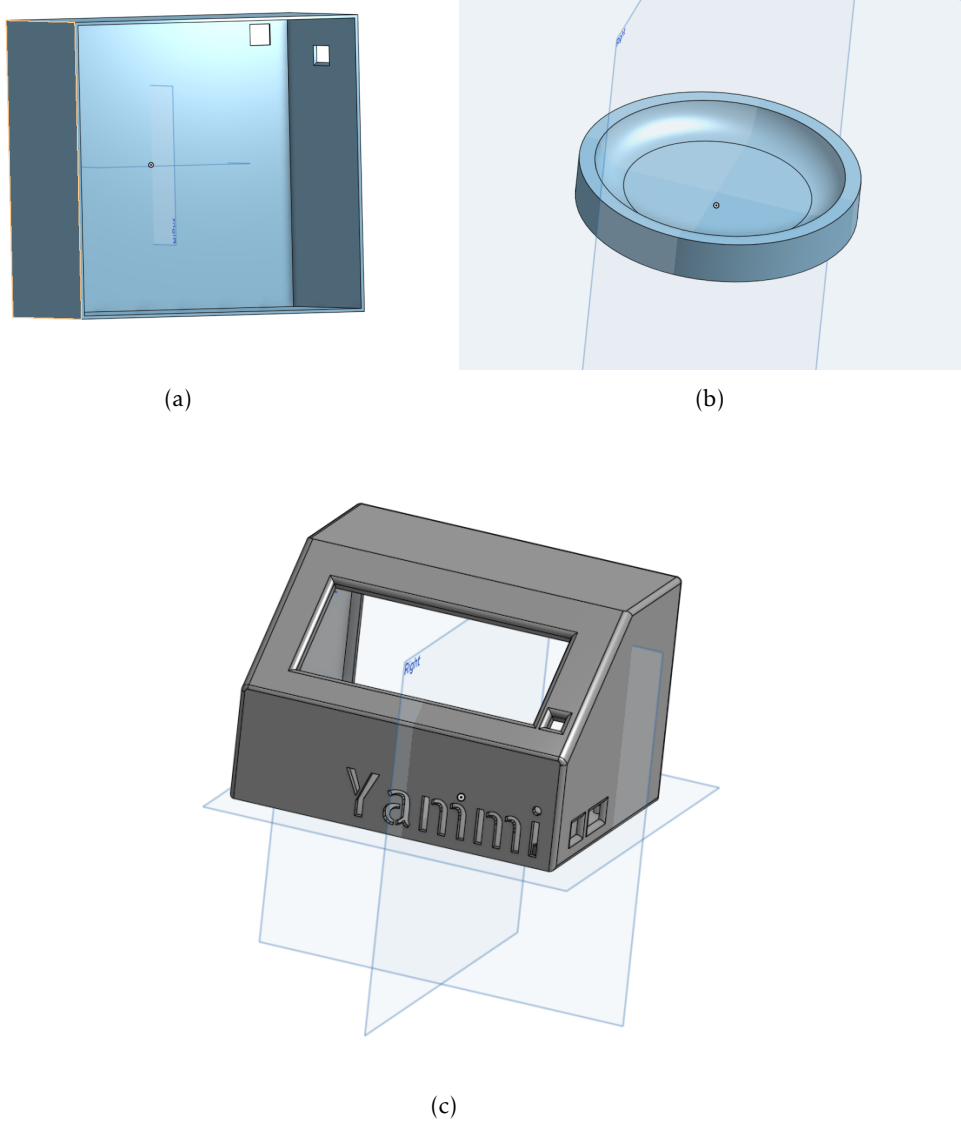


Figure 5.9: (a) - 3D model of the testing area which contained the web camera and the near-infrared sensor, SCIO. (b) - 3D model of the electronics case, which included the SBC and LCD Display. (c) - 3D Model of the base where the object that is under test is put.

The components that compose the model are:

- Web camera - Logitech HD Pro C920. This web camera can take high definition pictures that are ideal for both training the algorithm and running it.
- Near Infrared Sensor - SCIO[72]. The size and quality of this sensor were the key factors in choosing it.
- Processing Unit - Raspberry Pi 3B+. This SBC is equipped with 4x ARM Cortex-A53 CPU that runs at a maximum 1.2GHz, a Broadcom VideoCore IV GPU, wifi and ethernet networking, 40-pin header GPIO, and 1GB of LPDDR2 RAM. It also has four USB 2.0 ports, Camera Serial Interface (CSI), and HDMI.

The image recognition algorithms were trained in a desktop computer equipped with 16GB of RAM, a Ryzen-2600X processor unit, and an MSI GTX 1070 Ti graphics card. This computer was running a 64-bit Windows 10 operating system, and the dataset was trained in a virtual machine of 64-bit Ubuntu OS that used all the resources available on the computer. The algorithms were trained using Google Colaboratory that makes available for the user a Nvidia K80, where it is possible to run all types of programs. The dataset used to train the algorithms had 2800 pictures, 700 per each different fruit.

## EXPERIMENTAL RESULTS

This chapter describes the results obtained during the experimental phase of this dissertation as well as a critical analysis of the results obtained. A critical overview will also be performed for the module developed, as well as, the processing unit performance during this experimental phase. The image recognition algorithm will be analysed in different parameters, such as, detection time, precision, accuracy, and recall. For the near-infrared sensor it was designed two different tests in order to measure two important parameters, accuracy and repeatability.

### 6.1 Image Recognition Algorithm

In this section, it will be presented the analysis of the two models that were created for this dissertation. The first part of the chapter is related to the training of the different algorithms and the second will be focused on the detection evaluation.

#### 6.1.1 Training

In order to achieve this dissertation's objective, two different models were trained, using the same dataset. These models were trained using Google Colaboratory which consists of a online platform running on a google cloud machine and uses a Jupyter Notebook. Google Colaboratory uses a Nvidia Tesla K80 GPU with 12GB of memory.

The dataset used to train the two models was created by the author using a Logitech C920 Webcam and different pieces of fruit, namely, apples, lemons, pears, and tomatoes. The pictures were taken in the module created for this dissertation in order to control the background, and to focus all the features of the images in the object of training. The completed dataset has 2540 images split into 65% training and 35% validation. Every

picture in the dataset was resized by the author so that it would facilitate the training process, meaning that the resolution is the same in all the pictures in the dataset.

Regarding the training of the neural networks, ResNet50 and InceptionV3, the author had to do several tests to avoid the overfitting of the models.

	Batch Size	Number of Epochs
ResNet50	32	15
Inceptionv3	32	10

Table 6.1: Training parameters of each network.

Both models training were stopped when the model started to overfit. As it possible to see in 6.1, the ResNet model needed more epochs to start to overfit than the InceptionV3 model.

### 6.1.2 Detection and Model Metrics Evaluation

During the creating of the training and validation dataset the author also took some extra pictures of the different fruits that belong to the classes trained but also from some other fruits that are similar in colour, shape, or size to the fruits of the four classes in observation. This allowed the author to create a dataset to test all the different model metrics. This dataset is composed by forty eight pictures, thirty two of them are from the different classes studied, and sixteen of them are from different fruits similar to the ones of the model's classes, in this case, bananas and oranges.

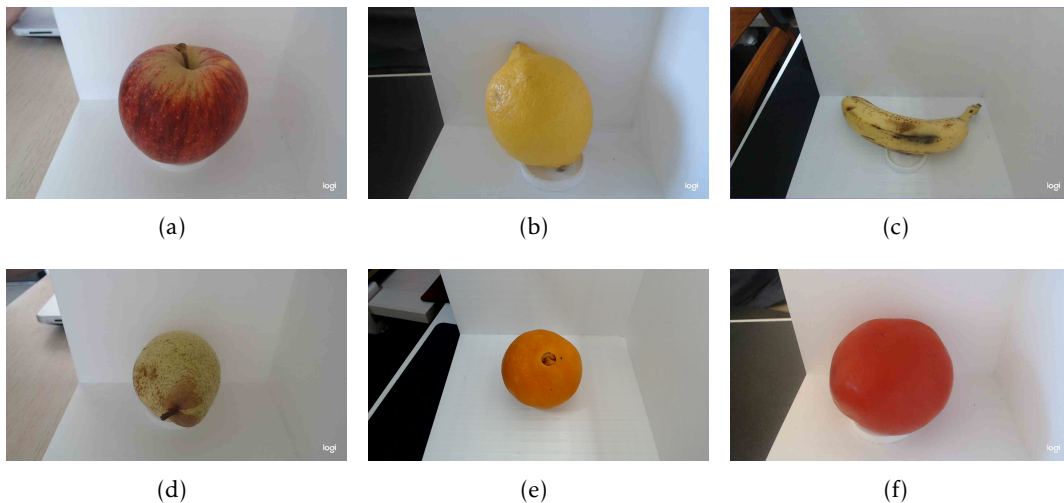


Figure 6.1: Example of the six different types of fruits present in the test dataset.

