Allegheny College
Department of Computer Science

Senior Thesis

Robotic Object Manipulation Assistant Guided with Computer Vision

by

Jordan Byrne

Project Supervisor: **Dr. Bonham-Carter**
Co-Supervisor: **Dr. Jumadinova**

September 28, 2022

Abstract

The use of robotics has revolutionized our world, but by integrating other technologies into a robotic agent it may improve its function. The combination of technologies includes computer vision (CV) and robotics would give the device awareness of the environment help guide how it can best interact with the objects in its surroundings. The robotic arm design that was utilized for the completion of this project was the Inmoov robotic arm[15]. The development of a robotic agent using specialized hardware is the most important aspects of this project. The second most vital step for this project is the computer vision (CV) software to enable it to understand what is in its environment. Computer vision (CV) and robotics have a wide range of applications which include agriculture, medicine, and others. The robotic arm design that was utilized for this research was Inmoov[15]. Three different 3D printer were used to output the model files[33][2]. The software that was chosen to handle the data processing tasks were created in Python and include the use of tools such as Open CV, Open AI, Pyfirmata and many others[29][21][20][24][6][17][10][9][11]. The main steps that need to be accomplished to recreate this research are the 3D printing of the chosen robotic design, develop or utilize software for the device while the parts are being created. Future areas of work include the improvement of the 3D printing process, improvements in the design of the robotic arm, software changes, and general hardware improvements. Research in the field of robotics has a bright future and has a lot of needed work to improve the technology so that it can improve human lives.

Acknowledgment

I would like to thank the following people for their help while completing this research.

- Dr. Bonham-Carter, for his help throughout the idea creation process and research work.
- Dr. Jumadinova, for her help with working on the thesis and feedback on writing and hardware progress.
- Jack Honeysett, for his help and tutorials in the ALIC lab and feedback on hardware assembly and 3D Printing.

Abbreviations

(CV) - Computer Vision

Glossary

I dedicate this research to the people that have supported me throughout the entire progress. My parents Suzanne Rogan and Jim Byrne supported me through this entire process and helped me with how I should convey this research.

Contents

List of Figures

List of Tables

Template Overview

You should first modify the documents in the preamble, things that appear before the main text as detailed below.

Front page: use the one provided in this template, after changing the values like names in the file `preamble/mydefinitions.tex`.

Abstract: There should be a single paragraph of about 250 words, which concisely summarizes the entire proposal, written in the file `preamble/abstract.tex`.

Acknowledgments, Abbreviations, Glossary, Dedication preamble pages are optional and can be used at the author's discretion.

The main text of the proposal should be stored in the "SeniorThesis.tex" document. The following descriptions are sections that must be included in the thesis document.

Bibliography: The bibliography should include all references cited in the text (as [**?**]) and it should not include references that have not been cited. ACM referencing style should be used when preparing the bibliography. We recommend using BibTeX or BibLaTeX and using the file `preamble/bibliography.bib`.

Chapter 1

Introduction

1.1 Motivation

Robots are able to perform and repeat exhaustive tasks for humans[27]. They are able to pick up objects guided with their sensors which inform them on the status of the environment[27] One potential application of robotics is to delegate robotic assistive devices to help people who struggle with performing everyday tasks. Robotics have advanced from ephemeral appearances in popular media to complicated and powerful machines which enable better and more accurate prosthetic devices, assist surgeons with surgery, and allow us to explore the universe around us without being physically present[4] [13] [18]. These advancements have ushered in leaps and bounds in improving outcomes of surgeries, scientific experiments and discoveries, and enhancing quality of life for people with disabilities[39] [18]. Combining a technology such as computer vision and robotics it could enable it to perform as an assistive robot for people with disabilities.

1.2 Background

Many innovations that have been made with robotics paired with multiple other technologies. Myo-electric upper limb prosthesis are able to work because of the combination of robotics, biology, and medicine[4]. The prosthetic device's creation involved the pairing of a robotic prosthesis equipped with electrical receptors that connect to the ends of nerves where the prosthesis connects to the person[4]. These electrical receptors measure the local electrical activity and then send the signals to the computer in the prosthesis which analyzes the input and then produces arm and hand movements that were made by the person in the brain and sent through the peripheral nervous system[4]. Another combination of robotics and medicine allowed for the development of tele-surgery devices that enable surgeon's to perform operations remotely[13]. Robotic surgical devices such as the da Vinci system were paired with cameras and controls that allowed for precise movements during the surgery to be made[13]. The combination of robotics, cross planetary telecommunications, and artificial intelligence powers rovers to drive on the surface of Mars and the Moon. [18]. They have enabled us to gain insights into what it's like on the surface of other planets[18]. These rovers have utilized an artificial intelligence (AI) algorithm called deep learning to calculate the best path it can drive to minimize damage to the wheels and overall structure[18]. By combining several technologies with robotics it could have a profound impact on the quality of people's lives.

1.3 Goals of the Project

There are several goals for this research. First, the development of a robotic agent using specialized hardware is the most important aspects of this project. The second most vital step for this project is the computer vision (CV) software to enable it to understand what is in its environment. This project also intends to carry out software and hardware testing to evaluate the effectiveness of the developed robot in carrying out specific tasks and it's object detection rate. From a macro perspective this project aims to investigate whether a low cost robot made with 3D printed parts, equipped with computer vision (CV) to assist people with specific tasks.

1.4 Conclusion

A potentially promising combination with robotics is computer vision (CV), sentiment analysis, voice recognition, and speech synthesis that could help people perform everyday tasks that are difficult to perform on their own. This proposed robot could gain semi-awareness by utilizing a variety of sensors and a deep learning algorithm that specializes in object detection. A type of deep learning algorithm that has been used extensively for object recognition is called computer vision[28]. With this type of algorithm it's able to detect different types of objects that are in a photo. For instance it could detect a sports ball, a plane, a fork, or any object that it has been trained to recognize[28]. Since robots are able to automate repetitive tasks this would make them especially adapted to perform a series of tasks that people need to perform every day that could otherwise be cumbersome. This project could have an appropriate application for assisting people with disabilities with vital tasks such as eating or bringing objects closer to them so that they could be used more easily. Since physical tasks would be demanding therefore the way this proposed machine should be controlled is by voice commands. By utilizing these technologies to create a robot that's able to look at the world and to some degree interpret what it's seeing, it would make it well equipped to assist humans.

Chapter 2

Related Work

2.1 Introduction

Previous research that includes computer vision guided robotic agents covers feasibility research, the strengths and trade offs of differing robotic hand devices, applications for therapeutics, and probabilistic methods of exploring an environment via a robotic hand. This section aims to describe robotic hands that have previously been constructed and how combining technologies from different areas can resolve currently existing limitations associated with robotic hands. Each of these preceding sections summarize existing data and information in order to illustrate previous successes and some of the most effective methods to implement this type of robot. The following findings aim to inform the direction of this project, give background on slightly different but related fields, and give background information for the decisions that will have been made during this project.

2.2 Robotics

. .

