Automated Industrial Inspection Workbench for Human Machine Interface (HMI) Consoles

Alison Shigueo Nakashima

Thesis Presented to the School of Technology and Management of Polytechnic Institute of Bragança to the Fulfillment of the Requirements for the Master of Science Degree in Industrial Engineering (Electronic Engineering branch).

Supervised by:

Prof. PhD José Luís Souza de Magalhães Lima

Prof. PhD Paulo Jorge Pinto Leitão

Prof. PhD Gilson Junior Schiavon

Bragança

2018

INSTITUTO POLITÉCNICO DE BRAGANÇA Escola Superior de Tecnologia e de Gestão

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Automated Industrial Inspection Workbench for Human Machine Interface (HMI) Consoles

Alison Shigueo Nakashima

Thesis Presented to the School of Technology and Management of Polytechnic Institute of Bragança to the Fulfillment of the Requirements for the Master of Science Degree in Industrial Engineering (Electronic Engineering branch).

Supervised by:

Prof. PhD José Luís Souza de Magalhães Lima

Prof. PhD Paulo Jorge Pinto Leitão

Prof. PhD Gilson Junior Schiavon

Bragança

2018

Dedication

This work is dedicated to my parents Gerson Nakashima and Marilene Nakashima, that makes the realization of this possible. And also, my grandparents Vicente Biazetto and Angelina Biazetto.

Acknowledgment

First of all, thank God, my parents Gerson and Marilene, my family, my girlfriend Leidyana and my friends that somehow make something to support me during this and others process during my life. Be into another country studying was a dream and, it's realization was only possible with your help.

Be far from home was difficult but is an experience that makes me learn many things. The miss of my family, friends and my girlfriend that always heard me, in bad or good moments, I LOVE YOU. To everyone around me, make me better understand the meaning of them in my life.

For those, strange ones that I know in the IPB, main the guys that are into the LSE. These guys, become like brothers to me. The same thing is to the people from my own university that I knew here. You all make my days better. I hope that our friendship never ends, and wish the best of everything for you guys.

To the IPB, mainly the professors and members from LCAR that became big friends. My supervisors José Lima, Paulo Leitão and my co-supervisor Gilson Schiavon for the support and help given during this period. And to thanks the UTFPR, principally the coordination from my course. Also, the professors that participate in this and other important parts from my study, these are Flávio Rossini, Reginaldo Nunes de Souza, Márcio Rodrigues da Cunha. In addition, the professor from IPB Vicente Leite for reception and friendliness into the LSE and Diego Bertolini professor from UTFPR-CM for the support given. I hope you all continue to make your job of teaching as good as possible.

"With good people right beside you, goals and objectives will be always attained."

Abstract

The actual moment of the industrial production is changing the way of production. Now the systems are adaptable to produce different items in the same production line with a very reduced time to setup the systems. In the same way, the quality control systems must be more adaptable and intelligent possible. The present work propose the creation of intelligent and adaptable inspection cell to inspect Human Machine Interface (HMI) consoles of different types. This cell is composed by an image acquisition system with controlled illumination, a force sensor installed on the robot tool to verify the buttons' functionality. The force tests are processed and classified using decision three, k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM) classification method. Besides, the Thin-Film Transistor (TFT) display uses Normalized Cross-Correlation (NCC) and Correlation Coefficients (CC) to check the display's regions. To Liquid Cristal Display (LCD) is used the same method and also be used a Neural Network Classification (NNC). In the experimental tests, four different types of consoles prototypes are tested, one of them has a TFT display and buttons, others two have only buttons and one has only a LCD display. In the inspection workbench is created, all the hardware necessary to execute the inspection was installed successfully. Moreover, the inspection methods obtained a precision higher than 90% to the buttons and display inspection.

Keywords: Adaptive industrial inspection, TFT and LCD display inspection, Adaptive workbench, Force sensor, Machine learning.

Resumo

O momento atual produção industrial está mudando a forma de produzir. Agora os sistemas são adaptativos para produzir diferentes itens na mesma linha de produção com tempo de mudança ou customização muito reduzido. No mesmo sentido, os sistemas de controle de qualidade devem ser o mais adaptativo e inteligente possível. O presente trabalho propõe o desenvolvimento de célula de inspeção inteligente e adaptativa para inspectionar consoles de *Human Machine Interface* (HMI) de diferentes tipos. Esta célula é composta por um sistema de aquisição de imagem com iluminação controlada, um sensor de força instalado na ferramenta de um manipulador para verificar a funcionalidade dos botões. Os testes de força são processados e classificados usando métodos de aprendizagem de máquina, nomeadamente, *decision tree*, *k-Nearest Neighbor* (k-NN) and *Support Vector Machine* (SVM). Além disso,é utilizada a *Nomalized Cross-Correlation* e *Correlation Coefficients* para checar as regiões do display do tipo *Thin-Film Transistor* (TFT). Em displays do tipo Cristal Líquido (LCD) é utilizado o mesmo metodo, sendo também utilizada a classificação usando Rede Neurais. Nos testes experimentais, foram testados quatro tipos de consoles HMI, sendo que um deles possui um display de TFT e botões, outros dois possuem somente botões e um tem somente um display de LCD. Na bancada de inspeção criada, foi devidamente instalado todo o *hardware* necessário para execução da inspeção. Além do mais, obteve-se precisão acima de 90% para os métodos de inspeção dos botões e displays.

Palavras-chave: Inspeção industrial adaptativa, Inspeção de displays de TFT and LCD, Bancada de trabalho adaptativa, Sensor de força, Aprendizagem de máquina.

Contents

List of Figures

Acronyms

3D Three Dimensional.

AI Artificial Intelligence.

CAD Computer-Aided Design.

CAM Computer-Aided Manufacturing.

CC Correlation Coefficients.

CPPS Cyber-Physical Production System.

CPS Cyber-Physical System.

GND Ground.

GUI Graphical User Interface.

HMI Human Machine Interface.

ISO International Organization for Standardization.

k-NN k-Nearest Neighbors.

LCD Liquid Crystal Display.

LED Light Emitting Diode.

NCC Normalized Cross-Correlation.

NN Neural Network.

NNC Neural Network Classification.

RGB Red, Green and Blue.

SVM Support Vector Machine.

TCP/IP Transmission Control Protocol – Internet Protocol.

TFT Thin-Film Transistor.

USB Universal Serial Port.

Chapter 1

Introduction

The current trend of the industry, called by 4^{th} Industrial Revolution, is introducing a new way of production, providing intelligence and connection to Cyber-Physical System (CPS) [1]. Using the Ethernet and other modes of connection different types of system are connected providing information to each other creating smart factories. In this context, in consequence of the increase of the production versatility, the mass production must decrease to avoid storage of production, thus, increasing the production value [2].

At the same time, with this connectivity, the use of collaborative robots in all levels of the production process is increasing globally and, consequently, security systems and rules are being studied to avoid accidents with humans and also damage the robots during their utilization [3]. Beyond that, increasing the flexibility of production systems, the collaborative works between robots and humans needs to complement each other, as seen that, both have their own weakness [4].

1.1 Motivation and Objectives

The project presented in this dissertation is focused on the creation of a standard workbench, introducing the 4.0 Industry in quality inspection of consoles (buttons, TFT and LCD displays), making an automated and collaborative work part of the process. Furthermore, the insertion of the technologies that are part from this new industry increase systems adaptability, the products' quality, the efficiency from process and simplify the products' traceability.

Quality inspection is an existent task in all production factories, besides, it also proves that all the process made before are doing well and according to what was foreseen. When this inspection is done, the products have a higher commercial value, meaning an increase of the process phases' value too.

To realize the inspection, is used a collaborative robot UR3 (Universal Robots), the Robotiq's Force Torque Sensor FT300, an industrial camera Mako G-125B, an Arduino Uno, a Light Emitting Diode (LED) illumination and a workbench to install the system. All these will be communicating by Ethernet and Universal Serial Port (USB), to achieve a simple process and to enable the expansion of the system and communication with other external devices. All these tools will work together to make the inspection of Human Machine Interface (HMI) consoles and the resulting system will be able to work online and also has an user interface that will provide the monitoring of the quality inspection process and the results.

This work made study and development of standard workbench to inspect four different types HMI consoles, inspecting the buttons and display using artificial vision and Artificial Intelligence (AI).

1.2 Work Organization

This work is distributed in six chapters with the present chapter as an introduction describing the actual scenario of the industrial production, the motivation of the work and the project's characteristics.

The second chapter is the state of the art related with industrial production and inspection, methods and procedures used in other projects to realize the identification of objects or production quality control.

The third chapter describes the hardware and software architecture used to accomplish the project and also, the communication applied to connect all the necessary devices.

The fourth chapter is a description of the programming methods and algorithms used to realize the inspection of the consoles. How different methods to classify the buttons results works, how the image acquired to make the inspection of the displays are processed to find problems into regions and, in addition, a description of Graphical User Interface (GUI) application.

The fifth chapter shows the result of each phase performed, and the output of the phases are grouped to create the GUI application.

Finally, the sixth chapter is a conclusion about the project and, also, a discussion about future works in this area.

Chapter 2

State of the Art

This chapter makes a review of the research that had been done until now. The research made, makes a survey of the historical evolution from the industrial machines until the actual state. Furthermore, are presented the changes in the way of production, through the collaborative work.

The adaptability of the system is also an important section, owing to the new way of industrial production. Many studies are being done in this area to improve its versatility and demonstrate some new approach at all levels of production.

2.1 The Machines Evolution

Mckerrow P. J. [5], defined "*robot* like a machine that can be programmed to do a variety of tasks, in the same way that a computer is an electronic circuit that can be programmed to do a variety of functions". Therefore, not all automated system is a robot, because for this it must be programmable and has connection with some problem or practical task [5].

However, the automation evolution during the history comes from the beginning from the First Industrial Revolution occurred approximately at half from the century XVIII [6]. This first revolution starts the mechanization of the production using water and vapour to move the machines. The second one utilizes the electric energy to supply those machines and create the mass production. In a third moment, were used electronic components and information technology to automatize the production.

The evolution from technology due to this succession of events made with “the use of industrial robots became identifiable from the 1960s, along with computer-aided design Computer-Aided Design (CAD) systems and Computer-Aided Manufacturing (CAM) systems” [7]. For, in 1959 George Devol e Joseph Engelberger developed the first industrial robot, next in 1961 the General Motors installed the first industrial robot in a production line [8].

It is perceived that the growing of the strategic role of technology, together with its development and applications that tend to operate an increasing number of industries and activities. From the 60’s, the evolution of robotics becomes very evident as well as the communication media, computer systems and their processing power. The introduction of process systems CAD and CAM characterized as a trend and, raised the industrial automation to another level [9], [7].

With the expansion and development of robotics, the robots’ generation could be better classified. First, the playback generation is composed by executor robots that only repeat recorded instructions. On the following generation, the robots have ability to make decisions, according to sensors’ response in a closed loop system and after they were improved to a vision system, in other words, they have image process actuating on the system [10].

The third generation, is composed by the robots with an adaptive control. These controllers can automatically reprogram their actions based on the sensors’ signal. Currently the AI uses programming techniques to make the robots solve the problems and make their own decisions [10].

Recently, started the Fourth Industrial Revolution also called “Industry 4.0”, is using as a foundation the technologies implemented in the third revolution, is applying the digital revolution in these equipments. It has been occurring since mid-century and, is characterized by a fusion of physical, digital and biological technologies [11].

In consequence of this revolution, some countries as Germany and United States, have the factory floor's decisions taken only by machines, that is, digital informations are created and then sent to a series of manipulators that perform all the fabrication process, quality inspection and correction of the errors. At this point the communication between the robots has been setup and the industry become practically independent from the human action. In addition, it is possible to produce different models of a product, without having to stop for a reprogramming of the production line [12].

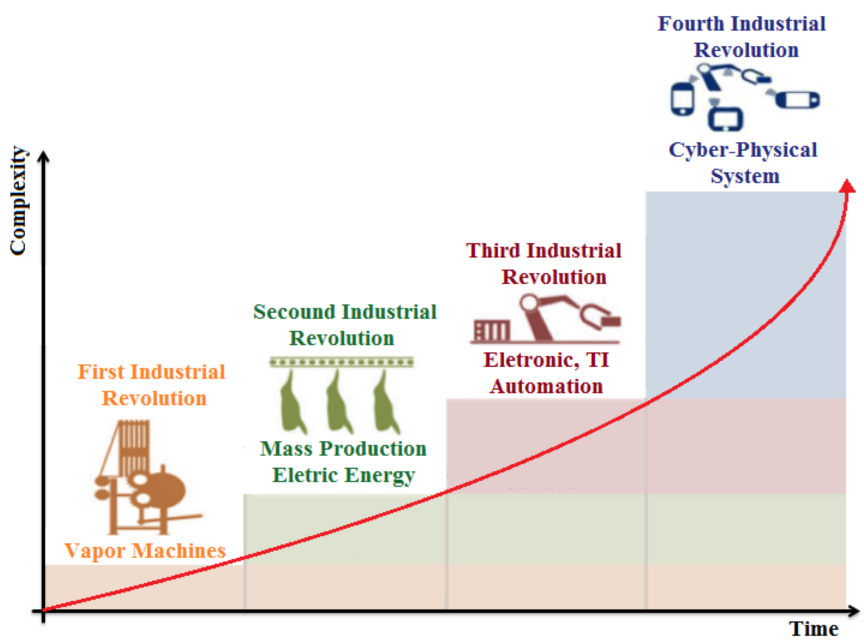


Figure 2.1: Industrial Machines Evolution (adapted from [13]).

The advancement of digitalization has the potential to revolutionize our daily lives, offering solutions to important challenges like the urban mobility, with the adoption of technologies linked to the concept of “smart cities”, energy efficiency (implementation of smart grids), solutions in the health area and industrial development [14]. When compared to the previous industrial revolution, the Industry 4.0 is evolving more exponentially than in linear rhythm, as can be seen on the Figure 2.1. Withal, it is affecting almost all sectors of industries in all countries [11].

2.2 Collaborative Work

Currently, the number of multipurpose robots developed by providers in Industry 4.0 and used in industrial production has almost doubled since 2004. So, smart robots do not go only substitute the human work in specific areas, now it can be said that humans and robots will work together, on interconnected tasks using sensors and with human-machine interfaces that can be remotely controlled. In general, this evolution brings five main economic effects related with: consumers' expectation, products enhancement, collaborative innovation and organization form of production and trade [14], [15].

Nowadays, one of the main aspects of this revolution is the collaborative work between human-robot, these robots were defined by collaborative robots or "cobots". They are projected to work alongside humans inside a shared space without a physical and conventional protection system such as safety case or light curtains. In addition, the dissemination of this technology is occurring in all sectors of production and work areas, whether in small tasks in a laboratory or in automotive industry [16].

In consequence of these advances, the International Organization for Standardization International Organization for Standardization (ISO) created the ISO/TS 15066:2016, that specifies the safety requirements to collaborative industrial robots and its environment of work, and complements some more requirements and orientations about the "cobots" operation [17].

There are many benefits of these robots application, but to make it happen are necessary some advanced solutions considering many aspects, because they affect the organization and work quality. There are many research areas about the different types of cooperative robots (Figure 2.2) and the way that these robots and their work are being applied increasingly due to the increasing of the fabrication process complexity, creating "multirobotic" systems [18].

To setup the cooperation between robots is necessary perform a communication to realize the connection with the controllers and the online platform. To the cooperation with the workspace is necessary actuators equipment as speed, force, position and vision

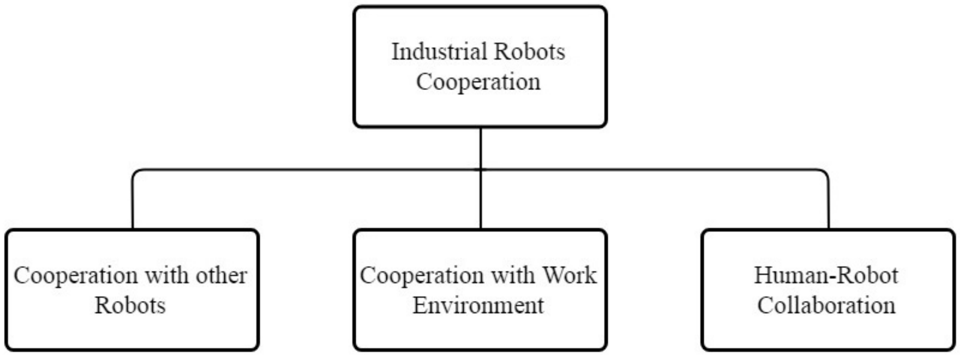


Figure 2.2: Industrial Robots Cooperation.

sensors, connected with the controller to coordinate the work without they affect each other. Dealing with cobots here are some existing examples:

- ABB Yumi: a collaborative robot with two arms used to assembly of small parts (Figure 2.3(a);
- Schneider Electric Baxter: human-friendly robot that helps to increase the production flexibility (Figure 2.3(b));
- Universal Robots UR3: a flexible robot for light tasks at automated assembly stations (Figure 2.3(c);

All of these are considered special solutions to collaborative work between human-machine [18].