In the next subsections the author is going to analyse all the different components that led to the final choice of the model to be used in this dissertation's module.

### 6.1.2.1 Classification Metrics

The classification metrics used to evaluate the results of the model in the test dataset were:

1. True positives - when the model predicts that the object belongs to one of the classes and it truly belongs to the class in question.
2. True negatives - when the model predicts that the object does not belong to any class and it truly does not belong to any class.
3. False positives - when the model predicts that the object belongs to one of the classes and in fact does not belong to the class in question.
4. False negatives - when the model predicts that the object does not belong to any of the classes but in reality it belongs to one of the classes.

		Prediction	
		0	1
True Label	0	2 True Negatives	10 False Positives
	1	2 False Negatives	34 True Positives

Figure 6.2: In this matrix it is possible to observe the results of the test dataset using the InceptionV3 model.

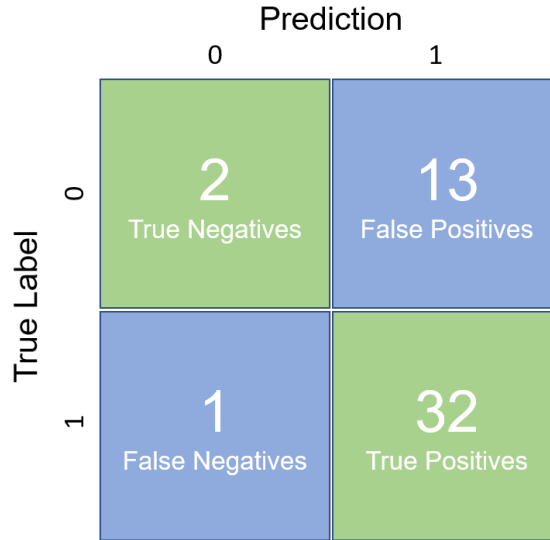


Figure 6.3: In this matrix it is possible to observe the results of the test dataset using the ResNet50 model.

### 6.1.2.2 Accuracy, Precision and Recall

After obtaining the number of true and false, positives and negatives, it is possible to determine the accuracy, precision, and recall of the two trained models. The values for this parameters are obtained using the mathematical formulas below:

1. Precision - describe the fraction of relevant examples (true positives) among the retrieved instances.

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (6.1)$$

2. Recall - defines the fraction of examples that were predicted to belong to that same class.

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (6.2)$$

3. F1 Score - represents the model's accuracy on a dataset. It combines precision and recall.

$$F1\ Score = 2 * \frac{precision * recall}{precision + recall} \quad (6.3)$$

	Precision	Recall	F1 Score
ResNet50	0.71	0.97	0.82
InceptionV3	0.77	0.94	0.85

Table 6.2: Results of the classification metrics of the two models, ResNet50 and InceptionV3.



### 6.1.2.3 Detection Time

The detection time was defined between the time that the algorithm received the input image and the time that it obtained the result. The tests were performed in a SBC that has 1GB of RAM. To test the detection time, both models were run five times in the SBC using the same fruit, in this case it was an apple. The prediction algorithm used was the same for both models.

	Detection Time (s)
ResNet50	42 ( $\pm$ 3)
InceptionV3	38 ( $\pm$ 2)

Table 6.3: Both models detection time using the same prediction algorithm, and an apple as the test subject

Upon analysing 6.2, and 6.3 the author conclude that the model trained using the InceptionV3 was the best option for the final module. It had better precision and overall F1 Score. In the matter of detection time, it also performed better than the ResNet50.

## 6.2 Near-Infrared Sensor

The primary objective was to integrate the SCIO API in the controller in order to use the hardware from the sensor, but use the controller to process the results and show them directly to the user in the lcd. However, the software of the SCIO sensor is closed and the company behind this sensor did not made available any API or developer kit in order to enable the incorporation of the sensor in the controller. This means, that the only way to use the sensor is through the mobile app made available by the company that developed the sensor. All the results shown below were obtained through this app using the process explained in 5.1.2.

In order to test the reliability and accuracy of the sensor, a set of tests was designed.

1. Test ten times the same piece of fruit of each category, which means probing ten times in a row the same apple, lemon, pear, or tomato. This is a repeatability test designed to test the reliability of the sensor.
2. Test ten different pieces of fruits in each category, which means ten apples, ten lemons, ten pears, and ten tomatoes. This test is designed to demonstrate the differences between the results inside the same category.

The pieces of fruit that were tested were bought fresh in the day from the supermarket. The tests were made using the module developed and the mobile application that is provided by the company that develops the sensor SCIO.

It is important to refer a few key aspects about the results that this tests produced:

1. All test results are "per 100g". This means that the values that are presented in the tables below are all related to a quantity by hundred grams of the food that is being analysed.
2. The carbohydrates and the water results are in grams. The calories results are in the unit of calories, cal, that is used to measured energy.
3. The calories results are divided in intervals of five.

### 6.2.1 Repeatability Tests

The repeatability test was made in the same day and the ten probes were made consecutively in the same piece of fruit. To analyse the repeatability, the mathematical methods used were the average value between all the fields, and the standard deviation. The methods chosen were used in all four fruits: apple, lemon, pear, and tomato.

The formulas used were:

1. Standard Deviation -

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \quad (6.4)$$

2. Average -

$$Average = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (6.5)$$

#### 6.2.1.1 Apple Repeatability Test Results

Apple per 100g of fruit	1	2	3	4	5	6	7	8	9	10
Carbohydrates(grams)	17	17	17	18	18	19	17	17	17	17
Calories	80	80	75	80	80	85	75	75	75	75
Water(grams)	81	81	82	79	81	79	81	81	81	81

Table 6.4: Results of the repeatability tests performed ten times in the same apple.

	Standard Deviation	Average
Carbohydrates(grams)	0,70	17,4
Calories	3,50	78
Water(grams)	0,95	80,7

Table 6.5: Standard deviation and average for the apple test results presented in 6.4.

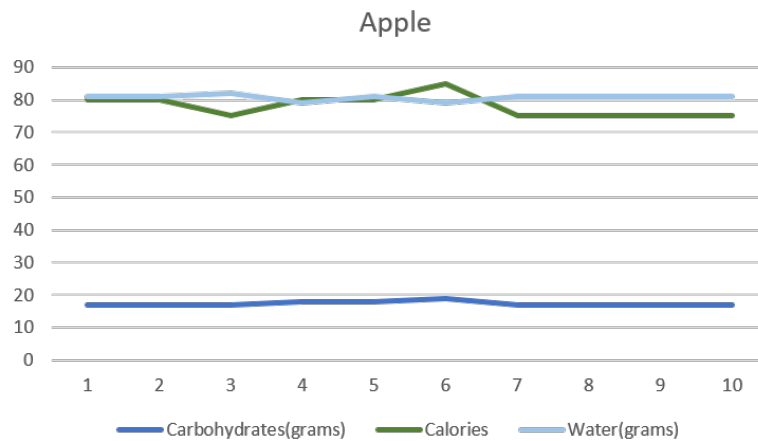


Figure 6.4: Apple Repeatability test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes.

Upon analysing the table 6.4 and 6.5 it is possible to conclude that in the apple tests the repeatability of the sensor is accurate since the standard deviation between all three fields of analysis are small compared to the average.

### 6.2.1.2 Lemon Repeatability Test Results

Lemon per 100g of fruit	1	2	3	4	5	6	7	8	9	10
Carbohydrates(grams)	11	13	12	14	12	13	12	12	12	12
Calories	50	60	55	65	60	60	55	60	55	55
Water(grams)	87	85	86	84	85	85	87	85	86	86

Table 6.6: Results of the repeatability tests performed ten times in the same lemon.

	Standard Deviation	Average
Carbohydrates(grams)	0,82	12,3
Calories	4,25	57,5
Water(grams)	0,97	85,6

Table 6.7: Standard deviation and average for the lemon test results presented in 6.6.