The first recorded appearance of robotic agents appeared in early to mid 20th century media such as the Jetsons and Issac Asimov's novel[13]. For practical use in the real world robotics have traditionally been used in the manufacturing process for different types of goods and machines[13]. For instance robotics assemble automobiles and for food packaging[13]. Additionally robotics have had a growing presence in the healthcare field for performing and assisting specific types of surgeries[13]. According to an article titled "A History Of Robotcs: From Science Fiction To Surgical Robots" the first use of robotics to perform and assist surgeries was in 1985 to perform a "brain biopsy"[13]. Since then robotic aided or performed surgeries that have been used since then include hip replacements, knee replacement surgeries where there's an implantation of a prosthesis[13]. A different type of robotic surgery that has been utilized is known as "Telerobotics" which is controlling a surgical robotic from a remote location by relying on the integrated sensors to perform the operation[13]. Medical researchers Joe Rosen and Scott Fisher pioneered tele-operation devices[13]. Currently the "daVinci" robotic surgery device is FDA approved for several different types of surgeries that include cardiac and general surgeries[13]. According to the authors of the history of robotics study reviewed clinical data from the "daVinci" surgery device saw "improved post-operative function, decreased blood loss, shorter hospital stays, and a favorable learning curve for newly trained robotic surgeons."[13]. Such findings demonstrate the strengths that robotic systems can have in the healthcare setting and the promise that new devices could produce further improvements for patients in other settings.

Currently there's an experimental method to obtain information on limb movements that aren't based on the technology inside of myo-electric prosthetics. Instead they rely on brain monitoring devices to deduce the intended movements of a specific prosthetic limb[35]. This bridge between a human and a computer from a brain monitoring device is called a human-computer interface[35]. There are two techniques for human computer interfaces, invasive and non invasive[35]. Invasive techniques aren't as popular as non-invasive techniques because of the risks associated with directly interfacing with the brain[35]. Non-invasive techniques of gathering data that have been researched mainly use a device called an electroencephalography (EEG)[35]. An (EEG) is a device that's used to read the electrical charge of specific parts of the brain, which correlate with brain activation of specific areas of the brain[35]. Additional noninvasive techniques include MEG, fMRI, and fNIRS[35]. These techniques scan the brain using magnetic technology to detect changes in blood flow and thus brain activation[35]. [35]. The required tools for developing a human-computer interface to control bionic limbs include a brain monitoring device and an artificial intelligence (AI) algorithm that can detect hidden patterns in data and improve data classification[35]. While experimentally human-computer interfaces could be a very effective way to develop more accurate prosthetics, there's currently some limitations that exist.

There are existing challenges associated with the use of current human-computer interfaces. The first issue has to do with the portability of the brain monitoring devices[35]. Many brain monitoring devices are large and encompass the entire scalp and necessitate the need for many wires and cables. There are some smaller devices like portable (EEG) that yield less accurate data while ear (EEG) devices could be more accurate devices, but are still in early development[35]. Online based (EEG) systems that use an (IOT) framework where a device sends data that has been recorded to a server for analysis return less reliable data when compared to local data analysis[36]. On average it's around 10 percent less accurate than local (EEG) systems[36]. These outlined challenges present some difficulties for developing a human-computer interface to control bionic limbs.

In conclusion there are opportunities to pair brain monitoring devices with artificial intelligence algorithms that can improve data interpretation to create a human-computer interface to control bionic limbs[35]. Tools such as an (EEG) and artificial intelligence algorithms can be used to gather data from the outside of the skull in a non invasive manner and thus interpret the results based on what the algorithm was developed to interpret or diagnose[35]. There are a multitude of applications of a human-computer interface such as bionic limbs, internet of things devices, seizure detection, among many others[35]. There are existing challenges that exist which are associated with the development of a human-computer interface that include the portability of brain monitoring hardware and the effectiveness when it comes to online (EEG) systems of detection[35]. [36]. A great opportunity exists in developing an artificial intelligence algorithms to more accurately detect changes in the brain to enable the use of an accurate bionic limbs that's controlled via a human-computer interface.

2.3 Computer Vision

.

Shifting to computer vision algorithms, there are a variety of applications that this technology has been used for. One such use of specialized computer vision (CV) algorithms is the self-driving car[12]. The concept of a self-driving car is based on a combination of data from several different sensors like cameras that help the vehicle make sense of its surroundings[12]. A Computer vision algorithm in this context would have to be able to detect different types of objects in the environment such as other cars, stop signs, green lights, and many other objects[37]. The use of computer vision (CV) algorithms brings up ethical concerns[1]. Additionally computer vision (CV) algorithms have been used in the healthcare field to analyze x-ray scans, bio-metric recognition software, and apple growth detection[37][7][32].

Computer vision's uses for self-driving cars allow for a machine to drive for people but also raises questions about the ethics of the decisions that it makes in dangerous scenarios[37].[5]. These ethical dilemmas arise when the algorithm has to make a decision that could lead to a person's injury or death[5]. Questions like should the algorithm: prioritize the riders inside of the vehicle, prioritize pedestrians, young people, old people appear in our minds and we are immediately uncomfortable with the consequences of a decision that an algorithm has made[5]. A widely reviewed paper Titled "The Moral Machine Experiment" explored these ethical questions and gathered data on the types of decisions that people make based on a specific set of criteria in different situations and how it illustrates how our own biases can effect how we expect machines to behave[5]. Computer vision algorithms have applications for self-driving cars but bring up ethical questions about the algorithms that make decisions that can permanently affect someone's life[5][37]. CV algorithms are also used in the healthcare field to assist in the diagnosis or detection of diseases from medical scans[8]. When one of these detection tools are used they are paired with a deep learning algorithm that has been trained on thousands of scans so that it can better detect an illness[8]. A strength of these tools is that the algorithm is able to use knowledge of many more scans than what a human doctor remembers[8]. This means that the algorithm could potentially pick up on abnormal presentations of diseases that a doctor could miss because they couldn't remember one type of scan variation[8]. The combination of deep learning and computer vision has become popular for uses in radiology[8]. Another example of computer vision and deep learning applications is facial recognition[7]. Facial recognition has been used for a variety of applications such as law enforcement, identity verification, and criminal investigations[7]. A major drawback of these facial recognition systems is algorithmic bias[1]. Research into this bias found that there are differences in the detection rate between race and gender[1]. Specifically, the results of one study called "Uncovering and Mitigating Algorithmic Bias Through Learned Latent Structure" found that women were correctly detected at a lower rate than men and

.[htp]

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

Figure 2.1: Yolo-V3 Performance against other computer vision (CV) algorithms, statistics . Obtained From[28].

.[htp]

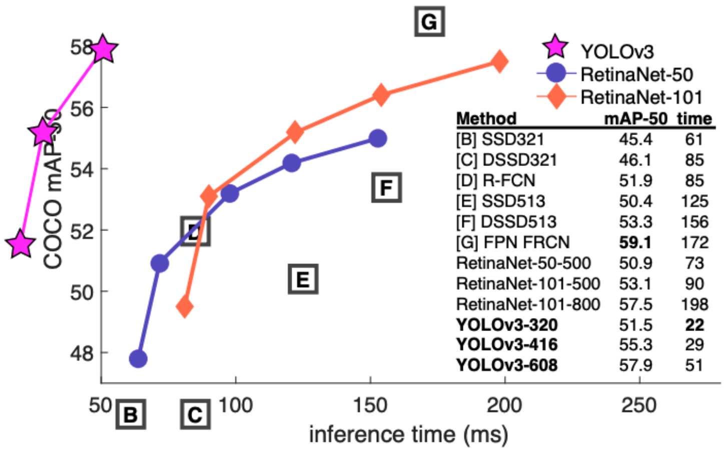


Figure 2.2: Yolo-V3 Performance against other computer vision (CV) algorithms. Obtained From[28].

people with a darker complexion were detected at a lower rate than those with a fair complexion[1]. Computer vision and deep learning uses in healthcare and law enforcement have shown some promising results but concerns of algorithmic bias exist that undermine the effectiveness of facial recognition algorithms that are currently used[1]. Moving on, computer vision (CV) algorithms have been investigated for uses in agriculture to determine the growth stage of apples in the setting of an apple orchard[32]. Specifically a study called "Apple detection during different growth stages in orchards using the improved YOLO-V3 model" used CV to determine when apples meet specific growth stages[32]. The researchers found that the YOLOV3-dense algorithm was the most accurate model to use for this problem[32][5].

