People's Democratic Republic of Algeria
Ministry of Higher Education and
Scientific Research

University Echahid Hamma Lakhdar of El-Oued

Faculty of Technology

Memory the end
Of Study In order to obtain the diploma of
MASTERS M ACADEMIC

Field : Science and Technology
Sector : Telecommunications

Theme

Virtual drag and drop using open CV And Arduino

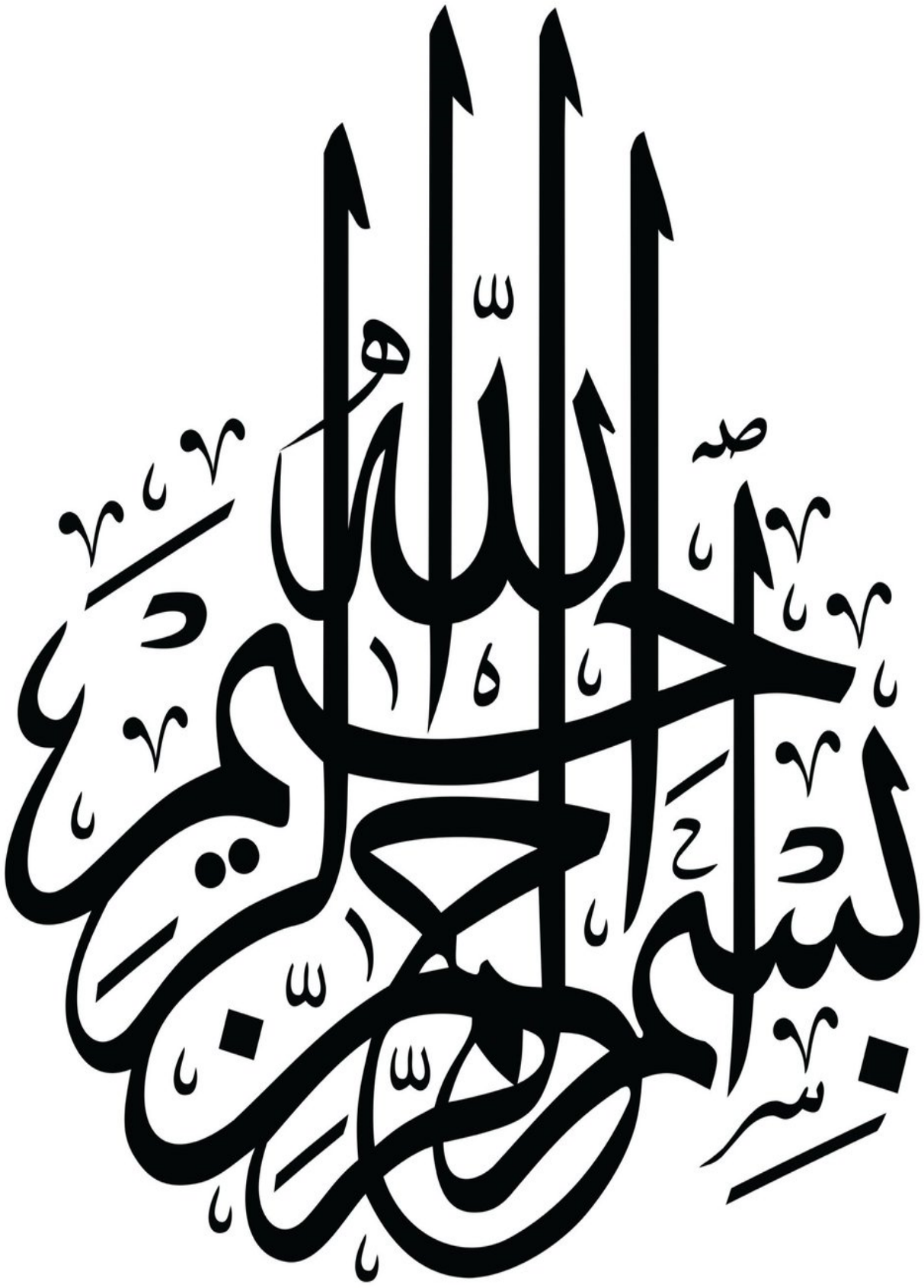
Presented by:

Fridjat AbdelFattah ,Aliat Boubakeur ,Imad Nattari.

Was debated in: Jun 2023 in front of The Examining Committee Composed from:

- **President Dr :**
- **Supervisor Dr: HIMA. A** **MCA**
- **Co-supervisor Dr: Tidjani . A** **MCB**
- **Co-supervisor Pr: Moufid. A** **Professor**

Academie Year: 2022/2023


بسم الله الرحمن الرحيم

Thanks

The work presented in this dissertation was carried out within the Faculty of Technology of Echahid Hamma Lakhdar. El Oued University.

First of all, I thank ALLAH, the almighty, for giving me the courage and the will to accomplish this work.

I thank my supervisors, Mr. Abdelkader Hima and Tidjani Amina, for having directed these works, and for their immense help, For having directed this work, for their immense help, for their simplicity, and for their unique values which have always supported me.) In the most difficult moments, they always received me with sympathy and put their time and knowledge at my disposal, they followed and encouraged the many corrections until the completion of this work, please find here the expression of my gratitude.

My thanks to:

All our gratitude to our teachers.

And our friends.

To all those who have contributed directly or indirectly to the realization of this work.

Dédicace

Je dédiée ce travail à :

Ma Mère,

A mon Père,

A ma femme

A mes fils Razan et Idris

A ma fille

À mon professeur **Tir Zuhair**

A mes frères

Et à tous mes Amis

Merci à tous.

عبد الفتاح

إهداء:

بعد مسيرة دراسية حملت في طياتها الكثير من الصعوبات والمشقة والتعب

اليوم نقطف ثمرها والحمد لله

اهدي ثمرة هذا العمل المتواضع أولا الى ابي الذي كان لي السراج الذي أنار درب حياتي وألهمني الصمود والتحدي لبلوغ مقاصدي

إلى أملي في الحياة وقرة عيني وسر نجاحي أمي الغالية ادامها الله وأطال في عمرها

وإلى كل من ساندني اصدقائي وإخوتي الذين وقفوا بجانبي

بوبكر

إهداء:

أهدي هذه المذكرة التخرج إلى عائلتي المحبّة، التي كانت الدعم الأساسي والملهم خلال رحلتي الأكاديمية. شكراً لوالدي ووالدتي على حبهما اللامتناهي وتشجيعهما الدائم، ولإخوتي وأخواتي على التفهّم والدعم المستمر. أنا ممتن لكل لحظة قضيتها معهم ولكل كلمة دعم وثقة قدموها لي.

وأخيراً، أهدي هذه المذكرة إلى أصدقائي الأعزاء، الذين شاركوا معي أوقاتًا ممتعة وصعبة على حد سواء. شكراً لكم على الدعم المعنوي والتشجيع المستمر، وعلى تقديم الأفكار والنصائح التي ساعدتني على التفوق والنجاح.

عماد

Summary :

Virtual drag and drop technology using Open CV and Arduino is one of the innovative technologies in the field of computer vision and electronic interaction. This technology allows users to drag and drop digital elements onto the screen using hand gestures.

This technology is based on a connection between a computer and an Arduino board and using the Open CV library for image processing and hand movement analysis. A camera is used to capture the movement of the hand and locate it on the screen. Arduino receives the signals sent from the computer and performs the actions required to achieve the drag-and-drop operation.
Keywords: Open CV, Computer Vision, hand movement, Arduino Python, Virtual drag-and-drop.

Résumé :

La technologie de glisser-déposer virtuel utilisant Open CV et Arduino est l'une des technologies innovantes dans le domaine de la vision par ordinateur et de l'interaction électronique. Cette technologie permet aux utilisateurs de faire glisser et déposer des éléments numériques sur l'écran en utilisant des gestes de la main.

Cette technologie est basée sur une connexion entre un ordinateur et une carte Arduino et utilise la bibliothèque Open CV pour le traitement d'images et l'analyse des mouvements de la main. Une caméra est utilisée pour capter le mouvement de la main et le localiser sur l'écran. Arduino reçoit les signaux envoyés par l'ordinateur et effectue les actions nécessaires pour réaliser l'opération de glisser-déposer.
Mots-clés : CV ouvert, vision par ordinateur, mouvement de la main, Arduino Python, glisser-déposer virtuel

---

ملخص:

تقنية السحب والإفلات الافتراضية باستخدام Open CV و Arduino هي إحدى التقنيات المبتكرة في مجال رؤية الكمبيوتر والتفاعل الإلكتروني. تتيح هذه التقنية للمستخدمين سحب العناصر الرقمية وإفلاتها على الشاشة باستخدام إيماءات اليد.

تعتمد هذه التقنية على اتصال بين جهاز كمبيوتر ولوحة Arduino وتستخدم مكتبة Open CV لمعالجة الصور وتحليل حركة اليد. تستخدم الكاميرا لالتقاط حركة اليد وتحديد موقعها على الشاشة. يستقبل Arduino الإشارات التي يرسلها الكمبيوتر ويقوم بالإجراءات اللازمة لأداء عملية السحب والإفلات.
الكلمات الرئيسية: السيرة الذاتية، رؤية الكمبيوتر، حركة اليد، Arduino، السحب والإفلات الظاهري

Liste des figures

Liste des tableaux

Sommaire

المحتويات

Introduction General:

Virtual drag-and-drop technology using Open CV and Arduino provides an intuitive Interactive interface for users to drag and drop digital elements onto the screen using hand gestures. This technology requires the use of a camera to analyze the movement of the hand and locate it on The screen, while the Arduino takes care of receiving the signals and performing the required Actions.

In the first chapter we talked about, you will learn the basics and overview of a computer Vision system. This chapter allows you to look at it from a broader perspective to address computer Vision problems. In the second chapter, computer vision has movement Recognition.

With advanced algorithms and technologies, computer vision systems can Accurately analyze and interpret human hand movements.

Finally, in the practical chapter, prepare the hardware and software components and then connect these components, Next, the Python code is written and tested, then the code is integrated with the Arduino board.

Finally, the system is tested with different motions and objects to evaluate its performance

Chapter I

General Overview of Computer Vision Systems

1. Introducing:

Computer vision systems are technologies that enable computers to perceive, analyze, and interpret visual information from digital images or videos. These systems mimic Chuman visual perception and use algorithms and techniques to extract useful information from visual data

In this chapter, we will talk about ways to solve computer vision problems

1.1. Definition of computer vision

Computer vision is a field in computer science and artificial intelligence that is concerned with developing techniques and tools for analyzing, understanding, and extracting information from digital images and videos. Computer vision aims to enable computers to understand and interpret visual content in a similar way to humans.

Computer vision encompasses a wide range of technologies and concepts, including shape recognition, people and object recognition, motion recognition, 3D imaging, face recognition, motion control, and many more. Computer vision is used in many fields such as robotics, face recognition, security and surveillance, medical image analysis, augmented and virtual reality, and others.

The way computer vision works depends on a set of steps that include converting images and videos into digital data, and then applying various techniques to analyze and process this data. This includes using algorithms, tools to detect landmarks and objects, recognize shapes and faces, track motion, and extract visual information.

The process begins with converting photos and videos into a digital representation using sensors or cameras. Digital processing techniques are then applied to enhance the image, reduce noise, and adjust color and contrast.

Then, analysis and recognition techniques are applied to extract the required information from the image. Machine learning and deep learning techniques can be used to train models to recognize and classify shapes, objects, and faces.

Then, motion tracking and analysis techniques are applied to track motion in a video or sequence of images and analyze it to extract motion-related information . [1]

1.2. Computer Vision problems

To solve a complex problem such as In order to solve a computer vision problem, for example, it is important to break it down into simple and manageable sub-steps and to understand the purpose of each step. The purpose of this chapter is to show how to approach a computer vision problem and how to model the problem with a generic model template. [2]

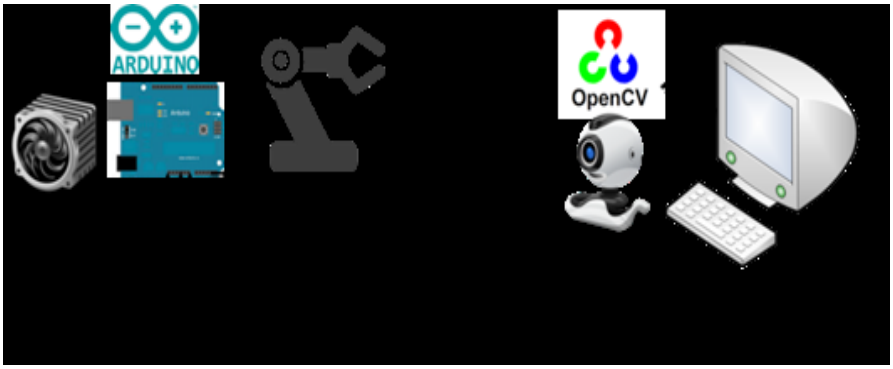


Figure 1.1computer vision architecture

Arduino is solely responsible for collecting sensory information such as temperature or humidity from the environment and transmitting this information to the vision controller's Open CV system. The communication between the vision system and the Arduino system can be wired or wireless as the Arduino easily handles both. After the vision system has processed the data coming from the Arduino and the webcam, the detection (or recognition) is complete. For example, can even recognize your face. The next step is to respond to this request by sending commands to the Arduino chip and taking the appropriate action. These actions could power a fan to cool the room, move a robotic arm to serve coffee, etc.

