وزارة البحث العلمي والتعليم العالي
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جامعـة عبد الحميد بن باديس مستغانم
University Abdelhamid Ibn Badis Mostaganem
كـليـة الـعـلـوم و الـتـكـنـولـوجـيـا
Faculty of Sciences andTechnology
DEPARTEMENT DE GENIE DES ELECTRONIQUE

N° d'ordre: M ....../GE/2022

THESIS

Presented to obtain a diploma in

MASTER IN ELECTRONICS

Field: Embedded systems

TITLE:
CONTROLING ROBOT HAND USING AI

Prepared by	Supervisor:
Bendref Abd el wadoud	**Merah Mostefa**

Board of Examiners

Président :	.	MC	Université de Mostaganem
Examinateur 1:	.	MC	Université de Mostaganem
Examinateur 2:	.	MA	Université de Mostaganem
Rapporteur :	.	MC	Université de Mostaganem
Sous-rapporteur :	.	MA	Université de Mostaganem

Abstract:

artificial intelligence is having an exponential effect on today's world ,it can used in a very various fields in human’s life to solve daily problems ,recently, a research center applied artificial intelligence to detect cancer and COVID-19 earlier, alot of pharmaceutical companies are using AI to accelerate drug and vaccine development, , From just these examples, it's obvious that AI is improving All the fields but “does factories use it to provide the protection for employees from work accidents ?” Artificial intelligence can reduce the amount of workplace accidents and provide the security and safety in this field

Keywords: AI; artificial intelligence; safety, security

Résumé :

l'intelligence artificielle a un effet exponentiel sur le monde d'aujourd'hui, elle peut être utilisée dans de nombreux domaines de la vie humaine pour résoudre des problèmes quotidiens, récemment, un centre de recherche a appliqué l'intelligence artificielle pour détecter le cancer et le COVID-19 plus tôt, de nombreuses sociétés pharmaceutiques utilisent L'IA pour accélérer le développement de médicaments et de vaccins, A partir de ces seuls exemples, il est évident que l'IA s'améliore dans tous les domaines mais "les usines l'utilisent-elles pour protéger les employés contre les accidents du travail ?" L'intelligence artificielle peut réduire le nombre d'accidents du travail et assurer la sécurité et la sûreté dans ce domaine**.**

Mots clés : l'intelligence artificielle :AI : sûreté : sécurité

الملخص :

للذكاء الاصطناعي تأثير أسي على عالم اليوم ، يمكن استخدامه في مجالات مختلفة جدًا في حياة الإنسان لحل المشكلات اليومية ، مؤخرًا ، قام مركز أبحاث بتطبيق الذكاء الاصطناعي للكشف عن السرطان و COVID-19 سابقًا ، تستخدم الكثير من شركات الأدوية الذكاء الاصطناعي لتسريع تطوير الأدوية واللقاحات ، من خلال هذه الأمثلة فقط ، من الواضح أن الذكاء الاصطناعي يعمل على تحسين جميع المجالات ولكن "هل تستخدمه المصانع لتوفير الحماية للموظفين من حوادث العمل؟" يمكن للذكاء الاصطناعي تقليل عدد حوادث العمل وتوفير الأمن والحماية في هذا المجال.

الكلمات الأساسية: الذكاء الاصطناعي :AI : الامن :الحماية .

Dedication:

I dedicate this dissertation to my family, I am deeply grateful to my parents for their love, patience, confidence, and their financial support through the many years of my education.

To my dear brother Abdeldjalil

To my English teacher Larbi Youcef Ouarda
For all her efforts, support and encouragement

To my friend Mustapha for his support

Acknowledgments:

First, I would like to thank Allah for giving me strength and capacity to complete this work.

I would like to express my deepest gratitude to my supervisor Mr. Mostefa MERAH for his support, guidance and advice. Without his generous help, I would never have successfully completed my dissertation. I am eternally thankful to him.

I am deeply grateful to all my teachers who have constantly been source of knowledge and enlightenment throughout the different stages of my education

In addition I would like to extend my special thanks to the members of jury for devoting a part of their precious time for reading and evaluating this work.

Table of content:

Abstract……………………………………….…………………………………………..…2

Dedications…………………………………..……………………………….………3

Acknowledgments…………………….…………………………………..………….……4

Table of content…………………………………………………………..….…..……5

List of Figures and Tables ……………………………………………….……………9

List of abbreviation ……………………………………………………………….……11

General Introduction ………………………………………………………...……12

Chapter I: HMI (Human–machine interface) ………………………………..…13

I.1. Introduction**……………………………………………………………...………14**

I.1.2 Human–computer interface**……………………………………….……..………15**

I.2.1 Visual-based**…….……………..…………..…………..…………..……………..……………16**

I.2.2 Audio Based**…….………………..……………..………………..…………..……..……………15**

I.2.3 Machine environment**…….………….……………..…………………..….………15**

I.2.4 Areas of the interface**…….………….………….………….………..…….……..……15**

I.2.5 Input flow**…….………….………….………….…………………………….…………15**

I.2.6 Output**…….………….………….………….……………………………………...…15**

I.2.7 Feedback**…………………………………………………………………………………15**

I.2.8 Fit**………………………..……………………………………………………….………15**

I.3 Different types of HMI**……………….………………………………………..……16**

Table of Content

I.4 the Pushbutton Replacer……………..…………………………..…………………….……16

I.5 the Data Handler…………..………………………………..……………………………16

I.6 the Overseer……………..…………………………………………………..…..……17

I.8 Properties of an HMI……………..…………………………………………..…..……17

I.10 Safe and secure HMIs……………..……………………………………………………17

I.11 Industrial Security…………………..…………………………………..……..……18

I.12 Conclusion…………………..………………………………………………………19

Chapter II: Image processing (python) ……………………………….…..………20

II.1 Introduction……………………………………………………………………..…………21

II.2 Applications………………………………………………………………………………21

II.3 Computer vision……………………………………………………………………..……22

II.4 Convolutional Neural Networks………………………………………….………………22

II.4.1 Definition…………………………………………………………..…………………22

II.4.2 how do convolutional neural networks work? ……………………………………………23

II.4.2.1 Convolutional Layer…………………………………………..………………23

II.4.2.2 Pooling Layer………………………………………………………………………24

II.4.2.3 Fully-Connected Layer………………………………………..…………………24

II.5 Hand detection……………………………………………………………………….……24

II.6 Handtracking……………………………………………………...…………………26

II.7 Hand Landmark Model…………………………………………………………………27

II.8 Right and Left Hand Detection: ……………………………………………..…………27

II.9 Finger state detection: ……………………………………………………….………29

II.10 Conclusion: …………………………………………………………………..………30

Table of Content

Chapter III: Software and hardware of the controlling system………………31

III.1 Introduction……………………………………………………………..………32

III.2 Hardware side……………………………………………………….………..32

III.3 The electronic card Arduino Uno………………………………..……………32

III.3.1 Arduino Uno definition…………………………..…………………………34

III.3.2 Arduino Uno specifics………………………………………………………34

III.4 Servo motors MG996R…………………………………….…………………35

III.5 3D printed hand……………………………………………………….…………36

III.6 fishing wire………………………………………………………………………37

III.7 the pc…………………………………………………………….…….…………37

III.8 Serial communication Protocol UART…………………….…..………………37

III.8.1 Definition…………………………………………………….…..…………….37

III.8.2 Data Transmission………………………………………………….……….39

III.8.3 Start Bit………………………………………………………….……………..39

III.8.4 Data Frame……………………………………………………………..………40

III.8.5 Parity…………………………………………………………..…………40

III.8.6 Stop Bits……………………………………………………………………..41

III.8.7 Steps of UART Transmission……………………………….……………41

III.9 Software side…………………………………………………………………41

III.9.1 Python 3……………………………………………………………………..41

III.9.2 PyCharm IDE………………………………………………………………..42

III.9.3 Arduino IDE…………………………………………………….………..42

Table of Content

III.10 Conclusion…………………………………………………..…………………………..44

Chapter IV: Experimental results……………………….…………………………………45

IV.1 Introduction……………………………………………..………………………………46

IV.2 The created language……………………………………………………………………47

IV.3 Conclusion **……………………………………………………………...……………..52**

General Conclusion……………………………………………………………………….53

References…..……………………………………………………………...………54

List of Figures and Tables

Figures:

Figure 1: classic visual based   HMI……………………………………………………….14

Figure 2: visual based HMI using VR…………………………….………………….15

Figure 3: Amazon Product Alexa…………………………..…………………………15

Figure 4: HMI diagram……………………………………………………………..…16

Figure 5: General process of our hybrid detection/reconstruction framework…………………26

Figure 6: 21 Hand Landmarks……………………………………………………..27

Figure 7: right left detection…………………………………………………………..28

Figure 8: left hand detection………………………………………………………….28

Figure 9: organizational chart of Arduino program……………………………………..29

Figure 10: Synoptic diagram………………………………………………….………32

Figure 11: history of Arduino…….…………………………………….……………33

Figure 12: Arduino Uno…………………………………………………………………34

Figure 13: Servo motors MG996R…………………………………….….…………35

Figure 14: robotic hand……………………………………………….……….……..36

Figure 15: Fishing wire…………………………………………….……..……..……37

Figure 16: Two UARTs directly communicate with each other…………..…….….….……37

Figure 16.1: UART with data bus…………………………………………….….…….38

Figure16.2: UART packet………………………………………………….….….……39

Figure 16.3: Start bit………………………………………..……………..…..………39

Figure 16.4: Data frame……………………………………..……………..………40

Figure 16.5: Parity bits…………………………………..………………….………40

Figure16.6: Stop bits……………………………...…………………………….…..…41

Figure 17: PyCharm logo………………………….………………………….…..….…..42

Figure18: Arduino IDE logo………………………………………………………..…..43

List of Figures and Tables

Figure 19: Organizational chart of Arduino program………………..……………………………..43

Tables:

Table 1: Arduino Uno specifics……………………………………………………………35

Table2: Servo motors MG996R specifics…………………………………………………….36

Table 3: UART Summary…………………………………………………………………….38

Table4: hand gestures signification...46

List of Abbreviations

AI: Artificial Intelligence

ML: Machin Learning.

HMI: Human-Machine Interface.

HCI: human-computer interface.

CNN: convolutional neural network.

RNN: recurrent neural network.

GPU: graphics processing unit.