Additionally computer vision has also been used in space rovers[18]. A relatively recent paper titled "Computer vision on Mars" concluded that CV is essential to ensuring space rovers can become more autonomous and become better at identifying objects of scientific interest[18]. Beyond planetary navigation, CV has been used to help researchers analyze astronomical data to better detect celestial bodies that have been picked up by telescopes[16]. Computer vision (CV) has board uses in healthcare, automobiles, law enforcement, identity verification, and agriculture[8][1][7][32]. Computer vision (CV) in each of these industries is often paired with an additional form of artificial intelligence such as deep learning so that those tools can gain insight and make connections between data[8]. Computer vision has potential ethical concerns that could negatively permanently impact a person's life[5].

Research that has examined methods and aspects of reinforcement learning outside of the setting of the laboratory has discovered that specific methodologies of reinforcement learning yield higher accuracy rates[38]. One paper that examined this paradigm is titled "The Ingredients Of Real-World Robotic Reinforcement Learning"[38]. The reason why this investigation is important is because it outlines how researchers and hardware and software developers could make their machines work better outside of optimal environments such as a laboratory and function in many different scenarios[38]. The way in which the researchers were able to allow for robotic devices to address some challenges with functioning in many different scenarios by learning from data from sensors, assign a reward based on trials, and "learn continuously in non-episodic settings" that don't require the supervision of a person[38]. The results that were found was that with a specific bead task determined with the more time that was spent training the deep learning algorithm the robotic hand became more successful at the task[38]. This research could apply to this idea because it states the steps required to ensure that the robotic device could perform its task outside of the laboratory physical setting.

CV algorithms have many applications that are applicable to many industries like healthcare, law enforcement, agriculture, and space exploration[18]. This is especially valid when paired with other forms of artificial intelligence such as deep learning which allow for computers to make connections between data to come to a conclusion about an input that has been submitted to the algorithm[8]. CV paired with other technologies could improve and save many people's lives through safer driving and more efficient disease detections[8][37]. There exists concerns about CV biases and concerns where based on a person's demographics it's less effective at detecting them[1]. Based on the findings of the research discussed previously, it is vital that any implementation of computer vision should equally value both the effectiveness of and the potential for algorithmic bias, and concerns for user privacy.

2.4 Robotics And Computer Vision

The research that has thus been published for combining these two technologies computer vision (CV) and robotics with the application for therapeutics has covered aspects of real life learning, vision based operation, the most optimal robotic hands that adhere to the shape and function of the human hand, future developments for this area and field, and a robotic benchmark for learning[27][14][38]. The importance that this area has for this project is that it gives an overview of how previous implementations have been completed and how to approach and solve specific areas when combining a camera and robotics for grasping and bringing objects.

In the research that have combined computer vision and robotics they have found that it's an effective way of teaching robotic agents how to complete specific tasks through observation and that this was able to be accomplished without the use of a specialized glove that measured the changes in movement in the hand in order to submit the movement changes to the robotic hand[27]. Specifically in one study titled "Vision-Based Multi-Task Manipulation for Inexpensive Robots Using End-To-End Learning from Demonstration" the researchers investigated whether it's possible using neural networks to interpret raw images to extrapolate the movement a person is making in order to teach a robotic hand and arm to complete the same movement[27]. The reason why this type of work is important is due to the need for robots to complete specific and complicated tasks that require them to manipulate the environment around them. Specifically this paper outlines that "activities of daily living" can be a "major challenge" for someone who is disabled or elderly and that robots could help them complete such tasks via learning from them[27]. The way in which this task was accomplished was via recurrent neural networks that were trained to interpret the series of images that contained the instructions for the task to complete[27]. Then this recurrent neural network gives specific instructions to the robotic hand and arm for the task that needs to be completed and the machine executes the task[27]. The results of this experiment found that it was able to achieve at least 76 percent success rate for each of the five tasks mentioned in this research paper[27]. The reason why this type of research applies to my senior thesis is because it could be very helpful to give background information in the related literature section of the paper on how to help teach a robotic arm and hand to complete a series of specific tasks. Additionally it could help illustrate how the robotic hand and arm are able to complete a specific task that I would like it to accomplish.

With any type of implementation using artificial intelligence algorithms a robotic hand could produce encouraging results for a specific set of tasks but it would be much more valuable to be able to rate the effectiveness of not just one robotic limb but every one using the same set of criteria[14]. The way in which researchers are able to do this is through a robotic hand benchmark examination[14]. The way in which this typically works is that there's a series of different tasks that vary on the type of movement and coordination that's required and assess the performance through several different types of criteria[14]. There are many types of benchmarks that are popular[14]. One example of a robotic arm benchmark is called RLBench. A research paper titled "RLBench: The Robot Learning Benchmark Learning Environment" introduced a series of benchmarks for the type of tasks that a robotic hand can complete[14]. The purpose of establishing this type of benchmark is to "accelerate progress" for vision guided tasks for robots and includes one hundred hand tasks[14]. Therefore such a benchmark could help determine the effectiveness of a created robotic system. The researchers were able to do this through using an already existing tool called PyRep which "is a toolkit for robot learning research"[14]. The results of this paper found that it can be used to provide information for a robotic agent to learn by imitation[14]. The reason why this is tied to this research is due to the ability that it affords for generating inputs for the system to learn how to perform precise and effective movements.

Robotic agents have been implemented with artificial intelligence tools to improve their performance on tasks. A form of computer vision has been used to allow robots to learn by example from people for a specific set of tasks[27]. One drawback of the current research that exists for computer vision based reinforcement learning is that it doesn't work as well in the real world when compared to the performance in a laboratory with a specific setup[38][27]. The performance of the tasks that robots complete can be judged on a variety of assessments that determine its overall and very specific strength and weaknesses[14]. The combination of robots and computer vision allows for the device to be somewhat aware of it's environment and allows it to learn how to perform specific tasks just by observation[27].

2.5 Conclusion

Overall many technologies that have been implicated in the use of a therapeutic device or assistant device consist of cutting edge research with brain monitoring devices and myo-electirc prosthetics[35]. Computer vision (CV) algorithms are able to perform a variety of detection tasks such as performing self driving tasks, detecting disease from a medical scan, and performing facial recognition[37] [8] [7]. This is especially true when CV is paired with deep learning[8]. There are ethical concerns with algorithmic bias that have been found in experiments[1]. The related literature has indicated that combining robotics and computer vision (CV) technologies enables robotic agents to perform a variety of tasks and to be aware of their environment[14].[27]. This technological combination also allows for real time reinforcement learning from watching a person perform a task and extrapolate those movements for the machine[27]. This also means that every task doesn't need to be built in, instead a robot could be asked to learn a new task and watch a person perform it several times in order to learn how to perform it[27]. Overall both computer vision and robotics have a wide range of applications to explore and experiment their current effectiveness in, especially for uses with people.