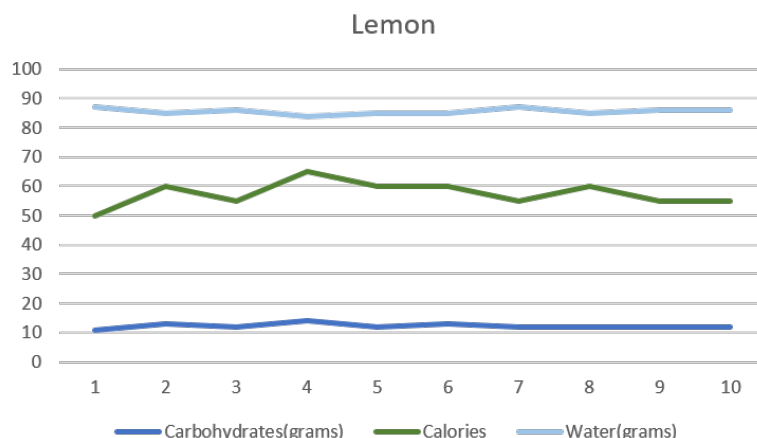


Figure 6.5: Lemon Repeatability test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes.

The results of the lemon repeatability tests 6.6 were analysed and the mathematical methods used are in 6.7. It is possible to observe that, the results are worse than in the apple tests. However, it is important to refer that calories are in intervals of five, so it is normal that the results in this field are worse. The values of standard deviation in the carbohydrates, and water are very small, less than one gram of deviation. Concluding the repeatability of the sensor in lemons is accurate.

### 6.2.1.3 Pear Repeatability Test Results

Pear per 100g of fruit	1	2	3	4	5	6	7	8	9	10
Carbohydrates(grams)	20	20	20	21	20	20	20	20	20	18
Calories	90	85	90	95	90	90	85	90	90	80
Water(grams)	78	78	78	78	78	78	79	78	78	80

Table 6.8: Results of the repeatability tests performed ten times in the same pear.

	Standard Deviation	Average
Carbohydrates(grams)	0,74	19,9
Calories	4,12	88,5
Water(grams)	0,67	78,3

Table 6.9: Standard deviation and average for the pear test results presented in 6.8.

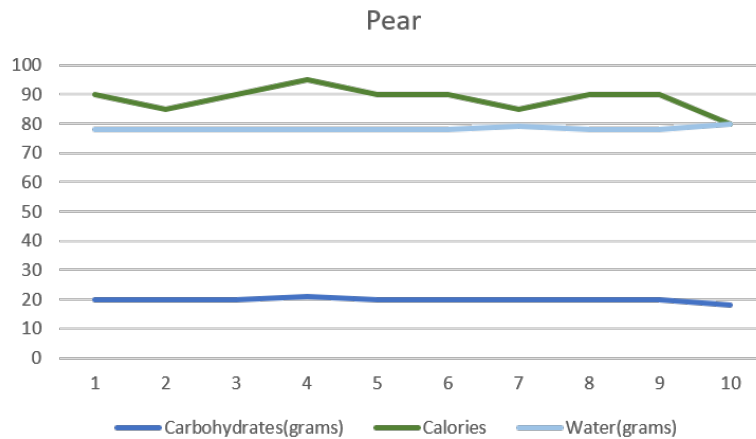


Figure 6.6: Pear Repeatability test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes.

Analysing the pear repeatability test results it is possible to conclude that the sensor was accurate after the ten probes. The results are better than the lemon tests but were not the best achieved from the first set of tests.

#### 6.2.1.4 Tomato Repeatability Test Results

Tomato per 100g of fruit	1	2	3	4	5	6	7	8	9	10
Carbohydrates(grams)	<5.0	<5.0	<5.0	<5.0	5	5	5	<5.0	6	6
Calories	25	25	25	25	30	25	25	25	30	30
Water(grams)	93	93	93	94	92	93	93	93	92	92

Table 6.10: Results of the repeatability tests performed ten times in the same tomato.

	Standard Deviation	Average
Carbohydrates(grams)	0,60	4,95
Calories	2,42	26,5
Water(grams)	0,63	92,8

Table 6.11: Standard deviation and average for the tomato test results presented in 6.10.

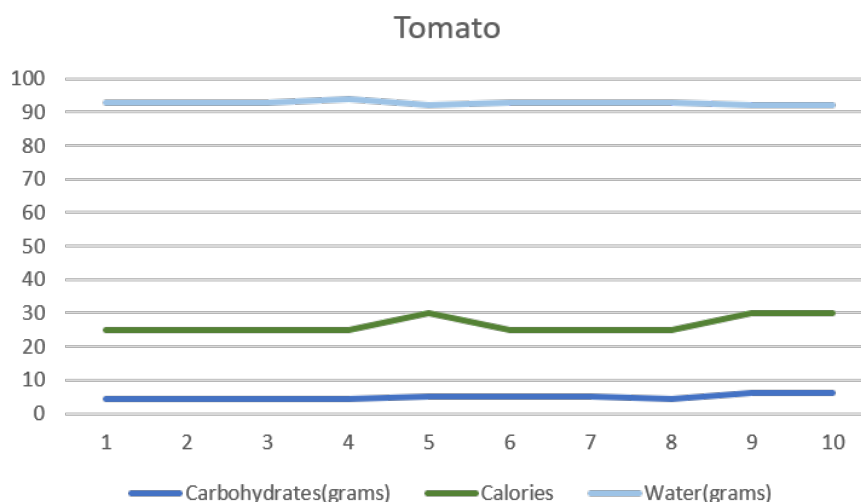


Figure 6.7: Tomato Repeatability test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes.

The tomato test results have a particularity because the sensor does not measure below five grams of carbohydrates. Since tomatoes, in general, have below five grams of carbohydrates, it is not possible to achieve accurate results so for the values that are below five grams it was assumed in the graph, and in the mathematical methods the value of four and a half grams (4,5).

As is possible to observe the tomato tests achieved the best results of all four fruits, by having the lower standard deviation across all fields carbohydrates, calories, and water.

### 6.2.2 Accuracy Test Results

In order to be able to make an analytical analysis of the second phase of tests it is important to have reference values. For this tests the values used are presented in the table below.

Apple per 100g of fruit	Reference Values	Lemon per 100g of fruit	Reference Values
Carbohydrates(grams)	13,8	Carbohydrates(grams)	9,3
Calories	52	Calories	29
Water(grams)	86	Water(grams)	89

Pear per 100g of fruit	Reference Values	Tomato per 100g of fruit	Reference Values
Carbohydrates(grams)	27	Carbohydrates(grams)	3,9
Calories	101	Calories	18
Water(grams)	83	Water(grams)	95

Table 6.12: Reference values for Apple[77], Lemon[78], Pear[79], Tomato[80].

The accuracy test were made during the course of a week always using fresh fruit

from the day. It was tested ten different pieces of the same fruit, this means: ten apples, ten lemons, ten pears, and ten tomatoes.

The analysis of the accuracy test results will be more focused on the mathematical method, average, compared with the reference values in 6.12. The values in this table were obtained using trustful sources due to the fact that there was no possibility to do the tests using a laboratory spectrometer.

### 6.2.2.1 Apple Accuracy Test Results

Apple per 100g of fruit	1	2	3	4	5	6	7	8	9	10
Carbohydrates(grams)	19	15	14	15	19	15	18	19	14	16
Calories	85	70	65	70	85	70	80	85	65	70
Water(grams)	79	83	84	83	79	83	81	79	84	82

Table 6.13: Results of the accuracy tests performed in ten different apples.

	Standard Deviation	Average
Carbohydrates(grams)	2,12	16,4
Calories	8,32	74,5
Water(grams)	2,06	81,7

Table 6.14: Standard deviation and average for the apple test results presented in 6.13.

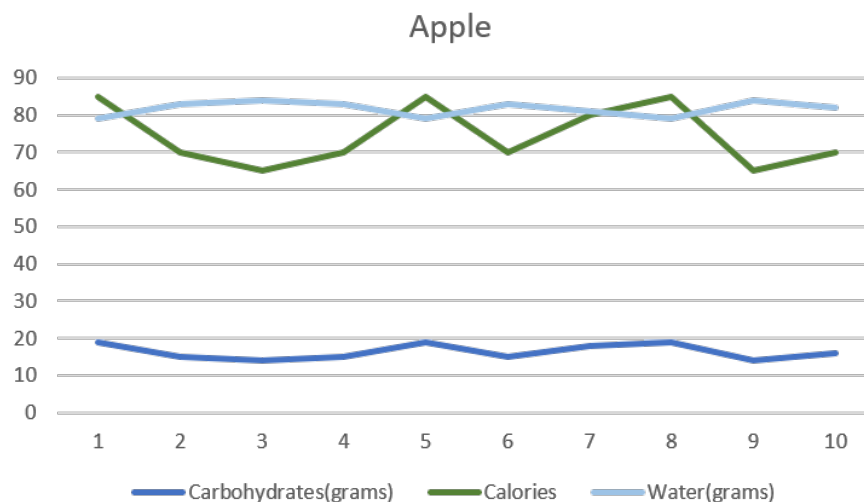


Figure 6.8: Apple Accuracy test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes in the ten different apples.