RPN: region proposal network

GAN: Generative Adversarial Network

RGB: Red-Green-Blue

ROI: region of interest

MCU: microcontrollers

IDE: integrated development. Environment

General Introduction

Artificial Intelligence is an approach to create a computer, a robot, or a product to think how smart human think. AI can be defined as a study of how human brain think, learn, decide and work, when it tries to solve problems. And at last this study outputs intelligent software systems. The aim of AI is to enhance computer functions which are associated with human knowledge, for instance, reasoning, learning, and problem-solving. Building an AI system may be a careful process of reverse-engineering human traits and capabilities in a machine, and using its computational prowess to surpass what we are capable of.

To understand How AI actually works, one has to deep dive into the varied sub-domains of artificial intelligence and understand how those domains might be applied to the varied fields of the industry. We use Machine Learning to teach a machine the way to make inferences and decisions based on past experience. It identifies patterns, analyses past data to infer the meaning of those data points to reach a possible conclusion without having to involve human experience. This automation to achieve conclusions by evaluating data, saves a human time and helps them make an improved decision. Form Machine Learning techniques there is Deep Learning it teaches a machine to process inputs through layers in order to classify, infer and predict the outcome.

With a similar principles to Human Neural cells it exist Neural Networks which is algorithms that catch the relationship between various important variables and processes the data as a human brain does. In order to make the system understand images computer vison is needed which is algorithms which study different parts of the object. Those algorithms helps the machine classify and learn from a data base of images to make a better output decision

Chapter I:

«HMI (Human–machine interface)»

Chapter I: HMI (Human-machine interface)

I.1 Introduction:

Human Machine Interface, often known by the acronym HMI, refers to a dashboard or display used to control machinery. Line operators, supervisors in industry rely on HMIs to translate complex data into useful information [1].

For example, they use HMIs to monitor machines to make sure they are working properly. Easy-to-understand visual displays give meaning and context to near real-time information about tank levels, pressure, engine and valve status, and other variables.

Today's HMIs allow supervisors to do much more than control processes. By using trend data, they offer vast new opportunities to improve product quality and make systems safer and more efficient.

I.2 Human–computer interface;

The human-computer interface can be defined as the point of communication between the human user and the computer. The flow of information between human and computer is defined as the interaction loop. The interaction loop has several aspects, including:

I.2.1 Visual-based: Visual-based human-computer interaction is probably the most prevalent area of human-computer interaction (HCI) research.

Figure 5: classic visual based HMI

Figure 6: visual based HMI using VR

I.2.2 Audio Based: Computer-human voice interaction is another important area of HCI systems. This area deals with information obtained by various audio signals. [2]

Task environment: smart home (Alexa) and other user defined conditions and goals.

Figure 7: Amazon Product Alexa

I.2.3 Machine environment: The computer environment is connected, for example, to a laptop computer in the student residence.

I.2.4 Areas of the interface: Non-overlapping domains include human and computer processes, not their interaction. Meanwhile, the overlapping regions are concerned only with their interaction processes.

I.2.5 Input flow: The flow of information in a task environment begins when the user has a task that requires the use of his computer.

I.2.6 Output: The flow of information that originates in the machine environment.

I.2.7 Feedback: Loops through the interface that evaluate, modify, and confirm processes as they go from human through the interface to the computer and back.

I.2.8 Fit: It matches the design of the computer, the user, and the task of optimizing the human resources needed to complete the task.

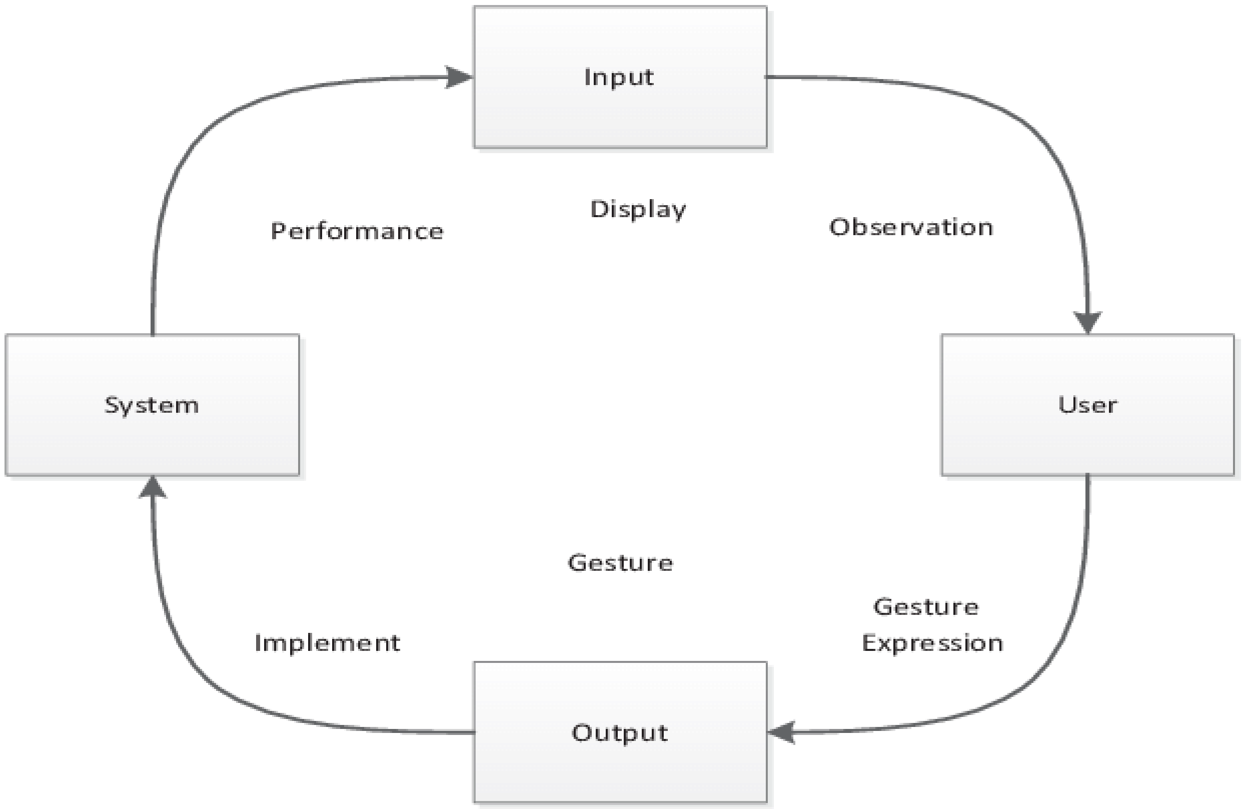


Figure 8: HMI diagram

I.3 Different types of HMI:

Human Machine Interfaces (HMIs) is a multibillion dollar industry that continues to grow and expand year after year. In fact, current estimates expect the industry to reach $6.31 billion by 2023[3].

I.3.1 the Pushbutton Replacer

One of the most common types of HMIs is push-button replacement. As the name implies, it is a central control panel made up of several buttons, each with a specific function. Many industrial facilities use push button alternatives to improve their efficiency and productivity. It allows workers to execute different commands on one or more devices by using one convenient interface. Centralizing all controls in one place is a huge improvement in machine operation manually.

I.3.2 the Data Handler

Another commonly used type of HMI is the data processor. HMI is mainly used to collect/retrieve data, in which case it can be sent to a hard disk or printed, depending on the user's command. Data manipulators are especially useful in applications that involve large amounts of data. If a company needs to collect data from a device or piece of equipment, it will likely use a data manager HMI.

I.3.3 the Overseer

The third type of HMI is the supervisor, which usually runs on the Windows operating system. It takes a visual approach to the interaction between a human operator and a machine, allowing a graphical interface via an electronic screen or touch screen.

I.4 Properties of an HMI

All HMIs contain a processor and memory. It is important to ensure that the processor and memory capabilities are sufficient to control a particular system.

Additional physical characteristics of HMI vary from model to model, application to application, and manufacturer to manufacturer. A water treatment plant HMI may have several water seals around its perimeter, as opposed to a drug depot HMI which does not require a seal [3].

HMI size is also a major physical property that varies, as not all applications require a large, high-resolution screen. Some applications may require a small black and white touch screen only. When it comes to choosing an HMI, the physical characteristics are very important as the operating environment and security measures that the HMI needs to protect itself must be considered. Also, a certain size may be required due to space limitations.

I.5 What are they used for?

HMI stands for Human Machine Interface. It is an electronic control panel that connects the user to a device or Machine. HMIs include anything connected to a machine, from MP3 players to industrial equipment.

HMIs are used for all types of machines, systems, and devices. They create a central control center to display data and perform more complex operations, such as turning machines on or off, by communicating with programmable logic controllers (PLCs) and input/output sensors. HMIs must be durable enough to handle fluctuations in temperature, humidity, and industrial environmental factors such as hazardous chemicals.

I.6 Safe and secure HMIs:

When it comes to human-machine interfaces (HMIs), the distinction between safety and security is often well defined. Security refers to "the controls built into PLC systems and security locks," says Stephen Garbrecht, marketing program manager for platform products and infrastructure for Wonderware [4].

I.7Keeping control

From an HMI perspective, Garbrecht explains, there are three main scenarios: "One is someone from outside the company coming in through firewalls, going through the network, and doing something with the HMI. The second is someone within the company who wants to do something malicious for some reason. The third would be someone who isn't trying to be malicious, works for the company, but just does something they shouldn't be doing and makes a mistake or causes security or some kind of glitch in the process .Clark calls this philosophy "having limited threat vectors." [5] An ideal secure control system, according to him: is isolated from all threats, including corporate business enterprises;

- is layered with aggressive anti-penetration devices;
- has only one point of ingress/egress;
- contains all the system automation within a secure bubble; and
- Allows each trusted machine within the enterprise to have unimpeded, unlimited access to any other trusted machine.

I.8 Industrial Security:

In order to protect the employees   it is necessary to use a visual based HMI to control machines and complete tasks without employee’s presence in with the danger zone in industry

I.9Industrial Security objectives:

- Preventing work-related fatalities, disabling injuries, illness, and damage to machinery or materials
- Ensuring continued production by preventing disruptive incidents
- Reducing workers compensation costs, maintaining lower insurance rates, and minimizing indirect costs associated with accidents
- Strengthening safety culture and increasing employee morale

I.10 Conclusion:

In this chapter, we've seen the different types of HMIs that exist, and HMIs are commonly used in industrial settings, such as food, manufacturing, and energy. Because of their ability to collect data, integrate information, and provide visuals, HMIs are useful for industrial companies that need a central system to interact with their equipment.