The communication interfaces for these robots are the link between the collaborators. These include:

- Digital input and output (24V);
- Analogic input and output (general 10V);
- Serial port (RS232, RS485, RS422);
- Buses (ASI, Profibus, FMS, ProfibusDP, Interbus S, CAN-Bus);

(a) ABB, Yumi [19]. (b) Schneider Electric, Baxter [20]. (c) Universal Robots, UR3 [21].

Figure 2.3: Collaborative robots examples (a) (b) (c).

- Network (Ethernet, Arcnet).

Through these means of communication the "cobots" communicate and transmit information to devices over a network in a standard way under specific industrial conditions [22].

Collaborative robots can work together with the users to make their job even when there is not connection with any network but, when a communication is setup, the collaborative work can be done using electronic devices (smartphone, laptop, tablets) commanded by a software and a standard data transmission channel as mediator. Therefore, the cobots communicate and transmit information to devices and network in a specific industrial condition, thus making the collaborative work as simple as possible to the user (see Figure 2.4) [18], [22].

To improve the robots work, different types of sensors and cameras are used with the aim of monitoring the work and enhancement of the precision and detection of the objects and surfaces. The vision system is responsible to manage the process according to each necessity [24].

To make the collaborative work possible and integrate as many systems as necessary the communication system is also evaluated, meaning that everything is getting smart and connected with different types of communication. So, in the 4th Industrial Revolution is composed by different types of connection, that are interconnected in a system and

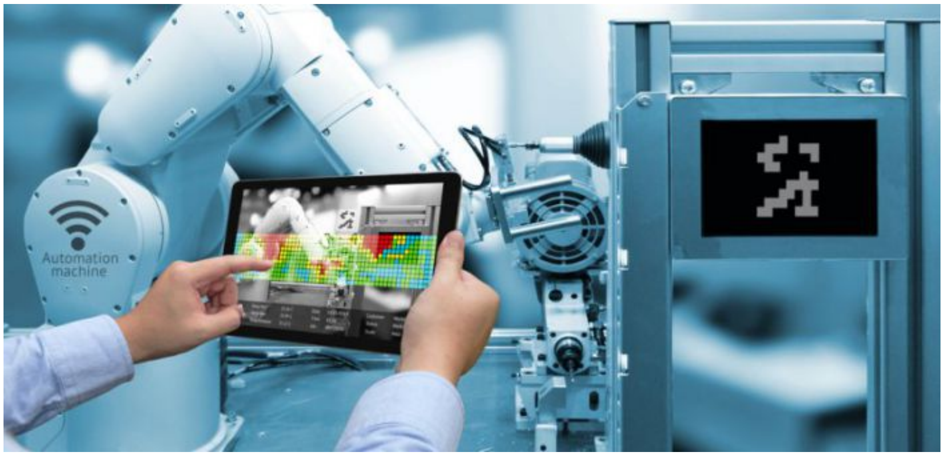


Figure 2.4: Interface with Collaborative Robots [23].

accessed for different users providing new kinds of production [25].

2.3 Adaptive System

The Industry 4.0 is supported by the creation of CPS that comprises a network of distributed entities interpreting physical and cyber counterparts. The Cyber-Physical Production System (CPPS) uses the internet and other methods to generate communication among different systems, to approximate the industrial production to the future, where the production is characterized to the individualization and flexibility of the production, with high-quality and increasing the process value [26]. But with this evolution, the quality manufacturing technologies need to be capable to improve their performance as fast as possible, to maintain the relation between flexible production and its quality [27].

This real-time communication and data transmission together with sensors and camera make this system and robots capable to execute different tasks and correct possible errors, in other words, creating an adaptive system. It differs from a common closed loop control because the parameters that are being controlled can varies over the time, so a mechanism for auto-adjusting it, is part of the closed-loop system [28].

Others adaptive system uses images obtained by the cameras, to be processed and depending on their characteristics (background color, luminosity, brightness, focus) resulting in a response to the algorithm applied, consequently, a signal to the control system [24].

2.4 Quality Inspection

Since the Egyptians, passing through many others civilizations the control quality can be observed, but up to 1900 it gets more evident and over the years getting more importance. Posteriorly, quality was defined as a “conformance to requirements or specifications” [29]. This task is also part of every production concerned [30], and also necessary in all types of industries. According to the industry type, a specific inspection method is applied to check the production quality.

The quality settings must balance the relation between the costs and the product’s quality. The process of search for nonconformities or errors must avoid poor-quality products, identification of how much the imperfections affect the product to be classified as a nonconforming and also, collect data from inspections to provide as soon as possible corrections in the manufacturing process before a large scale of poor-quality production [31].

Many types of vision system had been applied on industrial quality inspection and it is typically composed by a computer to process the images, a camera to acquire the images and an illumination system, to facilitate the image process and classification. An industrial systems need to be adaptive, and consequently very reliable. For this reason, machine learning can be applied to the image analyses [32] and adaptive motion to the robot system are being applied. In addition, this system is also applied in inspection of TFT liquid crystal displays (LCD), through the reflection of light [33]. To perform these display inspections the pattern recognition needs to be done to differ some specific defect from other defects [34].

Many other types of sensors have been applied to quality inspection systems, increasing their flexibility and inherit intuitive configuration capabilities [35]. The type of tool, even the sensor that is applied on quality system changes according to the quality characteristics that are being evaluated such as structural elements (length and weight of an object) and sensory characteristics (taste of the food and the smell), and after it will be analysed by the system like a specific variable [29].

Force sensors are applied into different types of industrial process and also for medical use. The integration these sensors position, speed and acceleration control possibilities the utilization of robots to extreme precision surgeries. Along with the robot, force sensors are applied to the system tool to provide a highest precision [36] and also, the sensors have been applied on products test. They can be used to test different types of products like buttons and gauntlets as many times as the user needs (in cases of durability tests) [37].

2.5 Machine Learning

The applications of learning machine systems are being extended to many research areas and industrial application, where machine vision systems are widely utilized. These image system based on a camera and an illumination system, they can perform inspection of external features of different types of fruits and vegetables, even it, internal quality can be checked using different types of lasers [38].

To create the classification methods is necessary a big amount of samples to process the results, extracting the characteristics and to create a pattern of recognition, all of this to create a method that is perceptible to the machine. The sequence of process done to create the machine learning method is on Figure 2.5. The feature extraction demands a big knowledge of the problematic, to avoid noises and overfitting the system, making it ineffective and unable to identify different types of classes [39].

The machine learning can be created by a supervised learning or unsupervised learning. Supervised learning uses category label or cost for each pattern to create a classification method, whereas the unsupervised learning do not use category label defined, this way, the system creates clusters with different cost functions [39].

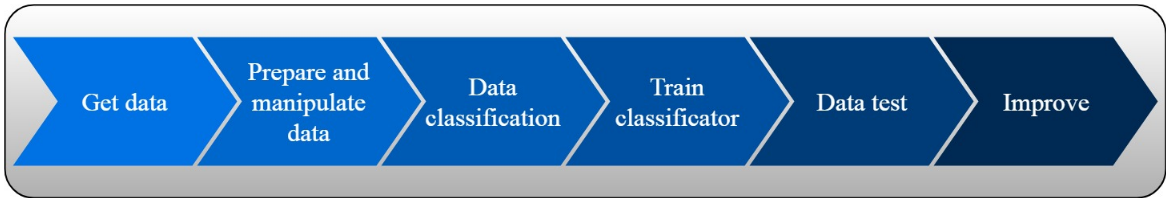


Figure 2.5: Supervised machine learning diagram.

Furthermore the machine learning has a decision method to create the classifier, as Decision Tree, k-NN, Support Vector Machine (SVM) and others. Decision Trees are based in a multistage decision making, the main idea is to divide a complex decision into a union of many simple decision, always finding a solution as similar as to the complex solution [40]. The k-NN decision rule uses the "k" nearest neighbors previous classified to set the sample classification. However, for many number of categories, the number of errors increase bounded by twice the Bayes probability of error [41]. And the basic idea of SVM is to find and create a hyperplane that separates the classes with a margin as big as possible [42].

Another algorithm to samples' classify and recognize is called Neural Network (NN), this one creates results using a process similar to the biological brain. Having the application of the inputs, the NN uses a large number of simple processing elements (neurons) and connect them to each other, leaving the signals from input to the better output. In [43], Accianiani *et al.* applied this method and others to classify solder joints and, as result, the NN got the best results.

The NN algorithm can also be applied into food industry as Shimizu *et al* [44] made. In this paper, the quality inspection of eggs is done based on image classification and extracting its characteristics to be processed by a Convolutional Neural Network. Another application was done by Li *et al.* [45] using NN Classification to inspect TFT LCDs, and the inspection gets great results.

Chapter 3

Study Cases and Inspection System Architecture

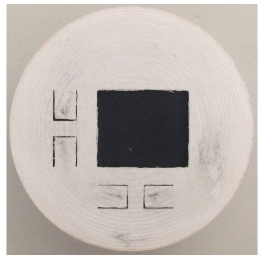
In this chapter, the hardware and software that compose the project are described together with their functions and tools needed to create the inspection system. Also, an overview about the inspection system scenarios.

3.1 Inspection System Scenarios

To the inspection process four scenarios are proposed, contemplating four types of consoles to be inspected, as can be seen on Figure 3.1. The consoles type II and III have only buttons to be tested, the console type IV has only a LCD display and the console type I has buttons and a TFT display. So, for each of the cases a specific sequence of steps are realized but to contemplate all the inspection procedures a flowchart was created (see Figure B.1 and is in the annex B).

To start the inspection process, the console needs to be placed on the inspection area and the start button pressed. On sequence a message a status message is showed and the connection with the robot is tested. With the connection established the system starts to adapt the illumination system to get the optimal illumination for the console type identification. After the console identified, according to its characteristics a signal is sent

(a) Console type I.



(b) Console type II.

(c) Console type III.

(d) Console type IV.

Figure 3.1: Console types to be inspected (a) (b) (c) (d).

to the UR3 to move to the next step of the process, if the console has buttons, so they are tested first. In case of the console does not has buttons the inspection sequence moves to the display inspection, on sequence, the system waits for the UR3 signal to confirm that the robot is in position to make the inspection of the display. According to the type of console a specific display inspection method is realized and, in to finish the process the results are showed on the user interface.

To support the robot installation and also improve the system's utilization, the creation of a workbench to the project was done. In this workbench the UR3 can be placed of different ways to the system became able to be used to different purposes (Figure 3.2).

Dealing with collaborative area of work and the need to change the console that will be inspected, some security instruments are installed in the workbench to improve the security, as an industrial barrier sensor, a status lights signal and emergency stop button (see Figure 3.3). These instruments are connected with the UR3's control box. Inside the control box, a panel to connect security instruments and any input and outputs from the robot as can be seen on Figure 3.4.

Other security method applied to the system is the limitation of force and speed,

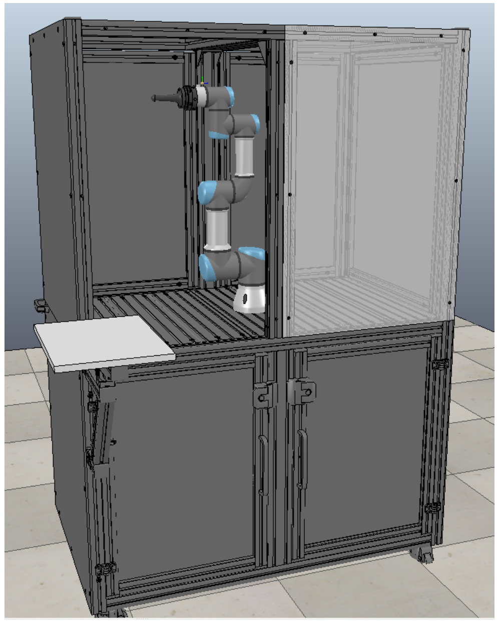


Figure 3.2: Workbench and UR3 installation.

(a) Industrial barrier sensor [46].

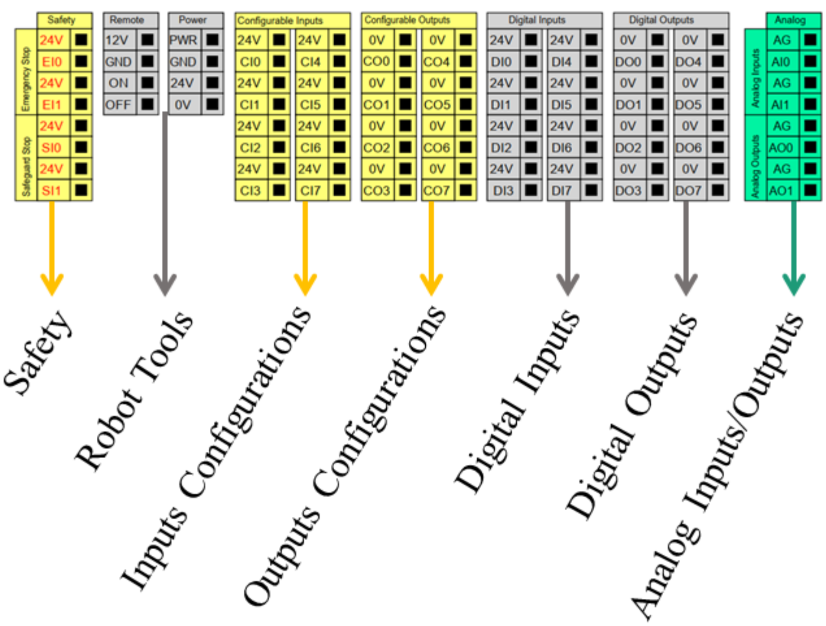
(b) Status light.

(c) Emergency stop button.

Figure 3.3: UR3 security external equipments (a) (b)(c).

it is done into the UR3's teach pendant console. This console is responsible for the programming of the robot and, also implements the control peripheral devices, the inputs and outputs.

(a) Control box.



(b) Electrical interface inside control box [47].

Figure 3.4: UR3's control box and electrical interface (a) (b).

Inside the control box there is a panel as showed on Figure 3.4(b), in this panel there are some colorful regions, each of this regions have an specif utility. There are two regions in yellow, the region in the middle is used to connect configurable safety devices and the other is to dedicated safety signals. The gray region deals with digital signals and the green one is for analog signals [47].

3.2 Hardware and Software Description

To build this system, the first step is to develop its architecture, this way, the system can be studied and analyzed in sections, facilitating the organization of the project's steps (Figure 3.5). The system is not only composed by hardware, to join all the system's parts and use their functions it's also necessary has a software architecture and application procedure, that is, the software used to construct the system functions and its working sequence as showed in Figure 3.6. The main program of the project is Matlab, based on it all the other functions are opened and closed. Thus, the system is composed by an image acquisition, an illumination control, image process, the UR3 inspection and an end analyses from the force sensor also made by Matlab. Also, the program send data to arduino and UR3 to control and indicates the next step of the inspection.

The UR3 programming can be done online and offline, each one of these have your own characteristics. In this work the programming will be done online, since the robot is able to be programmed, this way all the position for each type of consoles need to be identified, to the robot makes the analyses according with Matlab requesting.

With the robot installed in the workbench, the image acquisition is made by a camera. This camera is coupled to the UR3 through a support that was created in SolidWorks and printed in a Three Dimensional (3D) printer and, it is installed close to the robot's actuator. In the support is also coupled the adaptive illumination system that will operate to adjust the illumination to reach the setpoint. Also placed in the workbench there are another supports that were created to place the consoles on a standard position and, according to each kind of consoles a specific support was created as showed on Figure 3.7.