Analysing both tables 6.13, and 6.12 it is possible to observe that the sensor SCIO did not perform, as well as, in the repeatability tests, specially in the field of calories. The average value of calories in the ten apples is 74,5, and in the reference values table,

the value is 52. In the other fields of carbohydrates, and water the difference is not so significant between the accuracy test results and the reference values as in the calories fields but is significant enough to conclude that the accuracy of the sensor SCIO in apples is not satisfactory.

### 6.2.2.2 Lemon Accuracy Test Results

Lemon per 100g of fruit	1	2	3	4	5	6	7	8	9	10
Carbohydrates(grams)	10	10	13	12	11	14	12	11	14	10
Calories	45	50	70	55	55	80	55	55	70	50
Water(grams)	88	88	83	86	86	81	86	86	84	88

Table 6.15: Results of the accuracy tests performed in ten different lemons.

	Standard Deviation	Average
Carbohydrates(grams)	1,57	11,7
Calories	11,07	58,5
Water(grams)	2,32	85,6

Table 6.16: Standard deviation and average for the lemon test results presented in 6.15.

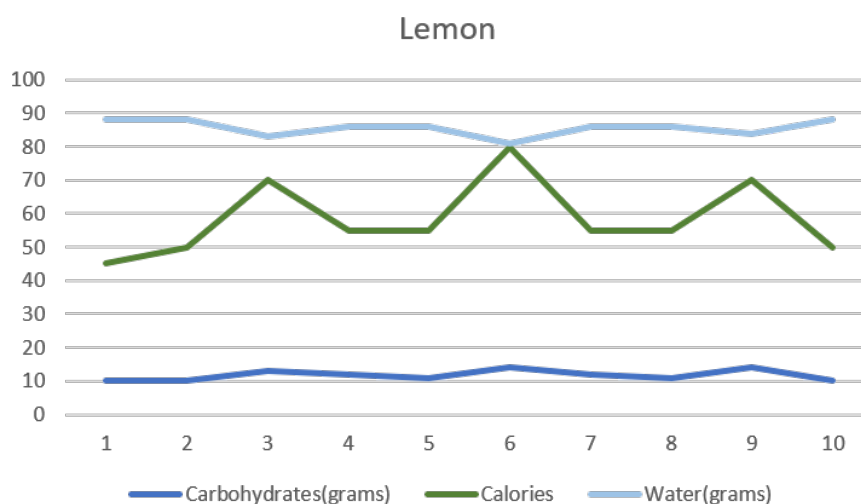


Figure 6.9: Lemon Accuracy test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes in the ten different lemons.

In the lemon accuracy test results it is possible to observe that the difference between the reference values in 6.12, and the average in the test results in 6.15 is low in the fields of the carbohydrates, and water. However, in the field of calories the difference is very significant having scored 58,5 average calories in the test results, and having a reference



value of just 29. Nevertheless, the sensor behaviour in the lemon accuracy tests was much better than the apples.

### 6.2.2.3 Pear Accuracy Test Results

Pear per 100g of fruit	1	2	3	4	5	6	7	8	9	10
Carbohydrates(grams)	16	16	14	17	16	17	17	17	15	16
Calories	70	70	65	75	70	70	70	75	65	70
Water(grams)	82	82	84	81	82	81	82	81	82	82

Table 6.17: Results of the accuracy tests performed in ten different pears.

	Standard Deviation	Average
Carbohydrates(grams)	0,99	16,1
Calories	3,33	70
Water(grams)	0,88	81,9

Table 6.18: Standard deviation and average for the pear test results presented in 6.17.

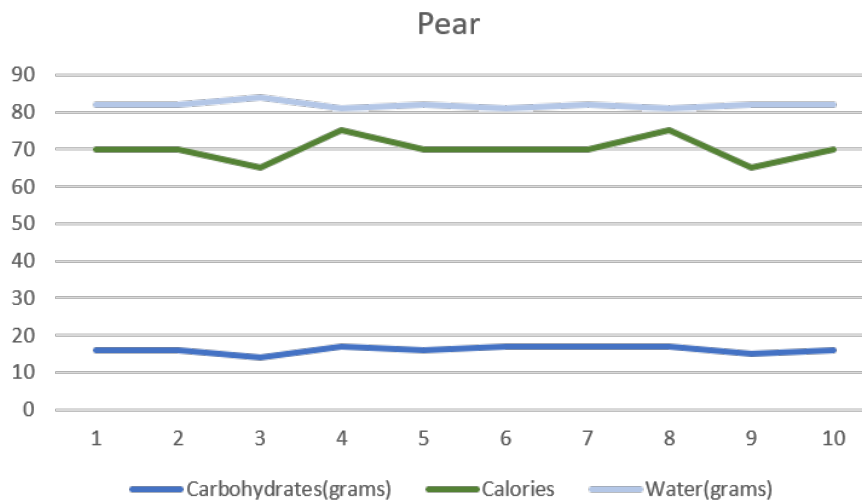


Figure 6.10: Pear Accuracy test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes in the ten different pears.

The field of water in the pear accuracy tests achieved the closest result out of the four fruits analysed. However, in the carbohydrates field the difference is very significant and, as it was possible to observe in the results of the two previous fruit tests, the calories unit presented again the biggest difference.

## 6.2.2.4 Tomato Accuracy Test Results

Tomato per 100g of fruit	1	2	3	4	5	6	7	8	9	10
Carbohydrates(grams)	<5,0	<5,0	<5,0	<5,0	<5,0	7	<5,0	<5,0	6	<5,0
Calories	20	20	20	20	25	30	20	20	25	25
Water(grams)	93	94	94	94	93	92	94	93	92	93

Table 6.19: Results of the accuracy tests performed in ten different tomatoes.

	Standard Deviation	Average
Carbohydrates(grams)	0,88	4,9
Calories	3,54	22,5
Water(grams)	0,79	93,2

Table 6.20: Standard deviation and average for the tomato test results presented in 6.10.

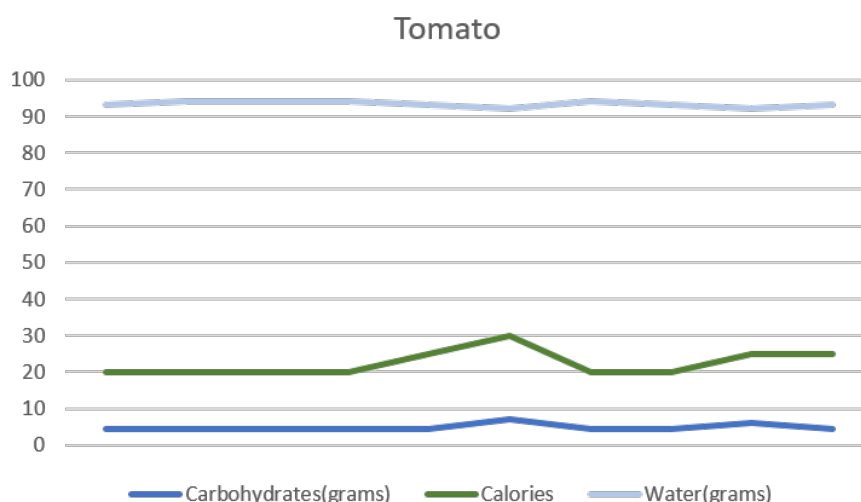


Figure 6.11: Tomato Accuracy test results graphic. The ten probes are in the x-axis. The results obtained are in the y-axis. In this graphic it is possible to observe the different fields of analysis throughout the ten probes in the ten different tomatoes.

Overall, the accuracy test results of the tomatoes achieved the best results of all the other fruits. It is possible to observe in 6.12, and 6.19 the difference across all the units of test, carbohydrates, calories, and water, is minimal.