Interaction with the machine is not only mastering the physical interface between the human user and the system, but also the mastery of the mental model applied in the engineering and logic of the machine. If the device design is well thought out and user-centric, it should reflect the mental model of the user. Communication with/supervision of a machine, or any automated system, is a matter of human performance, and as such must always include reasonable care to avoid laxity and excessive dependence with respect to the machine.

In the next chapter, we will focus on HMIs that are visible using Python.

Chapter II

«Image processing (python)»

Chapter II : Image Processing (python)

II.1 Introduction:

Image processing is a way to perform an operation on an image, to obtain an improved image or to extract useful information from it. It is a type of signal processing where the input is an image and the output can be an image or properties/characteristics associated with that image. Nowadays, image processing is one of the booming technologies. It is also a main area of research in the disciplines of engineering and computer science.

Image processing mainly includes the following three steps:

- Importing images via image acquisition tools.
- Image analysis and processing.
- The output through which the image can be modified or the report based on the analysis of the image.

Here are two types of methods used for image processing, analog and digital image processing. Analog image processing can be used for prints such as prints and photographs. Image analysts use different fundamentals for interpretation when using these visual techniques. Digital image processing techniques help in the processing of digital images using computers. The three general steps that all types of data must go through when using digital technology are preprocessing, enhancement, display, and information extraction.

II.2 Applications:

Below we list some of the main areas in which digital image processing is widely used.

- Sharpness improved and image restored
- Medical field
- Send and Encrypt
- Machine / Robot Vision
- Pattern Recognition
- Video processing

In this project, we focus on digital image processing for machine and robot vision and safety

II.3 Computer vision:

Computer vision is an area of artificial intelligence (AI) that allows computers and systems to extract meaningful information from digital images, video, and other visual inputs—and take actions or make recommendations based on that information. If artificial intelligence allows computers to think, then computer vision allows them to see, observe and understand.

How does computer vision work?

Computer vision needs a lot of data. It analyzes the data over and over until it distinguishes the differences and eventually recognizes the images. For example, to train a computer to recognize car tires, it must receive large amounts of tire images and tire-related items to find out the differences and recognize tires, especially flawless tires.

Two main technologies are used to achieve this: a type of machine learning called deep learning and a convolutional neural network (CNN).

Machine learning uses algorithmic models that allow a computer to learn about the context of visual data. If enough data is entered into the form, the computer will "look" at the data and learn to distinguish one image from another. Algorithms allow the machine to learn on its own, rather than having someone program it to recognize the image.

CNN assists machine learning or a deep learning "look" model by dividing images into pixels to which tags or labels are assigned. It uses labels to perform convolutions (arithmetic on two functions to produce a third function) and makes predictions about what it "sees". The neural network performs convolutions and validates its predictions in a series of iterations until the predictions start validating. It is, then, a matter of recognizing or seeing images in a way similar to humans.

Much like a human making an image from a distance, CNN first distinguishes hard edges and simple shapes, and then fills in the information as it runs its prediction iterations. CNN is used to understand individual images. A recurrent neural network (RNN) is similarly used for video applications to help computers understand how images in a series of images relate to each other.

II.4 Convolutional Neural Networks

II.4.1 Definition:

To quote from the Neural Networks Learn Hub article, neural networks are a subset of machine learning and are at the heart of deep learning algorithms. It consists of layers of nodes, containing an input layer, one or more hidden layers and an output layer. Each node is connected to another and has an associated weight and threshold. If the output of an individual node is greater than the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data will be passed to the next network layer.

Although we mainly focused on feed-forward networks in this article, there are different types of neural networks, which are used for different use cases and types of data. For example, recurrent neural networks are most commonly used for natural language processing and speech recognition, while convolutional neural networks (ConvNets or CNNs) are often used for classification and computer vision tasks. Before CNNs, manual and time-consuming feature extraction methods were used to identify objects in images. However, convolutional neural networks now offer a more scalable approach to image classification and object recognition tasks, by exploiting the principles of linear algebra, in particular matrix multiplication, to identify patterns in an image. However, they can be computationally demanding, and require graphics processing units (GPUs) to train models.

II.4.2 how do convolutional neural networks work?

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech or sound signal inputs. They have three main types of classes, namely:

- Convolutional layer
- Layer Assembly
- Fully Connected Layer (FC)

The convolutional layer is the first layer of the convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully connected layer is the final layer. With each layer, the CNN increases in complexity, identifying larger parts of the image. Previous layers focus on simple features, such as colors and outlines. As the image data progresses through the CNN layers, it begins to recognize larger features or shapes of the object until it finally determines the intended object.

II.4.2.1 Convolutional Layer

The convolutional layer is the central building block of a CNN, and is where the majority of computations occur. It requires few components, namely input data, filter, and feature map.

Suppose the input is a color image, consisting of a 3D matrix. This means that the input will have three dimensions - height, width, and depth - which correspond to the RGB in the image. We also have a feature detector, also known as a kernel or filter, which will cycle through the receptive fields of the image, to check if the feature is present. This process is known as torsion.

II.4.2.2 Pooling Layer

Bundling of layers, also known as downsampling, helps reduce dimensionality, thereby reducing the number of parameters in the input. Similar to the convolutional layer, the grouping operation sweeps a filter over the entire input, but the difference is that this filter has no weight. Instead, the kernel applies an aggregate function to the values in the receptive field, filling the output array.

II.4.2.3 Fully-Connected Layer

The name of the fully connected layer describes itself well. As mentioned earlier, the pixel values of the input image are not directly related to the output layer in partially connected layers. However, in the fully connected layer, each node in the output layer is directly connected to a node in the previous layer.

This layer performs a classification task based on the features extracted from the previous layers and their various filters. While convolutional and aggregate layers tend to use ReLu functions, FC layers typically make use of the softmax activation function to appropriately classify the input, yielding a probability of 0 to 1.

II.5 Hand detection:

Hand detection is an important preprocessing procedure for many computer vision tasks related to the human hand, such as hand position estimation, hand gesture recognition, hand gesture analysis, human activity, etc. However, reliable detection of multiple hands from crowded scenes remains a challenging task due to the variety of complex appearances of skilled human hands (eg, different hand shapes, skin colours, lighting, directions, scales, etc.) in color images. To address this issue, an accurate hand detection method is proposed to reliably detect multiple hands from a single color image using a hybrid convolutional neural network (CNN) discovery/reconstruct framework, in which hand regions are detected and hand semblances are reconstructed in parallel by sharing extracted features From the area proposal layer, the proposed model is trained from start to finish. Moreover, it has been observed that the Generative Adversarial Network (GAN) can improve detection performance by generating more realistic hand looks. Experimental results showed the superiority of the proposed method to the latest scientific findings in the field of detecting difficult hand landmarks to the public [8].

The human hand plays a very important role in communication when people interact with each other and with the environment in everyday life. Hand gesture recognition and human activities are closely related to the positions and directions of human hands. Therefore, reliable detection of the human hand [1] from monochrome images or videos captured from common image sensors plays an important role in many applications related to computer vision, such as human-computer interaction [2,3] and human hand posture estimation. [4, 5, 6], Recognition of human gestures [7, 8], Analysis of human activity [9].

In computer vision, the hand-related application pipeline typically contains three steps: (1) hand detection, (2) hand position estimation, and (3) static recognition gestures or dynamic action recognition. The second step is optional because the gesture/action recognition can be done with or without hand position estimation. About five years ago, manual position estimation and action recognition were among the most difficult steps (or bottlenecks) in the pipeline, even in restricted environments.

An environment (usually one hand and a simple background in an image) in which the hand can be easily detected or assumed to have already been cropped. Therefore, the hand-tied community focused mainly on the second and third phases in the past. However, at present, manual position estimation and gesture/action recognition in restricted environments are reaching maturity, and hand-related applications in an unconstrained environment (the complex background and number of hands in the image is unknown) will be an important trend in the future. Under this condition, hand detection in a free environment becomes a new bottleneck in hand-related business. Thus, a high-resolution hand detection method will be a critical step in the pipeline of hand-related applications in a stress-free environment. In this article, we focus on the hand detection algorithm. Conventional manual detection methods mainly use low-level image features such as skin color [10] and Fig. [5] to detect the area of the hand. Nowadays, detection methods based on Convolutional Neural Networks (CNN) [13, 14, 15, 16, 17, 18, 19] have proven to be more robust and accurate [20, 21, 22] due to the deeply learned discriminant properties. However, compared to common objects, human hands are highly articulated, appearing in different directions, scales, shapes, skin colours, and sometimes partial occlusions. Therefore, reliable detection of multiple human hands from massive unconstrained scenes remains a challenging problem.

The general process of the proposed approach is illustrated in Figure 1. First, the feature maps of the entire input image are extracted by the common convolutional layers. They are then fed into a region proposal network (RPN) to generate potential region proposals, also known as a region of interest (RoI). Finally, RoIs feature maps are used to classify the corresponding labels (hand/bottom), to optimize the locations of the detected scorpions and simultaneously reconstruct hand appearances [23].

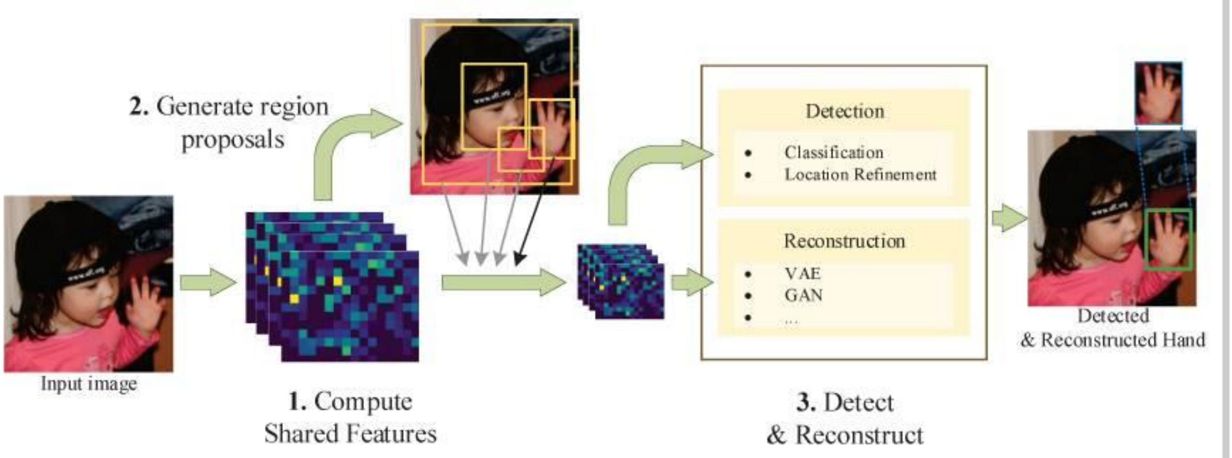


Figure 5: General process of our hybrid detection/reconstruction framework.