Any computer vision system consists of well-defined design In this chapter organized for data acquisition, pre-processing, image processing, post-filtering, detection (or acquisition), and commissioning. This research covers all of these steps in detail with a practical approach. We can draw a general outline of a computer vision system by mapping the steps to their implementation platforms. For an overview of the process of the vision system, see the figure below:

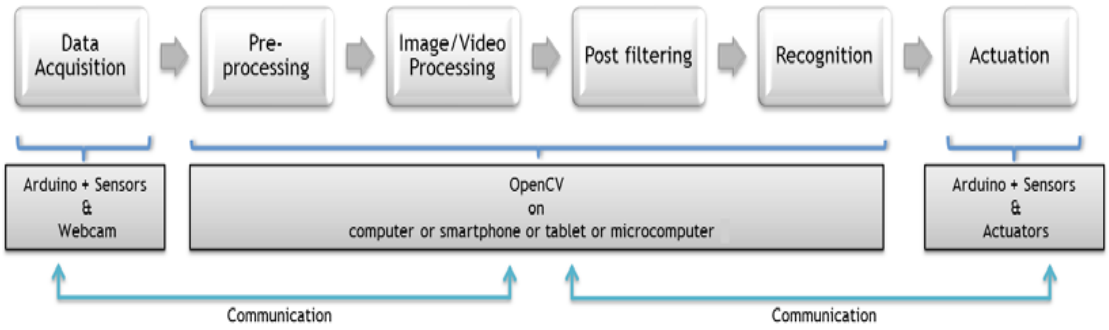


Figure 1.2.Process overview of the image processing system

1.3. Data acquisition

As you can see, the first step is data collection, which usually involves collecting sensory information from the environment. From the visual controller's point of view, there are two main sources of data: the camera and the Arduino system. The

Camera is the best sensor to mimic the human visual system and is directly connected to the vision controller in our schematic. Using Open CV is data acquisition capabilities; the vision controller reads video data from the camera. This data is a snapshot of an image or movie created from a series of images over a period. The camera can be of different types and categories.

In the simplest categorization, the camera can transmit analog or digital data. All of the cameras used in the examples in this book are digital because the computer environment and computer operations are also digital. Each element of the image is called a pixel. In digital imaging, a pixel, pixel, or -pixel is a physical point in a raster image, or the smallest addressable element in a all-point-addressable display device. it is therefore the smallest controllable element of the image displayed on the screen . For more information, cameras can also be classified based on their color recognition capabilities. RGB cameras are able to recognize both the main components of colors and a large number of combinations of these colors. Grayscale cameras can only see the scene in grayscale. Therefore, these cameras provide shape information instead of color information. After all, binary cameras only detect a black or white scene. By the way, a pixel in a binary camera can only have two values: black and white [3].

Another classification of cameras is their communication interface. Some examples are USB cameras, IP cameras, wireless cameras, etc. The communication interface of the camera also has a direct impact on the usability and capabilities of this camera. We usually have webcams with a USB interface. USB webcam use typically does not require external power sources or external components to get in the way of using the camera. It is therefore very easy to use a USB webcam for imaging tasks. Cameras also have properties like resolution, but we will cover camera properties in later chapters.

Common USB cameras, often used as webcams, provide 2D images. In addition to 2D camera systems, we now have 3D camera systems that can see the depth of every element in a scene. Probably the best-known example of 3D camera systems is the Kinect shown here**:**

Figure 1.3.systems camera 3D

Open CV supports different types of cameras and it is possible to read video information from all these cameras through simple interfaces, as this issue will be discussed in the following chapters with examples. Remember that capturing images is a crucial step in the viewing process and we have many options. In general, we need information beyond that provided by the camera to analyze the

environment around us. Some of this information relates to our other four senses. In addition, sometimes we need additional information beyond human capabilities. We can collect this information with Arduino sensors. Imagine you want to create an automatic door lock project with face recognition.

The system is likely to be activated by a knock on the door or a doorbell. You need a sound sensor that responds to a knock on the door or a doorbell. All this information can easily be gathered from the Arduino. We add a fingerprint reader to make it doubly secure! This allows you to combine the Arduino and camera data to get inferences about the scene in which the vision system is running. In summary, the camera and Arduino "with sensors" can be used by the Vision Controller to record the surroundings in detail! [4]

1.4. Preprocessing

Preprocessing means preparing something for processing. It can contain different types of steps, but the principle is always the same. We now explain preprocessing and why it is important in an image processing system. Let us clear something up first. This phase consists of preparing the collected video data for processing. Preprocessing is required in computer vision systems because raw data is usually noisy. In the image data we get from the camera, we have many unnecessary areas, and sometimes we get blurry images due to vibration, movement, etc. However, it is better to filter the image to make it more usable for our system make. For example, if you want to detect a big red ball in frame, you only need to remove the small dots or even the non-red parts. All these types of filter operations will make our life easier. In general, filtering is also performed when cameras collect data, but each camera has different pre-processing capabilities, and some of them have vibration isolation. However, as the number of built-in functions increases, so does the cost. So let us look at the filtering in our project using Open CV. Incidentally, robust vision systems can also be set up with inexpensive hardware such as a webcam. [5]

The same applies to sensor data. In real cases, we still get noisy data, so the noise has to be filtered out to get real sensor information. Some of this noise comes from the environment and some from the internal structure of the sensor. In any case, the data must be prepared for processing; this book will provide practical means of doing so. It is understood that the complexity of the image data is typically much greater than any conventional sensor, such as a temperature sensor or a humidity sensor. The dimensions of the data representing the information also vary. RGB images contain three-color components per pixel; red, green and blue. To display a scene with a resolution of 640x480, an RGB camera needs 640x480x3 = 921,600 bytes. Multiplying by three gives the size of each pixel. Each pixel contains a total of 3 bytes of data, 1 byte for each color .To represent the ambient temperature, and we typically need 4 bytes of data. This also explains why we need the very powerful devices to work with images. In addition, the complexity of image filters differs by from simple sensor filters. However, that does not mean we can't use complex filters in a simple way. If we know the purpose of the filter and what the filter parameters mean, we can use them easily. This book aims to make you more aware of the filtering process and how to easily apply advanced filtering techniques. As such,

filtering serves to extract real insights from data and is an integral part of the computer vision process. Many computer vision projects fail in development due to a missing filter layer. Even the best detection algorithms fail on noisy and inaccurate data. Therefore, note the importance of data filtering and preprocessing. [6]

1.5. Feature extraction by image processing

Probably the most challenging part of the computer vision project is the automated interpretation of scene. To extract meaning from an image, we use different types of image processing techniques. Sometimes we need more information than the we can get from a single photo. In this case, the relationship between the frames of the image becomes important. When such information between frames is extracted, it is called video processing. Some video processing applications also incudes' audio signal processing. All principles being the same, video processing is not significantly different form's image processing.

To understand the meaning of this chapter, it is useful to look at real-world applications of the. Imagine you want to build a robot with activated vision system that follows a line. A camera is placed in the middle of the robot and the robot follows a black line on the white floor. To do this, you need to recognize the line and know whether the line goes to the right or to the left. If the line is to the left of frame, rotate it to the left to move it to the center. If line is to the right of the picture, turn right as well. On the sideline, when the line is in the middle of frame, continue. You can also see the orientation of the lines to better plan your robot's movements. In this example, image-processing techniques shall be used to detect lines in the image. You will find that the book has a nice set of Techniques to show you the directions you need to take to solve your problem .At, using these techniques, certain candidate regions can be obtained in images that can be counted as a line. In order to correctly interpret the candidate lines, feature extraction must be performed for the image regions. By comparing the properties (characteristics) of the line candidate system, it is possible to separate the actual lines from the perturbations such as shadows. Feature Extraction is a pattern recognition and classification term that means extracting 1087 a small set of information that represents a larger set of information. When processing images, we extract so-called features such as length, position, image area, etc. Later we will use these functions to detect and detect all kinds of objects. There are several main approaches to extracting small groups of useful information from an image Segmentation is the term for it. Image segmentation divides a digital image into multiple segments (sets of pixels, also known as super pixels). The purpose of segmentation is to simplify and/or change the representation of the image into something more meaningful and easier to analyze. [7]

In our example, the candidate lines are image segments. Point analysis is a good way to identify segments of an image. This is a useful method for finding closed loops in an image. In general, closed loops correspond to objects in the image. Droplet analysis is also an image processing technique and will be discussed later. It is now important to understand the idea of feature mining

The information extracted from the image is used in the next stage of the artificial vision system. Because this processing step abstracts the image, doing it correctly is very important for the overall image processing system to work. Again, you do not need to know the complicated calculations under

the hood. Instead, you need to know where and how to use image processing techniques to extract valuable little bits of information from the scene. That is what the next chapters of this book are all about

1.6. Recognition or detection

The main purpose of a vision system is to conclude by interpreting pattern using images or arrays of images. The way to conclusion is cognition or sensing can be considered the basic form of cognition. The goal is to recognize an object or event. There are two types of conclusions. Objector event exists does not exist. Due to this binary nature of the find, it is a special classification process with two classes. The first class is existence and the second is non-existence. “To be or not to be, that is the question”. Recognition is a more complex term, also called classification, which speaks of the process of identifying one or more given or learned objects or classes of objects. Face recognition is a good example of this. The vision system should identify you

by your face. This is usually a complex classification process with many classes. In this case, each face is a class, so a complex problem. But with Open CV, we have many easy-to-use detection mechanisms, even for complex problems.

Sometimes complex algorithms take a long time. In addition, in some cases, very fast behavior is required, especially for real-time performance requirements. In of these cases we can also use simple but effective decision algorithms. As Leonardo da, Vinci said: "Simplicity is the height of sophistication". This book also teaches you how to build robust recognition systems using simple building blocks. [3]

Again, you need to be aware of the purpose of the recognition or classification. This consciousness shows you the path to success.

1.7. Acting in the real world

Every vision system has a purpose. There are different scenarios like; "When you encounter this event (or object), perform this action." After a long but enjoyable decision-making process for's vision system, the next step for would certainly be to act on the results. It is because of the "purpose of existence"; System. So far everything has been done so that we can take the right steps. Remember the example of our line-following robot.

The robot recognizes the line and the position of the line in its field of view. It also determines the direction in which it should go. Would it make sense if the robot knew which direction to move but could not move? This example also shows the importance of the action at the end. The physical action managed by the vision controller should affect real life in a physical manner. Good examples are driving motors to move on, heating an environment, and unlocking a door, triggering a device, and so on. To do this leads us to the Arduino. It has a huge amount of community support, lots of libraries on many devices and it is easy to use. Maybe for industrial design, you can go beyond Arduino but it is a fact that Arduino is a very good proof-of-concept platform for physical interaction with the world. Arduino is an integrated system that the uses to simplify and ease difficult engineering work. Therefore, we should like it!

The biggest challenge in using embedded systems to create various physical interactions for is that many of them are the same in terms of developing the software onboard the waves that can be used for some kind of dimming function .When outputting square waves with 50% duty cycle; this means that in a finite time t a pulse appears with a high logic state and in a finite time t with a low logic state.

The software applies the same control principle to completely different devices. If you apply this square wave to the motor at a high enough frequency, it will spin at half speed. Similarly, if you apply the same square wave to an LED, the will light up at half its light output. Even with completely different physical devices, we behave in the same way with the same software.

We should therefore see the inspirational nature of the tools we use to touch the world and if we have an idea of how to connect them to our system, everything will be fine.

In this research, we present a hardware-independent software approach to how the interacts with the world. Therefore, while the physical examples depend on the hardware, the same principles can be applied to any embedded system. There are also cutting-edge tips for artistically interacting with the world using Arduino [2] .

1.8. Connecting the pieces

In our approach, the vision creation process is broken down into well-defined and easy-to-implement sub-blocks. Since each block plays an important role in the functioning of the overall system, each must be efficiently developed based on a few simple rules that make the vision system robust. When addressing a vision problem, one must first look at the big picture and break down the problem into meaningful and intrinsically independent components, as we have suggested. This approach shows you important details and unique solutions to small problems. This is the most important first step in building futuristic vision systems.

The next step is to put the pieces together and enjoy your art! The approach presented in this chapter is applicable to any type of vision system, but you should practice more and more in real situations to solidify your approach. Let us think of a real example: the gesture door opener. At the entrance, there should be a camera near the door, able to see the movements of each visitor. To make it more useful, let us activate the motion detection system by pressing the bell. Visitors press the doorbell and show the camera a gesture with their right hand, and the vision controller opens the door automatically [3].