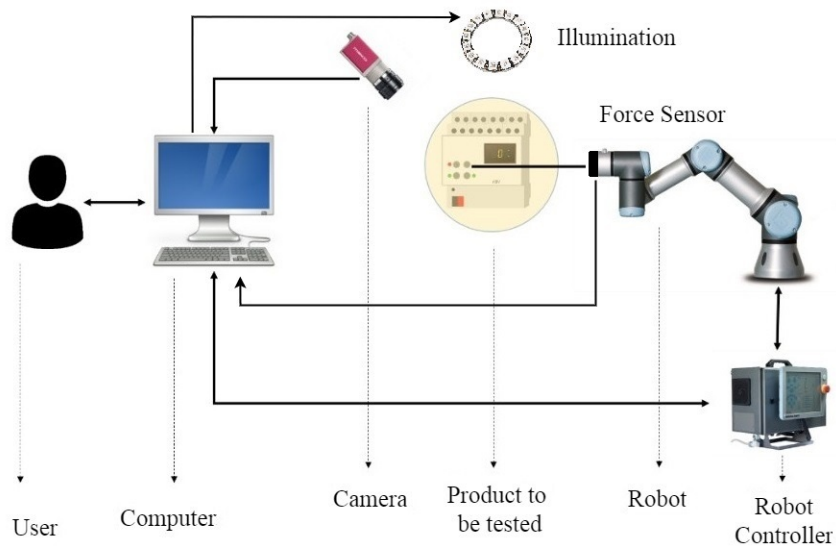


Figure 3.5: Hardware architecture.

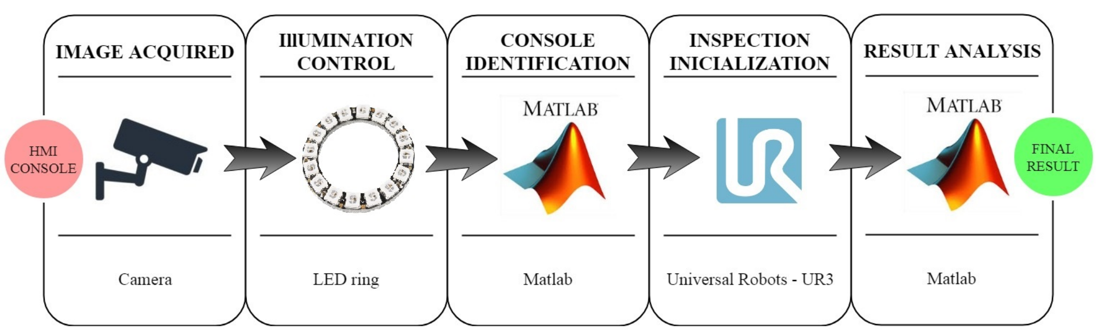
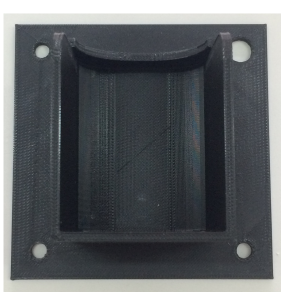
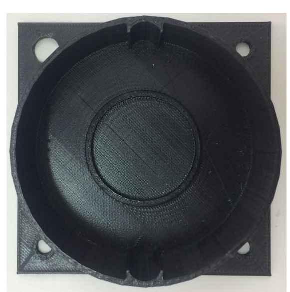


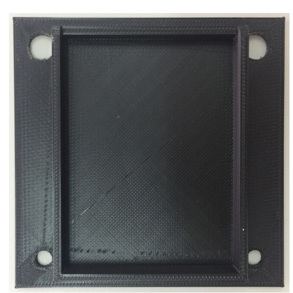
Figure 3.6: General application working sequence.



(a) Support for console type I.



(b) Support for console type II and III.



(c) Support for console type IV.

Figure 3.7: Support for console types (a) (b) (c).

These supports improve the image quality and consequently, the results of the inspection and there is a specific support for each type of the consoles inspected. This way, not only the results are improved but, also the programming of the UR3 becomes easier because using this supports the buttons and displays stay always at the same position.

3.2.1 Image Acquisition and Analyses

To acquire the images to be processed, an industrial camera Mako G-125b is inserted in the project (Figure 3.8). This is a monochromatic camera and has a resolution of 1296x964 and, it has an easily installation what makes it ideal to the project.

Figure 3.8: Industrial Camera Mako G-125 [48].

The image analyzes is made based on the calculation of the mean of brightness on the grayscale image, that is also represented numerically by an one-dimensional array with *MxN* elements. These elements composes the grayscale image and their value are between 0 and 255 (0 represents black color and 255 is the white one).

The image analyzes works into a loop until it gets the setpoint value. According to the mean of the image brightness the light can be increased or decreased until the setpoint is reached. The analyzes is made using an histogram, it shows the brightness frequency of an image, in grayscale, that is also represented numerically by an one-dimensional array with n elements. These elements contain a probabilistic value of brightness and is independent of its position on image. An edge detector is also applied to improve the identification of the consoles [49].

After capture the image with optimal illumination, a binarization and filling small holes on the binary image is made to avoid noises before identification of the regions. The identification of the regions is the process responsible for extract the image features as:

- Perimeter;
- Area;
- Eccentricity;

These proprieties are used to identify and differentiate the consoles type. Perimeter refers the pixel value that count the number of pixel that complain the perimeter of the figure, the area counts the number of pixel that compose the image and the eccentricity returns a value between 1 and 0, this is calculated base on the distance of the focus and the major axis length. The 0 is a perfect circle identification and 1 represents a line [50]. Getting all these properties the consoles types can be easy identified, since the camera position is aways the same.

3.2.2 Adaptive Illumination

The Adaptive illumination is applied to improve image quality and system's functions. Using the illumination system, the surface of the consoles can also be inspected, withal if the material of the consoles change or another types of console be added to the system, it makes the image process adapt to the changes.

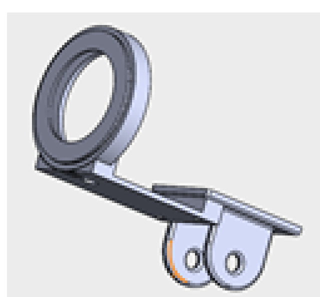


Figure 3.9: Camera and LED ring Support.

The image system is composed by an Arduino Uno, connected with an Red, Green and Blue (RGB) LED ring WS2812 [51]. To couple the illumination system together with the robot and the camera a support was created and printed in a 3D printer (see Figure 3.9).

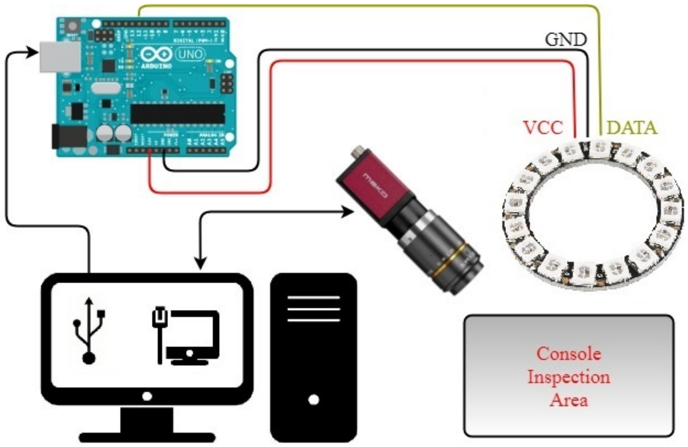


Figure 3.10: Adaptive illumination components.

Using USB connection with the Arduino Uno, the control of the illumination is done by communicating the LED lights with the Arduino as a Digital output 13, with the +3.3V supply and a Ground (GND) (Figure 3.10). The control of the illumination is done according to results of the last image acquired process, if the mean of the brightness is lower the light is increase to reach the setpoint or, if it is higher the illumination power is reduced (see Figure 3.11). This lights values are transmitted from Matlab to Arduino as a string and, sequentially interpreted to generate a value, sending it to the digital output.

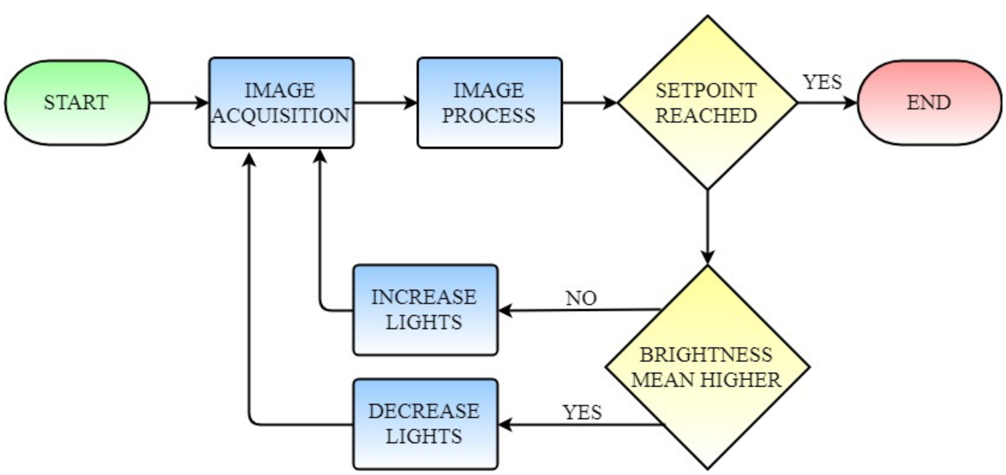


Figure 3.11: Adaptive illumination system flowchart.

3.2.3 The UR3 and Force Sensor

The UR3 needs to be installed and fixed in the workbench in order to have an optimal configuration of the components and improve the realization of the movements to perform the tasks. Also, in the UR3 User Manual [47] some specifications about the range of work and joint (see Figure 3.12) are done. So according to the type of inspection the robot can be installed in a defined position to improve its utilization.

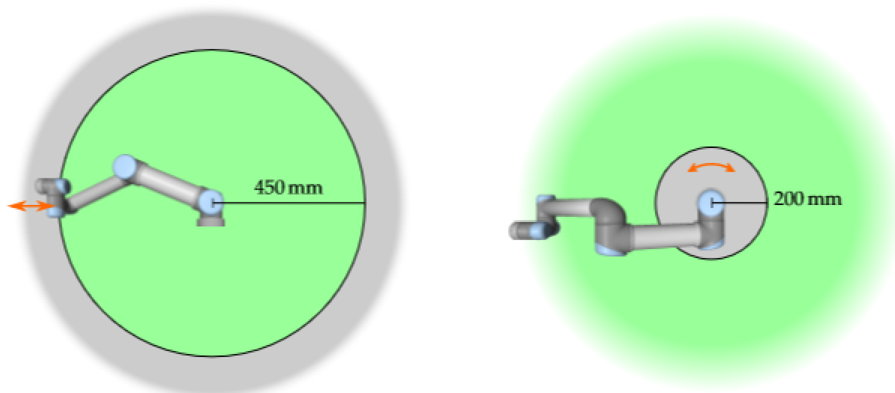


Figure 3.12: UR3 workspace [47].

The force sensor is installed in the robot tool and, on sequence, another tool was developed in Solid Works to press the buttons and it is coupled after the force sensor (see Figure 3.13), providing to the robot an easy way to press the buttons. In addition, to improve the utilization of the camera and the illumination, another support is used to attach then to the robot (see Figure 3.14).

These tools are very important to the realization of this work, what proves the importance of utilization of 3-D printers as the Industry 4.0 is demonstrating [52]. The use of these printed tools improves the versatility of the system, its reliability and the creation time.

To create a pattern that represents the pressing movement of the button some measures were made and a distance of 2.7 millimeters after the robot touch the button is used to press all the buttons and also identify its force. This process is done since the perception

Figure 3.13: Tool to press button.

of high variations in the values on force sensor. The pattern value to press the buttons was obtained using the measures showed on the UR3 console display. Upon the tool touch the the button, the value of the displacement was being decreased and when it moves more than 2.7 millimeters there is no displacement available. So, if it moves more than it, the values starts to increase like the robot was touching a rigid surface and activates the safety mode.

After apply this method, the creation and analyzes of a pattern in the results became perceptible, besides, the application of the mathematical methods to create a predictor to the classification method.

(a) CAD support to attach camera and LED ring to UR3.

(b) Support attached in UR3

Figure 3.14: Support to attach camera and LED ring on UR3 (a) (b).

After receiving the string signal from Matlab, with the identification of the console that is being inspected, the program made in UR3 console will read it and realize the inspection. The programming of the console is based on Universal Robots Language [53].

To teach the robot the position to move and the right sequence of movements to realize, a script will be created following the Matlab program steps. The main goal is control the process and synchronize the application with the robot program some status signals that will be read and according to its response the program moves to the next step. The program language is based on UR3 manual [47], where the functions to send, read, establish communication, move, loops, etc. are described.

3.3 Graphical User Interface (GUI) Application

Intending to make the process as intuitive as possible, the creation of an interface to the utilization of the program was proposed (see Figure 3.15). To realize it, the application has to join all the process explained before in a single program, so the user can keep up with the actual status of the process and also the main results informations like the image processed, the result of lights variation, the inspection of the buttons and the display have to be available and an *On/Off* button is required to control the inspection.

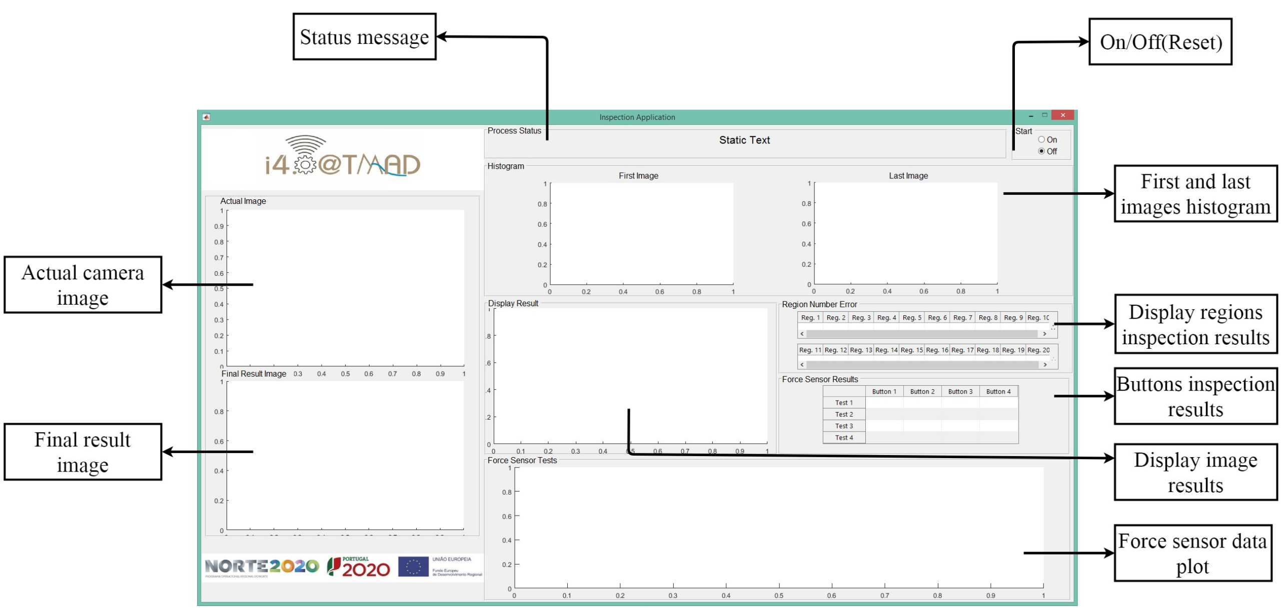


Figure 3.15: Final application window on Graphical User Interface (GUI).

This application in made following the purpose of the industry 4.0, that allows the workers to interact with production systems using an interface. The access to this interface can be done using different types of devices (tablet, smartphone, laptops, etc.) allowed to connect the local network were the system is installed.

3.4 Communication

Following the 4^{th} Industrial Revolution the system needs to be flexible to accept new information and even new tools, for this reason some parts of the system are connected by serial port and others by network communication. The camera and the UR3 are connected with Matlab by the network while the illumination system and force sensor by serial port. It is done to improve system capability of execute different task.