One possible justification for why this happens is the thickness of the skin of a tomato and the internal composition of a tomato. It was referred in 3.1, that the sensor works based on the insertion of light inside the object that is probed. Since the skin of a tomato is thinner than the other fruits, and the internal composition is also less dense than in the other fruits this can relate to achieving better results when using this type of sensor.

### 6.2.3 Spectral Images Comparison

The mobile application of the near-infrared sensor, SCIO, also provides a spectral image of the object that is being probed. Analysing the spectral image, it is also possible to draw some interesting conclusions about the effectiveness of the sensor. Considering that each different piece of fruit has its own spectral image, and there were two different set of tests, one with the same piece of the same fruit, and one with different pieces of the same fruit. In the next sections, the author goes over the repeatability test spectral images and the accuracy test spectral images. In order to make it easier to analyse the graphs of the ten different probes made in each step, the author used an image editing program to superimpose the ten different spectral images.

#### 6.2.3.1 Repeatability Spectral Images

In this section, the objective is to observe the repeatability test spectral image. In this set of tests, the same piece of each different fruit was probed ten times in a row, what is expected to observe is the same or, as close as possible, of the same spectral image inside each class of fruit.

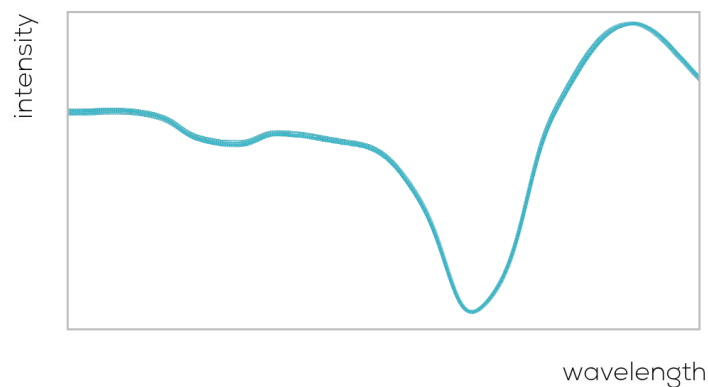


Figure 6.12: Apple repeatability test spectral image.

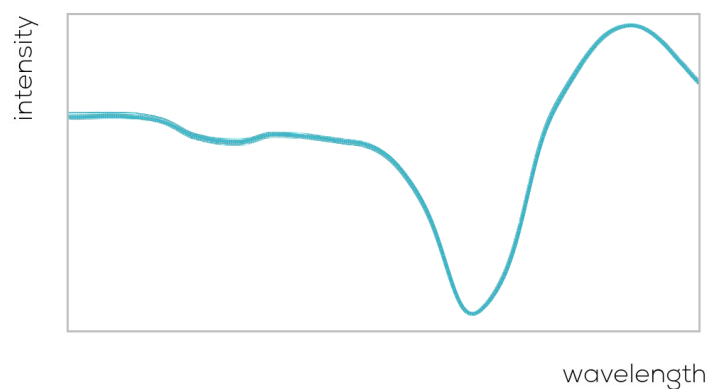


Figure 6.13: Lemon repeatability test spectral image.

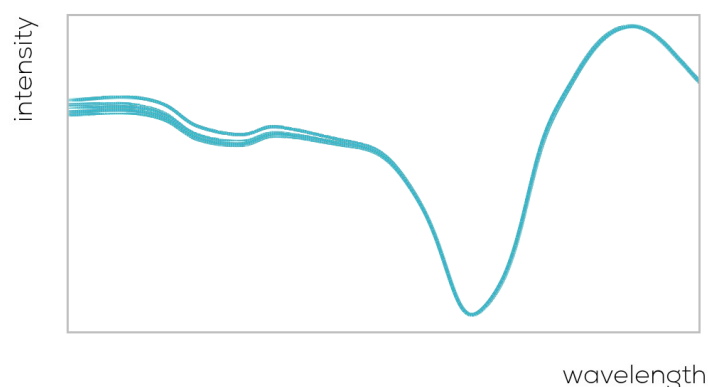


Figure 6.14: Pear repeatability test spectral image.

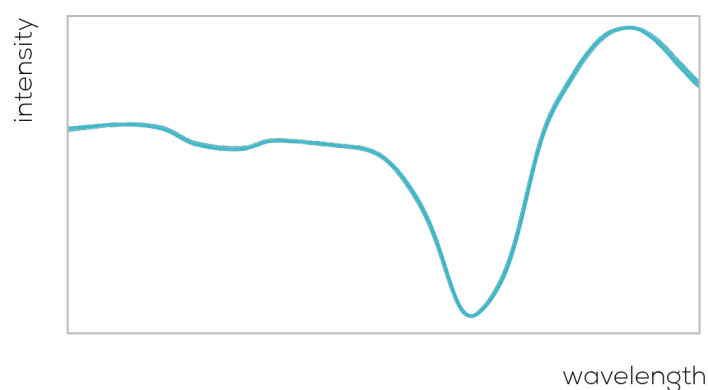


Figure 6.15: Tomato repeatability test spectral image.

After analysing all the repeatability spectral images it is possible to observe the sensor performed extremely well in all the different categories of fruits except in the pear. In 6.12, 6.13, and 6.15 the ten lines of the ten different probes are almost perfectly overlapping. This means, that for apples, lemons, and tomatoes the sensor is capable of providing a reliable spectral image.

However, in 6.14 it is possible to observe marked differences between the ten different probes of the same pear. This can be due to the position of the sensor when probing the pear. The shape of a pear is not ideal for this sensor because of some pears do not have enough flat surface to support the sensor.

### 6.2.3.2 Accuracy Spectral Images

In this section the objective is to overview the accuracy spectral images. The set of tests regarding accuracy was to probe ten different pieces of fruit in the same category, what is expected is to have spectral images that are similar but have slight differences to prove that the sensor can provide different spectral images for different types of fruit.

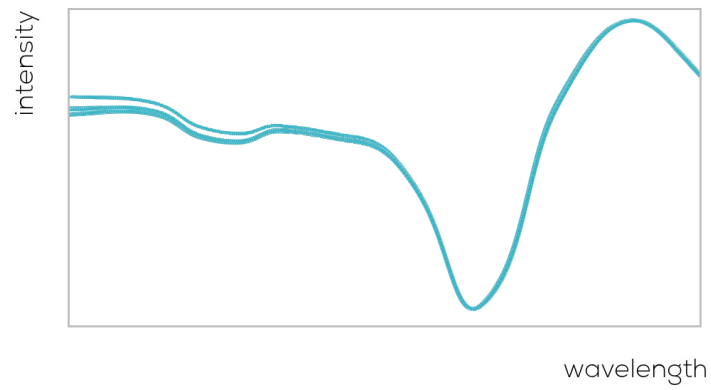


Figure 6.16: Apple accuracy test spectral image.

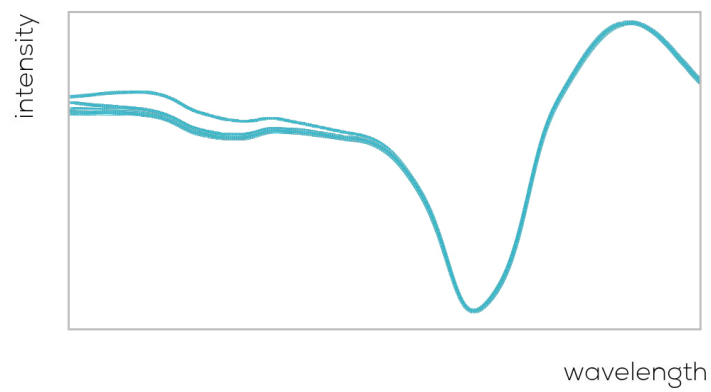


Figure 6.17: Lemon accuracy test spectral image.

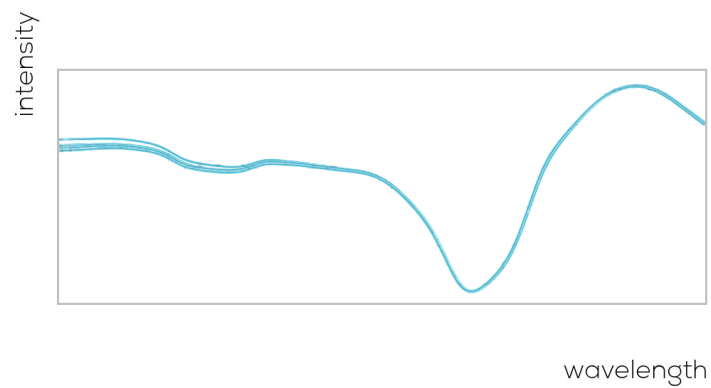


Figure 6.18: Pear accuracy test spectral image.