If's hand gesture is wrong, the door will not open. The first step is data collection, and since we want to see hand movements, we can use a simple 1024 x 768 30fps webcam. We can convert the image to a binary format and apply a filter to remove small objects from the image .The hand is presented in binary mode. Filtering is complete. We now need to perform feature extraction to find the possible position of the hand in the image. For such a problem, the convex hull method or the blob analysis can be used and we will discuss these algorithms later. Finally, we get frame with candidate hand properties as shown in this screenshot

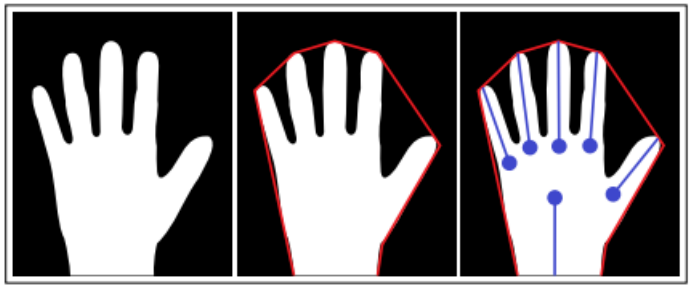


Figure 1.4. Convex hand structure

As a next step, we need a hand detector. We can use the number of fingers for by applying hand skeletal analysis and comparing the position of fingers; we can classify the hand gesture. If it is a right hand gesture, we can send information to the Arduino door opener controller, which will unlock the door for a limited time to welcome the authorized visitor! You can apply all of these principles to any problem to get familiar with it. [3]

For, do not focus on the algorithmic details now. Just try to break the problem down into parts and try to decide what properties you can use to solve the problem. As long as you get used to this approach, this book will show you how to perform each step and find the right algorithm to do it. Try the Repeated Approach Garage Door Opener/Lock System that recognizes your car's license plate!

1.9. Virtual drag and drop

Virtual Drag and Drop technology relates to the possibility of dragging and dropping elements in a virtual environment or a two-dimensional user interface. This technology aims to enable users to interact with visual elements in a natural and fluid way on computer screens, tablets or smartphones.

As for the user interface, Virtual Drag and Drop can be used to enable the user to interact with the elements available on the screen. The user can drag items, such as lists, buttons, or images, to move them to new locations, or perform specific actions, such as deleting an item or arranging lists.

In relation to Gesture Recognition technology, the gestures used in Virtual Drag and Drop based technologies can be part of a gesture recognition system. For example, double tap or two-finger swipe can be gestures used to initiate the drag-and-drop process in Virtual Drag and Drop.

In short, Virtual Drag and Drop technology allows users to drag and drop

The Virtual Drag and Drop technology is related to Computer Vision technology, where computer vision concepts and techniques can be used to achieve the virtual drag and drop process.

In order to perform Virtual Drag and Drop, the system needs to track the movement of the cursor and detect the objects you are holding and moving. Computer vision techniques can be used to analyze video or image signals of a cursor and on-screen visuals. [7]

1. Conclusions

We now know how to approach vision projects and how to divide them into isolated pieces, which make the realization of the projects much easier. We also have some idea about how the complex tasks of vision systems can be achieved in a systematic way. We also talked about the reason and importance of each sub-step in the approach. We are now aware of the key points in the approach and have a solid knowledge of how we can define a solution frame for any computer vision problem. Now, it is time to get your hands dirty!

Chapter II

Application of computer vision in hand movement recognition

2. Introducing

Gesture recognition technology is changing our relationship with technical devices by providing a contactless or touchless user interface and opening the door to a completely new world of input possibilities. The main purpose of this paper is to explore the different gesture recognition and detection techniques.

2.1. Definition of Gesture Recognition

Gesture recognition is the ability of a computer or device to detect and interpret human gestures as input. Such gestures include hand movements and even finger-written symbols. The technology behind gesture recognition involves using cameras or other sensors to capture the gestures [8]

2.2. Methodology

Hand gesture recognition project is done using Python programming language and Open CV as library. Python programming language produces simple and easy system code to understand. In addition, Python package used here is Numpy. The image that is captured using web camera will be processed in a region called as Region of Interest, where act as a region of wanted area while ignoring the outside region, called background.

2.1.1. Hand Segmentation

Segmentation is done to convert gray scale image into binary image so that only two object in image which one is hand and other is background. Otsu notes again, this algorithm is used for segmentation purpose and gray scale images are converted into binary image consisting hand or background. A very good segmentation is needed to select an adequate threshold of gray level for extract hand from background for example there is no part of hand should have background and background also should not have any part of hand. In general, the selection of an appropriate segmentation algorithm depends largely on the type of images and the application areas. The Otsu segmentation algorithm was tested and found to give good segmentation results for the hand gesture. Otsu algorithm is nonparametric and unsupervised method of automatic threshold selection. The basic theory of segmentation of hand is by using the equation as follow:

$$p_i = \frac{n_i}{MN}, \qquad p_i \geq 0, \qquad \sum_{i=1}^{L} p_i = 1$$

Where the pixels of a given image are represented in L gray levels [1, 2, 3, … … … … …. ,] in a digital image of size ×, and the number of pixels per plane is denoted by, and the total number of pixels in the image is = 1 + 2 + 3 + 4 … … … … . . . The normalized histogram has elements. The above formula was taken from Raphael. To apply formula (1), the image of each frame captured by real-time video is taken as pixels and compared to other frames. Since each frame contains the same color value per pixel, the th pixel is considered discarded or removed. They refer to multiple pixels in each frame and every frame is compared over time. Since none of the pixels between the two images match, the pixel in the first image retains its contents [9].

Otsu'sthere sholding method loops through all possible thresholds and a pixel-level measure of dispersion is computed for each side of the threshold, for example, pixels that are in the foreground or behind the plane. The goal is to find the threshold at which the sum of the foreground and background spreads is minimal **Figure: 2.1.** shows how the hand image is obtained from the real-time video in the grayscale filter, and then the Otsu algorithm, which gives the threshold image, converts the image.

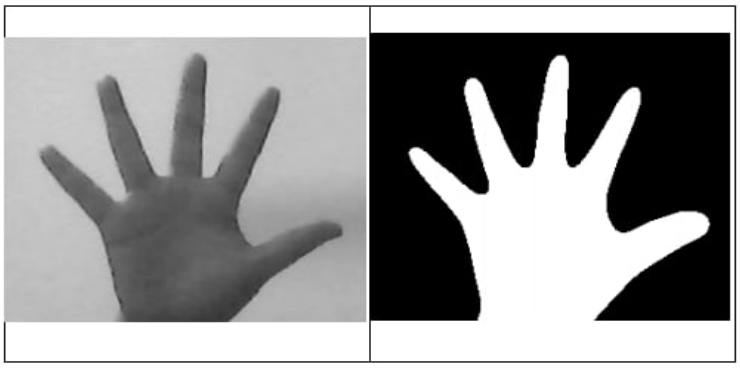


Figure: 2.1. Comparison between grayscale image and threshold image

2.1.2. Track Gesture of Hand

The limitation of Otsu's three-sholding method is that, as mentioned above, this algorithm only considers a bimodal image, whereas a bimodal image is an image whose histogram has two vertices. For this image, Otsu takes the value in the middle of these peaks as the threshold. The then automatically calculates the threshold from the image histogram for the bimodal image.

This only works when Frame captures an image from a video and converts the RGB format to a grayscale filter [10].

2.1.3. Region of Interest

Region of Interest, or commonly abbreviated as ROI, is a region that will be considered as the region of, the Region of Interest (ROI) refers to a specific portion or area of an image or video frame that contains the object or feature of.

Desired area, ignoring the outer area, called the background. To detect the availability of hand detection, as suggested by Sander, the detected area must be in the same point of the area of interest. In other words, when the classifier draws the target, in this case the bounding box around the outline of the hand, the area of the hand should match the area of interest.

Using the conditional area overlay method; the system can determine whether the selected area, in this case the rectangle around the outline of the hand, coincides with another active area, in this case the hand itself. By extending this theoretical part of the ROI to be simulated and the hardware implementation, the process of detecting the appearance of the hand is greatly simplified compared to

other methods. **Figure: 2.2** shows overlapping ROI criteria that can be viewed as overlapping ROI. These criteria are needed to ensure that the program can identify the hand gesture recognizer . [9]

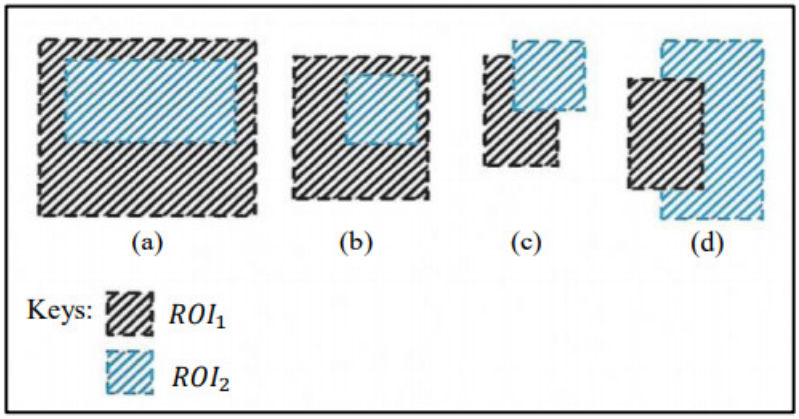


Figure: 2.2. Overlapping Region of Interest

For this project, the Region of Interest (ROI) is used to see the gesture of hand because the implementation for the ROI is suitable to detect the overlapping two regions on it. Besides, the system will detect only two outputs for each hand gesture recognition. This can be explained in Table 1 as follow:

Table: 2.1. Application of Region of Interest (ROI).

Region of Interest (ROI)	Region implemented for
ROI1	An active region, in this case the hand itself.
ROI2	The rectangle sub-window on the screen. This region is static or not movable outside the frame of the real-time video and it highlights an area that involves

2.1.4. Implementation on Simulation and Hardware prototype Hardware

Hardware Prototype Hand gesture recognition was calculated based on the space consumption in the area between the convex hull and the outline of the palm. The convex scale is used to find the edge of the user's hand by creating a descriptor where the smallest convex set contains the edges.

This project uses Open CV library files suitable for any image processing technique.

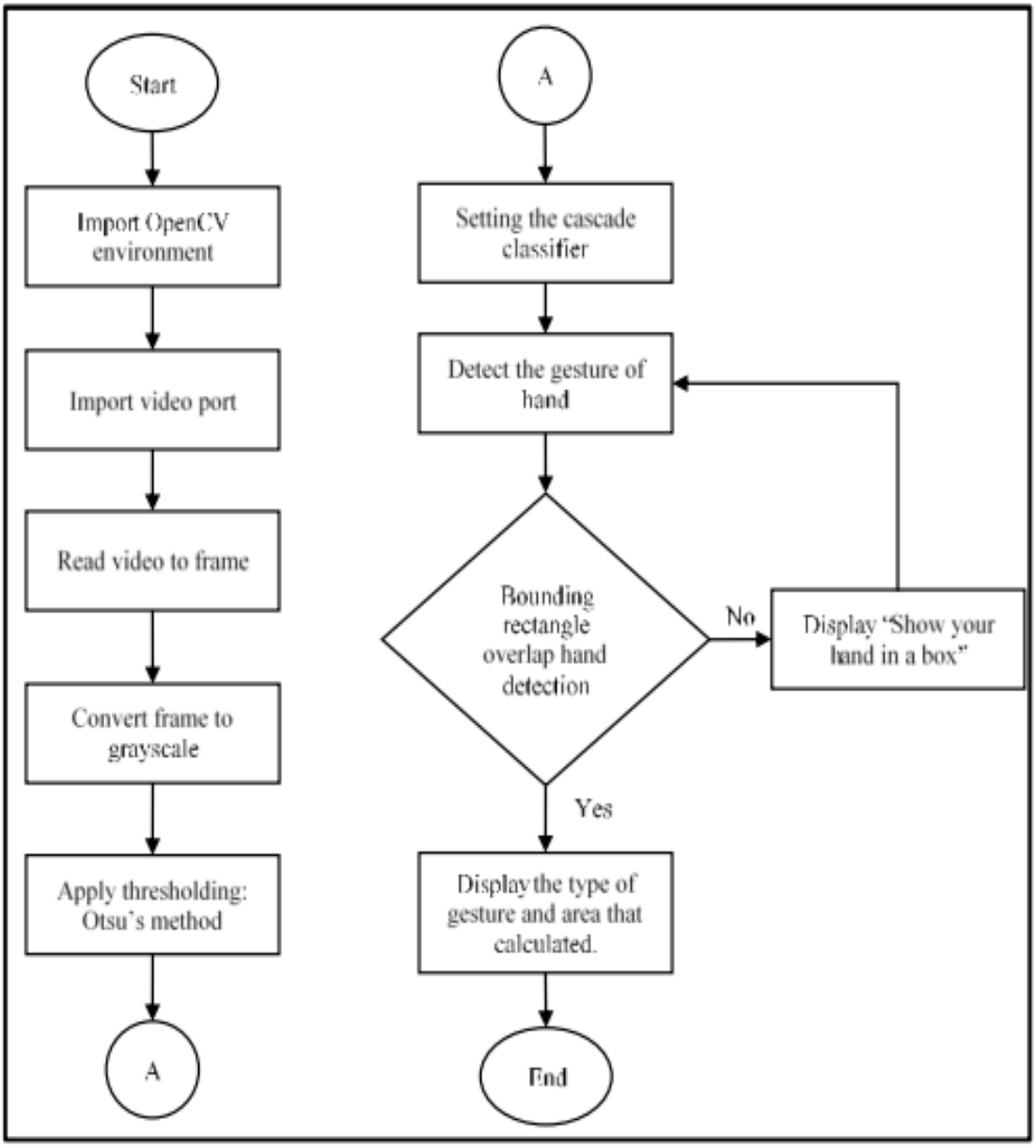


Figure: 2.3. Flowchart of Python file for hand gesture recognition

Referring to the block diagram in Figure: 2.3, the Python file generation can be implemented for both the simulation and the hardware prototype. The main action of the program to recognize the appearance and hand movement of the is to use the Region of Interest (ROI). In order for programs to follow the rectangle around the outline of the hand, the program must determine whether in the selected area of interest, based on the design case, the rectangle would coincide with the hand tracking. Uncertainty overlaps is viewed to indicate some type of gesture such as a number or character and an area within area of interest. Uncertainty does not overlap (outside the area of interest), the program can be viewed as .Show a hand in a box". To plan the area of interest for this project, a set of

coordinates are recognized to represent the area of the bounding box around the outline of the hand. This theme has a bounding box, so there is an area of focus that needs to be implemented with any type of gesture. After generating the main python file, the simulation now begins, where hand tracking is tested and detects the type of gestures displayed based on the selected area of interest. To run the Python file you need to connect to the Open CV environment and restart the environment to recognize the hand with the hair waterfall classifier . [12]