II.6 Handtracking:

To provide a stable manual tracking process, we use the mediapipe library that gives the user the ability to perceive the shape and movement of the hands can be a key element to improving the user experience in a variety of fields and technology platforms. For example, it could form the basis for understanding sign language and controlling hand gestures, and it could also allow digital content and information to be superimposed on the physical world in augmented reality. Although it comes naturally to people, strong real-time perception is a very challenging task for computer vision, as hands often clog themselves or each other (eg finger/palm occlusion, handshake) and lack high contrast patterns .

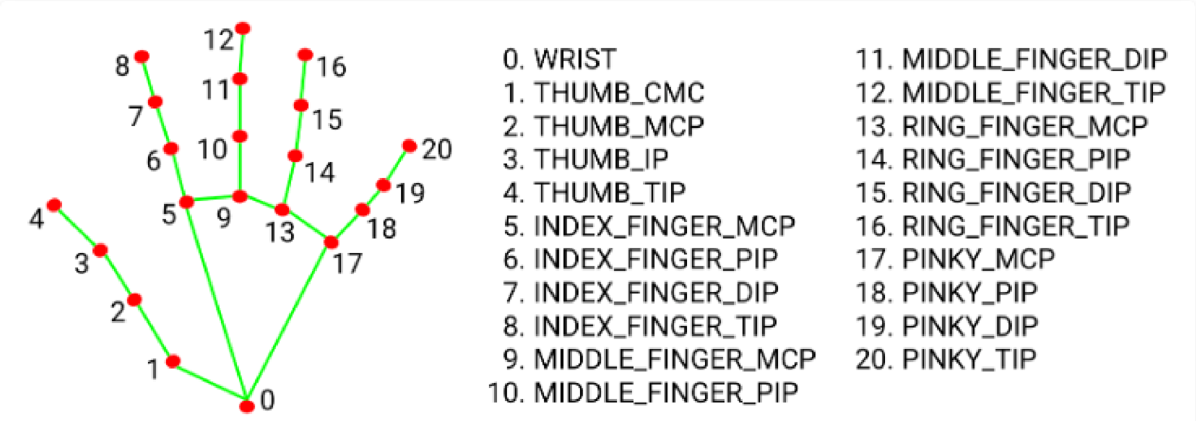
Mediapipe hands module:

MediaPipe Hands is a highly accurate hand and finger tracking solution. Machine learning (ML) is used to infer 21 3D landmarks of the hand from a single image. While current state-of-the-art methods rely primarily on robust desktop environments for inference, our method achieves real-time performance on a mobile phone and even accommodates multiple hands. We hope that introducing this manual perception function to the broader R&D community will lead to the emergence of creative use cases, and stimulate new applications and avenues of research.

II.7 Hand Landmark Model:

The manual landmark model performs precise localization of key points of 21 3D manual articulator coordinates within the regions of the hand detected via regression and is a direct prediction of the coordinates. The model learns a constant internal representation of hand position and is robust even for partially visible and self-obstructed hands.

For basic fact data, the model contains approximately 30,000 photorealistic images annotated with 21 3D coordinates, as shown below. To better cover potential hand positions and provide additional supervision of the nature of hand geometry, this method presents a high-quality artificial hand model on different backgrounds and mapped to the corresponding 3D coordinates [24].



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 6: 21 Hand Landmarks

II.8 Right and Left Hand Detection:

To implement this method, we used the mediapipe and OpenCV libraries in Python for right-hand and left-hand detection. We will be using the hands model from mediapipe solutions to detect hands, which is a palm detection model that runs on the whole image and returns a manually oriented chest.

Capture streaming images from your camera using OpenCV and make predictions using a custom hand model. The prediction made by the model is stored in the outcome variable where we can access the features using the results. Hands" on the other image of one hand, store the

MessageToDict () function on the label variable. If the label is “left”, put the text “left hand” on the image and if the label is “right”, put the text “right hand” on the image.

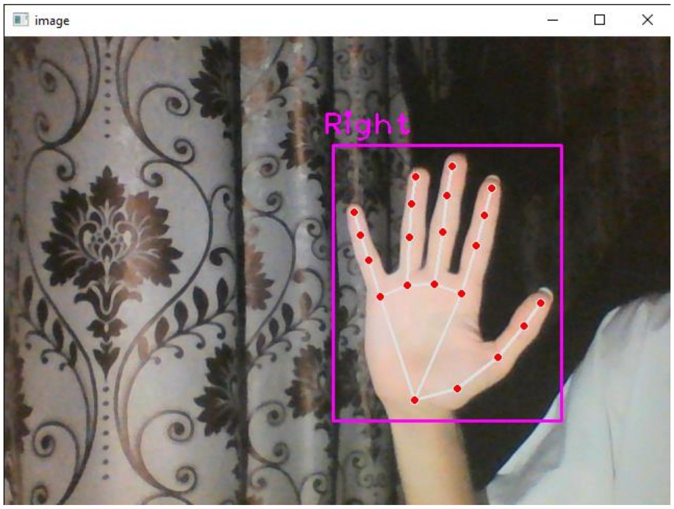


Figure 7: right left detection

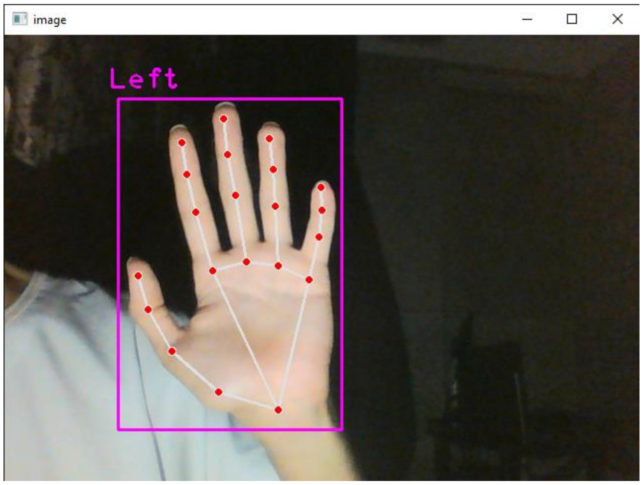


Figure 8: left hand detection

II.9 Finger state detection:

The used module Capture the frames continuously and detect the hand from the frame then detect the state of fingers, if any fingers is up it gives it the value 1 else returns 0.

-It returns the list of five elements and each element depends on the finger’s condition.

- The list order like [thumb, index finger, middle finger, ring finger, pinky/little finger]

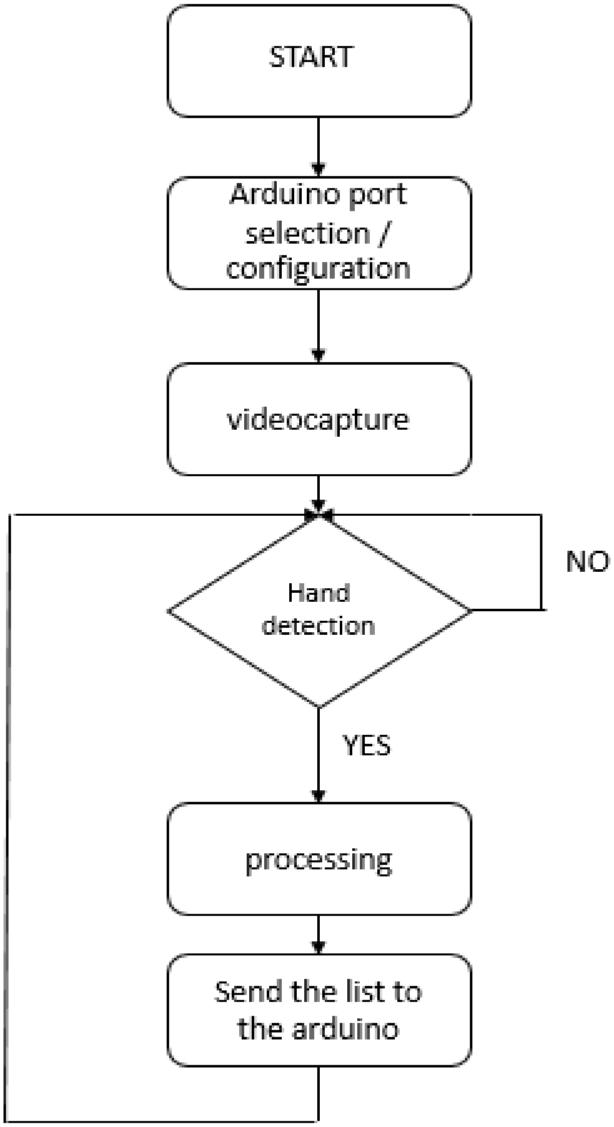

START
Arduino port selection / configuration
videocapture
Hand detection
NO
YES
processing
Send the list to the arduino

Figure 9: organizational chart of python program

II.10 Conclusion:

In this chapter, we focus on the method used in image processing with AI and CNN in both directions in general and in our specific project. As a result, we infer from the previous operation a 32-word language as the starting point, and receive as output a list containing the state of the finger and the order means a list of two states with 5 Boolean variables, through which we conclude 32 words (2 ^ 5) to control any remote device

Chapter III

« Software and hardware of the controlling system »

Chapter III: Software and hardware of the controlling system

III.1 Introduction:

In order to accomplish an electronic project a theoretical study is necessary and which will be followed by a practical concretization.

In this chapter we focus on explaining the hardware and software side of the project.