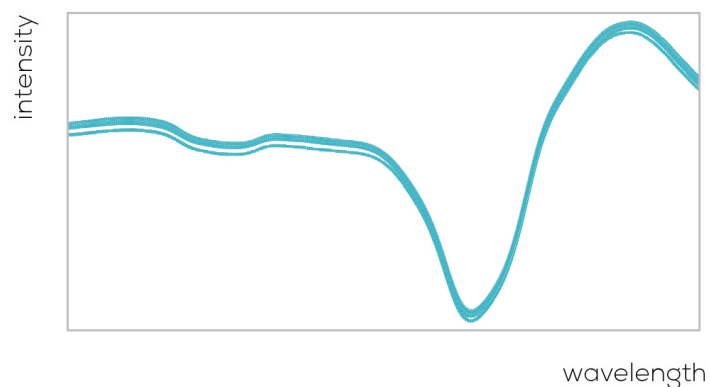


Figure 6.19: Tomato accuracy test spectral image.

After observing 6.16, 6.17, 6.18, and 6.19 it is possible to conclude that the near-infrared sensor is effective in the accuracy spectral images. Every spectral image of the different categories of fruits has some small differences specially in the beginning of the spectral image.

## CONCLUSION AND FUTURE WORK

### 7.1 Conclusion

The author achieved the main objective of this dissertation, that is the construction of a module capable of obtaining the nutritional value extraction of food exploiting computer vision and near-infrared spectrometry. The module obtained with all the parts incorporated presents a perfect solution and achieves good results.

This dissertation started with the objective of obtaining the nutritional value of food using near-infrared spectrography to solve the research problem. After analysing the several devices available in the market, it was possible to conclude that using only the near-infrared spectrometer, it would be impossible to solve the main objective. Therefore, it was necessary to introduce a first step to the process. After analysing the different solutions possible for this first step, the author decided to use computer vision, and image recognition algorithms.

Resuming, this dissertation was divided into two steps image recognition, and the use of near-infrared sensing. It was also necessary to add a controller to pass the information and run both devices. To perform the computer vision part, a camera was added to the testing module from which the dataset was obtained, to train, and test the different image recognition algorithms. These image recognition algorithms were developed using Python, ImageAI, Tensorflow, Google Colab, and Keras. These tools are developer-friendly allowing the author to implement the desired solution, and achieve quite good results in all the tests performed.

This solution resulted in an easy way to implement the image recognition algorithms in the controller, and solve the problem of detecting what type of food was being tested. As it possible to observe in 6.3, and 6.2, both models were eligible to be chosen for this dissertation's module. ResNet50, and InceptionV3 achieved results that the author

considers very satisfactory since both of the models had precision above 0.70, recall above 0.93, and F1 score above 0.81. The decisive factor was the detection time of both algorithms, in which, the InceptionV3 model performed much better than the ResNet50.

The near-infrared sensing part was implemented using the company's mobile application that developed the sensor SCIO. The mobile application was very well designed and is made in a very intuitive, and easy to use way. The original objective was to use the sensor API, and integrate the sensor in the controller, but since that was not possible, due to the fact that the sensor is closed software, the author had to resign to the use of the mobile application.

After making all the tests, it was possible to observe that regarding the repeatability, the sensor achieved good results. As is possible to see, in 6.4, 6.8, 6.6, and 6.10, the sensor had low values in the standard deviation across all fields. However, in the accuracy part and having into account 6.12, the sensor did not perform so well. The sensor had significant differences in value between the average value obtained in the tests and the reference values. This means, that despite the sensor achieving good results in the repeatability part of the tests, it has not performed well in the accuracy tests.

Since this sensor's objective is to be used daily by regular people to evaluate the quality of food they are buying, accuracy is the main goal that this sensor should achieve, and after analysing all the tests it is clear that this sensor is not the best solution for this task, for the moment.

The controller used to act as an intermediary between the different parts of this dissertation has also achieved excellent results. Due to the limited processing capacities of the SBC chosen, the image recognition algorithm took more time than expected to run. However, the SBC used was has prove of concept. There are already SBCs in the market dedicated to machine learning that are more effective in running image recognition algorithms.

The module was developed using 3D modeling software, and was 3D printed. It was used during the entire test series for both parts of the development. It was necessary to make some adjustments, but in general, the functionality and the way the module was designed and built is intuitive and presents all the relevant information for the user, and is very easy to use. For this module to become a final solution, it is necessary to develop the module using other materials to be food-safe and pass the different certifications.

## 7.2 Future Work

The work developed during this dissertation serves as a good starting point for the module that will be created and commercialised in the future. However, it is not a final solution, and with development, and additions can increase the quality. The list below is some of the improvements that the author believes will enhance the module.

- The fact that the user must introduce what type of category of the food that is being



analysed manually in the mobile application reduces the level of integration of all the different components. One solution would be to make a commercial partnership with the company responsible for developing the sensor to integrate the algorithms and databases in the controller or provide a connection to these same algorithms and databases from the controller.

- The processing time of the image recognition algorithm needs to be optimised. This means that either the controller needs to be more powerful in terms of processing capacity, or the image recognition algorithm needs to be optimised to run in a low capacity controller.
- The module itself needs to be redesigned to be mass-produced. The material with which the module was printed is food safe but is not very resistant or has the quality in terms of presentation to be a final product. Other solutions like plastic injection or thermoforming are much more durable and have the quality necessary to be sold to the general public.
- An important test that could not be done was to compare the near-infrared sensor, SCIO, with a laboratory level spectrometer. This test, would be essential in order to determine if the sensor has the level of accuracy needed.



## BIBLIOGRAPHY

- [1] *Obesity and overweight*. URL: <https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight> (visited on 11/27/2020).
- [2] P. P. Kalantri, R. R. Somani, and D. T. Makhija. "Raman spectroscopy : A potential technique in analysis of pharmaceuticals." In: *Der Chemica Sinica* 1.1 (2010), pp. 1–12.
- [3] ThermoNicolet. "FTIR vs. Dispersive IR." In: (2002). URL: [http://www.thermo.com.cn/Resources/200802/productPDF{\\\_}21615.pdf](http://www.thermo.com.cn/Resources/200802/productPDF{\_}21615.pdf).
- [4] L. H. Crawley. "Application of Non-Dispersive Infrared ( NDIR ) Spectroscopy to the Measurement of Atmospheric Trace Gases by." In: (2008).
- [5] J. B. Bates. "Fourier transform spectroscopy." In: *Computers and Mathematics with Applications* 4.2 (1978), pp. 73–84. ISSN: 08981221. DOI: 10.1016/0898-1221(78)90020-2.
- [6] W. D. Perkins. "Fourier transform-infrared spectroscopy: Part I. Instrumentation." In: *Journal of Chemical Education* 63.1 (1986), A5. ISSN: 0021-9584. DOI: 10.1021/ed063pA5. URL: <http://pubs.acs.org/doi/abs/10.1021/ed063pA5>.
- [7] A. Hirai, H. Matsumoto, D. Lin, and C. Tagaki. "Heterodyne Fourier transform spectrometer for the near- infrared region." In: *Opt. Express* 11.11 (2003), pp. 1258–1264. DOI: 10.1364/OE.11.001258. URL: <http://www.opticsexpress.org/abstract.cfm?URI=oe-11-11-1258>.
- [8] E. G. Learned-Miller. "Introduction to Computer Vision & Robotics." In: *Department of Computer Science University of Massachusetts* (2011), pp. 2–8.
- [9] *Computer Vision applications in Self-Driving Cars - Becoming Human: Artificial Intelligence Magazine*. URL: <https://becominghuman.ai/computer-vision-applications-in-self-driving-cars-610561e14118> (visited on 01/24/2020).
- [10] T. S. Huang. "Computer Vision: Evolution and Promise." In: *University of Illinois* (1997), p. 5. ISSN: 03029743. DOI: 10.5170/CERN-1996-008.21. URL: <http://cds.cern.ch/record/400313/files/p21.pdf>.