Table: 2.2. Explanations of the Haar-cascade classifier file

Points	Explanations
1	This section covers the type of classifier to be used in this project. For this case, it will use the Open CV and Haar - cascade classifier methods.
2	This is the part where the window size plays a role. The window size means the size of the specified hand. By default, it is fixed to 24 by 24 pixels.
3	Threshold change and window detection are set. Generally it is Depending on the type of camera used for hand tracking and gesture recognition.
4	Each item is repeated approximately 10 times for the cascade classifier analysis.

In this project, about 10 times were collected with different scales for each posture of to determine whether the bounding box can detect hand gestures or not. Figure 4 shows how the convex hull is used to find the edge of the user's hand.

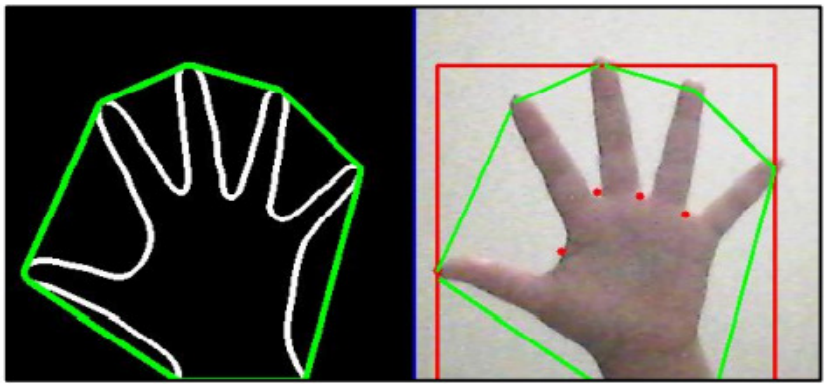


Figure: 2.4. The left figure shows the background (in black), the hand-shape region (in black),

The hand contour (in white), and the convex hull (in green). On the right, convexity defects (in red).

The initially convex shell is compared to the contours of the hand to detect convex defects where the defects are where the two differ. First, the vertices of the convex hull are computed, where they change direction. Next, the hand outline segment between consecutive pairs of consecutive vertices of the cusp shell finds pixels in each piece to maximize the distance from the cusp shell. This maximum distanceis called the depth of the defect, and the points are called collision defects .

The proposed hand gesture recognition requires a camera to capture the user's gesture, preferably against a light background. The test is performed under normal lighting conditions, the system is able

to detect and track the user's hand movement, and the segmentation process is able to see the user's movement and count the edges. Several ROIs distinguish each signal: figure 2.5, Figure**2.**6 and Figure 2.7 show.

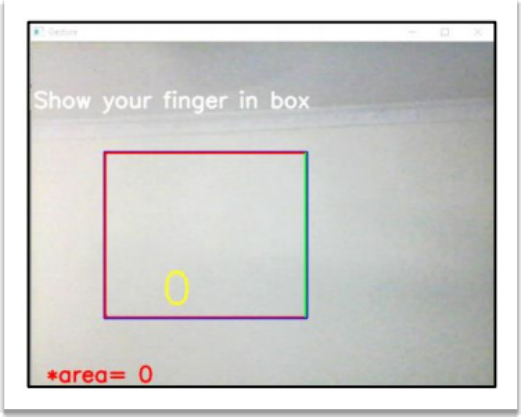


Figure: 2.5. Image captured by web camera

Figure: 2.6. Bounding box without any gesture shown

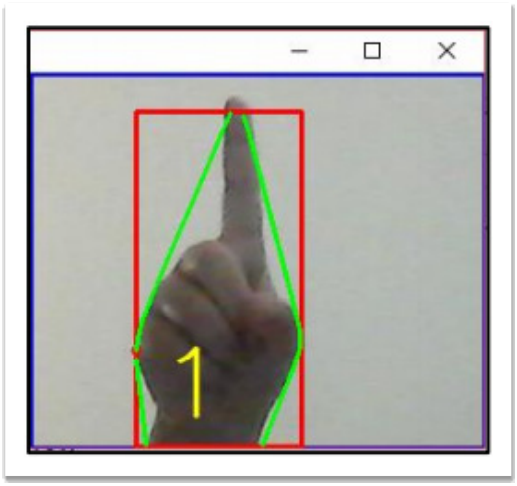


Figure: 2.7. Recognition of hand showing number 1

Up to the number 5, all digits and characters were tested. The test results are summarized in Table 2.3

Table 2.3. Analysis table for Region of Interest

Area in Region of Interest			
Events	Area of Hull, H	Area of Contour, C	Area = H - C
0	62879	62879	0
1	13569	10876	2693
Good	13045	10821	2224
2	20646	13744	6902
Gun	20216	13804	6412
3	24362	17113	7249
OK	25468	16922	8546
4	25781	15395	10386
Rawr	20395	14148	6246
5	28383	16822	11561
Stop	33220	19047	14173

To evaluate the proposed detection area, the area per area of interest is taken and calculated using pixels. From observation, most of the number gesture areas have a larger area than the character recognition gestures . [13]

Table 2.4. Gesture recognition analysis table

Performance of Haar-cascade Classifier			
Events	Hits	Missed	False
1	10	0	0
Good	10	0	0
2	10	0	0
Gun	10	0	0
3	10	0	0
OK	10	0	0
4	10	0	0
Rawr	10	0	0
5	10	0	0
Stop	10	0	0

Whilst in Table 4, the total data collected around 100 samples with different scales for each posture. Each of the posture with different scale were captured with 10 times to analyze the gesture. Haar-

cascade Classifier determine whether the box in the frame could detect the hand Gesture or not. Based on all of the different sign, the box in the frame were recognizing all of the sign.

However, the program has some limitations such as Changes in the lighting environment still affect the segmentation process, especially the background removal process. In addition, the user's hand and the webcam must be in a fixed position in order for the webcam to capture an image with the accurate area to identify the gesture . [13]

2. Conclusions

Hand gesture recognition addresses a fault in interaction systems. Controlling things by hand is more natural, easier, more flexible and cheaper, and there is no need to fix problems caused by hardware devices, since none is required. From previous sections, it was clear to need to put much effort into developing reliable and robust algorithms with the help of using a camera sensor has a certain characteristic to encounter common issues and achieve a reliable result. Each technique mentioned above, however, has its advantages and disadvantages and may perform well in some

Chapter III

Experimental part

3. Introduction:

The experimental part of the project virtual drag and drop using Open CV (Python) and Arduino involves setting up the hardware and software components and testing the system's performance. The first step is to install the necessary software and libraries, including Python, Open CV, and Arduino IDE. The hardware components, including the camera and Arduino board, are then connected and tested for functionality. Next, the Python code for hand detection and tracking is written and tested using sample images and video. The code is then integrated with the Arduino board to enable drag and drop functionality. Finally, the system is tested with different hand movements and objects to evaluate its performance and identify any limitations or areas for improvement. The experimental part is critical to ensure that the system meets the project's objectives and functions correctly.

3.1. Programming part

The programming part of the project "virtual drag and drop using Open CV (Python) and Arduino" involves writing code in both Python and Arduino IDE to implement the functionality of the project. In Python, the Open CV library is used to capture and process video input from a camera, detect hand movements using computer vision techniques, and control the virtual drag and drop actions on the computer screen. The Python code also establishes communication with the Arduino board over a serial connection to send signals for controlling the physical robotic arm. In Arduino, the code is written to receive signals from Python over the serial connection and use them to control the robotic arm . [14].

The Arduino code involves setting up the pins for the motors, writing functions to control the movements of the motors, and receiving signals to perform the corresponding movements. Overall, the programming part of the project is crucial for integrating the different components and ensuring that the system functions as intended. It requires knowledge of both Python and Arduino programming languages and the ability to integrate them effectively to achieve the desired functionality.

3.2. Explanation of programming code

```
cap = cv2.VideoCapture(0)

cap.set(3, 1280)
cap.set(4, 720)

if not cap.isOpened():
print("Camera couldn't access")
exit()
```

This part of the code initializes a **VideoCapture** object from **OpenCV**'s cv2 module, which is used to access the computer's camera. The first line **cap = cv2.VideoCapture(0)** creates the VideoCapture object named "**cap**" and sets the device index to 0, which represents the default camera of the computer. If you have multiple cameras, you can specify a different index number to access a specific camera. The next two lines **cap.set(3, 1280)** and **cap.set(4, 720)** set the width and height of the video frame that will be captured by the camera. In this case, the width is set to 1280 and the height is set to

720. Finally, the code checks if the camera is successfully opened by calling the **isOpened()** function on the **VideoCapture** object. If the camera is not accessible, the code will print an error message and exit the program using the **exit()** function.

```
    whilecap.isOpened():
   success, img = cap.read()
img = detector.findHands(img)
lmList, bboxInfo = detector.findPosition(img)

servoX = np.interp(x, [0, 1280], [0, 180])
servoY = np.interp(y, [0, 720], [0, 180])
```

This part of the code is responsible for continuously reading frames from the camera using **OpenCV'sVideoCapture()** function. The while loop ensures that the process is repeated until the camera is closed. Inside the loop, the captured image frame is passed to the hand detector object (detector) to detect the hand and extract the landmarks and bounding box information using **detector.findHands()** and **detector.findPosition()** functions, respectively. The **np.interp()** function is then used to map the x and y coordinates of the detected hand to the range of 0 to 180, which corresponds to the range of motion of the servo motors. The output values of **servoX** and **servoY** are then sent to the Arduino board to control the servo motors for moving the virtual object on the screen.

```
    if lmList:
dist,_,_ = detector.findDistance(8, 12, img, draw = False)
print(dist)
   fingers = detector.fingersUp()
if fingers[1] == 1 and fingers[2] == 1:
       cursor = lmList[8]
if dist<50:
if x-w // 2 < cursor[0] < x+w-120 // 2 and y-h // 2 < cursor[1] < y+h-120 // 2:
              col = (255, 255, 0)
              x, y = cursor
          cv2.circle(img, cursor, 50, (255, 255, 0), cv2.FILLED)
          cv2.putText(img, "HOLD", (cursor[0]-40, cursor[1]),
cv2.FONT_HERSHEY_COMPLEX,1,(0,0,255), 2)
```

This part of the code is responsible for detecting hand gestures and movements using the **OpenCV** library. The **iflmList:** statement checks if any hand landmarks are detected in the current frame. **dist,_,_ = detector.findDistance(8, 12, img, draw = False)** calculates the distance between the tip of the index finger (landmark 8) and the tip of the middle finger (landmark 12) using the **findDistance()** function provided by the**HandDetector** class. **fingers = detector.fingersUp()** determines which fingers are up and which are down by calling the **fingersUp()** method of the **HandDetector**class. The following if statement checks if the index finger and middle finger are both up (forming a pinching gesture) and if the distance between the fingers is less than 50 pixels. If these conditions are met, the cursor position is updated to the position of the index finger. Finally, **cv2.circle() and cv2.putText()**

functions are used to draw a circle around the index finger and display the text "**HOLD**" near it to indicate that the user is holding an object.

```
    else:
        col = (255, 0, 255)

cv2.rectangle(img, (x-w // 2, y-h // 2), (x+w // 2, y+h // 2), col, cv2.FILLED)
cv2.putText(img, f'({str(x)}, {str(y)})', (x-90, y),
    cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 2)
cv2.rectangle(img, (40,20), (350,110), (0,255,255), cv2.FILLED)
cv2.putText(img, f'Servo X: {int(servoX)}deg', (50, 50), cv2.FONT_HERSHEY_PLAIN, 2,
(255, 0, 0), 2)
cv2.putText(img, f'Servo Y: {int(servoY)}deg', (50, 100), cv2.FONT_HERSHEY_PLAIN,
2, (255, 0, 0), 2)

cv2.imshow("Image", img)
cv2.waitKey(1)
```