3.4.1 Universal Serial Port (USB)

With the dissemination of the USB connections in all computers and by its versatile, many devices use this type of communication to be easily installed. Also with its diffusion the price also decreased a lot what makes it more competitive if compared with others interfaces [54]. In this study the USB is used to send data to an Arduino Uno and read data from the force sensor. The Figure 3.16 shows the communication interfaces used in the project.

3.4.2 Transmission Control Protocol – Internet Protocol (TCP/IP) Communication

The Transmission Control Protocol – Internet Protocol (TCP/IP) protocol is the most popular protocol in network communication. It allows the connection of different types of devices, as: smartphones, computers, and embedded devices [55]. Having in mind the dissemination of the concept of "Internet of Things" [56], one from the most used means of communication might be the internet. In this work the TCP/IP is used to communicate

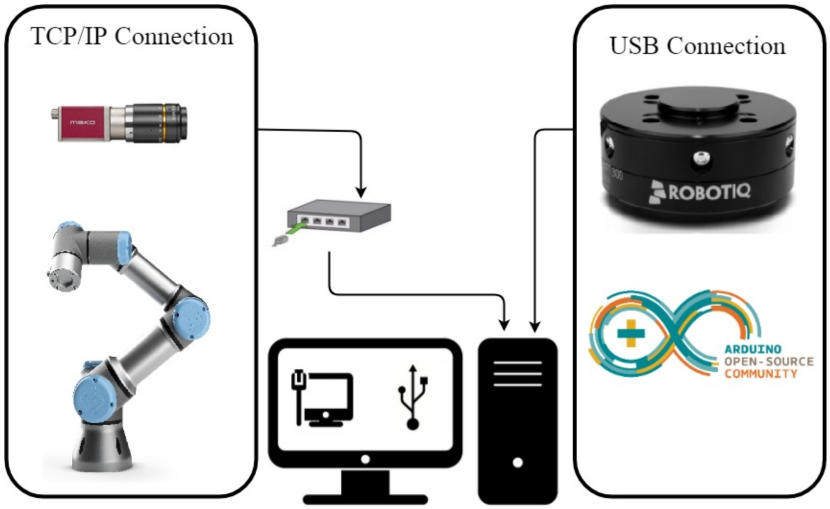


Figure 3.16: Devices communication mode.

the industrial camera and to read and write data from UR3 using a router to connect and direct the data to its address, as showed in Figure 3.16.

Chapter 4

Programming Methods to Perform Inspection

To contemplate all the project characteristics, a set of process and methodology are used to the creation of the final application. These steps focus on the main objectives that are: classification of the force sensor's data and the display images to be inspected. This chapter explains each one of the phases that compose the creation of software and programming functions made in Matlab to realize the inspection of the buttons and displays.

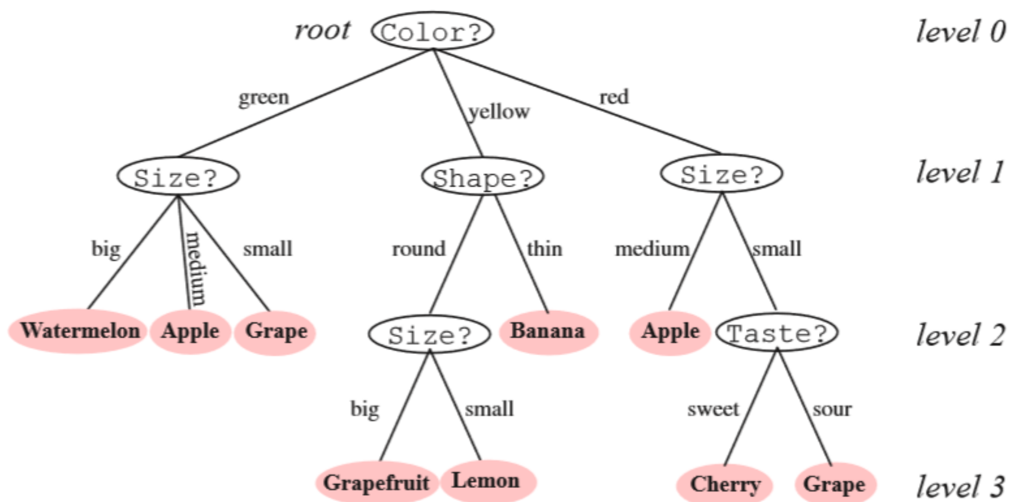
4.1 Machine Learning for Buttons Classification

When the buttons are pressed the force sensor detects variations and sends to the computer by USB communication to save this values. After finished the tests the values are analyzed by a standard deviation, mean and range. Using these characteristics is created a supervised machine learning classification (when a finite group of values are the output of the algorithm then, the learning method is called classification [57]).

To develop the classifier a set of data is necessary, this way improving the classification quality. Different types of classification can be used and to choose one, were created three types of classification in Matlab to be tested: a Decision Tree Classification, a SVM and a k-NN.

4.1.1 Decision Tree Classification

Decision tree is an intuitive way to classify pattern using a sequence of conditions (attribute dividers) extracted from the object, where depending from the answers a different question is used to follow the process (node). The nodes test a specific feature generally comparing with some constant value (extracted from the data base analyzes) [58].



,

Figure 4.1: Decision tree classification example [59].

This is a method to create simple solution to complex problems, based on the decisions made in the nodes until arrives the end (leaf) of the questions getting some result [60]. This decision process is created four hundred samples and on the quantity of characteristics extracted from the study (see Figure 4.1) that enable the creation of a set of decisions based on features values. Also, in this method small variations on values obtained generates big changes into the decision nodes, what can made the method instable. Another specification is that for more samples applied to the training, the results don't increase the method's precision significantly.

4.1.2 Support Vector Machine (SVM) Classification

The SVM is a supervised machine learning used for classification, also for regression tasks. After process the force sensor results and extract its features, these samples are plotted into a plane, then at this plane the algorithm creates a hyperplane to separate and classify this regions. This hyperplane has a margin, that is the distance between the hyperplane and the first sample in the train group (see Figure 4.2), the bigger is the margin better are the classification results [57].

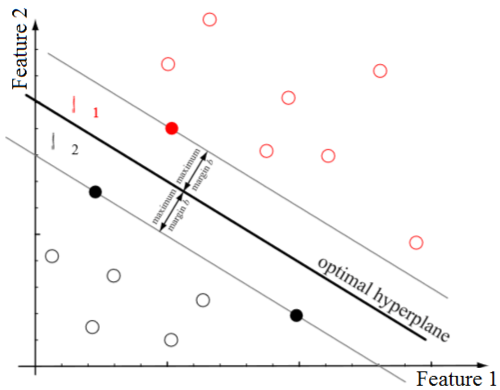


Figure 4.2: Support Vector Machine (SVM) classification example [59].

Together with SVM a device called "*kernel*" is used to improve the results and creation of the hyperplane. This device remap the samples to hyperspace intending to transform a non-linear separation into a linear one [61].

In this study, a SVM is applied with a Gaussian kernel (4.1), what means that after the analyses and calculation, data results are processed by the kernel [57], [62], plotted into a hyperspace and in sequence the creation of hyperplane is made.

$$G(x_1, x_2) = exp\left(-\left\|x_1 - x_2\right\|^2\right) \tag{4.1}$$

4.1.3 k-Nearest Neighbors (k-NN) Classification

This algorithm propose a classification based on 'k' nearest neighbors, since $k > 0$, this value should be proportional smaller (even k=1) but not so big to avoid to unclassified samples, improving the accuracy of the classification. For each case, a good value of k needs to be found, due to its variation for any specific application [41], [63].

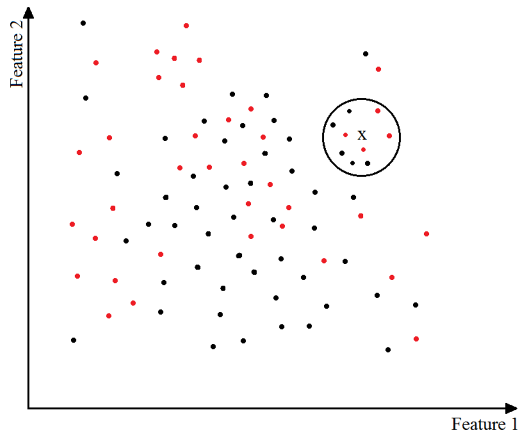


Figure 4.3: k-Nearest Neighbors (k-NN) classification example.

The nearest neighbors are found measuring its distance to the other samples, the 'k' nearest neighbors are counted and the most frequently class defines the result. In this method the algorithm might generate wrong rating for samples in regions with high density and variety of classes. Besides, the distance among the points needs to be measure using any method, in this work it is used an Euclidean distance [39], and it is defined by the equation (4.2). Also in this work, $k = 9$ is set to avoid draws in the classification (see Figure 4.3) and as was mentioned above k should not be so large to avoid non classified samples.

$$D(a, b) = \sqrt{\sum_{k=1}^{n} (a_k - b_k)^2} \tag{4.2}$$

4.2 Displays Inspection

After test the buttons, a next step is to check the displays. To test the displays, an algorithm is used to compare a good display to the others. Is perceptible that realize inspection using image require a complex process, since the image has a high amount of information to be interpreted. Consequently, an methodology named by "Divide-and-Conquer" was been studied to create a strategy to improve the display inspection. This method, works on recursively way, breaking the problem into several subproblems, being those more simple to be solved. The solution is a combination of all those small problems' results [64].

4.2.1 Thin-Film Transistor (TFT) Display

The image acquisition is also made by the camera coupled in the robot (Mako G-125b). This type of display has some special draws that represent the actual state of the system and the response of some changes by the press of the buttons. Consequently, a complete inspection of the display will be done to verify if all of the pixel regions are really working.

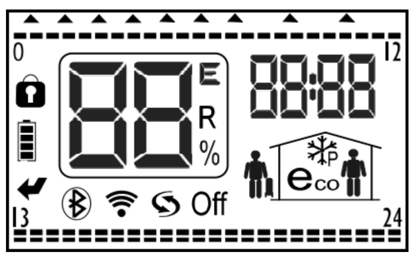


Figure 4.4: Illustration from Thin-Film Transistor (TFT) display.

The display is divided into 20 parts to be verified and further information about it can be found in the annex A, each of these parts are a template to be compared to a future image acquired [65]. These templates will be applied into a Normalized Cross-Correlation (NCC) [66] using the equation (4.3), as made in others papers [67], [68]. This function is applied to find regions with similarity to the template into an image. Thus, the result matrix is composed by values between -1 and 1 that corresponding to the relation of the image with the template [69].

$$\gamma\left(\upsilon,\vartheta\right)=\frac{\sum_{x,y}\left[f\left(x,y\right)-\bar{f}_{\upsilon,\vartheta}\right]\left[t\left(x-\upsilon,y-\vartheta\right)-\bar{t}\right]}{\left\{\sum_{x,y}\left[f\left(x,y\right)-\bar{f}_{\upsilon,\vartheta}\right]^{2}\sum_{x,y}\left[t\left(x-\upsilon,y-\vartheta\right)-\bar{t}\right]^{2}\right\}^{0.5}} \tag{4.3}$$

Afterward the NCC another search looking for the maximum value into the matrix of results is done and after find it, a region around this value with the same size of the template is copied to be directly compared to the template using the function Correlation Coefficients (CC) (as showed on the equation (4.4)). This comparison is a measure of the linear relation between the images having results values between -1 and 1, where -1 is a directly negative correlation and 1 means that the images are completely equal, and this function's results return as the equation (4.5) [70].

This process, is done to confirm if the image is really related with the template, considering that into a grayscale image it is possible to get good values into regions with similar values, when this directly comparison is made this type of error is avoided.

$$\rho\left(A,B\right) = \frac{1}{N-1}\sum_{N}^{i=1}\left(\frac{\overline{A_i - \mu_A}}{\sigma_A}\right)\left(\frac{\overline{B_i - \mu_B}}{\sigma_B}\right) \tag{4.4}$$

$$R = \begin{pmatrix} 1 & \rho\left(A,B\right) \\ \rho\left(B,A\right) & 1 \end{pmatrix} \tag{4.5}$$

Finally, after these comparisons the results obtained from the equation (4.5) is analyzed and according to the values a final response is set to an array with these analyzes results (1 to approved and 0 to reproved).

4.2.2 Liquid Crystal Display (LCD) with Neural Network Classification (NNC)

In order to validate the inspection process a LCD also is inspected, it is a 16x2 LCD generally used in Arduino. This process works similar as the TFT display inspection. However, the display do not have different regions, this one, is composed by 16x2 regions, each of this regions have 40 pixels (see Figure 4.5, the main intention is to find pixels that are not working.

In this inspection, the template will be used to count the number of regions that are working. On the other hand, this inspection will confirm and indicate the regions that are working well and the defective ones.

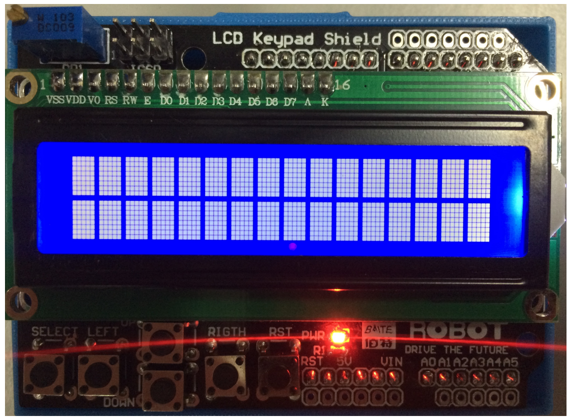


Figure 4.5: 16x2 LCD display.

The hole display is compared with the template to find large defects, then, this display is also divided into the areas of segmentation to verify small errors (see Figure 4.6). To realize it, a topology to analyze the regions is created using the grayscale pixels' value, making the calculations into four rows and five columns, that represents the pixel. Into these regions, the mean, standard deviation, minimum and maximum values into the regions are obtained to make an analyzes and, on sequence apply these into a Neural Network Classification (NNC).

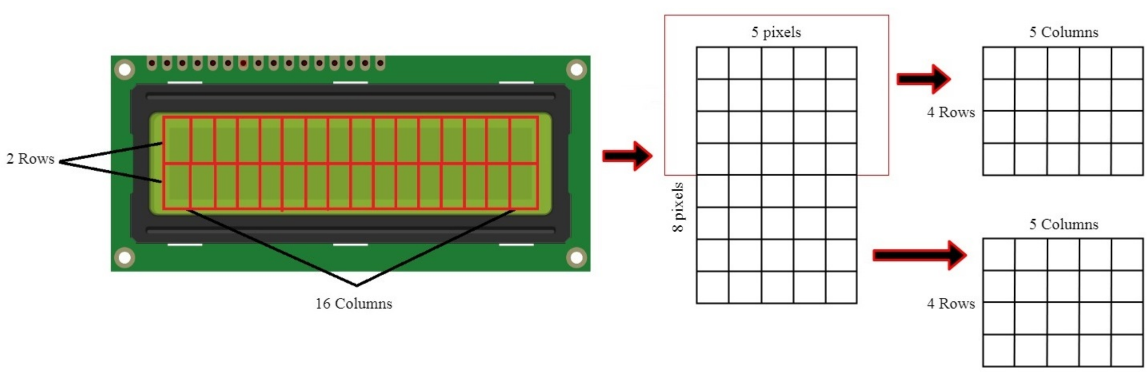


Figure 4.6: 16x2 Liquid Cristal Display (LCD) inspection workflow.

The NNC uses the data obtained by the image processing into the pixels regions to realize a pattern recognition and associate to a prescribed number of classes (outputs). The NN is represented group of neurons connected as can be seen on Figure 4.7.