- [11] J. Wu, B. Peng, Z. Huang, and J. Xie. “Research on computer vision-based object detection and classification.” In: *IFIP Advances in Information and Communication Technology* 392 AICT.PART 1 (2013), pp. 183–188. ISSN: 18684238. DOI: [10.1007/978-3-642-36124-1\\_23](https://doi.org/10.1007/978-3-642-36124-1_23).
- [12] N. Kulkarni. “Color Thresholding Method for Image Segmentation of Natural Images.” In: *International Journal of Image, Graphics and Signal Processing* 4.1 (2012), pp. 28–34. ISSN: 20749074. DOI: [10.5815/ijigsp.2012.01.04](https://doi.org/10.5815/ijigsp.2012.01.04). URL: <http://www.mecs-press.org/ijigsp/ijigsp-v4-n1/v4n1-4.html>.
- [13] M. J. Swain and D. H. Ballard. “Color indexing - Springer.” In: *International journal of computer vision* 7.1 (1991), pp. 11–32. ISSN: 15731405. DOI: [10.1007/BF00130487](https://doi.org/10.1007/BF00130487). arXiv: 0510473 [arXiv:cond-mat]. URL: <https://link.springer.com/content/pdf/10.1007%2FBF00130487.pdf>.
- [14] G. D. Brian V. Funt. “Color Constant Color Indexing.” In: ().
- [15] *HSL and HSV (Hue Saturation and Lightness / Value)*. [https://www5.in.tum.de/lehre/vorlesungen/graphik/info/csc/COL\\_23.htm](https://www5.in.tum.de/lehre/vorlesungen/graphik/info/csc/COL_23.htm).
- [16] *CMY and CMYK Color Spaces*. <https://msdn.microsoft.com/en-us/windows/desktop/dd371926>.
- [17] *1.2.2 The CIE Chromaticity Diagram*. URL: <https://www.mat.univie.ac.at/~kriegel/Skripten/CG/node9.html> (visited on 01/25/2019).
- [18] R. Wang. *Fourier Descriptor*. <http://fourier.eng.hmc.edu/e161/lectures/fd/node1.html>.
- [19] “The Hough Method for Curve Detection.” In: ().
- [20] *File:Hough transform diagram.svg - Wikipedia*. URL: <https://en.wikipedia.org/wiki/File:Hough%5Ctransform%5Cdiagram.svg> (visited on 03/15/2019).
- [21] H. F. Zhang and X. Zhang. “Shape recognition using a moment algorithm.” In: *2011 International Conference on Multimedia Technology, ICMT 2011* (2011), pp. 3226–3229. DOI: [10.1109/ICMT.2011.6002037](https://doi.org/10.1109/ICMT.2011.6002037).
- [22] Q. Yu, D. Mao, and J. Wang. “Deep Learning Based Food Recognition.” In: (2016). arXiv: [arXiv:1512.03385](https://arxiv.org/abs/1512.03385).
- [23] J. Bae and T. Song. “Image tracking algorithm using template matching and PSNF-m.” In: *International Journal of Control, Automation and Systems* 6.3 (2008), pp. 413–423. ISSN: 15986446.
- [24] S. R. Mangano. “A Genetic Algorithm White Paper.” In: *Man Machine Interfaces Inc* (1995), pp. 1–24.

- [25] A. Gidudu, G. Hulley, and T. Marwala. "Classification of Images Using Support Vector Machines." In: *Department of Electrical and Information Engineering, arXiv: 0709.3967v1, Cornell University, Library, 2007*. (2007), pp. 1–6. ISSN: 0016-8505, 1751-7656. DOI: 10.1680/geot.1979.29.1.1. arXiv: 0709.3967. URL: <http://arxiv.org/abs/0709.3967>.
- [26] *The Artificial Neural Networks handbook: Part 1 - Coinmonks - Medium*. URL: <https://medium.com/coinmonks/the-artificial-neural-networks-handbook-part-1-f9ceb0e376b4> (visited on 09/10/2019).
- [27] M. S. B. Maind and M. P. Wankar. "Research Paper on Basic of Artificial Neural Network." In: *International Journal on Recent and Innovation Trends in Computing and Communication* 2.1 (2014), pp. 96–100. ISSN: 2321-8169. DOI: 10.1109/Oceans-Spain.2011.6003625. URL: <http://www.ijritcc.org>.
- [28] J. E. Haugen and K. Kvaal. "Electronic nose and artificial neural network." In: *Meat Science* 49.SUPPL. 1 (1998). ISSN: 03091740. DOI: 10.1016/S0309-1740(98)90054-7.
- [29] *The Artificial Neural Networks Handbook: Part 4 - Jayesh Bapu Ahire - Medium*. URL: <https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e> (visited on 10/01/2019).
- [30] *Applied Deep Learning - Part 4: Convolutional Neural Networks*. URL: <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2> (visited on 11/23/2019).
- [31] *Convolutional Neural Networks (CNN): Step 1- Convolution Operation - Blogs SuperDataScience - Big Data | Analytics Careers | Mentors | Success*. URL: <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1-convolution-operation> (visited on 07/12/2019).
- [32] S. D.T.C.M. H. T.N. Krabach C. Staller and E. Fossum. "InGaAS detectors for miniature infrared instruments." In: 1874 (1993), pp. 214–223.
- [33] A. D. Mèrida, E. D. I. Geologica, U. D. L. Andes, U. M.R. I. Geophysique, P. Transcurrencia, and P. De. "Partición de la deformación interna a través de fallamiento transcurrente y pliegues de." In: (), pp. 1–16.
- [34] *1.3B: Monochromators - Chemistry LibreTexts*. URL: [https://chem.libretexts.org/Bookshelves/Analytical%20Chemistry/Supplemental%20Modules%20\(AAnalytical%20Chemistry\)/Analytical%20Sciences%20Digital%20Library/Active%20Learning/In%20Class%20Activities/Molecular%20and%20Atomic%20Spectroscopy/03%20Text%20%3A%20Molecular%20and%20Atomic%20Spectroscopy/1%20%3A%20General%20Background%20on%20Molecular%20Spectroscopy/1.3%20%3A%20Instrumental%20Setup%20of%20a%20Spectrophotometer/1.3B%20%3A%20Monochromators](https://chem.libretexts.org/Bookshelves/Analytical%20Chemistry/Supplemental%20Modules%20(AAnalytical%20Chemistry)/Analytical%20Sciences%20Digital%20Library/Active%20Learning/In%20Class%20Activities/Molecular%20and%20Atomic%20Spectroscopy/03%20Text%20%3A%20Molecular%20and%20Atomic%20Spectroscopy/1%20%3A%20General%20Background%20on%20Molecular%20Spectroscopy/1.3%20%3A%20Instrumental%20Setup%20of%20a%20Spectrophotometer/1.3B%20%3A%20Monochromators) (visited on 03/24/2019).

- [35] *How a Little Sensor is Making a Big Impact on the Global Cocoa Industry - AgFunderNews*. URL: <https://agfundernews.com/how-a-little-sensor-is-making-a-big-impact-on-the-global-cocoa-industry.html> (visited on 08/15/2019).
- [36] *TellSpec scanner identifies chemicals, allergens, nutrients in food. | Digital Trends*. URL: <https://www.digitaltrends.com/cool-tech/tellspec-food-scanner/> (visited on 08/15/2019).
- [37] E. Schaller, J. O. Bosset, and F. Escher. "Electronic noses' and their application to food." In: *LWT - Food Science and Technology* 31.4 (1998), pp. 305–316. ISSN: 00236438. DOI: 10.1006/fstl.1998.0376.
- [38] E. A. Baldwin, J. Bai, A. Plotto, and S. Dea. "Electronic noses and tongues: Applications for the food and pharmaceutical industries." In: *Sensors* 11.5 (2011), pp. 4744–4766. ISSN: 14248220. DOI: 10.3390/s110504744. arXiv: 031.
- [39] A. D. Wilson. "Review of Electronic-nose Technologies and Algorithms to Detect Hazardous Chemicals in the Environment." In: *Procedia Technology* 1 (2012), pp. 453–463. ISSN: 22120173. DOI: 10.1016/j.protcy.2012.02.101. URL: <http://linkinghub.elsevier.com/retrieve/pii/S2212017312001028>.
- [40] L. Capelli, S. Sironi, and R. Del Rosso. "Electronic noses for environmental monitoring applications." In: *Sensors (Switzerland)* 14.11 (2014), pp. 19979–20007. ISSN: 14248220. DOI: 10.3390/s141119979.
- [41] S. Ampuero and J. Bosset. "The electronic nose applied to dairy products: a review." In: *Sensors and Actuators B: Chemical* 94.1 (2003), pp. 1–12. ISSN: 0925-4005. DOI: 10.1016/S0925-4005(03)00321-6. URL: <https://www.sciencedirect.com/science/article/pii/S0925400503003216>.
- [42] *The World's Smallest Electronic Nose*. URL: <http://www.spacedaily.com/news/future-02a.html> (visited on 08/15/2019).
- [43] *Conducting Polymers*. <https://polymerdatabase.com/polymer%20physics/Polymer%20Conductivity.html>.
- [44] AZoSensors. *Piezoelectric Sensor Technology*. <https://www.azosensors.com/article.aspx?ArticleID=36>.
- [45] *ASTREE Electronic Tongue | Alpha MOS | CPhI Online*. URL: <https://www.cphi-online.com/astree-electronic-tongue-prod245372.html> (visited on 08/15/2019).
- [46] M. Podrazka, E. Bączyńska, M. Kundys, P. S. Jeleń, and E. W. Nery. "Electronic tongue-A tool for all tastes?" In: *Biosensors* 8.1 (2017), pp. 1–24. ISSN: 20796374. DOI: 10.3390/bios8010003.
- [47] Y. Tahara and K. Toko. "Electronic tongues-a review." In: *IEEE Sensors Journal* 13.8 (2013), pp. 3001–3011. ISSN: 1530437X. DOI: 10.1109/JSEN.2013.2263125.