Chapter 3

Method of Approach

3.1 Introduction

This research was accomplished by developing software to perform computer vision (CV), voice command analysis, and save the detected objects in computer memory. The robotic arm design that was utilized for this research was Inmoov[15]. Three different 3D printer were used to output the model files[33][2]. A 3D printer prints three dimensionally and allows for a variety of objects to be constructed from the model's file[33][2]. This project was completed by utilizing hardware and software tools to create a robotic arm.

3.2 Software And Algorithms

The implemented software in this research was done inside of the Python programming language[29]. It was chosen for its documentation, compatibility, and modern language features. Other software libraries used include Open AI, Open CV, Coco, and Google Cloud[10].[9].[11].[6].[20].[21]. The computer vision algorithm (CV) that was used was Coco[6]. It was chosen for it's ability to recognize many common objects in everyday life. These objects include assorted fruits, utensils, bowls, plates, sports balls, and many others[6]. One of the strengths of using this CV algorithm is that it runs locally on the device which reduces the latency between video capture and object detection[28].[6]. Matplotlib was used for it's ability to show detailed data[17]. Google Cloud was picked because it allows for code to interact with Google Cloud servers and send data for analysis[10].[9].[11]. Pyfirmata is a Python library that allows for your code to interact with an Arduino and control it like you would using Arduino's proprietary code editor[24]. Time was used so that the programs can keep track of time and use it for measuring run time latency[29]. The OS library was used in order to access it's useful functions for accessing locations and details in memory[25]. Open AI was utilized to allow the program to use sentiment analysis to understand the voice commands and generate a response based on sample data[20]. An algorithm that was created previously was utilized and adapted to perform the computer vision (CV)[19]. The use of developer environments allows for many developers to be using the same libraries and the version that they're specified out[22]. Basically they enhance and speed up the process of setting up computers for software engineering[22]. This project supports developer environments, and specifically the Poetry Python developer environment manager[22]. Software tools, computer vision (CV) algorithms, and developer libraries were utilized to implement the needed tasks for this device.

Utilized Software Libraries

- Open CV[21]
- Coco[6]
- Matplotlib[17]
- Torch Vision[26]
- Google Cloud[10][9][11]
- Pyfirmata[24]
- Time[29]
- OS[25]
- Open AI[20]
- Streamlit[31]

3.3 Open Source Implementation

When the assembly of the robotic hand has been completed it should look something like this. In the image you can see the variety of materials that were used in the assembly for this robotic hand.

In the following image you can see the developed interface for how the user interacts with the robotic device. The user interface shows the communication status with the micro-controller and when the camera is started up it shows the camera footage live and the detected objects in the scene. Additionally there's a list of detected objects that will show up from the code.

In the displayed code segment bellow shows the code that was written in order to support the user interface functionality of the program. The user interface utilizes a Python code package called Streamlit, which is a popular tool that enables developers to easily create user interfaces and display data in their web applications.[31]

```
class Detected_Entity():
"""Object to store the latest detected object
from the computer vision algorithm."""
def __init__(self, object):
    """Initialize the aspects of the object that stores the
    current detected entities from the CV algorithm."""
```

Figure 3.1: Image of Assembled Inmoov Robotic Hand[2]

```
        self.object = object

    def get_object(self):
        """Get the current object to the function
        that handles the user interface code."""
        return self.object

# Create an object entity.
det = Detected_Entity("")

class Video_Input(VideoTransformerBase):
    """Object to handle the computer
    vision manipulation of the live video feed."""
    def __init__(self):
        """Initialization function for the Video_Input class."""
        self.style = "color"

    def recv(self, frame):
        """Function to modify aspects of
        the video input from the camera."""

        # Convert frame data type into numpy array.
        img = frame.to_ndarray(format="bgr24")

        # Call the get computer vision algorithm from
        # the computer vision module.
        img, detected_object = computer_vision.run_computer_vision(img)

        # Define object data point from the
```

```
        # detected entity class.
        setattr(det, 'object', f'{detected_object}')

        # Return the frame of the video that was
        # manipulated with the CV algorithm.
        return av.VideoFrame.from_ndarray(img, format="bgr24")


def interface_elements():
    """Initialize and dynamically update aspects of
    the application's user interface."""
    # Define the specifications for the user interface.
    with column_1:
        # Section to allow the user to turn on the device.

        # This subsection handles all of the computer vision
        # related information that needs to be updated.
        st.subheader("Computer Vision")

        webrtc_streamer(key="computer_vision_feed",
        video_processor_factory=Video_Input)

        obj = Detected_Entity.get_object(det)
        st.text(f"Detected Object(s): {det.object}")


    with column_2:
        # This subsection shows the communication status of how
        # well the device is communicating with the hardware.
        st.subheader("Communication Status")

        # Initiate the connection status
        between the Arduino and the computer.
        connection_status = arduino_comms.setup_arduino_comms()

        # Update the status of the arduino computer connection #
        based on the results of the connection initialization function.
        if connection_status:
            # Show connection status with the
            Arduino device by checking the access to the port.
            st.text(f"Arduino Connected To Computer.")
        else:
            # Show connection status with the Arduino device
            by checking the access to the port.
            st.text(f"Arduino Not Connected To Computer.")
```

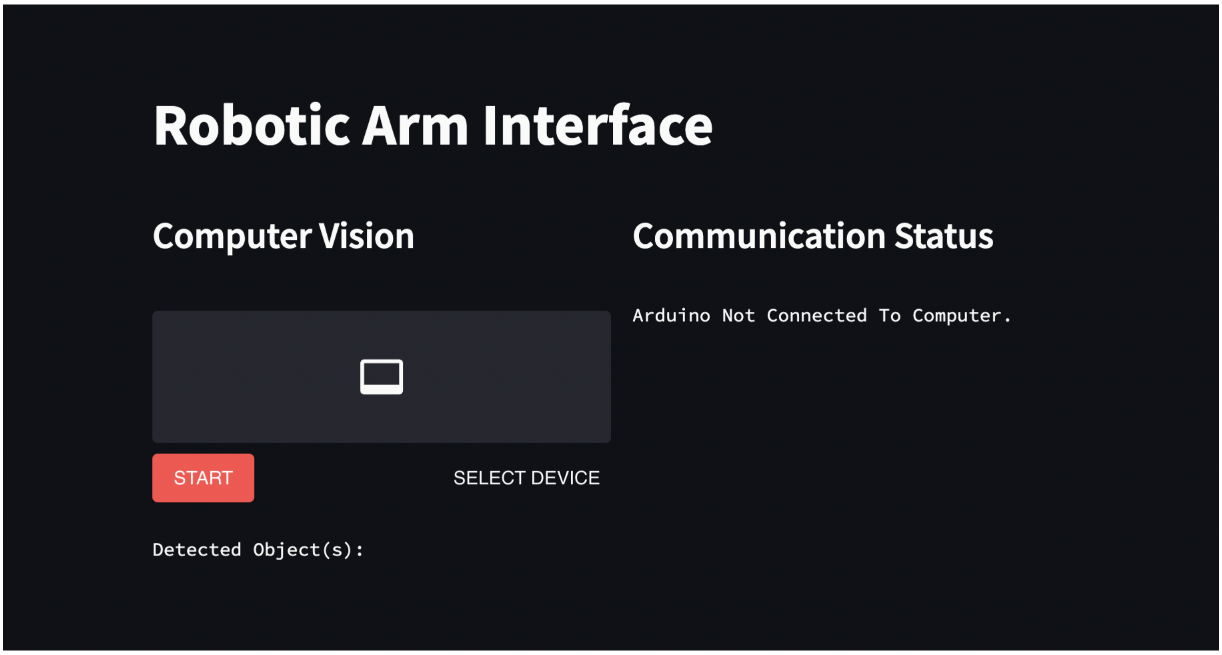


Figure 3.2: Image of Robot User Interface[2]