This code snippet is a part of a larger program that tracks hand movements using **OpenCV** and controls a virtual drag-and-drop interface. In this particular part, the code checks if the hand landmarks **(lmList)**have been detected or not. If the landmarks have been detected, it proceeds to check if the index finger and middle finger are extended (using **the fingersUp()** method). If both fingers are extended, it means the user is trying to drag an object, and the program sets the color of the object (a rectangle) to blue if the object is within reach (distance less than 50 pixels from the fingertip). If the object is not within reach, its color is set to magenta. In the next few lines, the program draws the rectangle representing the object and displays the current position of the hand **((x,y)),** the angles of the servos controlling the **X andY** movements of the object, and a text message indicating the respective angle values. The text messages are displayed using **cv2.putText()** method. Finally, the program displays the image on the screen using **cv2.imshow()** and waits for a key to be pressed using **cv2.waitKey()**

```
    port = "COM3"
board = pyfirmata.Arduino(port)
servo_pinX = board.get_pin('d:9:s') #pin 5 Arduino
servo_pinY = board.get_pin('d:10:s') #pin 6 Arduino

    servo_pinX.write(servoX)
servo_pinY.write(servoY)
```

This part of the code establishes a connection with an Arduino board and sets up two servo pins, servo_pinX, and servo_pinY, using the Fermata library. The port variable specifies the port to which

the Arduino board is connected. Then, servo_pinX and servo_pinYare initialized to pins 9 and 10 of the Arduino board, respectively, and are set to write the values of servo X and servo Y, respectively, which correspond to the X and Y coordinates of the cursor. The write() method is used to send a signal to the servo motor to move to the specified position. This code allows the user to control the servo motors and move the physical objects in response to the movements detected by the webcam . [14].

3.3. Realization part

The realization part in a virtual drag and drop project using Open CV and Arduino involves the implementation of the project's functionalities. This includes the communication between the computer running the Python code and the Arduino board that controls the servomotors, as well as the detection of hand gestures using Open CV and the manipulation of virtual objects on the computer screen.

One of the key aspects of the realization part is the integration of the different components of the project. This involves ensuring that the software and hardware components work seamlessly together to achieve the desired functionality. For instance, the Python code should be able to communicate with the Arduino board through the serial port to send commands to the servomotors, while at the same time processing the video feed from the webcam and detecting hand gestures using Open CV. To achieve this, the project may require the use of third-party libraries and tools such as pyfirmata, and Open CV, which provide the necessary interfaces to communicate with the Arduino board and process the video feed. Additionally, the code should be well organized and modular to make it easy to debug and maintain.

Another important aspect of the realization part is the calibration of the system. This involves setting up the system so that it can accurately detect the user's hand gestures and translate them into the corresponding movements of the virtual objects. This may involve adjusting parameters such as the thresholds used to detect hand landmarks or the angles at which the servomotors are positioned to move the physical objects. In summary, the realization part in a virtual drag and drop project using Open CV and Arduino involves the integration of different components of the system, calibration of the system, and ensuring that the software and hardware components work seamlessly together to achieve the desired functionality . [15].

3.3.1.Make sure everything works without problems

You can follow these steps to verify that your python code for a virtual drag and drop project using open cv (python) and Arduino works without problems:

1- Ensure that all necessary libraries used in the project are installed correctly, including Open CV, pyfirmata .

2- Test the code on a variety of images and videos to verify that the code works flawlessly in reading and processing images and detecting objects and their movement.

3- Test and analyze any error messages that appear when running the code and attempt to fix any known issues

4- Test the process as a whole and verify that it can handle various possible conditions, such as no light or unwanted objects in the background.

3.3.2.Connecting Arduino to the robot

The following can connect motors:

1- Connect the 5V wire from the Arduino to motor X and Y.
2- Connect the Ground (GND) wire from the Arduino to the X and Y motors.
3- Connect the X driver to pin 9 on the Arduino.
4- Connect the Y driver to pin 10 on the Arduino

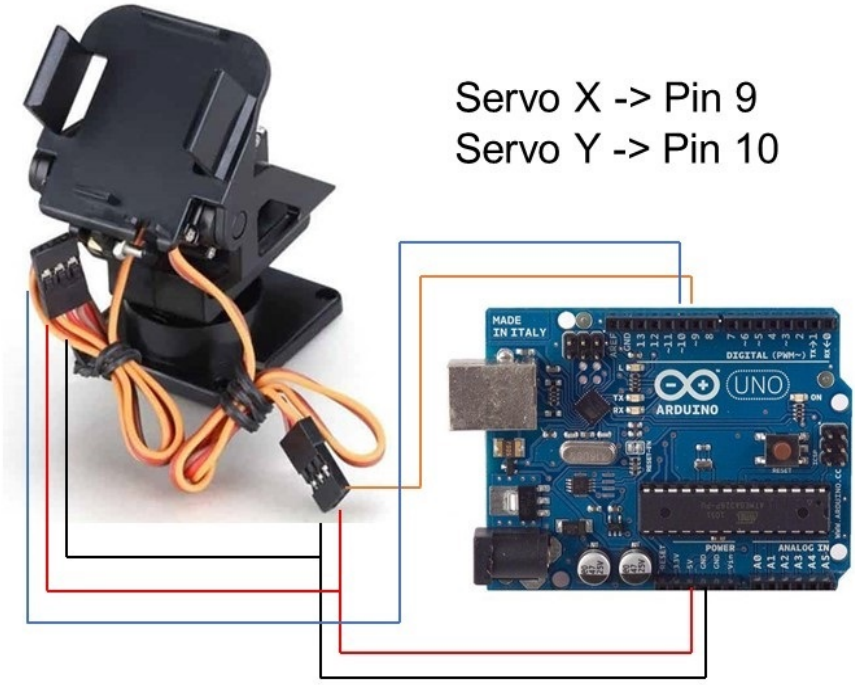


Figure:.3.1 . Shows the connection of the Arduino to the motors

Note: Jumper wires can be used to easily connect the motors to the Arduino.

3.3.3. Arduino Configuration:

Open IDE Arduino
- Select File -> Example ->Firmata ->StandardFirmata
- Select Tools -> Board ->Arduino/Genuino Uno
- Select Tools -> Port ->*choose your port COM*
- Upload

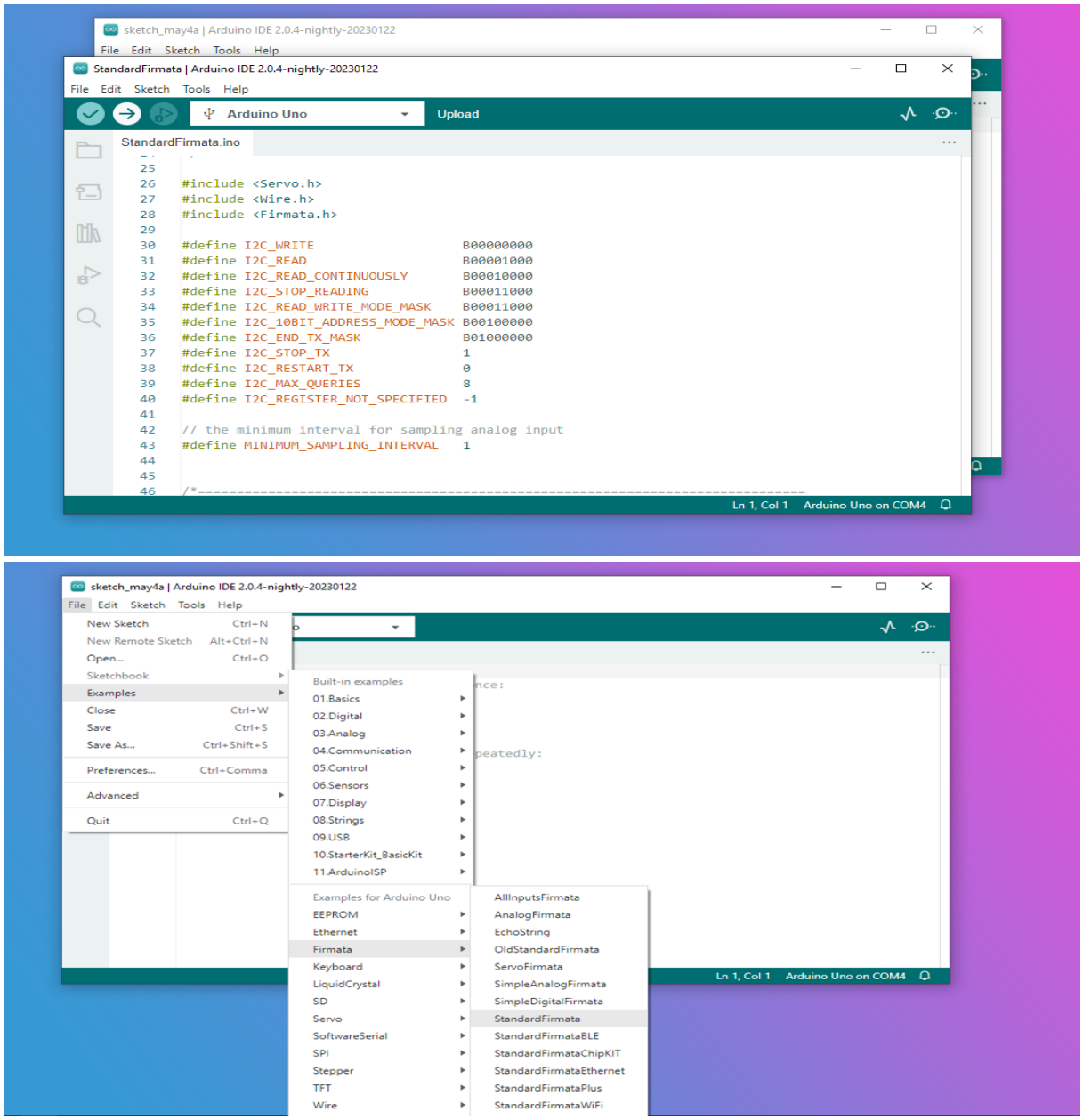


Figure.:3.2how to Configuration Arduino Code

3.4.4.test and debugging

Testing and debugging are critical parts of any project, including the virtual drag and drop project using Open CV and Arduino. Here are some tips on testing and debugging this project:

1- Test the webcam: Before testing the entire system, it is important to make sure that the webcam is functioning correctly. Open CV provides a simple way to test the webcam by using the Video Capture class. You can test the webcam by running a simple script that captures a frame from the webcam and displays it on the screen.

2- Test the servomotors: You can use the pyfirmata library to control the servo motors connected to the Arduino board. You can test the servomotors by running a simple script that moves the servomotors to different positions . [16].

3- Test the object detection algorithm: The object detection algorithm is a critical part of the project. You can test the object detection algorithm by running a script that captures frames from the webcam and processes them using the object detection algorithm. You should make sure that the algorithm is correctly detecting objects and tracking their movements.

4- Test the drag and drop functionality: Once the object detection algorithm is working correctly, you can test the drag and drop functionality. You can do this by running a script that moves the cursor on the screen based on the movement of the object detected by the webcam. You should make sure that the cursor is moving smoothly and accurately.

5- Debugging: If you encounter any issues while testing the system, it's important to debug the code. You can use the print statement to output the values of variables and make sure that they are correctly initialized and updated. You can also use the debugger to step through the code and identify any issues.