- [48] T. Brosnan and D. W. Sun. “Improving quality inspection of food products by computer vision - A review.” In: *Journal of Food Engineering* 61.1 SPEC. (2004), pp. 3–16. ISSN: 02608774. DOI: 10.1016/S0260-8774(03)00183-3.
- [49] “Machine Vision Technology for the Food and Beverage Industry Imaging Is Our Passion Vision Technology – a Key To Maximising Profit and Quality.” In: ().
- [50] S. Gunasekaran. “Computer vision technology for food quality assurance.” In: *Trends in Food Science and Technology* 7.8 (1996), pp. 245–256. ISSN: 09242244. DOI: 10.1016/0924-2244(96)10028-5. arXiv: arXiv:1011.1669v3.
- [51] A. Rafiq, H. A. Makroo, P. Sachdeva, and S. Sharma. “Application of Computer Vision System in Food Processing-A Review.” In: *Journal of Engineering Research and Applications* 3.6 (2013), pp. 1197–1205.
- [52] M. McFarland. *I spent 53 minutes in Amazon Go and saw the future of retail.* <https://edition.cnn.com/2018/10/03/tech/amazon-go/index.html>.
- [53] *New Toshiba image-recognition processors for ADAS; night-time pedestrian detection and 3D reconstruction - Green Car Congress.* URL: <https://www.greencarcongress.com/2014/11/20141114-toshibaimage.html> (visited on 08/20/2019).
- [54] Z.-Y. Ming, J. Chen, Y. Cao, C. Forde, C.-W. Ngo, and T. S. Chua. “Food Photo Recognition for Dietary Tracking: System and Experiment.” In: February (2018), pp. 129–141. DOI: 10.1007/978-3-319-73600-6\_12. URL: [http://link.springer.com/10.1007/978-3-319-73600-6\\_{\\\_}12](http://link.springer.com/10.1007/978-3-319-73600-6_{\_}12).
- [55] *Calorie Mama Food AI - Food Image Recognition and Calorie Counter using Deep Learning.* URL: <https://www.caloriemama.ai/> (visited on 08/21/2019).
- [56] *What Is a Calorie and Why Do We Count Them?* URL: <https://www.verywellfit.com/what-is-a-calorie-and-why-should-i-care-3496238> (visited on 02/16/2020).
- [57] *Fat | Eat For Health.* URL: <https://www.eatforhealth.gov.au/food-essentials/fat-salt-sugars-and-alcohol/fat> (visited on 02/16/2020).
- [58] *What are proteins and what do they do? - Genetics Home Reference - NIH.* URL: <https://ghr.nlm.nih.gov/primer/howgeneswork/protein> (visited on 02/16/2020).
- [59] *Good Carbs, Bad Carbs — How to Make the Right Choices.* URL: <https://www.healthline.com/nutrition/good-carbs-bad-carbs> (visited on 02/16/2020).
- [60] M. R. Kasaai. “Use of Water Properties in Food Technology: A Global View.” In: *International Journal of Food Properties* 17.5 (2014), pp. 1034–1054. DOI: 10.1080/10942912.2011.650339. URL: <https://doi.org/10.1080/10942912.2011.650339>.
- [61] *What is Python? Executive Summary | Python.org.* URL: <https://www.python.org/doc/essays/blurb/> (visited on 02/23/2020).

## BIBLIOGRAPHY

---

- [62] *TensorFlow*. URL: <https://www.tensorflow.org/> (visited on 02/23/2020).
- [63] *What is Keras? The deep neural network API explained | InfoWorld*. URL: <https://www.infoworld.com/article/3336192/what-is-keras-the-deep-neural-network-api-explained.html> (visited on 02/23/2020).
- [64] *Welcome To Colaboratory - Colaboratory*. URL: <https://colab.research.google.com/notebooks/intro.ipynb> (visited on 11/15/2020).
- [65] Moses and J. Olafenwa. *ImageAI, an open source python library built to empower developers to build applications and systems with self-contained Computer Vision capabilities*. 2018-. URL: <https://github.com/OlafenwaMoses/ImageAI>.
- [66] *Food Images (Food-101) | Kaggle*. URL: <https://www.kaggle.com/kmader/food41> (visited on 03/03/2020).
- [67] *Fruits 360 | Kaggle*. URL: <https://www.kaggle.com/moltean/fruits> (visited on 03/03/2020).
- [68] Israr Hussain, Q. He, Z. Chen, and W. Xie. "Fruit Recognition dataset." In: (2018). DOI: 10.5281/ZENODO.1310165.
- [69] *GitHub - tzutalin/labelImg: LabelImg is a graphical image annotation tool and label object bounding boxes in images*. URL: <https://github.com/tzutalin/labelImg> (visited on 03/03/2020).
- [70] H. Mikami, H. Suganuma, P. U-chupala, Y. Tanaka, and Y. Kageyama. *ImageNet/ResNet-50 Training in 224 Seconds*. Tech. rep. URL: <https://abci.ai/>.
- [71] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens. *Rethinking the Inception Architecture for Computer Vision*. Tech. rep. arXiv: 1512.00567v3.
- [72] *SCiO - The World's First Pocket Sized Molecular Sensor*. URL: <https://www.consumerphysics.com/> (visited on 03/08/2020).
- [73] *Brix as a Metric of Fruit Maturity | Tools for Applied Food Science | felixinstruments.com*. URL: <https://felixinstruments.com/blog/brix-as-a-metric-of-fruit-maturity/> (visited on 03/09/2020).
- [74] *FrontPage - Raspbian*. URL: <https://www.raspbian.org/> (visited on 02/23/2020).
- [75] *Complete All-In-One Product Development Platform | Onshape*. URL: <https://www.onshape.com/en/platform> (visited on 11/28/2020).
- [76] *Blocks One MKII*. URL: <http://blockstec.com/onemkii.html> (visited on 11/28/2020).
- [77] *Apples 101: Nutrition Facts and Health Benefits*. URL: <https://www.healthline.com/nutrition/foods/apples{\#}nutrients> (visited on 10/27/2020).
- [78] L. J. Wilcox, N. M. Borradaile, and M. W. Huff. *Antiatherogenic properties of naringenin, a citrus flavonoid*. 1999. DOI: 10.1111/j.1527-3466.1999.tb00011.x.



- [79] G. A. Spanos and R. E. Wrolstad. "Phenolics of apple, pear, and white grape juices and their changes with processing and storage. A review." In: *Journal of Agricultural and Food Chemistry* 40.9 (1992), pp. 1478–1487. ISSN: 0021-8561. DOI: [10.1021/jf00021a002](https://doi.org/10.1021/jf00021a002). URL: <https://pubs.acs.org/doi/abs/10.1021/jf00021a002>.
- [80] N. Kozukue and M. Friedman. "Tomatine, chlorophyll,  $\beta$ -carotene and lycopene content in tomatoes during growth and maturation." In: *Journal of the Science of Food and Agriculture* 83.3 (2003), pp. 195–200. ISSN: 00225142. DOI: [10.1002/jsfa.1292](https://doi.org/10.1002/jsfa.1292).