3.4 3D Printing

During the completion of this project there were two types of 3D printers that were used to print out the open source robotic arm design. They are Ultimaker 3, Anycubic: Photon Mono, and the Anycubic Kobra. X[33][2]. These two differ in how the prints are executed and how quickly they can complete. Throughout the project four types of plastic materials, both solid and liquid, were used to print out the parts of the robotic arm. The reason why this many different types of filament were used is due to inconsistent functioning of the Ultimaker 3 printer that was available. The time to complete all of the prints took about 6 weeks when the maintenance issues are factored into the time of completion. Although this presented an issue it helps elucidate what types of 3D printers could be most optimal for creating a robotic arm. The Anycubic Kobra was used sparingly for the printing out of some of the fingers. The use of multiple printers displays how different tools can yield different results.

3.4.1 Ultimaker 3

This was the first 3D printer that was used for this project. The way in which it works is that the nozzle heats up to approximately 200 degrees celsius, with default settings, and pushes the material through downwards onto the build plate[33]. It works layer by layer to sculpt the object until it has finished[33]. For any overhanging sections of the file it creates support to prevent part of the structure from falling down[33]. The strengths of using this printer are that it can output really high quality builds which are ready for use when the job has finished[33]. And for finding support online there's software and hardware documentation to help anyone out if they're stuck[33].

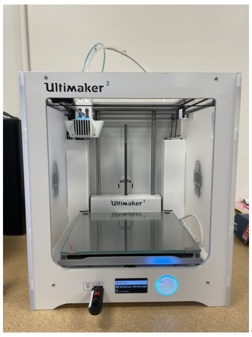


Figure 3.3: Image of the Ultimaker 3, 3D printer[33].

The support structures that are generated with the default structures were consistently great and work well with tough objects[33]. A drawback of using this printer is that it takes a long amount of time for the job to complete. Specifically when you combine multiple 3D models into the same job for a portion of the robotic arm it takes about one day to complete. There were also maintenance issues that were encountered during the completion of this project that caused significant issues. Specifically the issues encountered were inconsistent prints with a hairy texture or parts of a layer missing. The tip of the nozzle was easily clogged after having the nozzle disassembled and reassembled and thoroughly cleaned with hot and cold pulls. Generally the Ultimaker 3 printer produces high quality prints but when maintenance issues appear it can present a significant impairment to the progress of building the robotic arm[33].

3.4.2 Anycubic: Photon Mono X

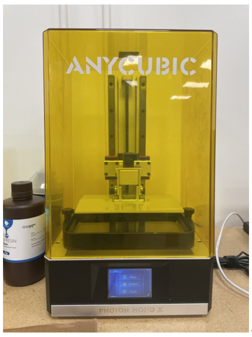


Figure 3.4: Image of the Anycubic: Photon Mono X 3D printer[2].

The Photon Mono X uses liquid resin that's UV sensitive[2]. Each layer is created by a beam of UV light that causes it to harden[2]. This process is repeated while the build plate is lowered to the bottom of the liquid resin[2]. Generally this printer was able to print out large files very quickly when compared to the Ultimaker 3 printer. Equivalent prints that would take twenty four hours took about five hours with this make and model. Quality is also very high which makes it ideal for parts that need the utmost precision like gears[2]. One of the drawbacks of using this type of printer is that finishing the prints and cleaning up can be difficult and time consuming. After each print is finished you have to unscrew the build plate and pull it out and scrape off the parts. Then it has to be washed in rubbing alcohol and then cured under UV lights[2]. The post print routine that was utilized for this project is as follows. The ethyl alcohol bath was done for ten minutes and the UV curing was carried out for thirty minutes[2]. Generally this routine was very good for finishing the prints. If the parts are still sticky and or not as desired, repeat the routine until you have removed the issue[2]. While working with the printer and post print routine wearing gloves and a face mask is necessary to keep you safe[2]. There's a slight risk that if the liquid resin comes into contact with your skin you might develop an allergic reaction[2]. Generally it's advised to wash it off immediately with soap and water[2]. Generally the software and hardware has good documentation but ensuring you have the latest release of the printing software is more difficult than necessary[2]. Overall this printer was effective for quickly creating high quality parts but have some drawbacks that should be considered for future work.

3.4.3 3D Printing Conclusion

3D printers are very effective devices for quickly creating and prototyping parts for any type of project[33].[2]. When working on builds larger than what can be held in the palm of your hand prepare to spend a good amount of time for the jobs to completed. Depending on what model and method of printing, the time to completion may vary by a large amount of time. For working with the Anycubic liquid emulsion printer prepare to spend time cleaning and curing the prints after they're initially finished[2]. Invest in safety equipment like gloves and masks to ensure your personal health while working[2]. When printing out parts for any project consider the benefits and drawbacks of the particular models that are available for use.

3.5 Hardware Integration

The robotic arm design that was utilized for the completion of this project was the Inmoov robotic arm[15]. This open source project was started in 2015 and has many robotic humanoid parts to pick from[15]. The micro controller that was used for the completion of this project was the Arduino Mega[3]. This version was chosen to allow for more data pins to be used to connect the motors and sensors[3]. A variety of hardware components were used to complete this research. In the following list you can see the details of each part. By using all of the parts specified this project can be successfully recreated.

3.6 Hardware Assembly Instructions

The first step in starting this type of project is to print out the needed 3D models for your intended robotic device. When starting out decide on the make and model of the 3D printer based on the benefits and drawbacks that are associated with it's design. From there install the workshop editor with the slicer. When you're selecting the settings for your print file use the ones recommended by the project you're utilizing

Hardware Item.	**Quantity**.
Servo Model: MG996r (Arm) (12)	4
Servo Model: MG996r (Hand) (4)	2
0.8 braided/fishing line	1
Extension Spring Pack (Need 10 .51mm)	1
Cable and Wires Kit	1
Silicone Grease Tube (White)	1
Camera Module	1
Ultrasonic Distance Sensor	1
Arduino Start Up Kit	1
USB A/B Arduino Connector Cable	1
Speaker Module Pack Of 4	1
Bread Board	2
Arduino Mega	1
Motor Driver Pack Of 4	1
Super Glue Bottle	1
Generic Pack of 8 Types Of Bolts	1

Table 3.1: List of hardware utilized in research

when it comes to infill percentage and support structures. If printing out a custom model use the default settings inside the software or use similar settings specified for other robotic parts. Many printers have a settings to create a base or a raft that makes it easier for the parts to be taken off the build plate. Generally choose the one that's easiest to take off. For the Anycubic Photon Mono X the base cannot be easily taken off and would need to be sanded off by either using sand paper or a power tool with a sanding attachment. Best results have been seen by using the slowest setting on the power tool and working slowly and checking the progress that's made for each individual part.