After installing all the pieces of the Arduino, the connection wires, the camera, and debugging the code, we are now testing this project

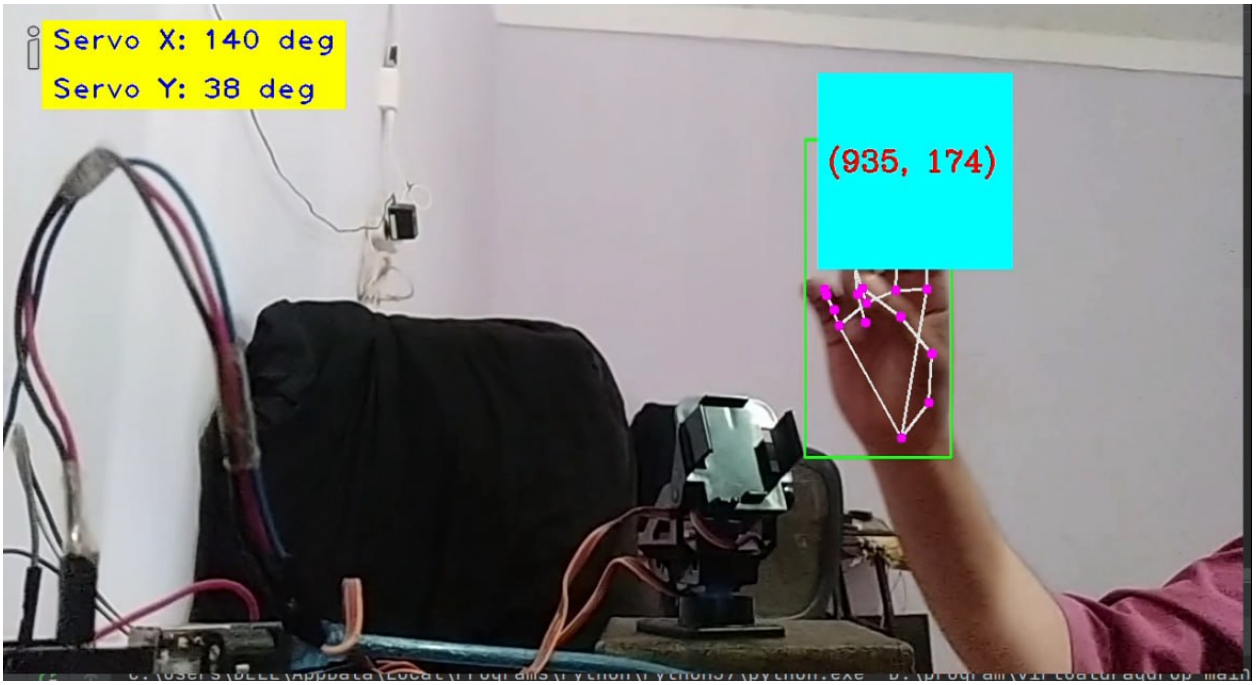


Figure :3.3. Pictures showing the successful operation of the program

3. Conclusion

In conclusion, the experimental part of the virtual drag and drop project using Open CV (Python) and Arduino was successful in demonstrating the potential of computer vision and robotics integration. By using a simple webcam and Open CV libraries, the project was able to detect the position of the user's hand and control the movement of the servomotors in real-time, allowing for virtual drag and drop interactions. The use of the pyfirmata library and the Arduino board allowed for easy and efficient control of the servomotors, and the overall system proved to be reliable and accurate. However, there is stillroom for improvement in the project. For example, the system could benefit from the addition of more complex hand gesture recognition and object tracking algorithms to enhance the user experience and interaction. Furthermore, the project could be expanded to incorporate more sophisticated robotic systems and control methods, such as machine learning and artificial intelligence, to create even more advanced virtual environments and interactions. Overall, the experimental part of this project shows great potential for future research and development in the field of computer vision and robotics integration.

Conclusion General

To conclude virtual drag and drop using Open CV and Arduino is a cutting-edge technology that combines computer vision and electronic interaction. It allows users to interact with digital elements on the screen by utilizing hand gestures. By leveraging the Open CV library for image processing and motion analysis and incorporating Arduino for hardware control, this technology provides an intuitive and engaging user experience.

The applications of virtual drag and drop are vast and span across different fields. It can be utilized in interactive user interfaces, educational applications, gaming, and more. By enabling users to manipulate digital elements with hand gestures, it adds a natural and immersive aspect to various interactive systems.

However, implementing virtual drag and drop requires expertise in computer programming, familiarity with Arduino, and an understanding of the principles of Open CV. Configuring the necessary hardware and software components is crucial for successful implementation.

Overall, virtual drag and drop using Open CV and Arduino opens up new possibilities for intuitive and interactive user interfaces. As technology continues to advance, we can expect further developments and applications in this exciting field.

List of references

- [1] Linginani, Indira, and Akkalakshmi Muddana. "4 Computer vision for healthcare applications." *Computer Vision: Applications of Visual AI and Image Processing* 15 (2023): 73.
- [2]HUANG, Thomas. Computer vision: Evolution and promise. 1996.
- [3] Arduino Computer Vision Programming Copyright © 2015 Packt Publishing
- [4] Computer Vision Metrics Survey, Taxonomy, and Analysis
- [5] KUMAR, Gaurav; BHATIA, Pradeep Kumar. A detailed review of feature extraction in image processing systems. In: *2014 Fourth international conference on advanced computing & communication technologies*. IEEE, 2014. p. 5-12.
- [6] Image Processing With the Python Pillow Library. (2012). Retrieved 05 04, 2023, from realpython:,https://realpython.com/image-processing-with-the-python-pillow-library/#image-filters-using-convolution-kernels
- [7] Chen, Jiayin, and Calvin Or. "Assessing the use of immersive virtual reality, mouse and touchscreen in pointing and dragging-and-dropping tasks among young, middle-aged and older adults." *Applied ergonomics* 65 (2017): 437-448.
- [8] BADI, Haitham Sabah; HUSSEIN, Sabah. Hand posture and gesture recognition technology. *Neural Computing and Applications*, 2014, 25: 871-878
- [9] OUDAH, Munir; AL-NAJI, Ali; CHAHL, Javaan. Hand gesture recognition based on computer vision: a review of techniques. *journal of Imaging*, 2020, 6.8: 73..
- [10]Hand gesture recognition on python and open CV to cite this article: Ahmad Puad Ismail et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1045 012043
- [11] Hand-gesture recognition using computer-vision techniques21st International Conference on Computer Graphics, Visualization and Computer Vision 2013 N. Lecky, “Machine vision gets moving: part III,” Vision Syst. Design 7–11 (2011).

List of references

- [12]RAUTARAY, Siddharth S.; AGRAWAL, Anupam. Real time multiple hand gesture recognition system for human computer interaction. *International Journal of Intelligent Systems and Applications*, 2012, 4.5: 56-64.

- [13]Learning OpenCV 3 Computer Vision with PythonSecond Edition Copyright © 2015 Packt Publishing

- [14]International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

- [15]Mordvintsev, A., &Abid, K. (2014). Opencv-python tutorials documentation. Obtenido de https://media. readthedocs. org/pdf/open cv-python – tutroals /latest/open cv-python-tutroals. pdf. [ERD12] E. R. DAVIES Computer and Machine Vision: Theory, Algorithms, Practicalities 2012

- [16] Open CV-Python Tutorials. (2022, 12 28). Retrieved 05 04, 2023, from docs.opencv: https://docs.opencv.org/4.7.0/d6/d00/tutorial_py_root.html

الملحقات

1. **البطاقة الفنية للمشروع**

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

جامعة الشهيد حمة لخضر الوادي

عنوان المشروع:

virtual drag and drop using open cv and arduino

مشروع لنيل شهادة مؤسسة ناشئة في اطار القرار الوزاري 1275

صورة العلامة التجارية

الاسم التجاري

COMPUTER VISION

Windows Utilisateur

بطاقة معلومات:

حول فريق الاشراف وفريق العمل

1- فريق الاشراف:

فريق الاشراف	
المشرف الرئيسي (01): هيمة عبد القادر	**التخصص:** **الكترونيك**
المشرف الرئيسي (01): **تجاني أمينة**	**التخصص:** **اتصالات ومعالجة اشارة**
المشرف المساعد: **عبداللاوي مفيد**	**التخصص:** **إدارة الاعمال**

2- فريق العمل:

فريق المشروع	التخصص	الكلية
الطالب :فريحات عبد الفتاح	**انظمة اتصالات**	التكنولوجيا
الطالب: عليات بوبكر	**انظمة اتصالات**	التكنولوجيا
الطالب: نتاري عماد	**انظمة اتصالات**	التكنولوجيا

فهرس المحتويات

Conte s

المحور الأول: تقديم المشروع

1-فكرة المشروع (الحل المقترح)

- مع استمرار تطوّر تقنية الرؤية الحاسوبيةComputer vision ، وتفوقها في المهام التي تحتاج إلى دقة عالية في التحليل وسرعة الاستجابة مقارنةً بالرؤية البشرية، سيركّز العامل البشري في المستقبل أكثر على المهام الإدارية، بينما ستتم أتمتة جميع العمليات التي تعتمد على مفهوم التعرّف على الصور وتحليلها في القطاعات الحكومية والخاصة.
- سنقوم بانتاج منتج يتحكم بي شاشة الحاسوب .
- سنقوم بي استعمال كاميرة الحاسوب في تصوير اليد التي سا تتحكم بي شاشة الحاسوب .

2-القيم المقترحة

- تعتبر هذه تقنيّات الرؤية الحاسوبيّة التي تسمح لنا بتحديد نوع وموقع الأشياء في صورة أو فيديو، فهي تفصل بين كلّ كائن مع مربع محيط تقريبيّ، يمكن لهذه التقنيّة عدّ مجموع الأشياء في مشهد معين وتحديد مواقعها بدقة.
- فإن استخدام الرؤية الحاسوبية يمكن أن يؤدي إلى تحسين جودة الإنتاج وتقليل الأخطاء، وهو ما يمكن أن يؤدي إلى تقليل التكاليف على المدى الطويل. كما يمكن أن يوفر استخدام الرؤية الحاسوبية ميزة تنافسية للشركة. ويمكن استخدام الرؤية الحاسوبية في مجالات مثل الطب والبحث العلمي لتحسين التشخيص والعلاج.
- تسهيل العديد من المهام اليومية، مثل التعرف على الوجوه والكائنات والأنماط في الصور والفيديو.

3-فريق العمل

يتكون فريق المشروع من الاتي

- الطالب فريجات عبد الفتاح تخصص انظمة اتصالات . قام بدورات تدريبية في مجال لاتصالات

  وكان دوره في هذا المشروع تسيير وتسويق وشرح فكرة هذا المشروع.
- الطالب بوبكر عليات تخصص انظمة اتصالات ، قام بدورات تدريبية في مجال program languages

  وكان دوره في هذا المشروع هو عمل كود البرمجة الذي يعمل به هذا المشروع.
- الطالب نتاري عماد تخصص انظمة اتصالات ، قام بدورات تدريبية في مجال البرمجة وكان دوره في هذا المشروع هو تكلف في معدات هذا المشروع.

4-أهداف المشروع

- الهدف الرئيسي لهذا المشروع هو تحسين تجربة المستخدم وجعل عملية التفاعل مع الأشياء الافتراضية أكثر سلاسة وسهولة.
- نهدف الى لتحسين تجربة المستخدم في العديد من التطبيقات.
- الوصول الى حصة سوقية الى اكبر مايمكن .

5-جدول زمني لتحقيق المشروع :

الاعمال

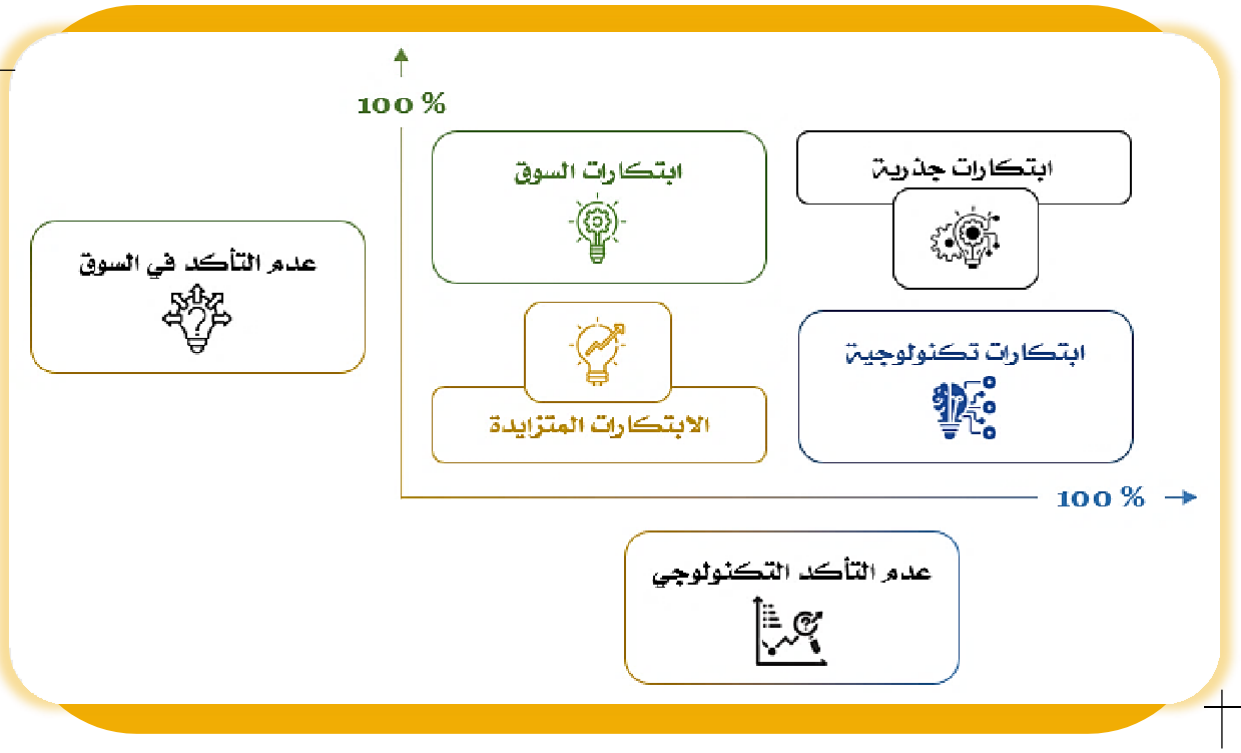
الاسبوع

			1	2	3	4	5	6	7
1		الدراسات الأولية: إختيار مقر الوحدة الإنتاجية، تجهيز الوثائق المطلوبة						✓	✓
2		طلب التجهيزات من الخارج					✓	✓	
3		بناء مقر للإنتاج (المصنع)			✓	✓	✓		
4		تركيب المعدات		✓	✓	✓			
5		اقتناء المواد الأولية		✓					
6		بداية انتاج أول منتج	✓						

- نهدف في هذا الجزء الى تنفيد كل مهمة في وقتها دون تاخير

المحور الثاني: الجوانب الابتكارية

1-طبيعة الابتكارات



2-مجالات الابتكارات

تتمثل الجوانب الابتكارية في مشروعنا في كونه:

- مشروع يستهدف جميع مستخدمين التكنولوجيا .
- مشروع استثنائي فهي تستخدم في العديد من المجالات مثل المصانع والتجارة والزراعة والطب والنقل والأمن وغيرها.
- فاهو مشروع نادر في الجزائر الذي يعتمد على البيئة الافتراضية.