III.2 Hardware side:

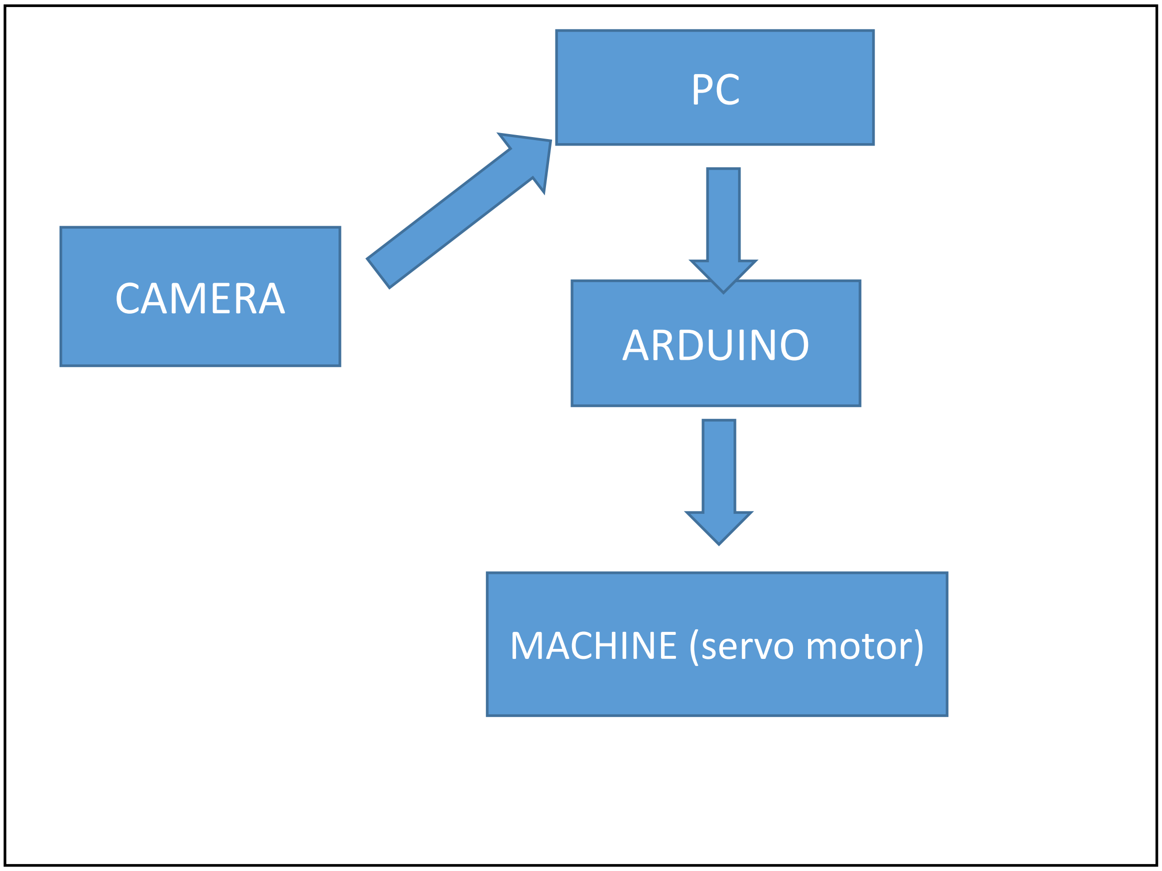


Figure 10: Synoptic diagram.

III.3 The electronic card Arduino Uno:

In 2005, building on the work of Hernando Barragán (the creator of Wiring), Massimo Banzi and David Quarteles created the Arduino, an easy-to-use programmable device for interactive art design projects, at the Ivrea Interactive Design Institute in Ivrea, Italy. David Mellis has developed a wire-based Arduino software. Not long after, Gianluca Martino and Tom Iguy joined the project, the five are known as the original Arduino founders. They wanted a simple device that was easy to connect to various elements (eg relays, actuators, sensors) and easy to program. It should also be inexpensive, as students and artists are not known to have a lot of spare money. They chose

Chapter III: Software and hardware of the controlling system

Atmel's AVR family of 8-bit microcontrollers (MCU or C), built a self-contained circuit board with easy-to-use connections, wrote a bootloader for the microcontroller, and did everything packaged in a simple integrated development. Environment (IDE) that uses software called "sketches". The result was an Arduino.

Since then, the Arduino has evolved in several different directions, with some versions getting smaller than the original and others bigger. Each has a certain niche that it is meant to fill. Common to everyone is the Arduino AVR-GCC runtime library that comes with the Arduino development environment, and the built-in bootloader firmware that comes preloaded on the microcontroller on every Arduino board.

The Arduino family of boards uses processors developed by the Atmel Corporation in San Jose, California. Most Arduino builds use the 8-bit AVR series of microcontrollers, with the main exception being a 32-bit ARM Cortex-M3 processor. We do not cover the Due in this book, as it is fundamentally different from AVRs in many basic ways and really deserves a separate discussion dedicated to it and similar microcontrollers based on the design of the ARM Cortex-M3.

Although the Arduino board, as mentioned by the Arduino team, is just a basic Atmel AVR development board, it is the Arduino software environment that sets it apart. It is a common experience for all Arduino users and the cornerstone of the Arduino concept. Chapter 5 covers the Arduino IDE, the libraries that come with the IDE, and the bootloader. Chapter 6 describes the process of creating a program for an AVR MCU without using the Arduino IDE.

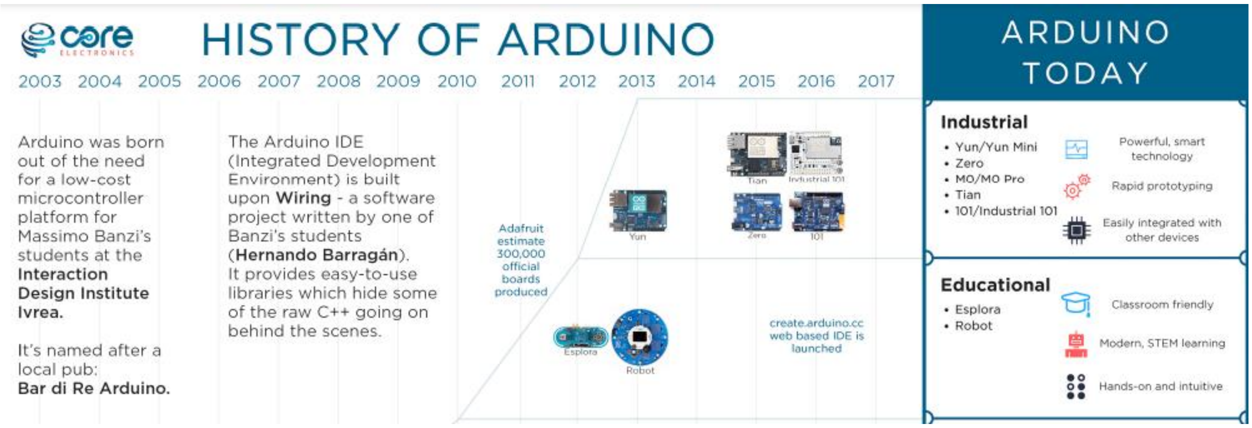

core ELECTRONICS
HISTORY OF ARDUINO
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
Arduino was born out of the need for a low-cost microcontroller platform for Massimo Banzi's students at the Interaction Design Institute Ivrea.
It's named after a local pub: Bar di Re Arduino.
The Arduino IDE (Integrated Development Environment) is built upon Wiring - a software project written by one of Banzi's students (Hernando Barragán).
It provides easy-to-use libraries which hide some of the raw C++ going on behind the scenes.
Adafruit estimate 300,000 official boards produced
Yun
Tian
Industrial 101
Zero
101
Esplora
Robot
create.arduino.cc web based IDE is launched
ARDUINO TODAY
Industrial
• Yun/Yun Mini
• Zero
• M0/M0 Pro
• Tian
• 101/Industrial 101
Powerful, smart technology
Rapid prototyping
Easily integrated with other devices
Educational
• Esplora
• Robot
Classroom friendly
Modern, STEM learning
Hands-on and intuitive

Figure 11: history of Arduino.

III.3.1 Arduino Uno definition:

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital I/O ports (6 of which can be used as PWM outputs), 6 analog inputs, a 16MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. Contains everything needed to support the microcontroller; just connect it to your computer with a USB cable or power it up with an AC adapter or battery to get started. You can DIY your own UNO without worrying too much about doing something wrong. The worst case is to replace the chip and start over.

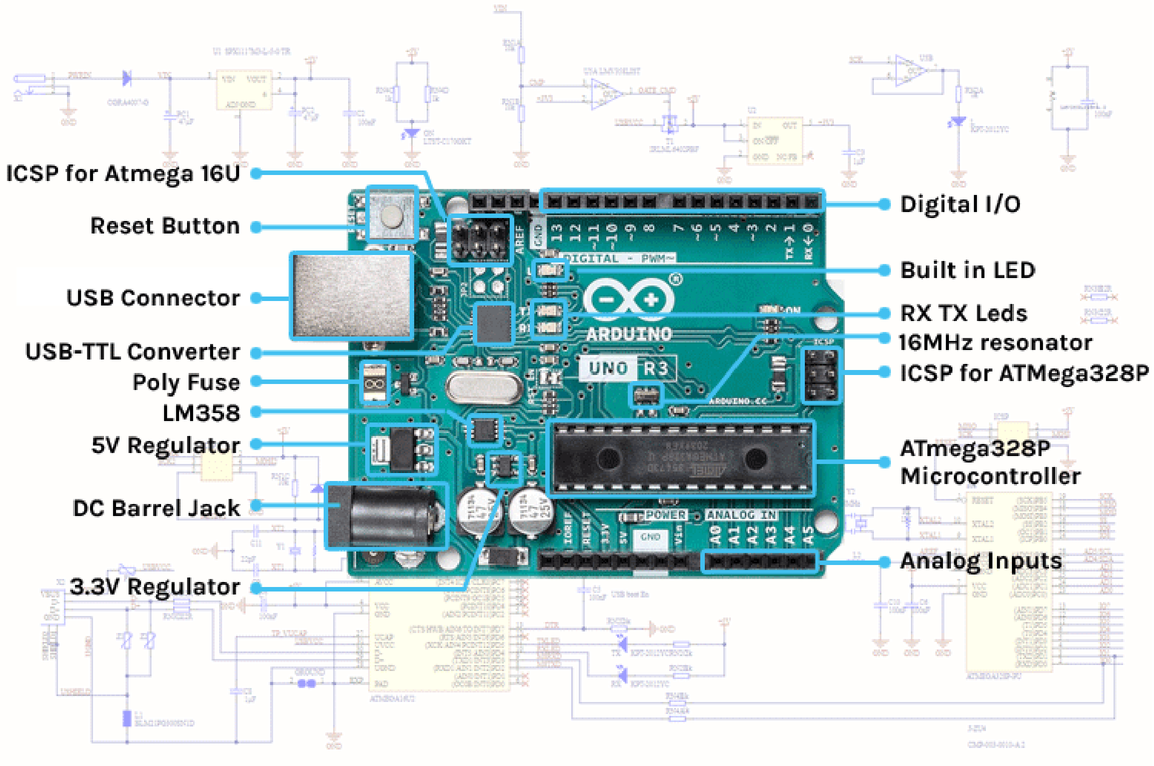


Figure 12: Arduino Uno

III.3.2 Arduino Uno specifics: this table presents the specifications of an Arduino Uno.