Once all of the parts have been created the assembly process can now take place. If following an open source project follow the instructions very closely to prevent any mishaps. Try to not apply too much pressure while assembling the parts to prevent breakage that would necessitate the printing out of another part. For the Inmoov robotic hand and forearm design these were the steps that were carried out[15].

3.6.1 Inmoov Hand And Forearm Construction

The following instructions detail how the robotic arm was constructed and the tools, techniques, and advice is recommended for this type of research. These steps are meant to be carried out once all of the parts have been 3D printed out and prepared for the construction of the robotic arm [15]. Depending on the settings used for the prints you may be able to skip some of the steps outlined bellow. This is by no means an exhaustive list that's applicable to every design but one way in which it has been completed. This section references the official instructions for constructing the Inmoov robotic arm but deviates in the listed steps of completion [15]. This list seeks to show the main steps in constructing a robotic arm.

3.6.2 Prepare Your Parts

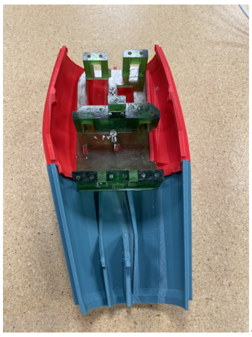
Ensuing that your parts are ready for construction is the most pressing task when starting out. Make sure that you have removed any of the rafting and support structures from your build. If you used the Anycubic, or any similar printer, with the support base you will likely need to individually sand it off manually [2]. To best accomplish this it's recommended to use a sanding power tool on the lowest speed and make gentle movements where the bulky support was generated. By checking all of the parts quality it helps ensure that the robotic arm functions as intended.

3.6.3 Assemble The Main Forearm And Servo Bed

To start out with the construction of the arm take out your upper and lower forearm pieces [15]. With these pieces ensure that you at first super glue one upper and one lower forearm to each other [15]. This will allow you to insert the servo-bed later on that will power the movement of the fingers [15]. Try not to use super glue on the outside as it will take away from the appearance [15]. Super glue inside and between the parts are recommended [15]. After they have been glued together let them sit for at least twenty minutes to allow for the bonding between the parts and to prevent any breakages of the adhesive seal [15]. In this research the servo-bed was slightly sanded down on the left side so that it can better fit into the servo-bed section. This modification of the part shouldn't impact performance in any noticeable way. They should now look like what's depicted in the image bellow.

Figure 3.5: Image of forearm section glued together.

After the parts have had enough time to seal together we can now move on to inserting the servo bed. Find the section that has the flat bed and two screws on the bottom of it [15]. This is where the servo-bed will be placed[15]. The Inmoov instructions ask that the servo-bed be screwed into this section but using adhesion to stick it in place also works well [15]. Now place each motor into the servo-bed. In total there should be enough space for five of them [15]. Make sure you insert them where's there space for the wires to come out of the back[15]. The use of screws or super glue also suffices for the placement of these motors [15].



.

Figure 3.6: Image of servo-bed glued into forearm section.

3.6.4 Build The Fingers, Wrist, And Hand

For building the fingers lay out all of the parts in front of you in an organized manner so that you know what parts correspond to each printer[15]. With the way in which each finger was design it was observed that there are three main types[15]. There's the connective structure that attaches the finger to the hand, the join structure that allows for the fingers to flex inward, and the supportive parts that help the structure of the joints[15]. You can determine that the joint section are the parts that are curved on the top and look like something could rotate from the inside of it[15]. It's also assume that the fingertip part falls under the supportive type of parts[15]. The way in which each finger is constructed is connective, joint, support, support, joint, support (fingertip)[15]. For each two parts they're glued together and each section is connected by the insertion of a 3mm PLA filament through the holes on the sides[15]. This filament ensures that each section of the fingers are attached[15].

Now that the fingers have been constructed you can now work on the hand. Insert each of the cables into the individual fingers that will control the movement of each individual finger[15]. For the wrist take the two outer shell parts of it and attach it so that the smaller one is on top of the larger section[15]. Inside of the smaller section place a motor and super glue or screw it into place so that the servo is peeking slightly outside of the circular hole on top[15]. This is what will allow the wrist to move around[15]. You have the option of two types of gears to use to allow for the movement of the wrist[15]. Whatever option that's chosen don't forget to use the white silicone grease to make sure that it's moving efficiently[15]. Finally place the final wrist part that connects the wrist to the hand and bolt or super glue it together[15]. Next, place the tensioner on the side that faces the wrist connection section of the part[15]. This will allow for the hand and finger movements to be more reactive to the slightest movement of the motors[15]. When this has been achieved move the strings through the wrist and into the forearm and connect them to the servos that have been previously installed[15]. Make sure that all of the cables go through the tensioner so that the movement sensitivities are synchronized[15]. The cabling for this project was directed from the inside of the forearm to the back, away from the hand[15]. From there it's able to be inserted into the breadboard which is connected to the Arduino Mega computer[15].[3]. When everything is connected test out the functionality of each sensor and make sure that you're able to read data from it and that it's as expected. As an example, for the ultrasonic distance sensor we would expect it to return positive integers instead of negative numbers. Once all of the sensors have been confirmed to function correctly then the assembly of the robotic arm has successfully been completed[15]. When this is completed you may super glue the remaining parts together. If all of these instructions have been followed the construction of the robotic arm has gone successfully.

3.6.5 Connect Hardware To Micro Controller

Depending on the extent to which sensors are included in the construction of the device, it will dictate how many breadboards, cables, and micro controllers are needed[3]. For this project having more than three large external sensors can fit on one small bread board, so determine if you have enough for the planned number of sensors. If integrated sensors inside of the fingertips are needed ensure that you have printed out the correct finger models so that you can safely insert the desired sensors[15].

3.6.6 Assembly Wrap Up

At this point you have constructed the robotic arm and have connected the hardware to a miniature computer or a micro controller[15][3]. Before you have completed the construction of the robotic arm have your parts organized into different sections to allow for a faster pace of assembly. The checking of each individual part is important to determine whether there are any potential issues that can come up with the support or raft structure.

3.7 Conclusion

During the research process for this project hardware like the Inmoov open source robotic design, Arduino, motors, and others were chosen to create the robotic arm[15].[3]. The software that was chosen to handle the data processing tasks were created in Python and include the use of tools such as Open CV, Open AI, Pyfirmata and many others[29][21][20][24][6][17][10][9][11]. The use of a developer environment manager, Poetry, was utilized to allow for ease of use when new computers are first running this software[22]. The main steps that need to be accomplished to recreate this research are the 3D printing of the chosen robotic design, develop or utilize software for the device while the parts are being created. When the hardware has been created, assemble it by following the instructions or use the intended procedure for a custom design. If all of these steps were followed and choices were made to best support the desired robotic project then this type of research can be successfully reproduced and hopefully yield better results in a shorter amount of time.

Chapter 4

Experimental Results

4.1 Introduction

To best show the experimental results of this research the results are divided into several different sections. These sections describe the type of testing that was done, how and why the implemented tool was created in the way that it was, and assumptions that could invalidate one or more results that were found. This section aims to show the type of hardware and software testing that would take place if the hardware and software testing if they had been completed. Each area seeks to best represent the results found in order to best assess how well the current implementation functions for the tasks that it was designed to function.

4.2 Experimental Design

The experimental design for this research was conducted on two aspects of the implemented tool. They are the hardware and software. For the software testing the computer vision's capabilities were evaluated to see how well it could identify objects that it was trained to spot inside of a laboratory section. Unfortunately hardware and software testing hadn't been completed due to the issues taking place with the inability to finish the hardware assembly of the robotic arm.