The hidden layers are functions to create an useful intervention between the inputs and

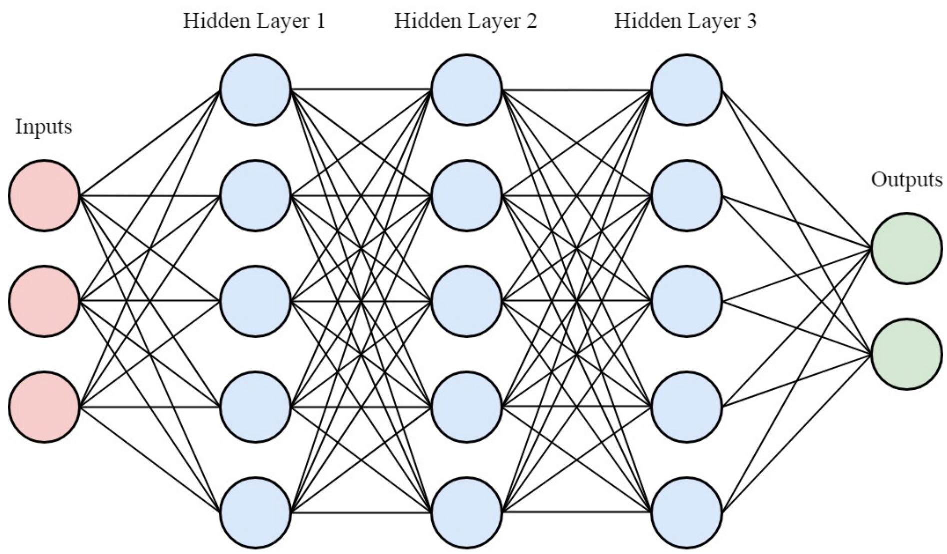


Figure 4.7: Multilayer neural network representation.

outputs, they perform some functions to extract more characteristics from data training. So, as can be seen on the Figure 4.7, the inputs of the system are pattern extraction from image and, also are the inputs from the first hidden layer. This hidden layer's output is the input signal from the second hidden layer, and so on until gets the output [71].

The math defines the Neural Network (Figure 4.8) as the equation (4.8). On the Figure 4.8 the first step is the data inputs acquired insertion, including bias (has effect of increasing or decreasing the net input activation function as weights does, but this gets the negative value from the activation function's threshold [72]), following, the application of the synaptic weights, these are the respective synaptic weights of the neuron k. The next step is to add all the signals and, the result of these is υ_k (equation (4.6)). This one is applied to the activation function [73], as the equation (4.7) shows. The result of this, is the output signal from the NN.

$$\upsilon_k = \sum_{m}^{j=0} \omega_{kj} x_j \tag{4.6}$$

$$\varphi(n) = \frac{2}{(1 + e^{2*n})} - 1 \tag{4.7}$$

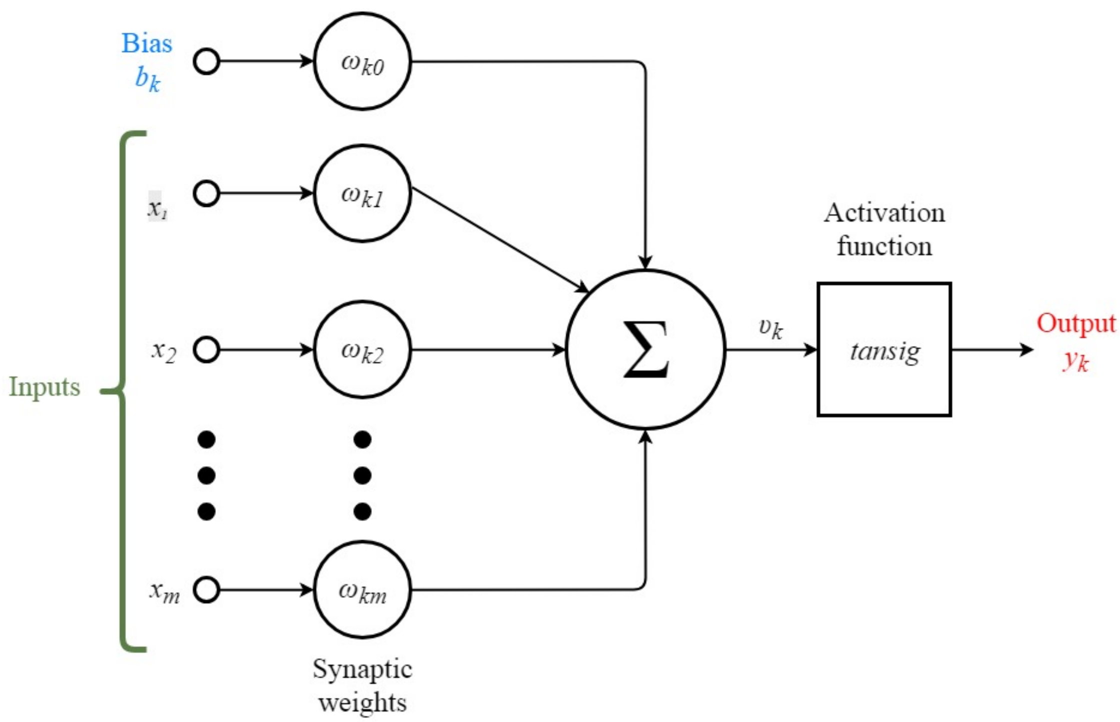


Figure 4.8: Multilayer neural network model (adapted from [71]).

$$y_k = \varphi(\upsilon_k) \tag{4.8}$$

Due to this, Hornik [74] shows that standards multilayer feedforward NN with continuous and smooth activation function are approximation function of an inputs signal. Thus, the NN results into an approximation function that will classify the samples into the output classes.

Chapter 5

Results

This chapter explains and discuss about the results obtained in all phases of the project until the final application. The first section shows the consoles recognition and it variations with the illumination system demonstrated by the using of a histogram plotting. The second one shows the functionality of the the adaptive illumination. The third section contains the results about the the UR3 programming, the next one shows the results obtained with the classification of the force sensor data. The fifth section contains the results of the displays inspection and the last section the results of the standard workbench and application.

5.1 Image Recognition

After the image acquisition and illumination established, the image is processed to the identification of the consoles type by its characteristics 5.1.

Table 5.1: Consoles image characteristics

Characteristics	**Console I**	**Console II**	**Console III**	**Console IV**
Area	180103	381077	53217	38904
Perimeter	1627.9	2188.6	815.1	980.1
Eccentricity	0.5879	0.0659	0.1635	0.9754

In the histogram the effect of illumination system can be observed by the increasing of the values of the pixels, the Figure 5.1 shows the result of the console identification and its histogram before and after the adaptive illumination works. The Figure 5.2 of image identification will be used in the final application. As can be seen on the image, with the image is adjusted to the set point there was an increasing of the light power, meaning that the external lights are not able to reach the set point.

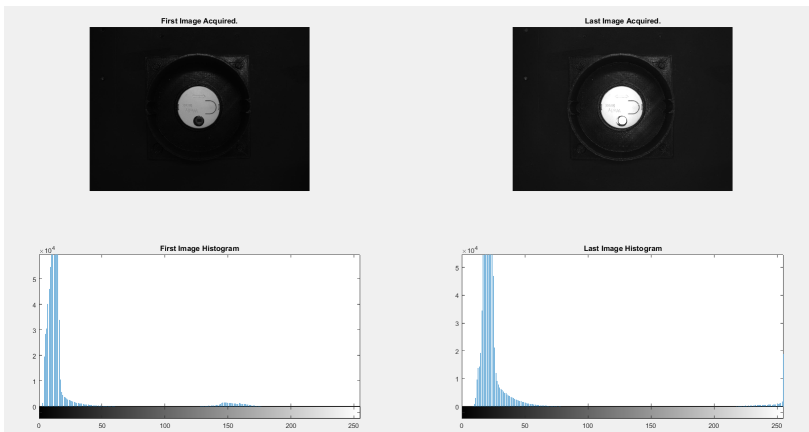


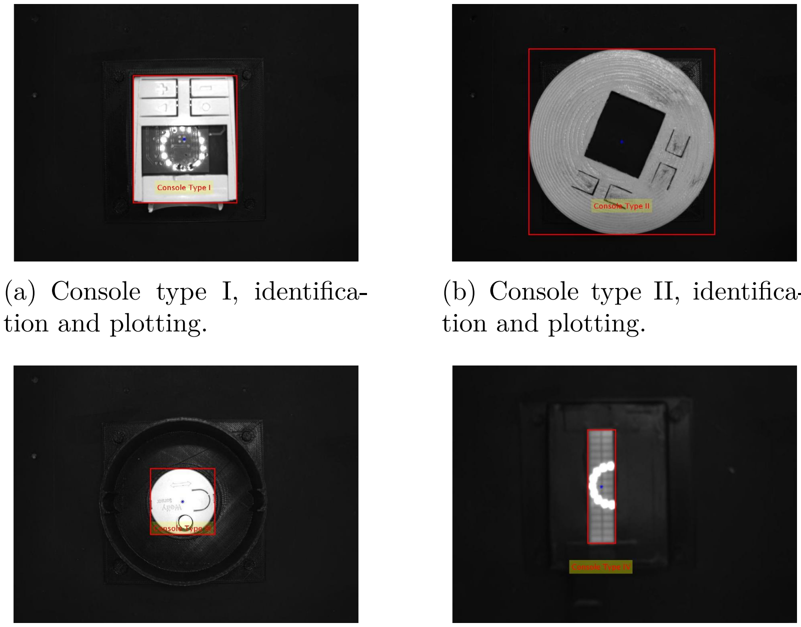
Figure 5.1: Image histogram before and after the initialization of the adaptive illumination.

In addition, the results of the recognition of the console shows the adaptability of the illumination system to perform the task using different console types.

5.2 Adaptive Illumination

The illumination system is done by the comparison of the brightness input inserted by the user and varies between 0 and 255, where 0 represents the minimum brightness on image (a complete black image) and 255 the maximum brightness on image (a complete white image). With this value sent to the Arduino, a specif level of brightness is activated. The illumination is initialized by the reading of a file that writes the last value applied on the light system.

One of the purpose of this system is provide an easy interface with the user, making possible to change the illumination according to his necessity and also have a template of illumination avoid noise, consequently, classification errors.



(a) Console type I, identification and plotting.

(b) Console type II, identification and plotting.

(c) Console type III, identification and plotting.

(d) Console type IV, identification and plotting.

Figure 5.2: Consoles identification (a) (b) (c) (d).

The algorithm calculates the image brightness and adjust it according to the necessity and finally take the picture adjusted according with the illumination calibration.

When the brightness on image is setup according to the specification a message is showed on the computer display, but if this value can't be obtained an alert message will appear according with the problem. The first problem is that the system has enough illumination with the ambient light and the second one is the system is not be able to increase the bright of image as much as the algorithm need.

5.3 UR3 Programming

The programming on the UR3 is based on sequence of steps showed below:

1. Start signal;

2. Move to the camera position to acquire image;

3. Send position acknowledge;
4. Wait signal of console type;
5. Start inspection of the console buttons;
6. Send end buttons inspection signal;
7. Move to inspection display position;
8. Move to initial position;

Following this sequence of steps, the main program is ready to work together with the UR3 scrip. The Figure C.1 in annex C shows the programming of the UR3 made on console. In this inspection system some types of consoles do not have buttons and others do not have display, so according to this the steps 5 and 7 are not used.

The UR3 program sequence gets a successful result and if any problem in connection happens the system gets stopped waiting for the signal to continue. If the problem persists, with the *Off* button on the application window the system is reseted an the process can normally start from the beginning.

5.4 Buttons Classification Results

This section shows the results obtained with the supervisioned machine learning using the three decision methods cited in last chapter. After the communication with UR3 is established, the tests are done and the values obtained are read and manipulated to be applied in the classifiers.

5.4.1 Decision Tree

This method of classifications gets a good result but as expected this isn't the most efficient if compared to the other classifications used. The Figure 5.3, shows the decision tree generated using the Matlab. This program create this nodes based on the inputs and the following figure (see Figure 5.4) demonstrate the confusion matrix from the classifier. This matrix has the true, false, *false-positive*, *false-negative*, the percentage of match for each case and the general percentage of match for the algorithm. The class *0* represents the buttons with defect and the class *1* represents the buttons without defect.

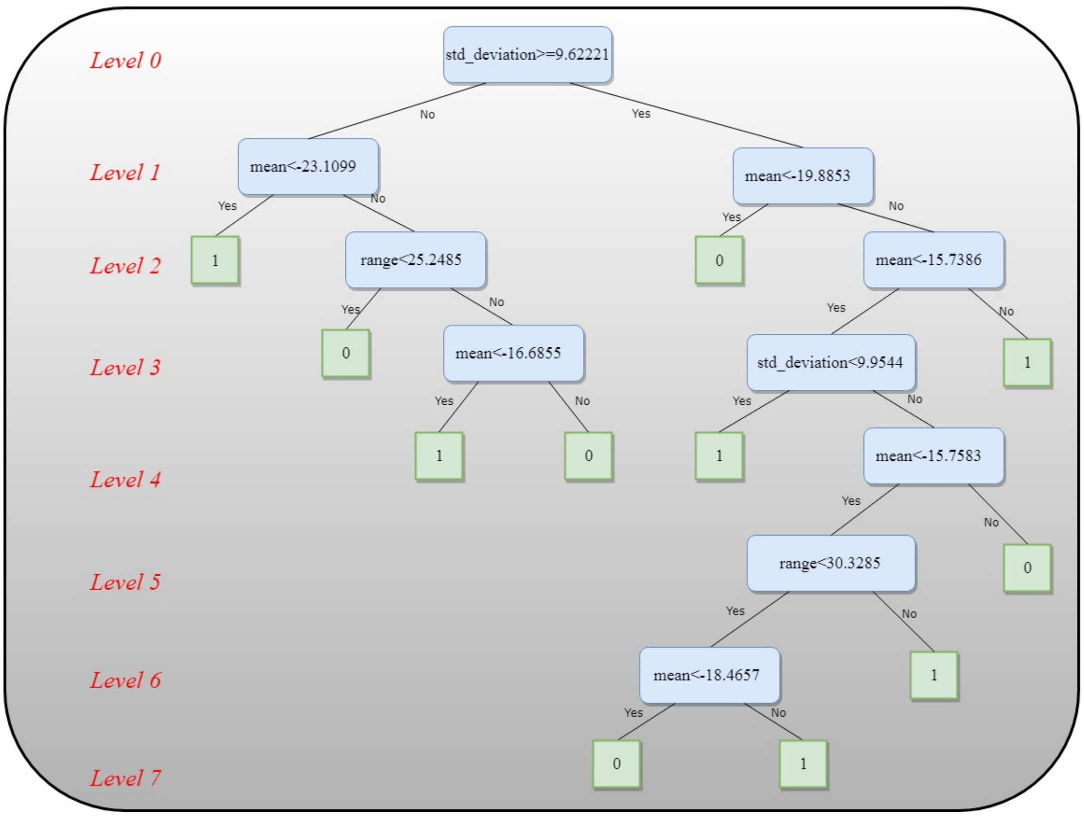


Figure 5.3: Decision tree classification.

The confusion matrix shows some errors in classifications into true and false classes, due to this, this isn't the algorithm applied in the final application to avoid the increasing of errors in future inspections.

Furthermore, another tests showed that in the classification tree the errors increase fast just with small de-calibration of the sensor, and it happens normally. So, after some time of use the ideal is to calibrate the sensor again (this is an easy and fast process, and leaves in the maximum 2 minutes). Another drawback is that decision trees starts the analyzes aways by the same conditional node, meaning that the conditions on the next levels of tree are depending of the node before.

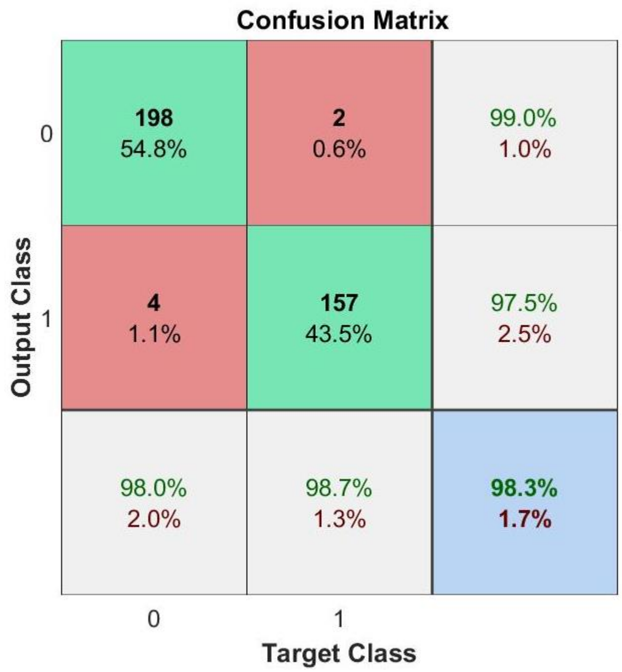


Figure 5.4: Confusion matrix from decision tree classification.