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Inout Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)

PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN:	13
Length	68.6 mm
Width	58.4 mm
Weight	25 g

Table 1: Arduino Uno specifics.

III.4 Servo motors MG996R:

MG996R is a metallic servo motor with a maximum torque of 11kg/cm. Like other RC devices, the motor rotates from 0 to 180 degrees based on the duty cycle of the PWM wave supplied to its signal pin.

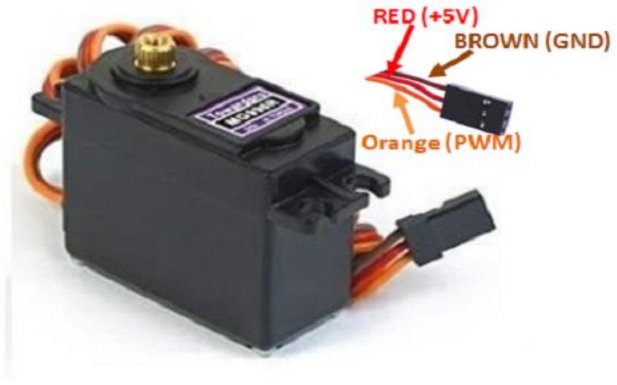


Figure 13: Servo motors MG996R:

And this table shows the specifications of a servo motors MG996R

Wire Number	Wire Color	Description
1	Brown	Ground wire connected to the ground of system
2	Red	Powers the motor typically +5V is used
3	Orange	PWM signal is given in through this wire to drive the motor

Table2: Servo motors MG996R specifics

III.5 3D printed hand:

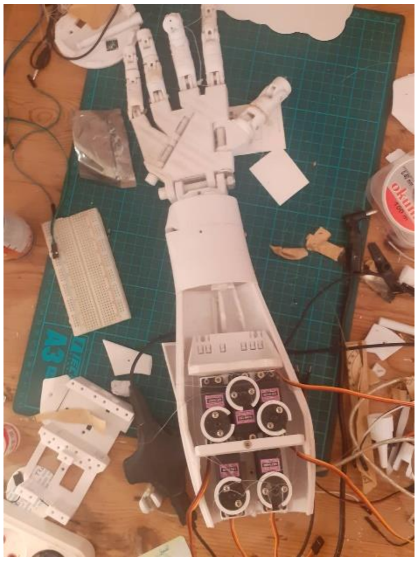


Figure 14: robotic hand

It’s the prototype, which shows to the user the result of the hand tracking.

III.6 fishing wire:

Figure 15: Fishing wire

The main role of the wire is to connect the 3D printed fingers to the servomotors to close and open the hand depending on the state of the servo (0 or 180 degrees)

III.7 The pc: web cam + the processing unit

III.8 Serial communication Protocol UART:

III.8.1 Definition:

By definition, UART is a hardware communication protocol that uses asynchronous serial communication at a configurable speed. Asynchronous means that there is no clock signal to synchronize the output bits from the transmitter to the receiving end.

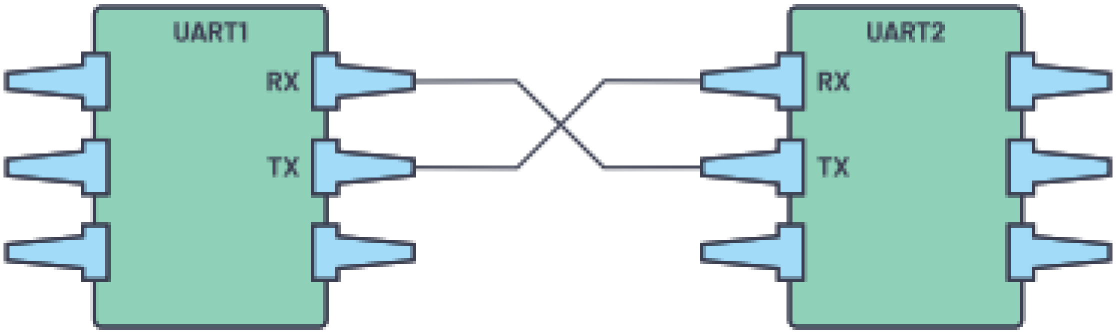


Figure 16: Two UARTs directly communicate with each other.

The two signals of each UART device are named:

-Transmitter (TX).

-Receiver (RX).

The main purpose of each device's transceiver line is to transmit and receive serial data for serial communication.

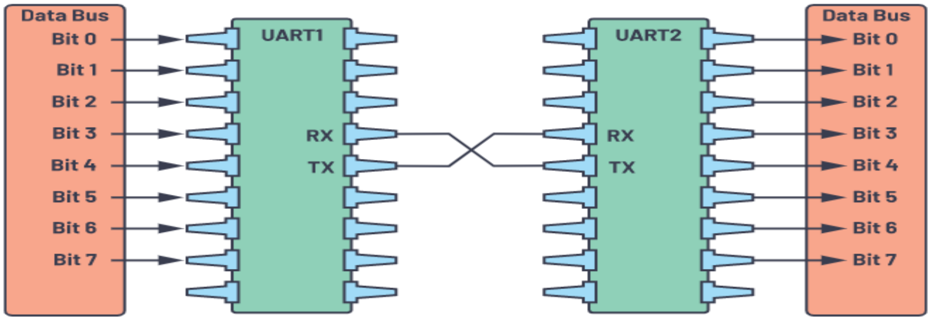


Figure 16.1: UART with data bus.

The sending UART is connected to the control data bus which sends the data in parallel form. From there, the data will now be transmitted over the serial transmission line (the wire), little by little, to the receiving UART. This, in turn, will convert the serial data to parallel to the receiver. UART lines serve as a means of communication for transmitting and receiving one piece of data to another. Note that the UART has a dedicated transmit and receive pin for transmit or receive. For UART and most serial connections, the baud rate should be the same on the transceiver. Baud rate is the rate at which information is transmitted over a communication channel. In the context of a serial port, the specified baud rate will act as the maximum number of bits per second to be transmitted.

Wires	2
Speed	9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 1000000, 1500000
Methods of Transmission	Asynchronous
Maximum Number of Masters	1
Maximum Number of Slaves	1

Table 3: UART Summary

The UART interface does not use the clock signal to synchronize the transmitters and receivers; it transmits data asynchronously. Instead of a clock signal, the transmitter generates a bit stream based on its own clock signal while the receiver uses the internal clock signal to sample the incoming data. The sync point is managed by having the same baud rate on both devices. Failure to do so may affect the timing of sending and receiving data, which may result in inconsistencies in data processing. The allowable baud rate difference can be as high as 10% before the bit timing gets too far.

III.8.2 Data Transmission:

In UART, the transmission mode is in packet form. The part that connects the transmitter and receiver understands the serial packet generation process and controls these physical device lines. A packet consists of a start bit, a data frame, a parity bit, and a stop bit.

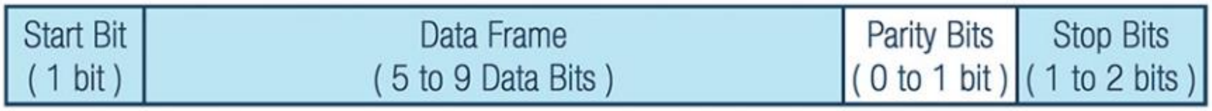


Figure16.2: UART packet.

III.8.3 Start Bit:

The UART data transmission line is usually kept at a high voltage level when no data is being transmitted. To initiate data transmission, the transmitter UART pulls the transmission line high and low for one (1) hour cycle. When the receiver UART detects a high-to-low voltage transition, it starts reading the bits from the data frame at the baud rate frequency.

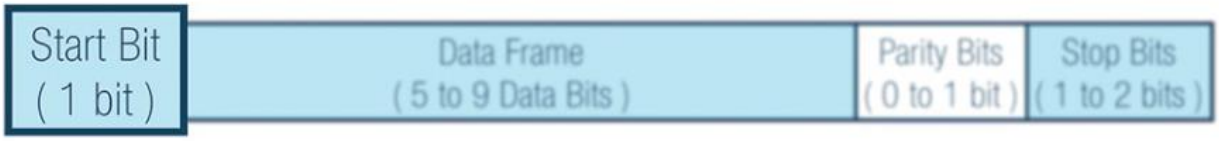


Figure 16.3: Start bit.

III.8.4 Data Frame

The data frame contains the actual data transferred. They can be five (5) bits up to eight (8) bits long if a parity bit is used. If a parity bit is not used, the data frame can be nine (9) bits long. In most cases, the least important data is sent first.

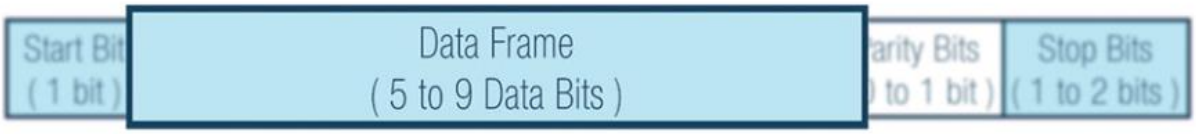


Figure 16.4: Data frame.

III.8.5 Parity:

Parity describes the symmetry or change in a number. The parity bit is a way for the receiver UART to see if data has changed during transmission. Bits can be altered by electromagnetic radiation, incompatible baud rates, or long distance data transmissions.

After the receiving UART reads the data frame, it counts the number of bits with a value of 1 and performs a procedure to determine whether the total is an even or odd number. If the parity bit is 0 (even parity), bit 1 or the logical high bit in the data frame must add up to an even number. If the parity bit is 1 (odd parity), then bit 1 or logical heights in the data frame must add up to an odd number.

When the parity bit matches the data, the UART knows that the transmission was error-free. But if the parity bit is 0 and the total is odd, or if the parity bit is 1 and the total is even, then the UART knows that the bits in the data frame have changed.

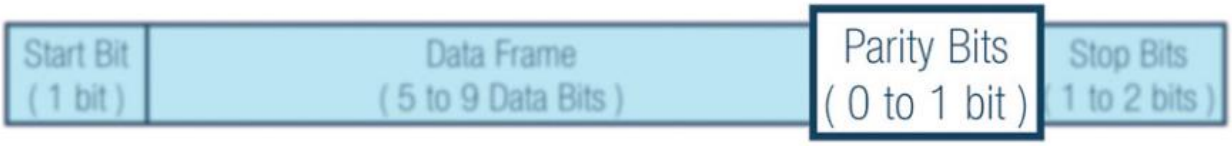


Figure 16.5: Parity bits.