4.3 Evaluation

This is the way in which the system functions.

- Connect Robotic Arm And Hardware To Computer
- Run Program To Start The Robotic Arm
- Follow The Prompts In The Command Terminal And Use The Arm As Directed
- Check The Status Of The Device On The Control Panel Local Host
- Click Stop To Shut Down The Robotic Arm

The hardware and software evaluations were picked in order to assess how well the developed tool functions in specific areas that are implicated in the general use of the device. The system starts when the code is run and can be communicated through voice commands. The device's status can be monitored through the local host web page that opens when the program's finished starting up. These are the ways in which the tool functions and the steps that are required to successfully start it up and use it.

4.4 Threats to Validity

The main threats of validity involve potential problems that can appear in situations in which the robotic arm was not tested in. The environment in which a device performs in can greatly impact how well it performs. For instance certain scenarios can make the sensors unusable temporarily. During the creation and evaluation of this research there were several assumptions that were made that could influence how well it performs in a multitude of areas.

Potentially there are certain factors that could influence how reproducible this research is based on the factors that influence how the robotic arm was constructed. During this research two 3D printers were used and a total of four materials were used[33][2]. Two were PLA and the other two were UV sensitive resin liquid[33][2]. One of the liquid filaments were plant based and the other was synthetic[2]. In addition to the differing types of materials that were utilized different method of removing the bases were done[33][2]. For objects created using the Ultimaker 3 printer the base raft could easily be peeled off[33]. But for the Photon Mono X each part had to be individually sanded using sand paper and a power tool with a metal sanding attachment[2]. Depending on how well the individual parts were sanded and the materials that were used to construct them, it could impact the functioning of the robotic arm.

The focus of testing can also potentially make some of the work difficult to recreate if it's used and or tested in different scenarios. Based on the selection of objects to evaluate the Computer Vision Algorithm it could mean that different objects can perform better or worse for object detection tasks[6][16][18]. Additionally the sensors can be influenced if the environment they're tested in are two loud, dark, or bright[6][16][18].

Depending on the environment in which the robotic device is assessed the results may vary from the results of the previous hardware and software testing that was conducted. The best way to determine the extent of this issue is through future experiments on this sector of research. These issues are the main threats to the reproducibility of this research.

4.5 Conclusion

The situations in which the robotic arm is recreated and or further evaluated can influence how well it performs at specific tasks. This can be influenced by the amount of brightness or noise[10][6][18][16]. The use of different or multiple materials to construct the arm could influence how well it performs[2][33]. The best way in which to determine the way in which these potential issues can affect the performance of the robotic arm is through future research. The experiments that took place during this experimental section include hardware and software testing.

Chapter 5

Discussion and Future Work

5.1 Introduction

Through this research several things were able to be successfully accomplished. This includes the construction of a robotic hand and forearm that's controlled from a micro-controller and is powered by a software tool that allows sensors to gather information about objects in the environment and inform the user on their existence and distance to them[3][15]. During this experiment there are several ethical considerations that should be considered for not only this project but with future work in this area. There are many things that can be changed or improved in like the hardware, software, application user interface, and customizability. This research was able to create a robotic arm, create software to power it, and evaluate how well the implementation performs in specific tasks.

5.2 Summary of Results

If the hardware assembly had been completed hardware and software testing would take place it would measure the effectiveness of the developed machine by determining the object detection rate, and the effectiveness of object gripping. There are some speculated threats to validity such as the environment of the testing could influence how effective the would be completed device would function. Overall the machine would likely be influenced by where it's being operated and how they influence the sensors such as the camera, connected to the robotic arm.

5.3 Ethical Considerations

There are a variety of ethical concerns that are associated with this type of research. They include data privacy, computational bias, and general human well being [1]. Any future work conducted in this area should consider each of these concerns careful to ensure that research is being conducted with the highest moral code.

Additionally there are concerns about data privacy of the information collected from these intensely personal devices[1][13]. For the implementation in this experiment it doesn't collect data that's no longer deleted and any collected data after it's shut off it immediately deleted. There could potentially theoretically be a way the data stream locally or in the cloud could be compromised by a rogue actor but that's highly unlikely due to the secure nature of Google Cloud with it's precise credentials and difficulty of compromising a physical device while it's in use[10]. This issue should be considered and decisions should be made to balance user privacy and the use of data to improve the functioning of the device.

Computational bias is an important issue in computer vision and facial recognition algorithms that are commonly used[1]. In a recent study it was found that facial recognition algorithms vary in their effectiveness in correctly identifying people based on their demographics such as gender and race[1]. In order to make sure that this discriminatory issue can be removed, training data for these algorithms should have more samples of people of every demographic to help ensure fairness[1]. Additionally these detection algorithms should be evaluated for their bias in order to assess if this has addressed the algorithmic bias issue[1].

Human well being should also be considered to ensure nobody can be harmed by these robotic devices. In this research the only movement that the robotic device can perform is a grasping motion. Therefore it could be possible that it could grasp a person and slightly hurt them. Notes that in the implementation of the software there are safeguards to prevent this from happening. To best determine how robotic devices could impact the well being of a person their movements and range of motion should be considered in order to determine what safeguards should be in place and whether to design the moving parts to be weaker to minimize any injury if it happens.

These ethical issues outline what can appear. Each of these issues should be carefully thought through during the design and production of any robotic part in order to prevent any harm. To address some of these issues built in software or hardware safeguards could prevent harm against anyone or data leakage. These ethical concerns outline the different areas in which ethics intersects with the construction of a robotic agent.

5.4 Future Work

There are many things that can be improved in future research. They include the improvement of the 3D printing process, improvements in the design of the robotic arm, software changes, and general hardware improvements. Robotic limbs would also become more complicated and suitable for replacing limbs that were lost and or creating independent robotic actors[23]. These future areas of work are mostly speculative and based on current evidence of where the technology is currently at. If these future improvements in the technology were reached it would make the creation of robotic parts much faster and operate with a higher degree of accuracy.

Arguably of the most complicated and sometimes difficult process of this type of research besides the assembly of the robotic arm is using the 3D printers. The ability that this technology gives us in incredible but is hindered with several persistent issues. Slow print times, restricted material use, and general issues with maintenance and trouble shooting. The issue with cleaning and maintenance could be addressed with an artificial intelligence (AI) assistant integrated into the printer that can automatically diagnose what's going wrong and attempt a fix such as cleaning the nozzle or replacing parts on its own. Improvements in printing times could be made in by using multiple print nozzles in a large 3D printer where the device could collect all the individual parts and heat up the edges on them and solidify them into one part.

The next area for improvement in robotics is general improvements in robotics. One of the issues with using screws and bolts to connect parts together is that you need a tool to disassemble it for maintenance[23]. This could be solved by using a magnetic system on the inside that keeps the parts together[23]. Inside of a phone app that's paired with the device the user could select the option to disable the magnets, with several layers of validation, and allow them to clean out the inside of it or perform maintenance. Additionally robotic arms should be easily able to be modified in terms of their measurements to best fit people who need intelligent prosthetic arms.

Software can be improved by using a voice recognition algorithm locally on the robotic arm so that the device can function without an internet connection. In order to make this device usable for much more people around the world there should be language options to allow voice commands to be given in any other language. Finally having a control panel on the device or on a smart phone could allow the user to better control.