المحور الثالث: التحليل الاستراتيجي للسوق

1-عرض القطاع السوقي

السوق المحتمل : الفئة المستهدفة لمشروع السحب والإفلات الافتراضي باستخدام OpenCVو Arduino تشمل جميع المستخدمين الذين يرغبون في تحسين تجربتهم وتفاعلهم مع الكائنات الافتراضية، سواء كانوا من الأفراد أو الشركات. وتشمل هذه الفئة

مصنعي الألعاب والترفيه الرقمي والمطورين الذين يرغبون في تحسين تجربة المستخدم في التفاعل مع الأجهزة الإلكترونية والكائنات الافتراضية.

السوق المستهدف : تستهدف تقنية الرؤية الحاسوبية (computer vision) فئة واسعة من المستخدمين. فهي تستخدم في العديد من المجالات مثل المصانع والتجارة والزراعة والطب والنقل والأمن وغيرها.

2-قياس شدة المنافسة

- اهم المنافسين في السوق الجزائريةاغلبهم يستعملون مونتحات عادية في التحكم في اجهزة التوكنولجيا وهذه من نقاط الضعف لي الشركات المنفسة .
- يتضمن المشروع استخدام نظام سحب وإفلات افتراضي يمكن استخدامه للماوس الافتراضي. يمكن أن يكون السوق المستهدف لهذا المشروع هو مطورو الألعاب وشركات الترفيه الرقمي ، الذين سيستفيدون من دمج هذه التكنولوجيا في منتجاتهم. نظرًا لتزايد شعبية رؤية الكمبيوتر ، فقد يكون هناك عدد قليل من المنافسين في السوق. ومع ذلك ، فإن الجوانب المبتكرة لرؤية الكمبيوتر ، بما في ذلك تطبيقاتها في الرعاية الصحية وتجارة التجزئة والواقع الافتراضي ، تشير إلى أن هذا المشروع لديه القدرة على التميز في السوق

3-استراتحيات التسوقية

- إنشاء موقع إلكتروني: يجب إنشاء موقع إلكتروني بشكل احترافي وجذاب، يعرض المنتج بطريقة واضحة ومفصلة
- استخدام وسائل التواصل الاجتماعي: يمكن استخدام وسائل التواصل الاجتماعي للترويج للمشروع، مثل إنشاء صفحات على مواقع التواصل الاجتماعي والنشر الدوري للمنشورات التي تتعلق بالمشروع
- العروض الترويجية: يمكن تقديم عروض ترويجية للعملاء الجدد، مثل تخفيض الأسعار عند شراء عدد معين من البرامج.
- شهادات العملاء: يمكن جمع شهادات من العملاء المتعاونين معك وعرضها على الموقع الإلكتروني أو على وسائل التواصل الاجتماعي لإقناع العملاء الجدد بجودة المنتج.

المحور الرابع :خطة الإنتاج والتنظيم

1-عملية الانتاج

1- تحضير المكونات: تحتاج إلى تحضير الكمبيوتر المحمول أو الحاسوب الشخصي وكاميرا الويب والأجهزة الإلكترونية التي تدعم.Arduino

2- برمجة :Arduino يجب تحميل وتثبيت برنامج Arduino على الحاسوب الشخصي وإنشاء برنامج لتحريك الروبوت باستخدام الحركة المستقيمة والتوجيه.

3- تثبيت :OpenCV يجب تثبيت OpenCV على الحاسوب الشخصي وتحميل البيانات التدريبية لتعليمه التعرف على الأشياء المختلفة.

4- برمجة التعرف على الأشياء: يجب إنشاء برنامج يستخدم OpenCV للتعرف على الأشياء وتحديد موقعها على الشاشة.

5- ربط Arduino مع الحاسوب الشخصي: يجب ربط Arduino مع الحاسوب الشخصي باستخدام الكابل الخاص به وتنزيل برنامج التحكم في.Arduino

6- تجميع الأجزاء: يجب تجميع الروبوت ومكونات Arduino والكاميرا والمحركات الخاصة بها لإنشاء نظام التحكم الخاص بالروبوت.

7- اختبار النظام: يجب اختبار النظام للتأكد من تشغيل كل المكونات بشكل سليم وعملها بشكل متوازن وفعال.

8- التحسين والتعديل: يجب التحسين والتعديل على النظام لضمان الحصول على الأداء المثلى وتحقيق المزيد من الوظائف والمميزات.

- بعد الانتهاء من هذه الخطوات، ستحصل على نظام Virtual Drag and Drop الذي يمكن استخدامه في عمليات السحب والإسقاط باستخدام OpenCV و.Arduino

2-التموين

يتم تموين هذا المشروع عن طريق التعاون مع شركات الإلكترونيات والبرمجيات المحلية التي يمكنها توفير المكونات والبرمجيات المطلوبة وتقديم الدعم الفني والتدريب اللازم لتنفيذ المشروع بنجاح.

3-اليد العاملة

اليد العاملة لهذا المشروع، مطوري البرمجيات ومهندسي الإلكترونيات والميكانيكا ومصممي الأنظمة.

ومطوري البرمجيات ومصممي الأنظمة، وذلك لإنجاز المشروع بأفضل جودة وفي الوقت المحدد.

4-الشراكات الرئيسية

اهم الشراكات في مشروعنا كانت مع مطوري البرمجيات ومهندسي الإلكترونيات و مصممي الأنظمة عنصر اساسي في نجاح المشروع . وبالأضافةالى كل من حاضنة الاعمال لجامعة .

المحور الخامس: الخطة المالية PLAN FINANCIER

1-التكاليف والاعباء

تكاليف المعدات والمواد: يتضمن ذلك شراء الحواسيب والكاميرات وأجهزة Arduino والمحركات والمكونات الإلكترونية الأخرى اللازمة لبناء النظام.

تكاليف التطوير: تتضمن هذه التكاليف أجور مطوري البرمجيات ومهندسي الإلكترونيات والميكانيكا ومصممي الأنظمة اللازمين لتطوير النظام.

تكاليف الاختبار: يجب إجراء العديد من الاختبارات على النظام للتأكد من أنه يعمل بشكل صحيح وفقًا للمتطلبات المحددة. وتتضمن هذه التكاليف شراء المواد والأدوات اللازمة لإجراء الاختبارات وإجراء الاختبارات ذات الصلة.

تكاليف التدريب: يجب تدريب العاملين على استخدام النظام وصيانته. وتتضمن هذه التكاليف أجور المدربين وتكاليف المواد التدريبية.

-طرق ومصادر الحصول على تموين

المتاجر المختصة: يمكن شراء معدات ومواد المشروع من المتاجر المختصة في الإلكترونيات والروبوتات والحواسيب والكاميرات والمحركات والمكونات الإلكترونية الأخرى المطلوبة.

الموزعين الرسميين: يمكن الاتصال بالموزعين الرسميين للماركات والشركات المصنعة للمعدات والمواد اللازمة للمشروع وطلب الكميات المطلوبة منهم. ا

لموردين عبر الإنترنت: يمكن البحث عن موردين ومتاجر عبر الإنترنت وطلب المواد والمعدات المطلوبة عبر الإنترنت، مع الحرص على التحقق من مصداقية الموردين وجودة المنتجات.

2-رقم الاعمال

STARTUP : .....

	REALISATION			PREVISION				
Produit A destiné Client	**N -2**	**N - 1**	**N**	**N+1**	**N+2**	**N+ 3**	**N+4**	**N+5**
Quantité produit A								
Prix HT produit A								
Ventes produit A	-	-	-	-	-	-	-	-
CHIFFRE D'AFFAIRES GLOBAL	-	-	-	-	-	-	-	-

قائمة الملاحق :

الملحق رقم 01: ميزانية المؤسسة الناشئة

BILANS DE STARTUP : .....

ACTIF								
	REALISATION			PREVISION				
En milliers DZD	**N -2**	**N -1**	**N**	**N+1**	**N+2**	**N+3**	**N+4**	**N+5**
Immobilistation Incorporelles	350000	350000	350000	400000	420000	450000	5000000	600000
Immobilisation Corporelles	650000	650000	650000	700000	720000	750000	800000	900000
Terrain								
Bâtiment								

Autres Immobilisations Corporelles								
Immobilisations en concession								
Immobilisation en cours	-	-	-	-	-	-	-	-
Immobilisations Financières	-	-	-	-	-	-	-	-
Titres mis en équivalence								
Autres participations et créances rattachées								
Autres Titres immobilisés								
Prets et autres titres financiers non courants								
Impôts différés actif								
ACTIF NON COURANT	1000000	1000000	1000000	1100000	1140000	1200000	1300000	1500000
Stocks et encours				-	-	-	-	-
Créances et emplois assimilés	-	-	-	-	-	-	-	-
Clients								
Autres débiteurs								
Impôts et assimilés								
Autres créances et emplois assimilés								
Disponibilités et assimilés	-	-	-	-	-	-	-	-
Placements et autres actifs financiers courants	2900000	2900000	2900000	3370000	3336000	3480000	3720000	4150000
Trésorerie								
ACTIF COURANT	-	-	-	-	-	-	-	-
TOTAL ACTIF	3900000	3900000	3900000	4370000	4336000	4480000	4720000	5150000
PASSIF								

	REALISATION			PREVISION				
En milliers DZD	N -2	N -1	N	N+1	N+2	N+3	N+4	N+5
CAPITAUX PROPRES								
Capital émis	1000000	1000000	1000000	1000000	1000000	1000000	1000000	1000000
Capital non appelé								
Ecart de réevaluation								
Primes et réserves- Réserves Consolidées								
Résultat net- RN part du groupe	90000	900000	900000	370000	336000	480000	720000	150000
Autres capitaux propores- report à nouveau								
Part de la société consolidante (1)								
CAPITAUX PROPRES	3900000	3900000	3900000	43370000	4336000	4480000	4720000	5150000
PASSIFS NON-COURANTS								
Emprunts et dettes financières								
impôt différé passif								
Autres dettes non courantes								
Provisions et produits constatés d'avance								
PASSIFS NON-COURANTS	-	-	-	-	-	-	-	-
PASSIFS COURNATS								
Fournisseurs et comptes rattachés								
Impôts								
Autres dettes								
Trésorerie passif								
PASSIFS COURANTS	-	-	-	-	-	-	-	-

TOTAL PASSIF	350000	350000	390000	4370000	431000	4480000	4730000	5150000
Verification de l'équilibre Actif/Passif	-	-	-	-	-	-	-	-

الملحق رقم 02: جدول حسابات النتائج المتوقعة

COMPTE DE RUSULTAT PREVISIONNELDE STARTUP :

.....