III.8.6 Stop Bits

To indicate the end of the data packet, the transmitter UART rotates the data transmission line from low voltage to high voltage for one (1) bit to (2) bits.

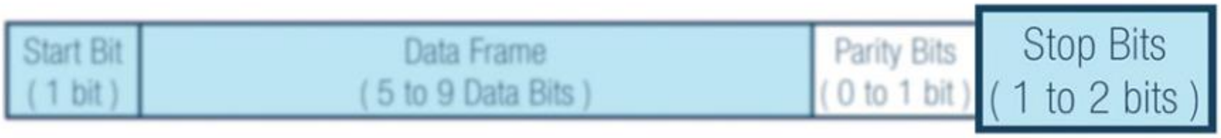


Figure16.6: Stop bits.

III.8.7 Steps of UART Transmission

- First: the sending UART receives data in parallel from the data bus.

-Second: The sender UART adds the start bit, parity bit, and stop bit(s) to the data frame.

-Third: the entire packet is sent sequentially from the start bit to the stop bit of the UART sent to the receiving UART. The receiving UART samples the data line at a preconfigured baud rate.

-Fourth: The receiver UART removes the start bit, parity bit, and stop bit from the data frame

- Fifth: The receiver UART converts the serial data back into parallel and transmits it to the receiver data bus.

III.9 Software side:

III.9.1 Python 3:

Python is a multi-paradigm, general-purpose, interpreted, and high-level programming language. Python allows programmers to use different programming styles to create simple or complex programs, get faster results, and write code as if they were speaking a human language. Some of the popular platforms and applications that have used Python during development include Google Search, YouTube, Bit-Torrent, Google App Engine, Eve Online, Maya, and iRobot.

Python 3 is a newer version of the Python programming language that was released in December 2008. This version was released primarily to address issues in Python 2. The nature of these changes is that Python 3 was incompatible with Python 2.

Applications of Python:

-Scientific and numerical applications

- Artificial intelligence and machine learning

- Software development

-Image processing and graphic design applications

III.9.2 PyCharm IDE:

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language. It was developed by the Czech company JetBrains (formerly IntelliJ). Provides code analysis, graphical debugger, built-in unit tester, integration with version control systems, and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with versions for Windows, macOS, and Linux. The community edition is released under the Apache license [7], there is also an educational edition, as well as a Professional edition with additional features (released under a special subscription-based license).

Figure 17 : PyCharm logo

III.9.3 Arduino IDE:

The Arduino integrated development environment - or Arduino software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and serial menus. Connects to Arduino hardware to download and communicate with software.

Programs written using the Arduino software (IDE) are called graphics. These graphics are written in a text editor and saved with the .ino file extension. The editor has functions to cut/paste and find/replace text. The message box provides comments during saving and export and also displays errors. The console displays text output from the Arduino program (IDE), including full error messages and other information. The lower right corner of the window displays the configured serial card and port. Toolbar buttons allow you to scan and download programs, create, open and save diagrams, and open the serial monitor.

Figure18: Arduino IDE logo

And this figure represents the organizational chart of Arduino:

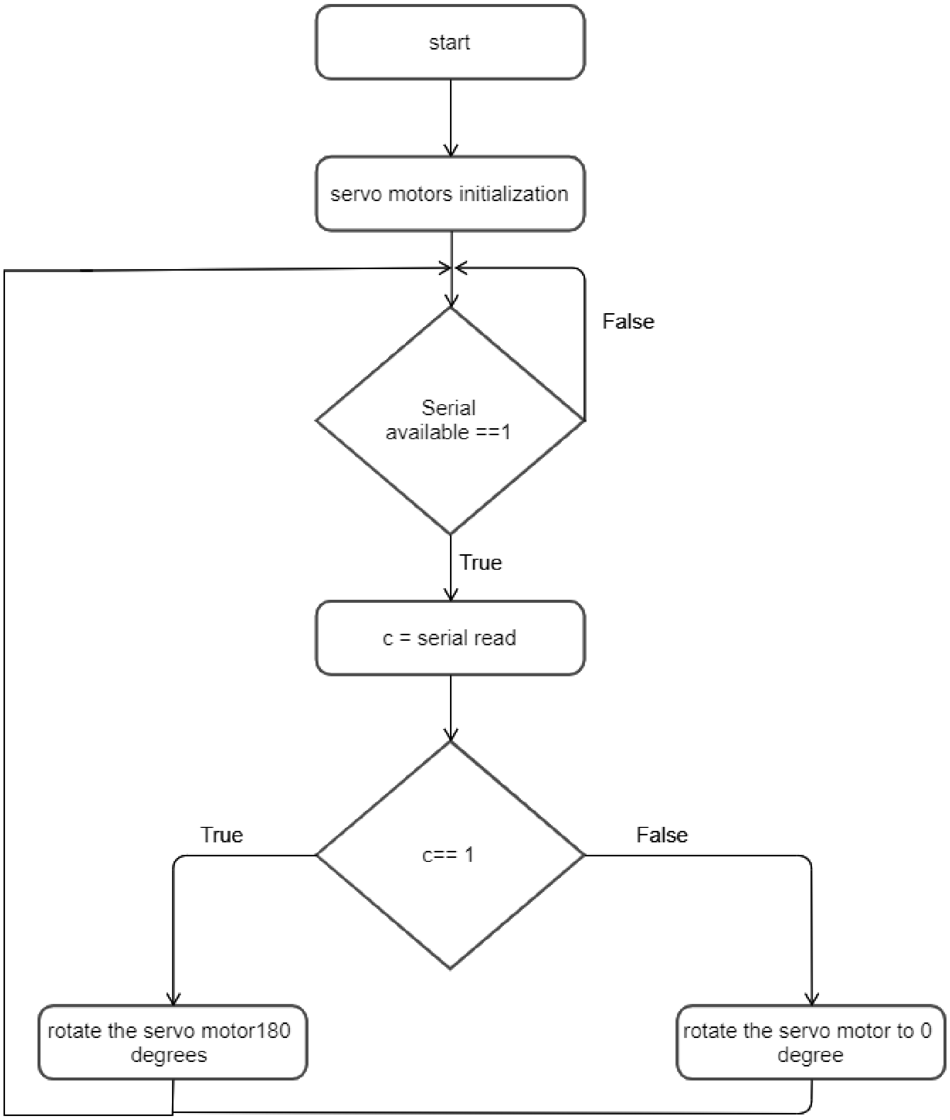

start
servo motors initialization
False
Serial available ==1
True
c = serial read
True
c== 1
False
rotate the servo motor180 degrees
rotate the servo motor to 0 degree

Figure 19: organisationnel chart of arduino program

III.10 Conclusion:

This chapter allowed us to learn about the different hardware used and the communication protocol on the other side, we got to know the software side and the different methods used in image processing and the Arduino program.

Chapter IV:

«Experimental results»

IV.1 Introduction:

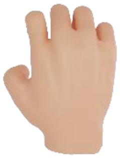
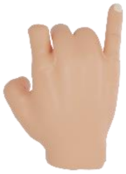
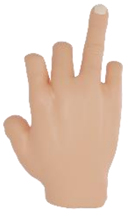
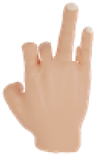
This chapter presents the experimental results of image processing using python language and process control by robotic hand by user hand gestures using the Arduino Uno board.

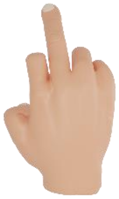
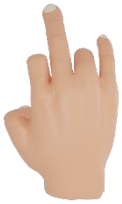
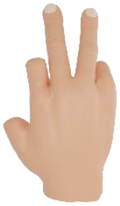
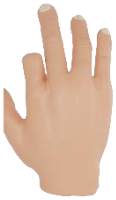
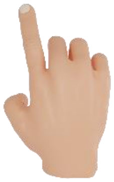
IV.2 The created language:

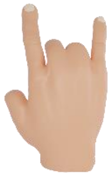
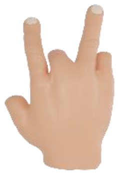
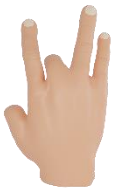
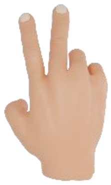
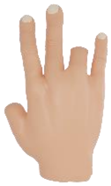
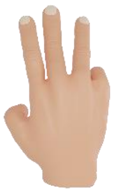
The module used constantly picks up frames and detects the hand from the frame and then detects the state of the fingers, if the finger is raised it gives it the value 1 otherwise it returns 0.

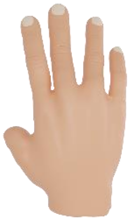
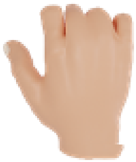
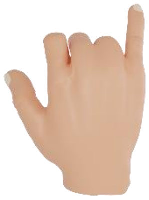
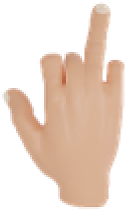
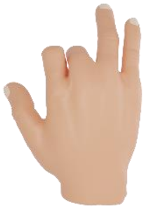
We deduce from the above process a language of 32 hand gestures to control any machine remotely.