5.4.2 Support Vector Machine (SVM)

The SVM gets the better result than the other classification method. The creation of hyperplane works as good as expected, the Figure 5.6 shows the plot of the hyperplane and its position in the hyperspace where the samples are distributed. On sequence, the characterization of the result, the confusion matrix of the classification was created (see Figure 5.5). For this type of problem, this solution proved to be the most efficient.

More than the creation of the hyperplane, the samples circulated with green are support vectors in the margin of the hyperplane included to improve the results and separation of samples.

The hyperplane created makes an excellent separation of the samples' group and, jointly with the support vectors on the samples located on the margin of the hyperplane, the solution shows an precision of more than 99.4%, with a higher probability of error on samples that are not true but are classified as it (*false-positive*).

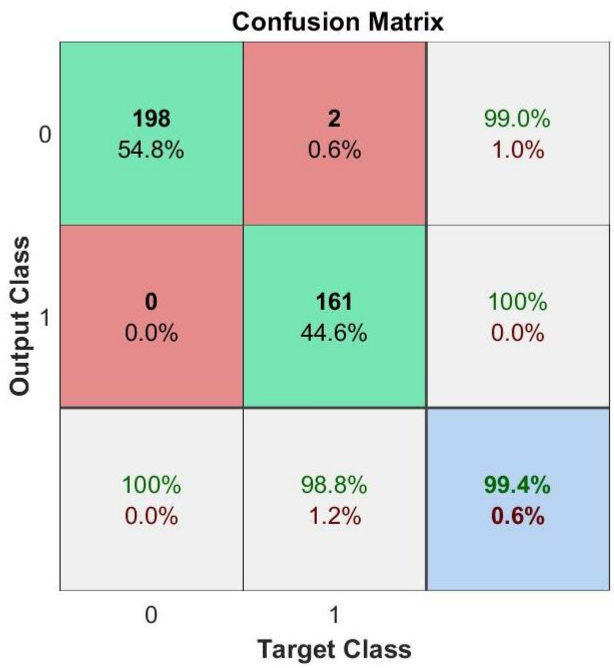
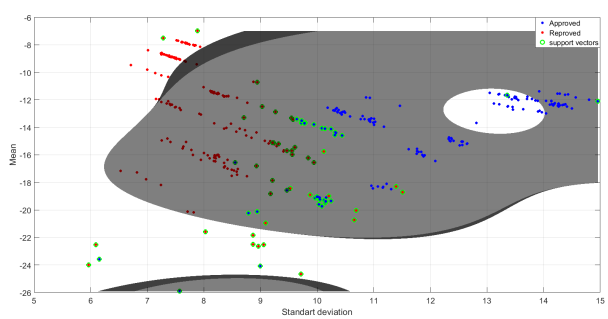
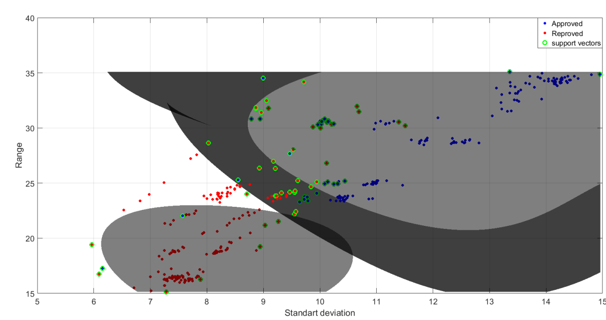


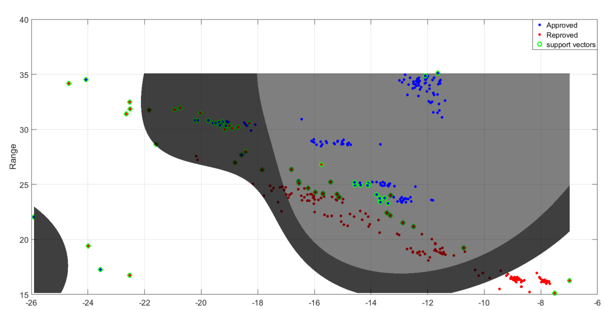
Figure 5.5: Confusion matrix from SVM classification.



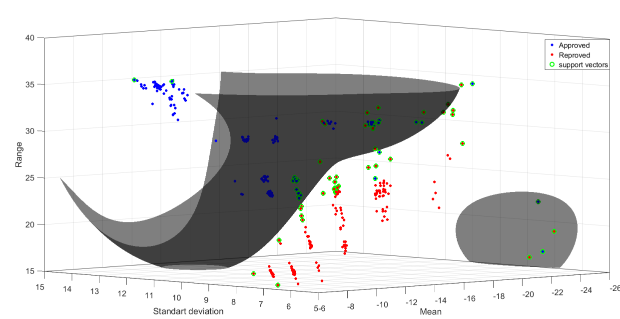
(a) SVM standard deviation and mean plane vision.



(b) SVM standard deviation and range plane vision.



(c) SVM mean and range plane vision.



(d) 3D vision from SVM hyperplane.

Figure 5.6: Support Vector Machine (SVM) Hyperplane result (a) (b) (c) (d).

5.4.3 k-Nearest Neighbor (k-NN)

As described in the previous chapter, the k-NN will be applied with k=9 and Euclidean equation to calculate the distance between points. Subsequently, the samples were plotted into a plane and trained to generate the confusion matrix (see Figure 5.7).

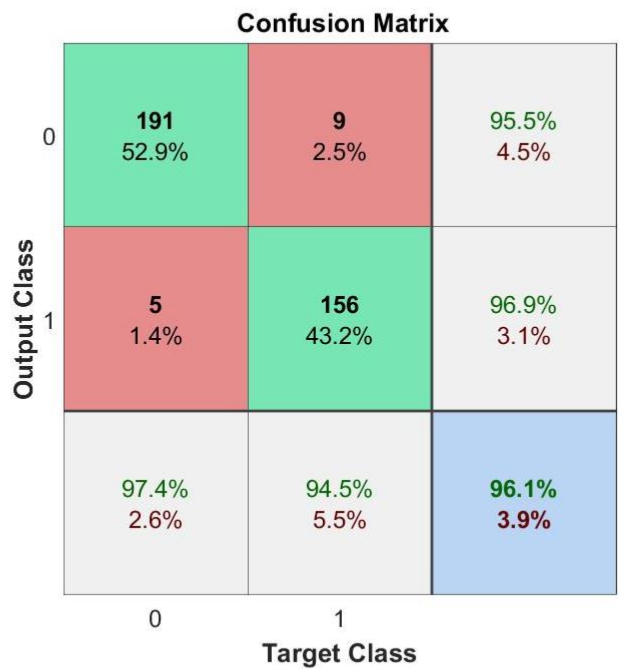
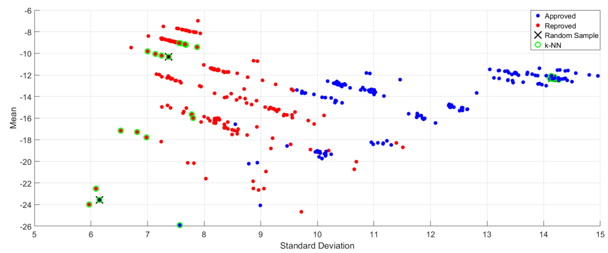


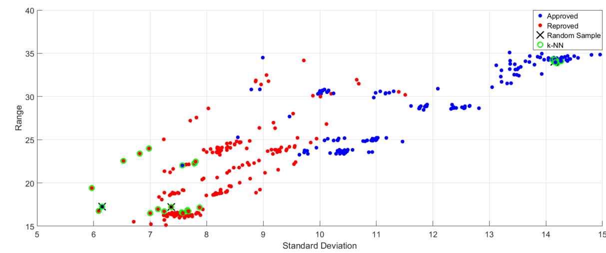
Figure 5.7: Confusion matrix from k-Nearest Neighbor (k-NN) classification.

To visualize some results, random samples were plotted an classified according to the model. Besides, the nearest neighbors (circulated with green) and the samples are plotted into the plane as the Figure 5.8 shows. According to the count of each 9 nearest samples group, the result is obtained. Also, in the Figure 5.8 the identification of most common areas to each class is easy done, but this classes gets variations too.

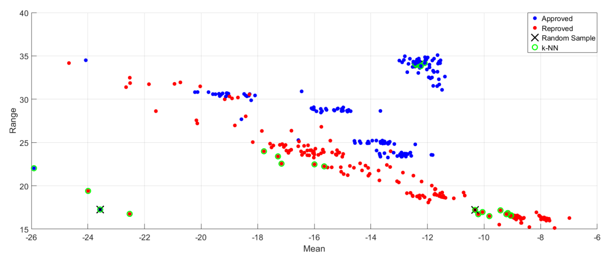
Even with the good results obtained, as can be seen on the confusion matrix, with the distribution of the samples many regions can be critical to be classified and can not generate *true-positive* results. This type of error is common in this method and also affects the Decision Tree methodology, because these methods are simpler than the SVM, that creates a hyperplane to separate the output classes.



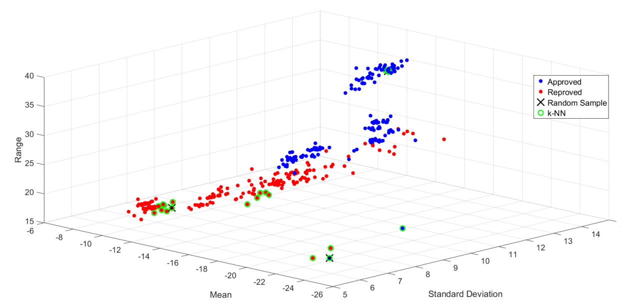
(a) k-NN standard deviation and mean plane.



(b) k-NN standard deviation and range plane.



(c) k-NN mean and range plane.



(d) 3D k-NN vision.

Figure 5.8: k-NN plane result with random samples (a) (b) (c) (d).

5.5 Display Inspection Results

At this section is presented the results obtained with the process to inspect different types of display intending to validate this method to inspect screens.

5.5.1 TFT Display Inspection

With the display inspection methodology defined the next step is extract the real images of each template described on the previous chapter. The comparison of the templates was made and the results get a precision of almost 90%. Besides it, is important to note that as other images process, external illumination and shadows can generate noises in the images acquired and, that is why the system illumination gets important.

The results obtained for display errors are as good as expected but in some regions the inspection is more difficult to be done, generating some errors (*false-positive* and *false-negative*). The results from the display inspection are showed on the Figure 5.9.

The TFT display's inspection verify that some regions as regions 19 and 20 are more difficult to be inspected because they are too small and are almost out of the displays glass. Besides it, how this console is a prototype a proposed solution can be change this pixels to other type of indicator (as LED external the display), or replace then in other region.

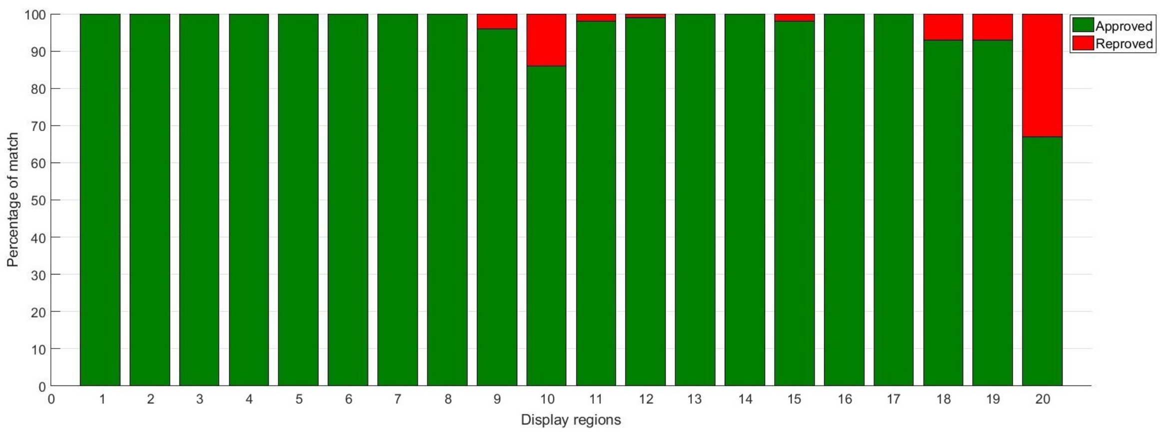


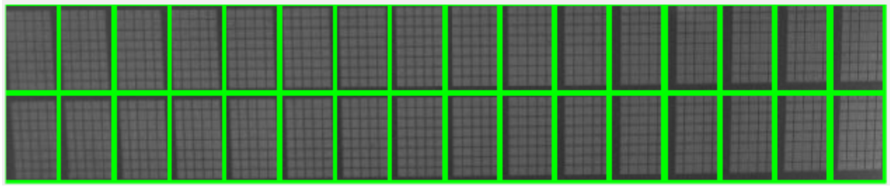
Figure 5.9: Matches and errors percentage of Thin-Film Transistor (TFT) display inspection.

As expected the structure and format of the glass and printed glass can directly interfere in the inspection, can be noted that some regions localized too near from some printed glass images, from the top and button edges of the display gets the worst results, as happens with the display region 18, 19 and 20. Besides, other region that gets problems is the region 10, this region is too small and has noise from the pixels by your side.

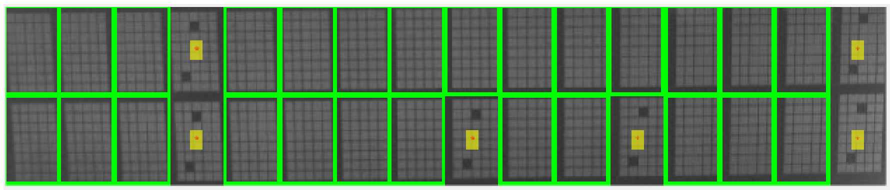
The other regions show that the adaptive illumination together with the algorithm created to inspect this type of display got an impressive result, with matches over than 90%, demonstrating the accuracy of the presented method.

5.5.2 LCD Inspection

The inspection of this type of display do not shows exactly the localization of the pixels with problems, but if any error inside a group of pixels is perceived, the algorithm shows the error in that group as the Figure 5.10 shows.



(a) LCD inspection with no errors.



(b) LCD inspection with errors into random regions.

Figure 5.10: Final image obtained with inspection in LCD (a) (b).

The same way that the previous display inspections does, the LCD inspection is based on the value of correlation between this particular region and the template of this LCD. For this reason, the error is located in that region what makes it easier to be seen and, beside that, the image of the display shows the error in region by writing a red '*' on it. Otherwise, the areas that are working normally have a green contour, as can be seen on Figure 5.10. Also, to verify and identify small errors, the inspection method described in Chapter 4.2.1 was done an applied to the NNC, the Figure 5.11 shows the NN created with the 8 inputs (row and column, mean, standard deviation, min and max, respectively), 20 hidden layers and 1 output (as the outputs is 0 or 1, with or without defect).

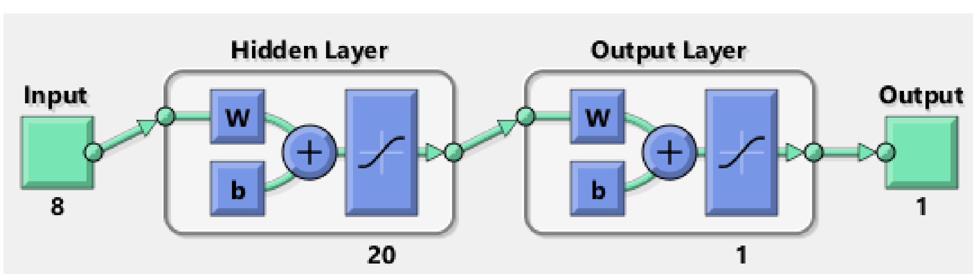
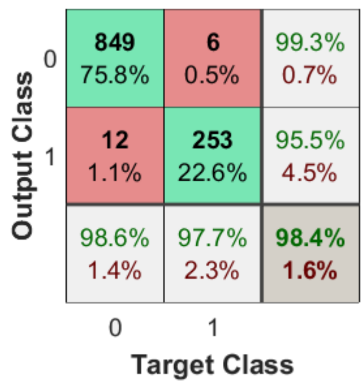
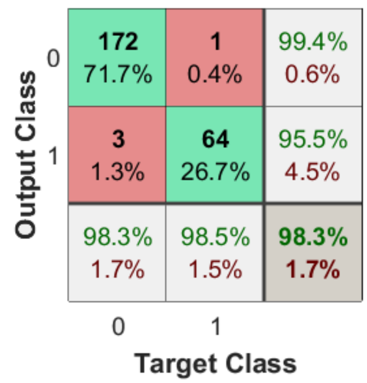


Figure 5.11: Neural network layer.