	REALISATION			PREVISION				
En Milliers DZD	N -2	N -1	N	N+1	N+2	N+3	N+4	N+5
Vente et produits annexes	50000000	50000000	50000000	5500000	5500000	5700000	6000000	6500000
Variation des stocks produits finis et en cours								
Production immobilisée								
Subvention d'exploitation								

Production de l'exercice	5000000	50000000	500000000	55000000	55000000	57000000	60000000	65000000
Achats consommés								
Services Extèrieurs et autres consommations	200000	200000	200000	220000	2500000	300000	350000	400000
Consommation de l'exercice	200000	200000	200000	220000	2500000	300000	350000	400000
Valeur ajoutée d'exploitation	800000	800000	4800000	280000	250000	400000	650000	6100000
Charges de personnel	1800000	1800000	1800000	1800000	1800000	1800000	1800000	1800000
Impôts et taxes et versement assimilés								
Excédent Brut d'Exploitation	000000	000000	3000000	480000	450000	3600000	850000	300000
Autres produits opérationnels								
Autres charges opérationnelles								
Dotations aux amortissements, Provisions	100000	100000	100000	110000	1140000	120000	130000	150000
Reprise sur pertes de valeurs et provisions								
Résultat opérationnel	290000	900000	2900000	370000	336000	480000	720000	150000
Produits Financiers								

Charges financières								
Résultat financier	-	-	-	-	-	-	-	-
Résultat Ordinaire avant impôt	290000	900000	2900000	900000	370000	480000	720000	150000
Impôt exigible sur résultat ordinaire								
Impôt différé (variation) sur résultat ordinaire								
TOTAL DES PRODUITS DES ACTIVITES ORDINAIRES	-	-	-	-	-	-	-	-
TOTAL DES CHARGES DES ACTIVITES ORDINAIRES	-	-	-	-	-	-	-	-
RESULTA NET DES ACTIVITES ORDINAIRES	290000	900000	2900000	900000	370000	480000	720000	150000
Eléments extraordinaire (produits)								
Eléments extraordinaire (charges)								
Résultat extraordinaire	-	-	-	-	-	-	-	-

RESULTAT NET DE L'EXERCICE	**290000**	**900000**	**2900000**	**900000**	**370000**	**480000**	**720000**	**150000**

الملحق رقم 03: حسابات الخزينة

TABLEAUX DE FLUX DE TRESORERIE

STARTUP : ....

المحور السادس : النموذج الاولي التجريبي

1-المراحل الاساسية لي الحصول على نموذج اولي.

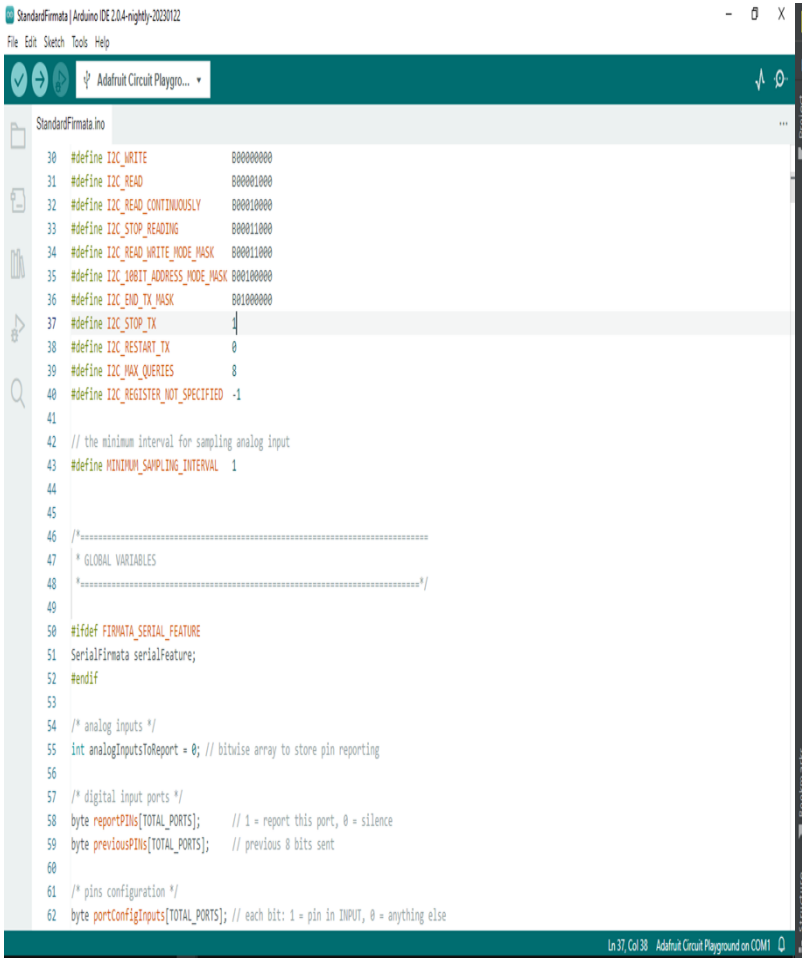
عملية إنتاج مشروع Virtual Drag and Drop باستخدام OpenCV و Arduino تحتاج إلى الخطوات التالية:

- تحضير المكونات: تحتاج إلى تحضير الكمبيوتر المحمول أو الحاسوب الشخصي وكاميرا الويب والأجهزة الإلكترونية التي تدعم Arduino
- برمجة Arduino: يجب تحميل وتثبيت برنامج Arduino على الحاسوب الشخصي وإنشاء برنامج لتحريك الروبوت باستخدام الحركة المستقيمة والتوجيه.
- تثبيت OpenCV: يجب تثبيت OpenCV على الحاسوب الشخصي وتحميل البيانات التدريبية لتعليمه التعرف على الأشياء المختلفة.
- برمجة التعرف على الأشياء: يجب إنشاء برنامج يستخدم OpenCV للتعرف على الأشياء وتحديد موقعها على الشاشة.
- ربط Arduino مع الحاسوب الشخصي: يجب ربط Arduino مع الحاسوب الشخصي باستخدام الكابل الخاص به وتنزيل برنامج التحكم في Arduino. 6-
- تجميع الأجزاء: يجب تجميع الروبوت ومكونات Arduino والكاميرا والمحركات الخاصة بها لإنشاء نظام التحكم الخاص بالروبوت.
- اختبار النظام: يجب اختبار النظام للتأكد من تشغيل كل المكونات بشكل سليم وعملها بشكل متوازن وفعال.
- التحسين والتعديل: يجب التحسين والتعديل على النظام لضمان الحصول على الأداء المثلى وتحقيق المزيد من الوظائف والمميزات.

* بعد الانتهاء من هذه الخطوات، ستحصل على نظام Virtual Drag and Drop الذي يمكن استخدامه في عمليات السحب والإسقاط باستخدام OpenCV و Arduino.

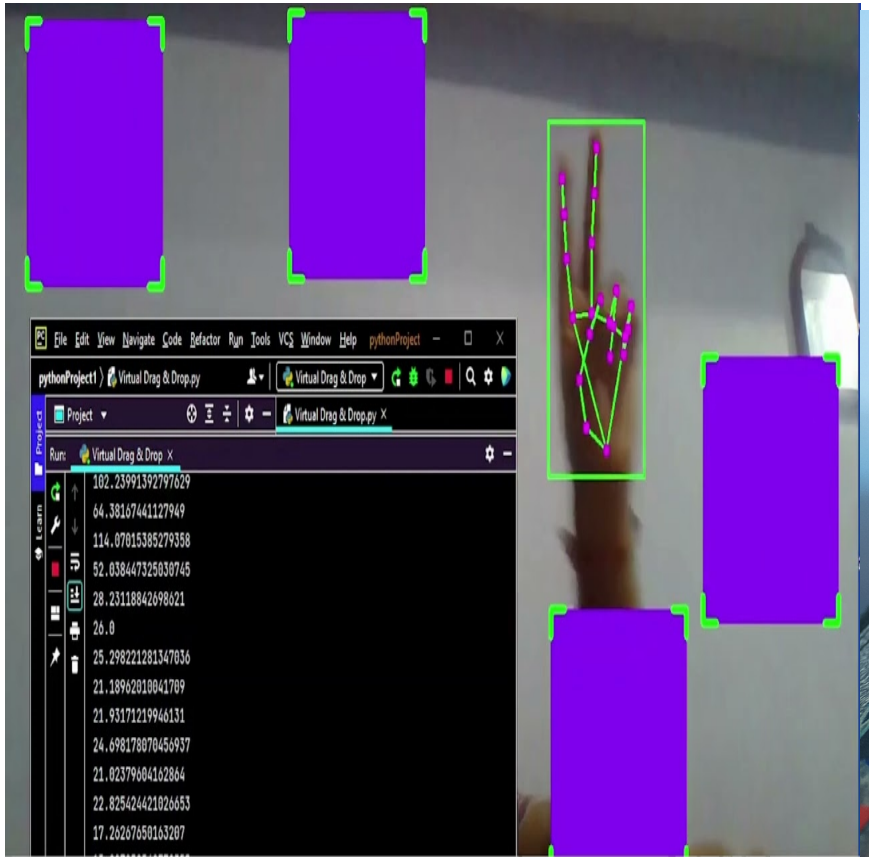
2

1


StandardFirmata | Arduino IDE 2.0.4-nightly-20230122
File Edit Sketch Tools Help
Adafruit Circuit Playgro...
StandardFirmata.ino
#define I2C_WRITE B00000000
#define I2C_READ B00001000
#define I2C_READ_CONTINUOUSLY B00010000
#define I2C_STOP_READING B00011000
#define I2C_READ_WRITE_MODE_MASK B00011000
#define I2C_10BIT_ADDRESS_MODE_MASK B00100000
#define I2C_END_TX_MASK B01000000
#define I2C_STOP_TX 1
#define I2C_RESTART_TX 0
#define I2C_MAX_QUERIES 8
#define I2C_REGISTER_NOT_SPECIFIED -1
// the minimum interval for sampling analog input
#define MINIMUM_SAMPLING_INTERVAL 1
/*==
* GLOBAL VARIABLES
==/
#ifdef FIRMATA_SERIAL_FEATURE
SerialFirmata serialFeature;
#endif
/* analog inputs */
int analogInputsToReport = 0; // bitwise array to store pin reporting
/* digital input ports */
byte reportPINs[TOTAL_PORTS]; // 1 = report this port, 0 = silence
byte previousPINs[TOTAL_PORTS]; // previous 8 bits sent
/* pins configuration */
byte portConfigInputs[TOTAL_PORTS]; // each bit: 1 = pin in INPUT, 0 = anything else
Ln 37, Col 38 Adafruit Circuit Playground on COM1
File Edit View Navigate Code Refactor Run Tools VCS Window Help
projectPython - test.py - Administrator
projectPython test.py
draganddrop.py new.py test.py drag.py
import cv2
from cvzone.HandTrackingModule import HandDetector
import numpy as np
import pyfirmata
cap = cv2.VideoCapture(0)
cap.set(3, 1280)
cap.set(4, 720)
if not cap.isOpened():
print("Camera couldn't access")
exit()
detector = HandDetector(detectionCon=0.8)
#port = "COM3"
#board = pyfirmata.Arduino(port)
#servo_pinX = board.get_pin('d:9:s') #pin 5 Arduino
#servo_pinY = board.get_pin('d:10:s') #pin 6 Arduino
x, y = 150, 230
w, h = 200, 200
col = (255, 0, 255)
while cap.isOpened():
success, img = cap.read()
img = detector.findHands(img)

0

4


File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject
pythonProject1 Virtual Drag & Drop.py
Virtual Drag & Drop
Project
Virtual Drag & Drop.py
Run: Virtual Drag & Drop
102.23991392797629
64.38167441127949
114.07015385279358
52.038447325030745
28.23118842698621
26.0
25.298221281347036
21.18962810041709
21.93171219946131
24.698178070456937
21.023796041628644
22.825424421026653
17.26267658163287

3


WOLF

2.نموذج العمل التجاري BMC

رقم المشروع: ....40.....

الشركاء الأساسيون

- المطورون البرمجيات المتخصصون في OpenCV.
- مصنعون وموردون Arduino.

المهام الأساسية

- تصميم وتطوير البرمجيات التي تسمح بتعقب الحركة باستخدام OpenCV.
- تصميم وتصنيع المكونات الأساسية للمشروع باستخدام Arduino.
- تجميع واختبار المكونات البرمجية والأجهزة الخاصة بالمشروع.

المصادر الأساسية

- الموظفون المؤهلون في البرمجة والهندسة.
- المعدات اللازمة لتطوير البرمج

القيمة الأساسية

- جهاز Virtual Drag and Drop القابل للتحريك والذي يمكن التحكم به بواسطة الحركة.
- البرمجيات التي تدعم عملية التعقب باستخدام OpenCV.

العلاقة مع العملاء

- الاستجابة لاحتياجات العملاء وتوفير تحديثات ودعم مستمر للمنتج.

القنوات

- الموقع الإلكتروني الخاص بالشركة والذي يوفر المنتجات والدعم.
- المشاركة في المعارض والفعاليات التقنية.
- الاشهار في مواقع التواصل الاجتماعي

الشرائح المستهدفة

- تطبيقات تعليمية: لتعزيز التفاعل بين المستخدم والمحتوى التعليمي
- المنازل الذكية
- المصممين الجرافيك: لإنشاء واجهات مستخدم تفاعلية تسمح للمستخدمين بسحب العناصر وترتيبها على الشاشة،
- مطوري تطبيقات ترفيهية
- للتحكم بالروبوتات
- الأطباء: في تطبيقات الطب والصحة لتمكين المستخدمين من التفاعل مع البيانات الطبية أو تحريك العناصر في الواجهة الطبي

هيكلة التكاليف

- تكلفة تصميم وتطوير البرمجيات والأجهزة الخاصة بالمشروع.
- تكلفة الإنتاج والتصنيع.
- تكلفة التسويق والدعاية.

مصادر الدخل

- بيع الأجهزة الخاصة بالمشروع والبرمجيات المصاحبة.
- عقود الصيانة والدعم للعملاء.
- الاشهارات