Hardware could be improved by integrating sensors inside of the fingertip and through the miniaturization of the electronics on board[30]. The use of fingertip sensors would allow the hand to determine how much pressure it's exerting while grasping an object. Shrinking the electronics could allow for more internal batteries to be placed and allow the device to operate without being connected to a power source[34].

By using batteries inside of the robotic arm it would allow it to function without being connected to a power source[34]. Additionally the improved power efficiency of all the parts and sensors would improve the theoretical battery life of the device[34]. A future type of battery technology, solid state batteries, could make the device much safer and last for a substantially longer amount of time[34]. The drawback of using this battery currently is that it's really expensive to produce[34].

Improved customizability would make these types of robotic devices more tailored for each individual user and allow them to better control it. Improved control could be gained through the use of EMG or EEG sensors[35][4]. This would allow for the user's thoughts of movements to control the robotic arm[35][4]. Due to the personal nature of these types of assistive tools there should be many ways to change the settings[35]. Here's an example of the types of settings that should be able to be modified. Changes in the language used to communicate with the device and the recognition of local accents. Recognition of voices from particular people to best identify who the user is and only respond to their commands. Modifications of the voice reaction speed and type. The arm movement speed and sensitivity should also be able to be adjusted. The future device could be able to learn and detect the users preferences over time by using a machine learning algorithm.

These areas of future work are great opportunities for improving the quality of the results of this type of research with robotic limbs. It will also likely improve the quality of life for the people who need the support of these types of devices in everyday life tasks[39]. The improvement in 3D printing technology could make prototyping and manufacturing of the parts much faster and improve the rate at which design can be improved[2][33]. These changes would greatly improve the technologies involved in the production of a robotic arm and it's functioning.

5.5 Conclusion

The general outcome of this research was that a robotic arm was produced that can detect objects by using a computer vision algorithm. In this type of research there are several ethical implications such as computational bias, human well being, and data privacy[13][1]. There are many avenues through which the technologies involved in this research can be improved and include the 3D printing process, software and hardware improvements, and improved customizability by the user. The overall attempted accomplishments are that the robotic arm can detect objects and perform a simple grasping motion, the experiments outline the strengths and weaknesses of the implementation. The future work described gives a potential guide map of where future advances can be made. The future is very bright for this type of research and eventually the gap between human and robotic limbs could be insignificant.

Bibliography

[1] Amini, A., Soleimany, A. P., Schwarting, W., Bhatia, S. N., and Rus, D. Uncovering and mitigating algorithmic bias through learned latent structure. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (2019), pp. 289–295.

[2] Anycubic, 2022.

[3] Arduino. Open-source electronics platform, 2021.

[4] Atzori, M., and Müller, H. Control capabilities of myoelectric robotic prostheses by hand amputees: a scientific research and market overview. *Frontiers in systems neuroscience 9* (2015), 162.

[5] Awad, E., Dsouza, S., Kim, R., Schulz, J., Henrich, J., Shariff, A., Bonnefon, J.-F., and Rahwan, I. The moral machine experiment. *Nature 563*, 7729 (2018), 59–64.

[6] Context, C. C. O. I., 2022.

[7] Elhoseny, M., Selim, M. M., and Shankar, K. Optimal deep learning based convolution neural network for digital forensics face sketch synthesis in internet of things (iot). *International Journal of Machine Learning and Cybernetics* (2020), 1–12.

[8] Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., and Socher, R. Deep learning-enabled medical computer vision. *NPJ digital medicine 4*, 1 (2021), 1–9.

[9] Google. Natural language ai, 2021.

[10] Google. Speech to text, 2021.

[11] Google. Text to speech, 2021.

[12] Hee Lee, G., Faundorfer, F., and Pollefeys, M. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 2746–2753.

[13] Hockstein, N. G., Gourin, C., Faust, R., and Terris, D. J. A history of robots: from science fiction to surgical robotics. *Journal of robotic surgery 1*, 2 (2007), 113–118.

[14] James, S., Ma, Z., Arrojo, D. R., and Davison, A. J. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters 5*, 2 (2020), 3019–3026.

[15] Langevin, G. Open source 3d printed life size robot, 2021.

[16] Martens, P., Attrill, G., Davey, A., Engell, A., Farid, S., Grigis, P., Kasper, J., Korreck, K., Saar, S., Savcheva, A., et al. Computer vision for the solar dynamics observatory (sdo). *Solar Physics 275*, 1 (2012), 79–113.

[17] Matplotlib, 2022.

[18] Matthies, L., Maimone, M., Johnson, A., Cheng, Y., Willson, R., Villalpando, C., Goldberg, S., Huertas, A., Stein, A., and Angelova, A. Computer vision on mars. *International Journal of Computer Vision 75*, 1 (2007), 67–92.

[19] Mehedi, S., 2022.

[20] OpenAI, 2022.

[21] OpenCV, 2022.

[22] Poetry, 2022.

[23] Post, M. A., Yan, X.-T., and Letier, P. Modularity for the future in space robotics: A review. *Acta Astronautica 189* (2021), 530–547.

[24] pyFirmata Contributers. pyfirmata, 2021.

[25] Python, 2022.

[26] PyTorch, 2022.

[27] Rahmatizadeh, R., Abolghasemi, P., Bölöni, L., and Levine, S. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *2018 IEEE international conference on robotics and automation (ICRA)* (2018), IEEE, pp. 3758–3765.

[28] Redmon, J., and Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[29] Sanner, M. F., et al. Python: a programming language for software integration and development. *J Mol Graph Model 17*, 1 (1999), 57–61.

[30] Shalf, J. The future of computing beyond moore's law. *Philosophical Transactions of the Royal Society A 378*, 2166 (2020), 20190061.

[31] Streamlit, 2022.

[32] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., and Liang, Z. Apple detection during different growth stages in orchards using the improved yolo-v3 model. *Computers and electronics in agriculture 157* (2019), 417–426.

[33] Ultimaker, 2022.

[34] Xiao, Y., Wang, Y., Bo, S.-H., Kim, J. C., Miara, L. J., and Ceder, G. Understanding interface stability in solid-state batteries. *Nature Reviews Materials 5*, 2 (2020), 105–126.

[35] Zhang, X., Yao, L., Wang, X., Monaghan, J., Mcalpine, D., and Zhang, Y. A survey on deep learning based brain computer interface: Recent advances and new frontiers. *arXiv preprint arXiv:1905.04149* (2019), 66.

[36] Zhang, X., Yao, L., Zhang, S., Kanhere, S., Sheng, M., and Liu, Y. Internet of things meets brain–computer interface: A unified deep learning framework for enabling human-thing cognitive interactivity. *IEEE Internet of Things Journal 6*, 2 (2018), 2084–2092.

[37] Zhao, J., Liang, B., and Chen, Q. The key technology toward the self-driving car. *International Journal of Intelligent Unmanned Systems* (2018).

[38] Zhu, H., Yu, J., Gupta, A., Shah, D., Hartikainen, K., Singh, A., Kumar, V., and Levine, S. The ingredients of real-world robotic reinforcement learning. *arXiv preprint arXiv:2004.12570* (2020).

[39] Zuniga, J., Katsavelis, D., Peck, J., Stollberg, J., Petrykowski, M., Carson, A., and Fernandez, C. Cyborg beast: a low-cost 3d-printed prosthetic hand for children with upper-limb differences. *BMC research notes 8*, 1 (2015), 1–9.