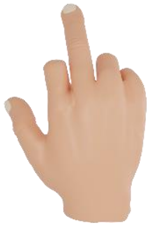
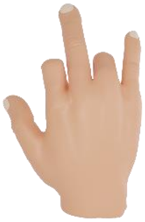
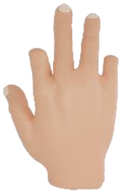
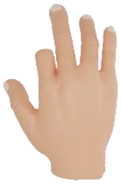
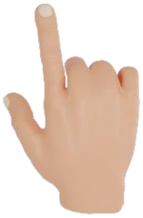
This table represents hand gestures signification:

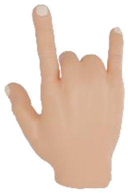
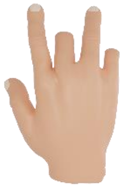
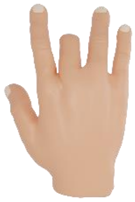
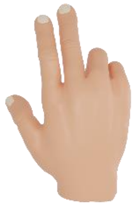
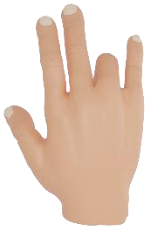
Hand gestures	the sent code to the Arduino	The order
	**[0, 0, 0, 0, 0]**	Order 1 activated [customized task]
	**[0, 0, 0, 0, 1]**	Order 2 activated [customized task]
	**[0, 0, 0, 1, 0]**	Order 3 activated [customized task]
	**[0, 0, 0, 1, 1]**	Order 4 activated [customized task]

	**[0, 0, 1, 0, 0]**	Order 5 activated [customized task]
	**[0, 0, 1, 0, 1]**	Order 6 activated [customized task]
	**[0, 0, 1, 1, 0]**	Order 7 activated [customized task]
	**[0, 0, 1, 1, 1]**	Order 8 activated [customized task]
	**[0, 1, 0, 0, 0]**	Order 9 activated [customized task]

	[0, 1, 0, 0, 1]	Order 10 activated [customized task]
	[0, 1, 0, 1, 0]	Order 11 is impossible
	[0, 1, 0, 1, 1]	Order 12 activated [customized task]
	[0, 1, 1, 0, 0]	Order 13 activated [customized task]
	[0, 1, 1, 0, 1]	Order 14 activated [customized task]
	[0, 1, 1, 1, 0]	Order 15 activated [customized task]

	**[0, 1, 1, 1, 1]**	Order 16 activated [customized task]
	**[1, 0, 0, 0, 0]**	Order 17 activated [customized task]
	**[1, 0, 0, 0, 1]**	Order 18 activated [customized task]
	**[1, 0, 0, 1, 0]**	Order 19 activated [customized task]
	**[1, 0, 0, 1, 1]**	Order 20 activated [customized task]

	**[1, 0, 1, 0, 0]**	Order 21 activated [customized task]
	**[1, 0, 1, 0, 1]**	Order 22activated [customized task]
	**[1, 0, 1, 1, 0]**	Order 23activated [customized task]
	**[1, 0, 1, 1, 1]**	Order 24 activated [customized task]
	**[1, 1, 0, 0, 0]**	Order 25 activated [customized task]

	[1, 1, 0, 0, 1]	Order 26 activated [customized task]
	[1, 1, 0, 1, 0]	Order 27 is impossible
	[1, 1, 0, 1, 1]	Order 28 activated [customized task]
	[1, 1, 1, 0, 0]	Order 29activated [customized task]
	[1, 1, 1, 0, 1]	Order 30 activated [customized task]

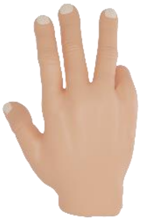
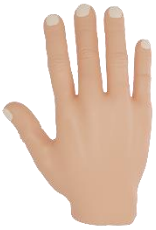
	**[1, 1, 1, 1, 0]**	Order 31activated [customized task]
	**[1, 1, 1, 1, 1]**	Order 32activated [customized task]

Table4: hand gestures signification

IV.3 Conclusion

In this chapter, we got to know each gesture and its meaning and furthermore, we noticed some impossible hand gestures for the user (Command No. 11 and 27) as shown in the previous table.

Results are obtained with a maximum of 1 tracking hands for added safety with detection set to 0.8. The detection hoax is the result of a comparison of the entered image to the database made with 30,000 images of hands.

For better system performance, security is needed, this project is scalable to be secure, recognize the user's hand and categorize according to tasks.

General Conclusion:

In this work we presented a solution for safety issues in workspaces which include: communicable disease, transportation accidents, slipping and falling, toxic events, particularly chemical and gas exposure, getting struck by objects, getting struck by objects, repetitive motion and ergonomic injuries, and hearing loss. this solution is based on the following concept: create smooth HMI to facility the interaction between the employee and the machine in addition to full sign language made with 32 gestures can be customized to control any machines remotely except that not every gesture can be applied by human hand after a test we conclude that just 30 gestures are applicable, this amount of gestures is sufficient to control any machine easily. this language is based on a whole process of hand tracking by both hardware and software devices which is a web cam and hand tracking module in python wich provide a real time tracking of 21 point of the hand in addition of the finger state with that amount of information we are able to control machines remotely , this project's target is factories and smart houses in the near future

References:

[1] https://csrc.nist.gov/glossary/term/human_machine_interface

[2] Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale (2003): Human–Computer Interaction. 3rd Edition. Prentice Hall, 2003. http://hcibook.com/e3/ ISBN 0-13-046109-1

[3] https://nelson-miller.com/comparing-the-different-types-of-hmis/?fbclid=IwAR2q-8jgGZl-mGZX4LFKaGFdz587bEbWSPYH0Rnsg0t_pAQ1GHwIpN1PjjI

[4] https://www.inductiveautomation.com/resources/article/what-is-hmi#:~:text=HMIs%20are%20used%20to%20optimize,systems%2C%20all%20through%20one%20console.

[5] https://www.bakerhughes.com/nexus-controls/distributed-control-systems-dcs/hmi-control-systems-software

[6] Chi Xu , Accurate Hand Detection from Single-Color Images by Reconstructing Hand Appearances

Published online 2019 Dec 29. doi: 10.3390/s20010192

[7] 1. Li C., Kitani K.M. Pixel-level hand detection in ego-centric videos; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Portland, OR, USA. 23–28 June 2013; pp. 3570–3577. [Google Scholar]

[8] Xu C., Cheng L. Efficient Hand Pose Estimation from a Single Depth Image; Proceedings of the IEEE International Conference on Computer Vision; Sydney, Australia. 1–8 December 2013; pp. 3456–3462. [Google Scholar]

[9]Ge L., Ren Z., Li Y., Xue Z., Wang Y., Cai J., Yuan J. 3D Hand shape and pose estimation from a single RGB image; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Long Beach, CA, USA. 16–20 June 2019; pp. 10833–10842. [Google Scholar]

[10] Lin H., Hsu M., Chen W. Human hand gesture recognition using a convolution neural network; Proceedings of the IEEE International Conference on Automation Science and Engineering; Taipei, Taiwan. 18–22 August 2014; pp. 1038–1043. [Google Scholar]

[11] Kirishima T., Sato K., Chihara K. Real-time gesture recognition by learning and selective control of visual interest points. IEEE Trans. Pattern Anal. Mach. Intell. 2005;27:351–364. doi: 10.1109/TPAMI.2005.61. [PubMed] [CrossRef] [Google Scholar]

[12] Meena Y.K., Cecotti H., Wong-Lin K., Dutta A., Prasad G. Toward optimization of gaze-controlled human–computer interaction: Application to hindi virtual keyboard for stroke patients.

IEEE Trans. Neural Syst. Rehabil. Eng. 2018;26:911–922. doi: 10.1109/TNSRE.2018.2814826. [PubMed] [CrossRef] [Google Scholar]

[13] Shahroudy A., Liu J., Ng T.T., Wang G. Ntu rgb+ d: A large scale dataset for 3d human activity analysis; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Las Vegas, NV, USA. 27–30 June 2016; pp. 1010–1019. [Google Scholar]

[14] Sigal L., Sclaroff S., Athitsos V. Skin color-based video segmentation under time-varying illumination. IEEE Trans. Pattern Anal. Mach. Intell. 2004;26:862–877. doi: 10.1109/TPAMI.2004.35. [PubMed] [CrossRef] [Google Scholar]

[15] Dalal N., Triggs B. Histograms of oriented gradients for human detection; Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; San Diego, CA, USA. 20–26 June 2005; pp. 886–893. [Google Scholar]

[16] Guo J., Cheng J., Pang J., Guo Y. Real-time hand detection based on multi-stage HOG-SVM classifier; Proceedings of the IEEE International Conference on Image Processing; Melbourne, Australia. 15–18 September 2013; pp. 4108–4111. [Google Scholar]

[17] Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Columbus, OH, USA. 24–27 June 2014; pp. 580–587. [Google Scholar]

[18] Girshick R. Fast r-cnn; Proceedings of the IEEE International Conference on Computer Vision; Santiago, Chile. 13–16 December 2015; pp. 1440–1448. [Google Scholar]

[19Ren S., He K., Girshick R., Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks; Proceedings of the Advances in Neural Information Processing Systems; Montréal, QC, Canada. 7–12 December 2015; pp. 91–99.

[20]He K., Gkioxari G., Dollár P., Girshick R. Mask r-cnn; Proceedings of the IEEE International Conference on Computer Vision; Venice, Italy. 22–29 October 2017; pp. 2961–2969. [Google Scholar]

[21] Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Las Vegas, NV, USA. 27–30 June 2016; pp. 779–788. [Google Scholar]

[22] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y., Berg A.C. Ssd: Single shot multibox detector; Proceedings of the European Conference on Computer Vision; Amsterdam, The Netherlands. 8–16 October 2016; pp. 21–37. [Google Scholar] [23]

[23] Lin T.Y., Goyal P., Girshick R., He K., Dollár P. Focal loss for dense object detection; Proceedings of the IEEE International Conference on Computer Vision; Venice, Italy. 22–29 October 2017; pp. 2980–2988. [Google Scholar]

[24] Deng X., Zhang Y., Yang S., Tan P., Chang L., Yuan Y., Wang H. Joint hand detection and rotation estimation using CNN. IEEE Trans. Image Process. 2017;27:1888–1900. doi: 10.1109/TIP.2017.2779600. [PubMed] [CrossRef] [Google Scholar]

[25] Le T.H.N., Quach K.G., Zhu C., Duong C.N., Luu K., Savvides M. Robust hand detection and classification in vehicles and in the wild; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops; Honolulu, HI, USA. 21–26 July 2017; pp. 1203–1210. [Google Scholar]

[26] Yang L., Qi Z., Liu Z., Liu H., Ling M., Shi L., Liu X. An embedded implementation of CNN-based hand detection and orientation estimation algorithm. Mach. Vis. Appl. 2019:1–12. doi: 10.1007/s00138-019-01038-4. [CrossRef] [Google Scholar]

[27] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative adversarial nets; Proceedings of the Advances in Neural Information Processing Systems; Montréal, QC, Canada. 8–13 December 2014; pp. 2672–2680. [Google Scholar]

[28] Paper: MediaPipe Hands: On-device Real-time Hand Tracking (presentation)

[29] https://www.oreilly.com/library/view/arduino-a-technical/9781491934319/ch01.html?fbclid=IwAR1jmk3fFHnbk3HSAQV95wSroFMTuFIIwK4xFh0rBBnY9393Wjx9EGSbPZU

[30] https://www.arduino.cc/en/Guide/Introduction/

[31] Eric Peňa , Mary Grace Legaspi UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter

[32] https://www.python.org/doc/essays/blurb/

[33] https://www.jetbrains.com/help/pycharm/quick-start-guide.html#summary

[34] https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics