ECE 445

Spring 2024

Senior Design Design Document

Automated Multi-Mode Garment Folding System with Raspberry Pi Control

Team 48
Tyler Hirsch
Bryson Maedge
Nolan Opalski
May 1st 2024
TA: Angquan Yu

Table Of Contents

Link to Github Notebook: Github Notebook

Abstract

The goal of this project was to create a machine that can assist less than able persons in the common task of folding clothes. The prototype uses AI to identify articles of clothing, which then sends instructions to servo motors that fold and pile the garments placed onto it.

1. Introduction

1.1. Problem

No one likes to fold laundry. It's dull, boring, and tedious. The only positive is that it gives you an excuse to listen to your favorite podcast that you may have had a backlog on. On top of that, folding laundry can prove to be a difficult task for the elderly and disabled. In commercial settings, employees of large retail clothing stores have reported getting carpal tunnel syndrome from the repetitive and manual task of folding clothes. Referencing the ADA, there are a long list of reasons that folding laundry can be difficult for people with physical disabilities. Our project would help mitigate these issues for people with physical disabilities. Our solution expands upon previous designs by offering multiple modes, a more compact size, and a cheaper alternative.

1.2. Solution

To solve this crisis, we want to create an automated multi-mode clothes folding system. This will allow the user to decrease the time and effort involved in folding laundry. The whole process takes place in three steps. A user will place one of three preset clothing items on the machine. An overhead camera is used to image the clothing item and send the data to a Raspberry Pi controller. The controller will run the image through a python program that will identify the item of clothing and send a series of instructions out to servos. The servos will operate mechanical folding panels that will properly fold the clothing item. A final panel will place the folded item into a pile.

2. Design

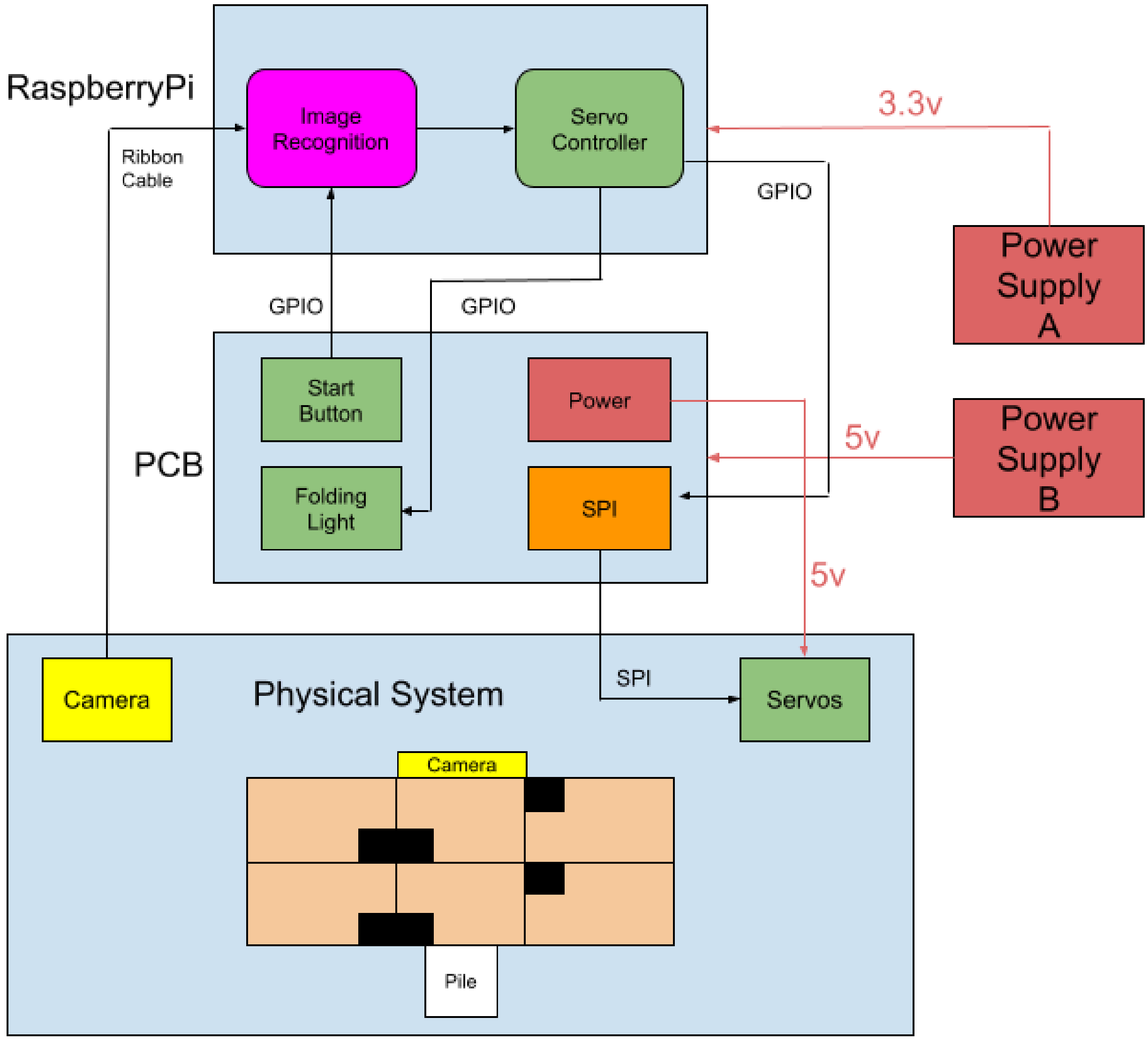

Figure 1: Our block diagram

2.1 Design Procedure

2.1.1 Physical System

The components that make up the physical system are panels, servos, camera mount, PCB board, wiring, and a baseplate for the components. There were many design considerations and ways we could have physically built the project with all of these components.

2.1.1.1 Panels and Camera Mount

The panels were our first consideration along with the camera mount. We initially wanted to make both out of 3D printed material so that we could have precisely cut and professional looking material. The reason why we had to pivot in a different direction is the cost of 3D printing. Even with a student discount, it was far too expensive for our budget. On top of that, 3D printing offered us a lot less of an ability to make changes to our design if something had gone other than planned. Looking back at the choice now, we are very grateful that we chose to use a different material due to all the frequent changes we had to make. After realizing 3D printed material will not work, we had to find a material that was readily available, able to be shaped, and durable enough to handle the weight of the clothes. The material we settled on was cardboard. Despite its less than optimal appearance, it was able to fulfill all the requirements we needed it to. The next step in the panel and camera mount subsystem was sizing. For the panels, we initially planned to fold adult sized clothes. At the professor's recommendation, we lowered our scale and set out to fold toddler sized clothes. We made the middle panel the size of the width of the toddler T-shirt and pullover and then the side panels long enough to cover the pant legs of the jeans. Finally, we had to cut out corner slots for the servos to be able to attach. The camera mount sizing was a little more difficult to accomplish and required a bit more precision. We needed the camera to be able to capture the whole area of the board with as little extra area captured as possible for image recognition. To accomplish this we took many pictures through the console while holding the camera above the board. We found the height that we thought fit the frame the best and marked it. We then cut the wooden camera mount to that height and attached the camera. We chose wood for the camera mount due to its cheap price tag and ability to stand upright despite a load on the end.

2.1.1.2 Servos

The most challenging portion of the physical system were the servos. Many different approaches could have been used to get the servos working, but I believe we chose the best one. The initial plan was to have two servos for each panel so that we did not have to worry about not having enough torque for the clothes. Due to the cheap nature of the servos we purchased, the signals to turn did not register to both at the same time. This caused one servo to begin turning before the other and essentially rip the other one off the cardboard platform it was set on. We tried everything to get the servos to act in unison by trying to amplify voltages using transistors but we concluded it was not a power issue but a construction issue of the servo. We had to then make the decision of purchasing new servos or try each panel only using one servo. We decided to just use one servo for each panel because if it worked it would simplify the wiring, construction, and power delivery by quite a bit. If it failed, we would have to purchase new servos anyway so it seemed like a high reward low risk choice. We did have to remake each panel accounting for only one servo attachment per panel, but that did not take long. When tested using the toddler clothes, the servos were more than capable of folding the clothes and that large problem was solved.

2.1.1.3 PCB Board

The PCB board was designed as a sort of breaker board to help organize our wiring as well as handle the button/LED system of the project. It was designed using KiCad 7.0. There were 8 total through-hole vias that would be used to connect components to ground and 7 through-hole vias that would be connected to power. On the bottom, there were 6 through-hole vias that connected the GPIO outputs to the servos. We chose through-hole vias because they would be easier to solder, and so that we could run wires on both sides of the board in case we needed to avoid an overlap. We chose a diameter of .25mm because it was slightly larger than the diameter of the wires we needed to solder and made the process extremely easy to do. Finally, we have a diode and resistors connecting the ground corner of the button so that the machine would activate when the signal is high and the button is pressed. Without the resistors, we often found ourselves shorting our Raspberry Pi when the button was pressed. The “diode” in the diagram is a footprint place holder for our LED light. The light turns on when the button is pressed. We could’ve used our PCB to handle anything and one issue we found that we ran into was not having a microcontroller. Due to user error and not reading the rubric, we forgot to include one on our PCB causing us to lose some points during the demo. In the future, we would know and add a microcontroller to potentially take over for our Raspberry Pi system.

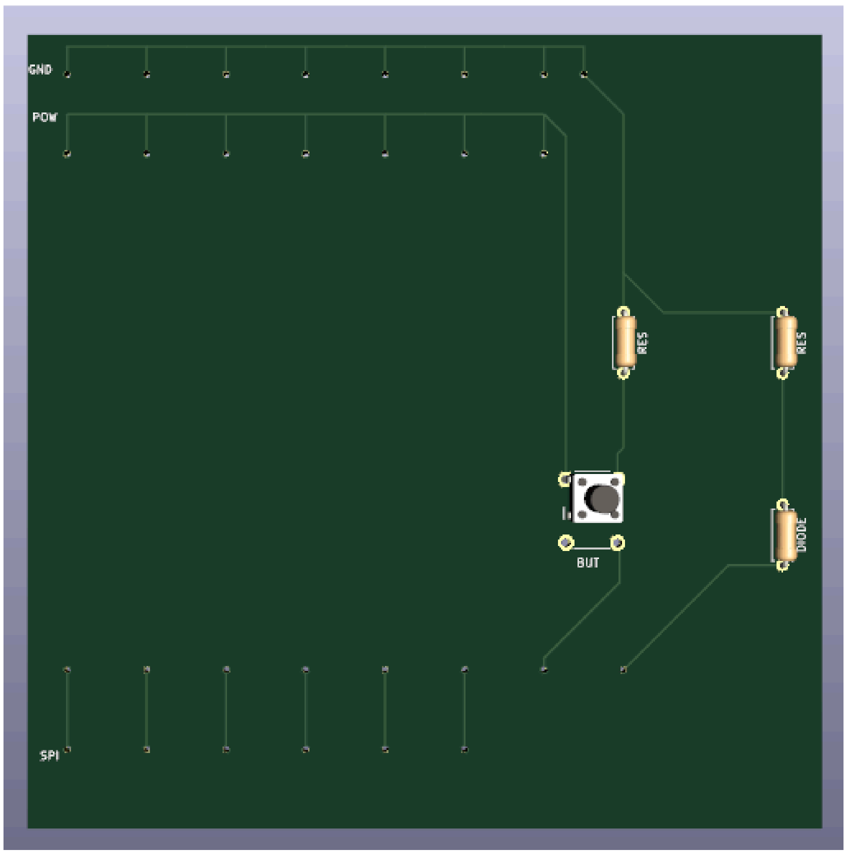

Figure 2: Our final PCB board design

2.1.1.4 Wiring

The wiring was straightforward and handled in a way that made it very convenient for us. We chose to use wires that offered serial connection.This allowed us to break and replace a wire after it was soldered instead of having to use a new PCB board. One downside to this approach was for longer connections it was more difficult to connect four wires together than just running one long wire all the way. The ability to connect and disconnect wires greatly outweighed that downside.

2.1.1.5 Baseplate

The baseplate was possibly the easiest portion of this project. The material was initially debated. For the same reasons we chose cardboard for the panels, we chose cardboard for the baseplate as well. All else we had to figure out was the size of the baseplate and how we attached the other components to it. The size of the baseplate was decided by laying out the panels and cutting the baseplate four inches larger on each side. This gave us plenty of room for the camera mount, Raspberry Pi, and PCB board. We attached the camera mount by gluing cardboard squares together and cutting a hole in the middle to lower the mount into. The PCB and Raspberry Pi were attached to the baseplate by creating bumpers on four corners of each component and gluing them down so that the components could just be lowered into the container. Finally, the servos were put on two cardboard square thick risers to prevent any interference with hitting the board and allowed wiring to fit below the panels.

2.1.2 Raspberry Pi Controller

2.1.2.1 Main Function

A main function integrates the 3 helper systems, image recognition, servo control, and utilities, together. The function accepts two parameters, two booleans *train_model* and *evaluate_model*. The *train_model* parameter controls whether the program will retrain and overwrite the current model that is saved to the SD card. The *evaluate_model* parameter controls whether to output the model's accuracy tested on Keras' fashion MNIST.

The main function then enters a loop that continues forever. Upon receiving a high signal from the button, the functional part of the program begins. It starts by signaling for the camera module to capture an image of the folding board. It then passes this image into a normalization function provided by the Image Recognition system, and then passes this image to a predict function, also provided by the Image Recognition system. The main function will pass the output of the predict function to the servo control system which will handle the folding. The system then waits for the next button press.

2.1.2.2 Image Recognition

The image recognition model is built on a Convoluted Neural Network (CNN). This is achieved by creating three 2-dimensional convoluted layers. The layers contain 32, 64, and 64 kernels respectively. Each layer has a size (3, 3) which limits the amount of information each kernel can inspect at once. Each layer uses a rectified linear unit (ReLU) activation function. The ReLU activation function picks the maximum of the input and 0, essentially ensuring the output will never be negative. The model is then compiled with an adaptive moment (ADAM) optimizer on a categorical cross entropy loss function. This means that the ADAM optimizer will adjust the learning rate for each parameter to achieve minimal loss.

Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Figure 3: The different categories for clothing items

The data used to train the model was taken from Keras' fashion Modified National Institute of Standards and Technology (MNIST). The Keras fashion MNIST contains 60,000 train images and 10,000 test images over 10 different labels. Each image is 28x28 pixels where each pixel is a single channel (grayscale) and corresponds to 1 of the 10 labels. The program loads all train and test data through the Keras *load_data* function. Our project focuses on t-shirts, trousers, and pullovers, therefore we only utilize about 18,000 train images and 3,000 test images. This is done by iterating through all train and test data and removing the image and label if the label is not a 0 (t-shirt), 1 (trouser), 2 (pullover). Next the program manipulates all images of trousers. When a user lays trousers on our folding machine, they will be laid horizontally rather than vertically. Therefore, all images of trousers are rotated 270 degrees. The model is then fit to the train data.

The next responsibility of the image recognition system is providing a method to save and load the trained models from the SD card. Keras provides two useful functions *load* and *save* which allows our program to easily read and write a trained model to the card. In deployment, our machine would only utilize the *load* function. Our program incorporates this to reduce the booting time. If the machine had to train a model every time, it would need internet access and about 10 minutes of time to train the model. Loading a model from storage happens in milliseconds, significantly faster than training a new model.

The image recognition system also supplies a normalization function. Images captured by the camera module are 3 channels, 1280x720 pixels. This needs to be reduced to a 28x28 pixel, grayscale image. 280 rows of black pixels are added above and below the image. The image is then resized and converted to grayscale by using *resize* and *grayscale* methods provided by Pillow (python imaging library). The normalization function returns this image to the main function.

Lastly, the image recognition system provides the predict function. The predict function accepts an image, converts it to a numpy array and passes it through the trained model to retrieve a prediction. The prediction, which is a 0, 1, or 2 is converted to a string indicative of the type of clothing and returned to the main function.

2.1.2.3 Servo Control

The servo control system contains a dictionary, where each value is a list of panels to fold. The *main* function of the servo controller iterates through each item of the list, passing the value as a parameter to the *fold* function provided by the utilities helper system.

2.1.2.4 Utilities

The utilities helper system provides all the communication methods between the physical ports on the Raspberry Pi and the program. These methods include: *setup_board*, *turn_diode_on*, *turn_diode_off*, *fold*, and *button*.

setup_board sets the board mode to broadcom (BCM). It initializes all pins connected to servos as a PWM output signal, the pin connected to the button as an input signal, and the pin connected to the LED as an output signal. It sets the duty cycle of all the servo pins to 0%.

turn_diode_on and *turn_diode_off* just set the output voltage to the LED pin to 5V or 0V.

fold accepts an integer, 1 to 6 inclusive, as a parameter. It will set the duty cycle of the pin connecting to the corresponding servo to 70%. The function then sleeps for 2 seconds before changing the duty cycle back to 0%. Finally, the function sleeps for 1 more second before returning.

2.1.3 Power

The power supply for the entire system was originally supposed to be provided by one source. However, as the project progressed, time constraints prevented the design, testing, and implementation of a power system. Instead of one power supply, two power supplies would be used. One source supplies the Raspberry Pi with 3.3V and the other source supplies 5V to the servos.

2.2 Design Details

2.2.1 Panels

The panels are what manipulate the garments on the board. There are two types of panels: wings and center. The wing panels flank to the left and right, while the center panels are the two in the middle. The center panels have dimensions of 7.5 x 8.5 inches, while the wing panels have dimensions of 7.5 x 8.5 inches. The discrepancy is to allow for more of the shoulder of shirts and pullovers to rest on the wing panels prior to folding. This allows for more area of the garment to be folded, making sure a proper fold gets created. The panels were able to hold and manipulate the garments satisfactorily.

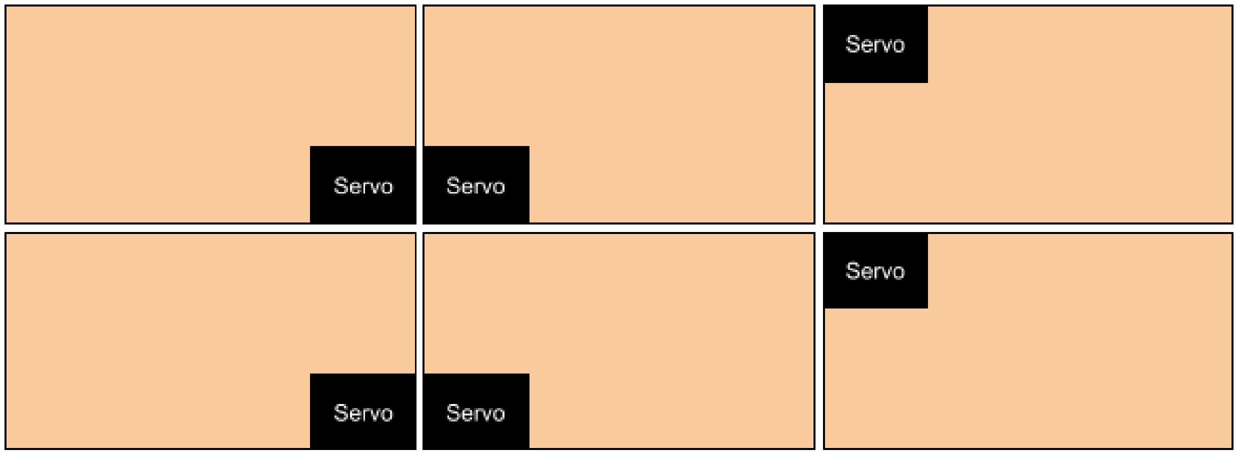

Figure 4: Layout of Panels

2.2.2 Servos

Servo motors are mounted to the base plate of the prototype through risers that give space to allow the panels to actuate without running into the base plate. This is demonstrated in the figure below. The servo is mounted to the panel through cut outs as seen in the figure above. Servos are programmed to rotate to 126° at a speed of roughly 252 degrees per second, completing each fold in about half a second. This gives enough force to keep the garment on the panel as it is rotating and enough rotation to fold the garment.

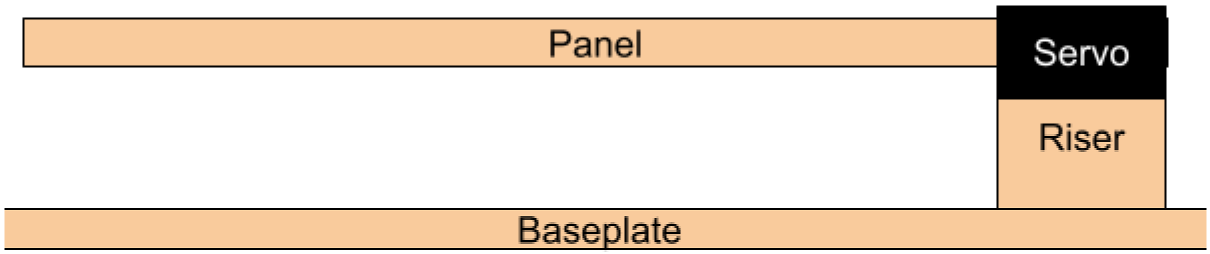

Figure 5: Diagram of Riser

2.2.3 Baseplate

The baseplate is the backbone of the whole project. The project sits on top of the baseplate with certain components blocked in by cardboard bumpers and others sitting atop risers.

2.3.4 Power

By using two power sources, stable voltage was able to be provided to the Raspberry Pi (3.3V) and servo motors (5V). Power to the servos were connected in parallel. To test our power delivery system, we activated the two servos (the most our design would ever be required to activate) together to see if constant power could be delivered. The servos moved smoothly without any disruption. We repeated this test multiple times and received the same result.

2.3.5 Folding Button and Light

The folding button and light and on the PCB. Pressing the button starts the automated folding process, while the light is used to indicate when the prototype is currently folding. We can tell our requirement was verified because the light would be on while the machine was folding and off while it was not. Finally, the button could be verified as meeting the requirement because the machine activated flawlessly after each push.

3. Design Verification

3.1 Raspberry Pi Controller

See Table 1 and 2 in Appendix A to view the requirement and verification table

3.1.1 Image Recognition

The image recognition model we trained achieved 97.9% accuracy when tested on the Keras fashion MNIST test data. Our model was significantly less accurate in predicting clothing items in an actual functional test. The software was able to distinguish trousers from t-shirts and pullovers about 90% of the time. The software was unable to distinguish between t-shirts and pullovers. Classifying both as pullovers. This resulted in a functional test accuracy of 63.33%.

3.1.2 Servo Control

The software was correctly able to output PWM signals of 0% and 70%. This resulted in the proper actuation of the panels. The servo control system was also able to control these outputs in a manner that neatly folded clothes.

3.1.4 Camera

The camera successfully completed its requirements as well. As previously discussed earlier, the method we used to capture the whole surface area of the board and little else was successful. Each picture limited background exposure and was able to fully view each clothing item. The other requirement of being able to send the image to the Raspberry Pi was also successful. Through a ribbon cable, the camera was able to be recognized by the Raspberry Pi and when told to an image was able to be taken and viewed.

3.1.5 Edge Cases

Due to wanting to keep our image recognition system accurate we decided to abandon the edge case requirements. Implementing them would have required us to train our model on images of the blank panel system. This would have caused some situations, especially in low lighting, where the machine is unable to recognize the clothing item. To maintain practicality all edge cases were not implemented.

3.2 Physical System

See Table 3 in Appendix A to view the requirement and verification table

3.2.1 Panels

Our panels met every requirement that we set out for them as well. The clothes did not sag through the panels. This was not an issue since we did not need to cut holes within the panels for weight reasons. The panels are also easily connected to the servos through attachments that came in the servo packaging. This was easily verified by looking at the panels and testing their strength by giving them a load. This load would obviously be the clothes set on top of the panels. This brings us to our next requirement that the panels are strong enough to support the clothing item. Both these requirements can be tested by folding a piece of clothing. When we did this, the panels stayed attached and did not bend or fold, indicating a success.

3.2.2 Servos

Servo operates with roughly 12 kg/cm torque at 5v. Each panel was about 80g, when loaded with a garment the total weight would be about 180-280g. Thus, the torque provided by the servos was more than enough to achieve rotation at 252 degrees/sec. Servos were able to read the PWM signal sent by the Raspberry Pi to actuate to 126° in about half a second.

3.3 Auxiliary Systems

See Table 4 in Appendix A to view the requirement and verification table

2.3.4 Power

Stable power was successfully verified by using a voltmeter. Raspberry Pi was continually supplied with 3.3V and the servos was supplied with 5V.

2.3.5 Folding Button and Light

The folding button successfully started the program. The folding light was off initially, then was on during the folding process, and then finally off when the folding ended.

4. Costs

4.1. Labor Costs

We determined the hourly value of each individual group member's time by utilizing the UIUC success report from 2021-2022, as shown in figure 6.

MAJOR	Employed Graduates	Graduates Reporting Salary	Average Salary	25th Percentile	50th Percentile	75th Percentile	Graduates Reporting Signing Bonus	Median Signing Bonus
Aerospace Engineering	41	27	$82,373	$70,000	$80,200	$85,000	14	$3,500
Agricultural & Biological Engineering*	14	10	$69,182	$70,000	$72,500	$75,000	--	--
Bioengineering	28	15	$76,979	$71,000	$75,730	$81,000	8	$5,000
Civil Engineering	54	38	$69,037	$65,000	$69,100	$73,500	19	$3,000
Computer Engineering	135	104	$109,176	$90,000	$110,000	$127,500	78	$15,000
Computer Science	93	70	$129,377	$112,500	$124,000	$142,000	61	$30,000
Electrical Engineering	59	48	$87,769	$76,000	$82,000	$95,000	34	$6,000
Industrial Engineering	37	20	$81,275	$75,000	$77,000	$88,000	17	$5,000
Materials Science and Engineering	26	15	$71,627	$70,000	$72,000	$79,200	9	$5,000
Mechanical Science & Engineering†	105	58	$79,118	$71,417	$79,000	$84,000	35	$5,000
Nuclear, Plasma & Radiological Engineering*	15	14	$76,486	$69,850	$75,000	$75,500	--	--
Physics†	26	12	$84,855	$50,000	$75,100	$121,000	5	$27,775
Systems Engineering & Design	36	20	$84,136	$76,000	$81,009	$92,500	12	$7,250
All Grainger Engineering	653	443	$92,813	$75,000	$83,000	$112,500	297	$8,500

Figure 6: Graduate starting salaries of student graduating from University of Illinois

Assuming an average employee works 40 hours a week for 52 weeks of the year, we can estimate hourly wages. Computer engineers would average $52.49/hr and electrical engineers would average $42.20/hr.

	Tyler Hirsch	Bryson Maedge	Nolan Opalski	Total
Hourly Rate ($/Hr)	52.49	42.20	42.20	45.63 avg
Labor (Hrs)	96	96	96	288
Total Cost ($)	5039.04	4051.2	4051.2	13141.44

4.2. Parts Costs

Part	Part Number	Number Purchased	Total Cost ($)
Cardboard Box	-	2	13.00
Super Glue	-	2	8.00
Servo Motor	MG996R B0BYD9M1P3	16	68.00
Raspberry Pi	Model B B07TC2BK1X	2	124.00
Camera	Arducam 5MP Camera B012V1HEP4	1	10.00
Micro SD Card	PNY 32GB Elite Class B07R8GVGN9	1	5.00
PCB	-	5	30.00
Estimated Cost			$199.00
Total			$258.00

4.3. Total Costs

Total Labor	$13141.44
Total Part	$258
Total	$13,399.44

5. Conclusions

5.1 Summary

In conclusion, while our final project did not fully work, the process and subsequent feedback from professors, peers, and the TAs proved to be invaluable for our future development as lifelong students. Our total design worked well, but due to some last-minute issues involving a broken Raspberry Pi and then our files not being able to install on the second Raspberry Pi the image recognition software was not able to be applied to the final demo. The most valuable feedback we received was to better advocate the solutions we applied to the problems we faced whether they worked or not. Professor Gruev stated to us that if we don't advocate for ourselves then no one else will. During our demo when we were facing our challenges, we stated the problem we faced but did not expand on what we did to try and solve it. This was shown to us to be a crucial mistake because while we personally know what we attempted, if we do not explain what we attempted it looks as if we gave up. We mitigated this mistake in our presentation by further expanding on our exhaustive attempts to try and solve our installation problems, but the lesson is one that will stick with us. Given more time and a new Raspberry Pi we would be able to install our software and connect all the systems to result in a complete and useful product.

5.2 Accomplishments

Our accomplishments are numerous despite the result not being where we wanted it. The first accomplishment is setting the bar high for us and tackling a project that not only is useful but technologically difficult. Each small issue we encountered we deployed a tactical and logical response that involved debate, discussion, and teamwork to plan and accomplish. We believe we did not fail at the result but just ran out of time. We integrated two of our three systems and accomplished many of our high-level requirements. We also made sure to apply a lot of more theoretical concepts we learned in class to our very physical design. Using components like transistors and diodes which we previously have usually only seen on circuit designs and breadboards, into a hardwired PCB was a challenge we accomplished. Finally, at every step we integrated the feedback from the communication-oriented TA's, professors, and the rest of the course staff to improve on all aspects of presenting and designing a large-scale project. We greatly improved our technical writing skills and feel far more prepared to take these skills to our post-college careers.

5.3 Future Work

The prototype has lots of features that can be enhanced to increase functionality across the board. The first thing to improve is to expand the size to fold adult sized garments. To increase accuracy of the image recognition, the panels and camera can be modified to reduce glare. The AI model can be trained on more garments allowing for the expansion of the roster of clothes the prototype can fold, leading to more folding instructions to be programmed. The quality of the prototype can be improved as well. Better materials, such as plastics, can be used for construction to allow for a water-resistant design that is more rugged and customizable. High quality servos can solve issues where the prototype would shake during program start-up and shut down to a fluctuation in the voltage of the GPIO ports.

5.4 Ethical Considerations

Due to the nature of the project, many of the ethical concerns listed in the IEEE and ACM Code of Ethics are upheld naturally (for example, the codes on intelligent systems don't apply because we are not implementing that capability.) In the IEEE code of ethics, there are two codes that are extremely important. These two codes are I.1 and I.5. The team wants to hold the safety and welfare of the public paramount while accepting criticism and updating our machine to fulfill the initial focus of helping the user of our machine. To fulfill this, the team initially designed our project to help people with health conditions or impairments like arthritis, cerebral palsy, and other physically limiting conditions. Throughout the designing and construction phases, the team has made sure to keep that in mind and not implement anything that competes with that focus. Finally, the team will thoughtfully consider improvements and feedback that is offered to us from the recipients of our project.

In the ACM Code of Ethics, the most unique and relevant ethical concerns that are not previously mentioned in the IEEE's list are the Professional Leadership Principles. These have been paramount to uphold since ultimately this has been a learning opportunity. These codes of ethics pertain to allowing each member of the group to be able to contribute and learn best from this project. To abide by these rules, a policy of open and considerate communication between group partners was adopted. This allowed us to voice concerns while also considering one another's interests and overall experience throughout the project.

For the safety concerns, our project does not fulfill any of the special concerns listed on the website. We did not use any batteries, large voltages, and no currents will be going through human subjects. Every member of the group has completed the Lab Safety Guidelines training and will strictly adhere to the rules and overall ethos of safety first. Our system is designed in a way that no bodily harm could result from its regular use.

6. References

[1] ACM. “”ACM Code of Ethics and Professional Conduct”.” (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 5/1/2024).

[2] ADA. “”Laundromat tips for People with Disabilities”.”(2020),[Online]. Available: https://thedutchmanslaundry.com/laundromat-tips-for-people-with-disabilities/#:~:text=Folding%20and%20sorting%20clothes%20may,a%20more%20inclusive%20laundromat%20environment.

[3] dfrobot. “”DSS-P05 Standard Servo”.” (2024), [Online]. Available: https://www.dfrobot.com/product-236.html (visited on 5/1/2024).

[4] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 5/1/2024).

[5] Team, K. (n.d.). *Keras Documentation: Fashion Mnist Dataset, an alternative to mnist*. https://keras.io/api/datasets/fashion_mnist/

[6] UIUC. “”ECE 445 SAFETY GUIDELINES”.” (2024), [Online]. Available: https://courses.engr.illinois.edu/ece445/guidelines/safety.asp (visited on 5/1/2024).

[7] UIUC “”New Graduate Average Salary””(2022), [Online]. Available: https://illinisuccess.illinois.edu/themes/custom/sasubtheme/files/2021-2022_Illini%20Success_FullReport_AllCampus%20_FINAL.pdf (visited on 5/1/2024)

7. Appendix A Requirement and Verification Tables

Table 1 Raspberry Pi System Requirements and Verification

Requirement	Verification	Success
Image Recognition subsystem will identify three different articles of clothing	- System can properly crop picture so that only clothing item is in view - System can resize cropped picture to 28x28 pixels - System can predict clothing type from resized, cropped images with at least 90% accuracy.	???
The servo controller will contain three different programs, one for each type of clothing	- The servos are able to move in a way indicative of the three programs that are programmed	The most efficient way of folding our garment selection was the same for each, so there is only one folding pattern
Camera correctly captures image of the clothing item	- It is far back enough to correctly capture the folding area - Camera correctly captures and sends the image to the Raspberry Pi	Successful

Table 2 Edge Case Requirements and Verification

Requirement	Verification	Success
If an item is not recognized, folding light will turn red and blink three times	- Image identification sends out error code for unidentified object - Raspberry Pi Controller sends out error code	Removed

	signal - LED lights up	
If there is no item on the folding table, the folding light will turn red and blink once.	- Clothes do not sag through panels - Panels have mounting point to connect to servos - Panels are strong enough to support clothing item	Removed

Table 3 Physical System Requirements and Verification

Requirement	**Verification**	**Success**
Servos actuate to fold the clothes	- Raspberry Pi Controller sends out signals for servo actuation - Servo's move to correct angle at correct speed	Successful
Panels can physically fold the clothes	- Clothes do not sag through panels - Panels have mounting point to connect to servos - Panels are strong enough to support clothing item	Successful

Table 4 Auxiliary System Requirements and Verification

Requirement	**Verification**	**Success**
Power system will supply stable voltage and accurate voltage to all the subsystems	- Voltage on both side of linear voltage regulators are accurate - Power supply is	Linear regulators were not used as it was decided that two power sources would be used, one for the servos

	receiving power from wall outlet - Power supply is supplying power to its output	supplying 5v and another for the Raspberry Pi supplying 3.3v
Folding button will start the entire folding process	- Raspberry Pi controller receives signal to start the processes - Raspberry Pi controller correctly starts once the start signal is received	Successful
Fold light will be active when the folding process is occuring	- Raspberry Pi controller sends power to folding light while running - Raspberry Pi controller sends no power while idling	Successful