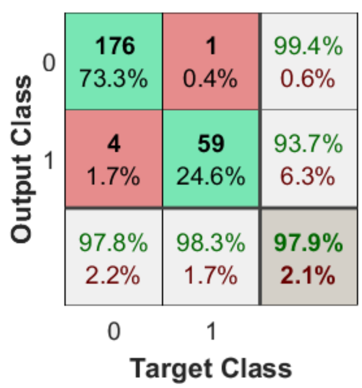
Additionally, with the training samples was generated a confusion matrix from the training, validation and test with 70%, 15% and 15% of the samples, respectively. Another confusion matrix also was created and shows the results with all the samples to demonstrate the method accuracy (see Figure 5.12).



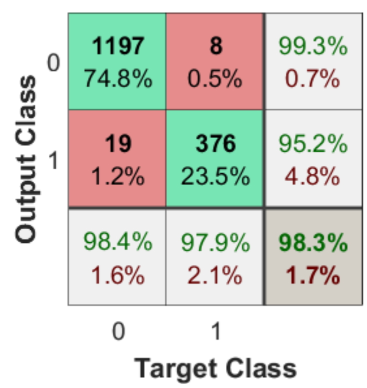
(a) Confusion matrix from training samples.



(b) Confusion matrix from validation samples.



(c) Confusion matrix from test samples.



(d) Confusion matrix from all samples.

Figure 5.12: Confusion Matrix from Neural Network Classification (NNC) (a) (b) (c) (d).

The NN gets great results, with the correctness of the proposed method, the features extraction are resulting into a excellent classification and even small errors as missing of one pixel are being detected using NNC.

5.6 Standard Workbench and Application

After the development stage an *.exe* file was created, to run into multiples platforms like: a industrial computer that can be used directly on it or using a tablet or another portable device connected to the computer by the internet.

The workbench that is fixed all the system and tools need to run the inspection process, as can be seen on Figure 5.13. The security devices are also installed to assure the security of the users. The status light shows the actual state of the robot (is working or stopped), the emergency button stops the robot automatically and the light curtains reduce the motion of robot.

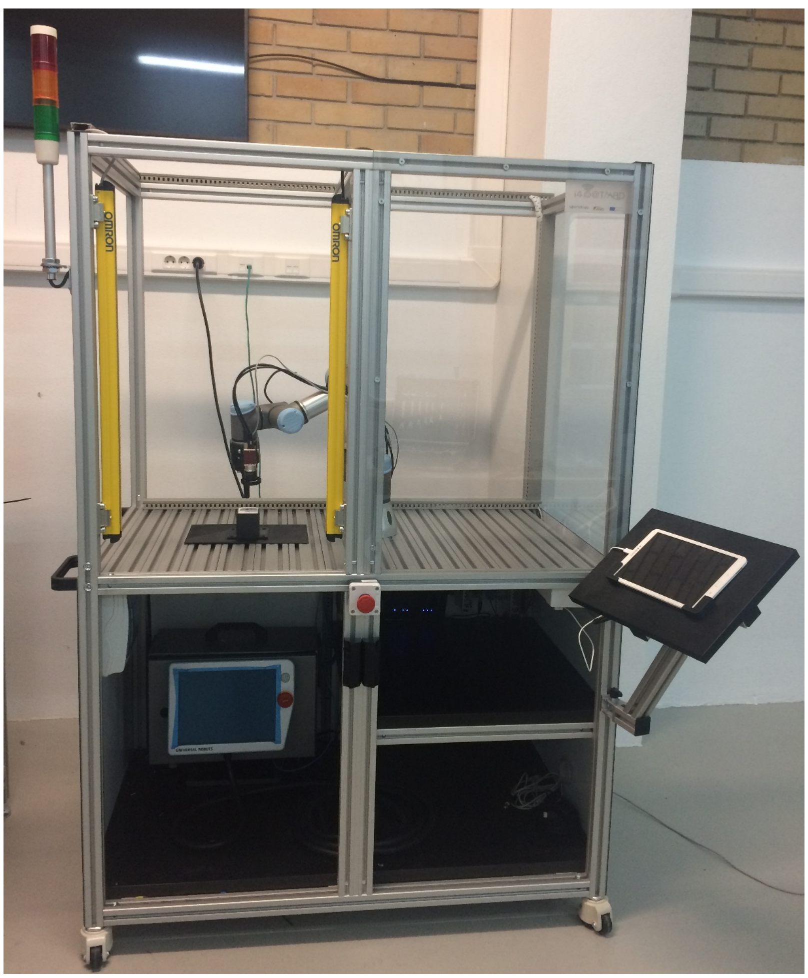


Figure 5.13: Workbench complete mount.

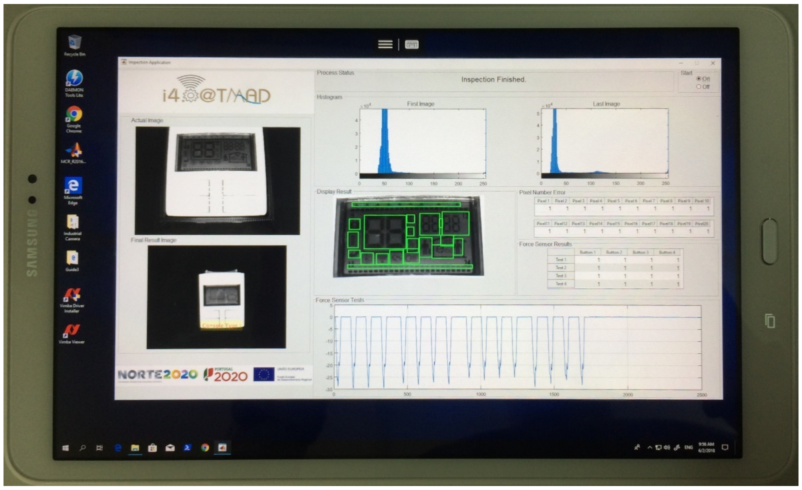


Figure 5.14: Application access by tablet.

A local interface into a tablet is available to the user (see Figure 5.14), this realize the communication with the industrial computer installed in the workbench. Also is possible access the interface remotely if the user is connect into the same network.

To change setup to inspect other type of console, the adaptation time can varies according to the necessity of the variation of the illumination, in general cases the complete inspection time can leave about one minute and half. This is a small time, since you can inspect different types of displays and, also buttons.

This workbench is part of a project named as i4@TMAD (from the portuguese name "Promoção da Indústria 4.0 na Região de Trás-os-Montes e Alto Douro") that is proposing show to the businessman the main purposes and technologies applied in the Industry 4.0. This way, an exhibition with complete presentation of the technologies and algorithm applied was done in the cities: Vila Real, Chaves, Bragança (see Figure 5.15) and also, for especial visit from the Cape Verde's President.

(a)

(b)

Figure 5.15: Exhibition from workbench to the businessman in Bragança (a) (b).

Chapter 6

Conclusion and Future Work

In this work, the creation of an inspection workbench was made and the accuracy and fiability of the inspection was successfully proved. Due to this, the methods used to classify each of the samples prove to be efficients, some of them better than others.

In the case of the buttons inspection, all methods proved to good but as expected the SVM is best one. It happens because, this method get better responses even with small de-calibration of the sensor, what happens during use. The k-NN method also shows good results but this method can easy misclassify the samples into mixed regions and transition regions between the output classes (as showed into the chapter 5). In the case of decision tree, good results were obtained, but in contrast to the SVM this classification does not get good responses to small de-calibration.

To the displays inspection the adaptive system demonstrated to be extremely important for the system accuracy, because it reduces a lot the external illumination noise into the console identification and mainly the displays inspection.

For the TFT display inspection be a prototype, the regions 19 and 20, that were the most problematic, the errors during the verification process were expected as these regions are difficult to be seen even by the user, in consequence of the overlay from the draws glass printed and the pixels regions. To the LCD display, the NNC make the system more effective and accurate, as expected from the use of this type of classification.

The workbench made possible the installation of the robot and the other devices necessary to the system operation. The application running into the Industrial computer is working and by remote desktop using a tablet what makes the process more intuitive to the user.

For the improvement of this work, apply the inspection workbench on another real tasks, create a function to auto-extract the consoles characteristics and, in conjunction with these, the creation of new loops to enumerate the consoles automatically and identify then using code bar for save the results of each console into the product characteristics file. This way, making an improvement of the production, traceability, quality control and, also facilitating the use of the system. Another improvement can be done applying deep learning method in all inspection process.

Bibliography

[1] R. Anderl, “Industrie 4.0 - advanced engineering of smart products and smart production 09 october 2014”, Jan. 2015.

[2] M. Aiman Kamarul Bahrin, F. Othman, N. Hayati Nor Azli, and M. Farihin Talib, “Industry 4.0: A review on industrial automation and robotic”, vol. 78, Jun. 2016.

[3] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris, “Design considerations for safe human-robot collaborative workplaces”, *Procedia CIRP*, vol. 37, pp. 248–253, 2015.

[4] M. Rooker, M. Hofmann, J. Minichberger, M. Ikeda, G. Ebenhofer, G. Fritz, and A. Pichler, *Quality inspection performed by a flexible robot system*, May 2014.

[5] P. J. McKerrow, “Robotics, an academic discipline?”, *Robotics*, vol. 2, no. 3, pp. 267–274, 1986.

[6] A. L. Branco, *"revoluções industriais: Primeira, segunda e terceira revoluções"*, `http://educacao.uol.com.br/disciplinas/geografia/revolucoes-industriais-primeira-segunda-e-terceira-revolucoes.htm`, Oct. 2007.

[7] J. J. Craig, *Introduction to robotics: Mechanics and control*, 3th. Upper Saddle River, New Jersey, United States of America: Pearson Prentice Hall, 2005.

[8] I. F. of Robotics, *"robot history"*, `https://ifr.org/robot-history`.

[9] H. Rattner, “Inovação tecnológica e planejamento estratégico na década de 80”, pt, *REVISTA DE ADMINISTRAÇÃO DE EMPRESAS*, vol. 24, pp. 86–93, Dec. 1984.

[10] V. M. F. Santos, *"robótica industrial"*, http://www.ece.ufrgs.br/~rventura/RoboticaIndustrial.pdf, 2003.

[11] K. Schwab, *"the fourth industrial revolution: What it means, how to respond"*, https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/, Jan. 2016.

[12] Exame, *"o brasil está pronto para a indústria 4.0?"*, http://exame.abril.com.br/publicidade/siemens/conteudo-patrocinado/o-brasil-esta-pronto-para-a-industria-4-0?utm_source=home&utm_medium=chamada&utm_campaign=nativa-siemen, May 2016.

[13] J. F. Neto, *"a 4ª revolução industrial"*, http://avozregional.com.br/2018/03/06/joao-francisco-neto-4a-revolucao-industrial/, Mar. 2018.

[14] C. N. da Indústria (CNI), *"desafios para a indústria 4.0 no brasil"*, http://www.portaldaindustria.com.br/publicacoes/2016/8/desafios-para-industria-40-no-brasil/, Aug. 2016.

[15] "Industry 4.0: A review on industrial automation and robotic", *Teknologi*, Sep. 2016.

[16] R. Bogue, "Europe continues to lead the way in the collaborative robot business", *Industrial Robot: The international journal of robotics research and application*, vol. 43, no. 1, pp. 6–11, 2016.

[17] I. O. F. S. (ISO), *"iso/ts 15066:2016, robots and robotic devices – collaborative robots"*, https://www.iso.org/standard/62996.html, Feb. 2016.

[18] Ł. Sobaszek and A. Gola, "Perspective and methods of human-industrial robots cooperation", in *Theory and Practice of Industrial and Production Engineering*, ser. Applied Mechanics and Materials, vol. 791, Trans Tech Publications, Oct. 2015, pp. 178–183.

[19] *"abb industrial robots"*, http://new.abb.com/products/robotics/industrial-robots/yumi.

[20] *"schneider electric implementa robots baxter en su cadena de montaje"*, http://www.infoplc.net/historias-exito/item/102267-schneider-electric-implementa-robots-baxter-cadena-montaje.

[21] *"universal robots products"*, https://www.universal-robots.com/media/442094/ur3_left.png.

[22] T. N. Andrzej Piotrowski, "Setting up a work centre based on the profibus network", *Applied Mechanics and Materials*, vol. 282, pp. 51–58, 2013.

[23] E. Costa, *"10 motivos que explicam a necessidade da indústria 4.0 no brasil"*, https://conteudo.startse.com.br/empreendedores/elena/10-motivos-que-explicam-necessidade-da-industria-4-0-no-brasil/, Oct. 2017.

[24] H. Herrero, A. A. Moughlbay, J. L. Outón, D. Sallé, and K. L. de Ipiña, "Skill based robot programming: Assembly, vision and workspace monitoring skill interaction", *Neurocomputing*, vol. 255, pp. 61–70, 2017.

[25] L. Bartevyan, *"industry 4.0 - summary report"*, https://www.cenit.com/fileadmin/dam/Corporate/PDFs/2015_5_Expertenwissen_E.pdf, 2015.

[26] L. Monostori, "Cyber-physical production systems: Roots, expectations and r&d challenges", *Procedia CIRP*, vol. 17, pp. 9–13, 2014.

[27] P. Leitão, V. Mařík, and P. Vrba, "Past, present, and future of industrial agent applications", *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2360–2372, Nov. 2013.

[28] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration", in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, p. 147.

[29] A. Mitra, *Fundamentals of quality control and improvement, solutions manual*, Jan. 2012.

[30] M. Kefer and J. Tian, “An intelligent robot for flexible quality inspection”, in *2016 IEEE International Conference on Information and Automation (ICIA)*, Aug. 2016, pp. 80–84.

[31] S. Y. Nof, *Handbook of industrial robotics*. New York, NY, USA: John Wiley & Sons, Inc., 1985.

[32] E. N. Malamas, E. G. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat, “A survey on industrial vision systems, applications and tools”, *Image and Vision Computing*, vol. 21, no. 2, pp. 171–188, 2003.

[33] A. Y. Jazi and J. J. Liu, “Handling class imbalance and multiple inspection objectives in design of industrial inspection system”, in *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, May 2017, pp. 606–611.

[34] Y.-H. Liu, S.-H. Lin, Y.-L. Hsueh, and M.-J. Lee, “Automatic target defect identification for tft-lcd array process inspection using kernel fcm-based fuzzy svdd ensemble”, *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 1978–1998, 2009.

[35] A. Pichler, S. C. Akkaladevi, M. Ikeda, M. Hofmann, M. Plasch, C. Wögerer, and G. Fritz, “Towards shared autonomy for robotic tasks in manufacturing”, *Procedia Manufacturing*, vol. 11, pp. 72–82, 2017.

[36] T. Haidegger, B. Benyó, L. Kovács, and Z. Benyó, “Force sensing and force control for surgical robots”, *IFAC Proceedings Volumes*, vol. 42, no. 12, pp. 401–406, 2009.

[37] M. Bélanger-Barrette, *"robotic product testing with force torque sensors"*, `https://blog.robotiq.com/bid/73329/Robotic-Product-Testing`, Feb. 2015.

[38] S. Cubero, N. Aleixos, E. Moltó, J. Gómez-Sanchis, and J. Blasco, “Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables”, *Food and Bioprocess Technology*, vol. 4, no. 4, pp. 487–504, May 2011.

[39] R. O. Duda, P. E. Hart, D. G. Stork, C. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification, 2nd ed*, 2001.

[40] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, May 1991.

[41] T. Cover and P. Hart, “Nearest neighbor pattern classification”, *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[42] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: Sentiment classification using machine learning techniques”, in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP ’02, Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86.

[43] G. Acciani, G. Brunetti, and G. Fornarelli, “Application of neural networks in optical inspection and classification of solder joints in surface mount technology”, *IEEE Transactions on Industrial Informatics*, vol. 2, no. 3, pp. 200–209, Aug. 2006.

[44] R. Shimizu, S. Yanagawa, T. Shimizu, M. Hamada, and T. Kuroda, “Convolutional neural network for industrial egg classification”, pp. 67–68, Nov. 2017.

[45] T. Y. Li, J. Z. Tsai, R. S. Chang, L. W. Ho, and C. F. Yang, “Pretest gap mura on tft lcds using the optical interference pattern sensing method and neural network classification”, *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 3976–3982, Sep. 2013.

[46] *"omron industrial automation"*, `https://industrial.omron.us/en/products/f3sg-re`.

[47] U. Robots, *"ur3 user manual"*, `https://www.universal-robots.com/media/207442/ur3_user_manual_en_global.pdf`.

[48] A. Vision, *"mako g data sheet g-125 (pdf)"*, `https://www.alliedvision.com/en/support/technical-documentation/mako-g-documentation.html`.

[49] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision.* USA: Thomson, 2008.

[50] *"matlab support regionprops"*, `https://www.mathworks.com/help/images/ref/regionprops.html`.

[51] *"neopixel ring - 16 x 5050 rgb led with integrated drivers"*, `https://www.adafruit.com/product/1463`.

[52] F. Almada-Lobo, "The industry 4.0 revolution and the future of manufacturing execution systems (mes)", vol. 3, pp. 16–21, Jan. 2016.

[53] U. Robots, *"the urscript programming language"*, `https://s3-eu-west-1.amazonaws.com/ur-support-site/15731/scriptmanual_en.pdf`, "January" 2015.

[54] J. Axelson, *Usb complete: The developer's guide*, 4th. Lakeview Research, 2009.

[55] W. R. Stevens and K. Fall, *Tcp/ip illustrated: The protocols v. 1*, 2nd. USA: Addison-Wesley Profissional, 2011.

[56] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey", *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[57] S. Russell and P. Norvig, *Artificial intelligence: A modern approach*, Third. Upper Saddle River, NJ: Prentice Hall, 2010.

[58] I. H. Witten, E. Frank, and M. A. Hall, *Data mining: Practical machine learning tools and techniques*, 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011, ISBN: 0123748569, 9780123748560.

[59] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, Secound. New York: Wiley-Interscience, 2001.

[60] B. Ferreira, R. G. Silva, and V. Pereira, "Feature selection using non-binary decision trees applied to condition monitoring", in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–7.

[61] L. Zhang, W. Zhou, and L. Jiao, "Wavelet support vector machine", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 34–39, Feb. 2004.

[62] *"support vector machines for binary classification"*, `https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html`.

[63] T. Denoeux, "A k-nearest neighbor classification rule based on dempster-shafer theory", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 5, pp. 804–813, May 1995.

[64] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to algorithms*, 2nd. McGraw-Hill Higher Education, 2001.

[65] C. Jian, J. Gao, and Y. Ao, "Automatic surface defect detection for mobile phone screen glass based on machine vision", *Applied Soft Computing*, vol. 52, pp. 348–358, 2017.

[66] J. Lewis, "Fast normalized cross-correlation", vol. 10, Oct. 2001.

[67] J. Zheng, J. Tian, K. Deng, X. Dai, X. Zhang, and M. Xu, "Salient feature region: A new method for retinal image registration", *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 221–232, Mar. 2011.

[68] Y. Hel-Or, H. Hel-Or, and E. David, "Matching by tone mapping: Photometric invariant template matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 317–330, Feb. 2014.

[69] *"normalized 2-d cross-correlation"*, `https://www.mathworks.com/help/images/ref/normxcorr2.html`.

[70] *"correlation coefficient"*, `https://www.mathworks.com/help/matlab/ref/corrcoef.html`.

[71] S. S. Haykin, *Neural networks and learning machines*, Third. Upper Saddle River, NJ: Pearson Education, 2009.

[72] L. Fausett, Ed., *Fundamentals of neural networks: Architectures, algorithms, and applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994.

[73] *"hyperbolic tangent sigmoid transfer function"*, `https://www.mathworks.com/help/nnet/ref/tansig.html`.

[74] K. Hornik, “Approximation capabilities of multilayer feedforward networks”, *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

Annex A

TFT display regions.

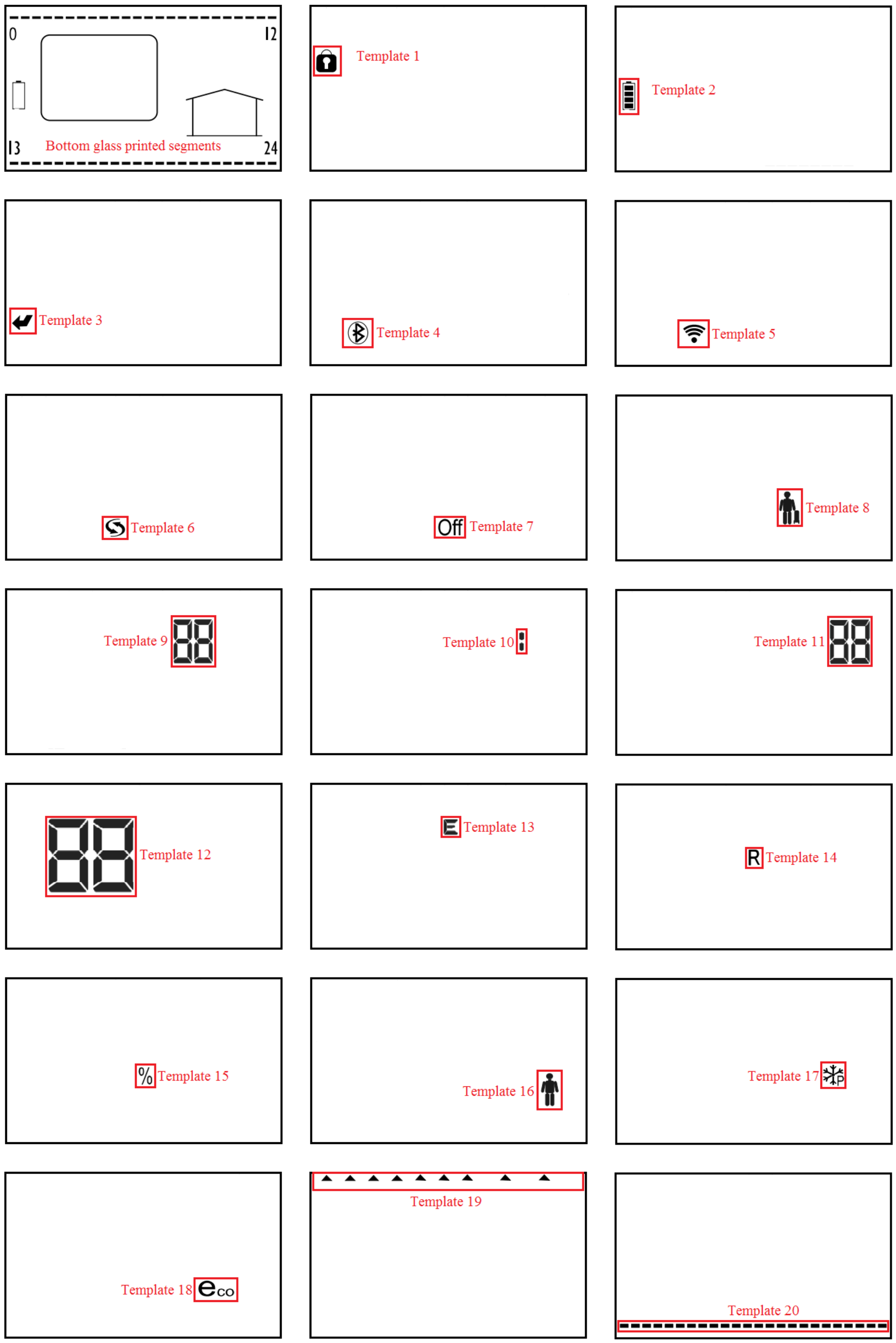


Figure A.1: Illustration from TFT display templates

Annex B

Inspection Workbench Flowchart.

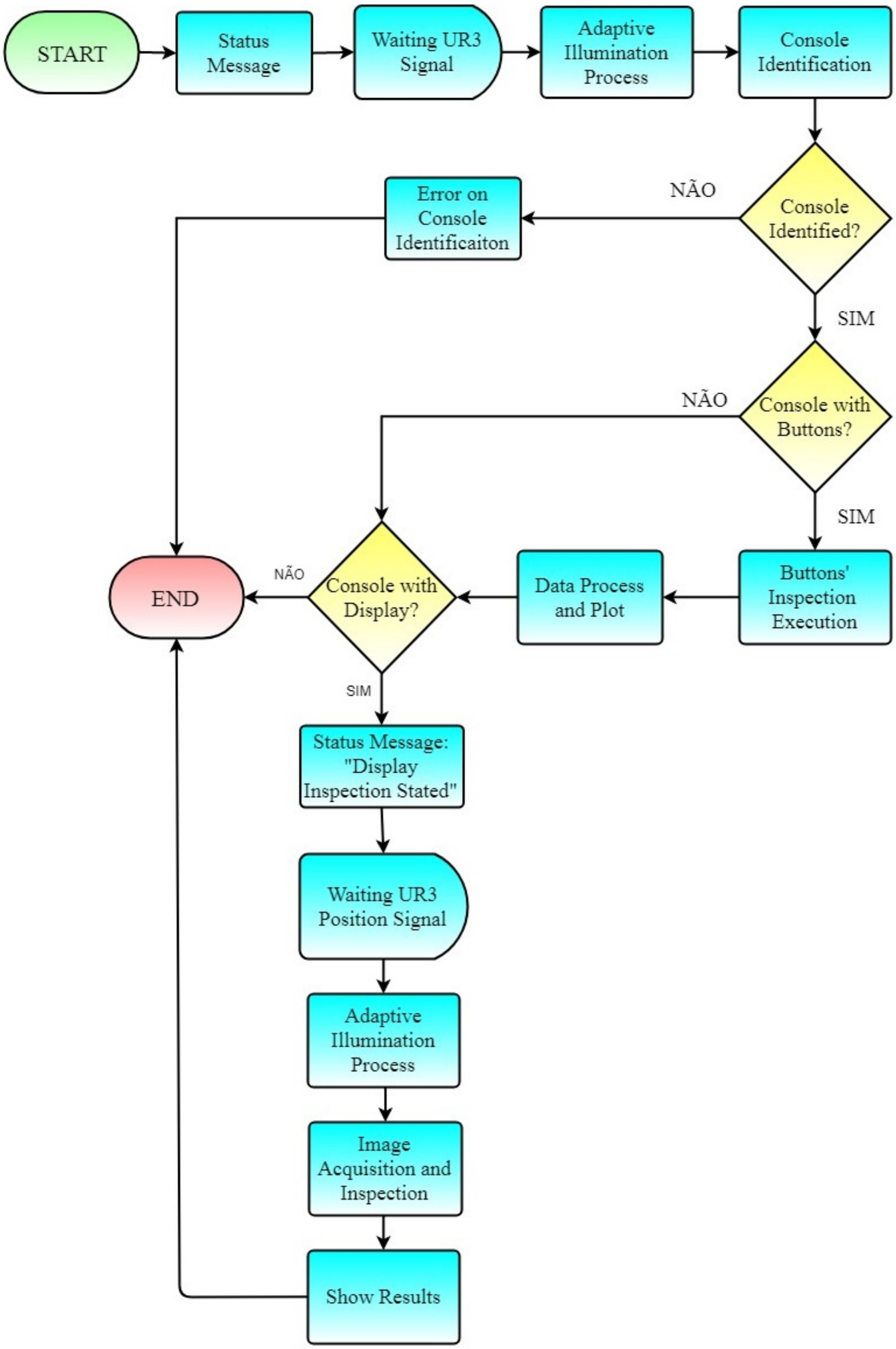
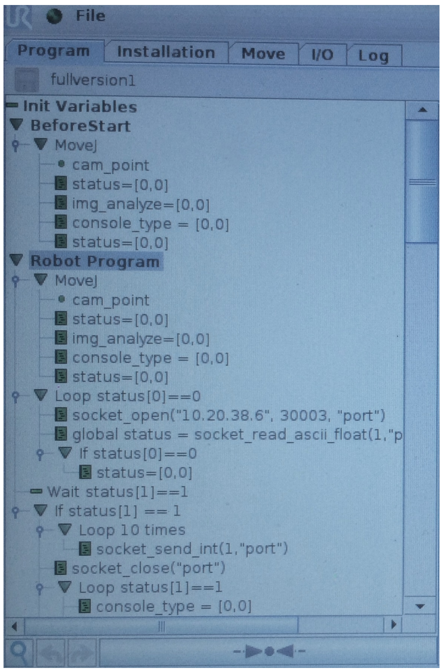


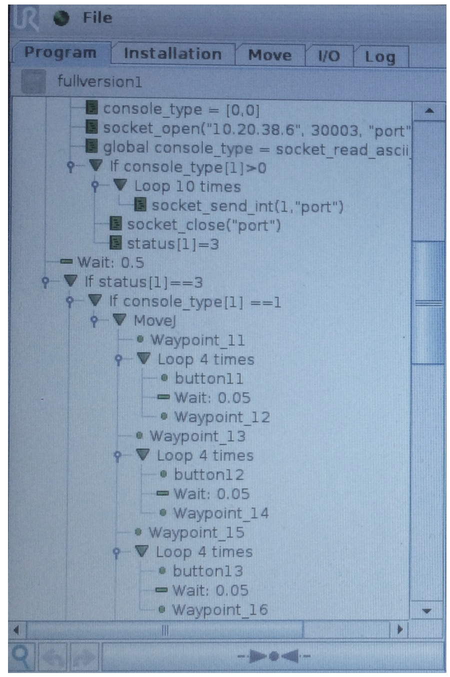
Figure B.1: Generic flowchart to accomplish the entire inspection method.

Annex C

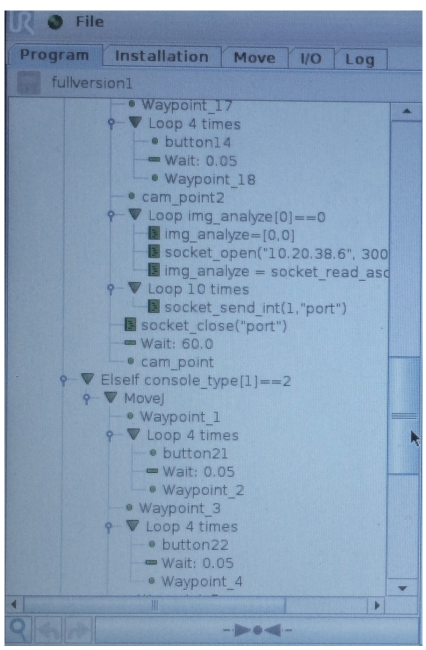
Programming script on UR3 console



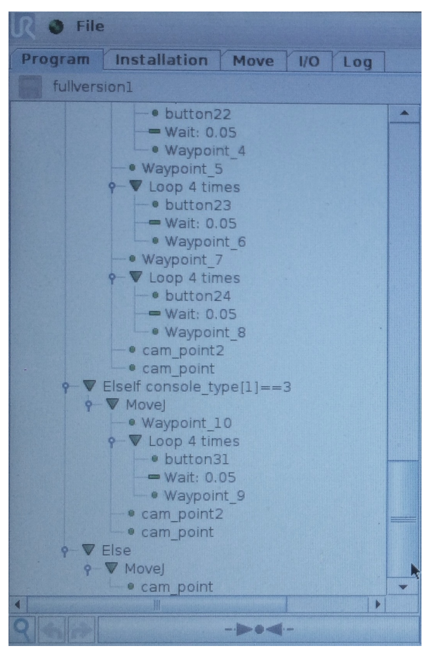
(a) UR3 Console program (part 1).



(b) UR3 Console program (part 2).



(c) UR3 Console program (part 3).



(d) UR3 Console program (part 5).

Figure C.1: Console program scrip.