Spring 2021

Automated Mixed-Dose Pill Dispenser with Image Verificaton

Nairu Garcia-Acevedo

Rohan Bhatt

SANTA CLARA UNIVERSITY

Department of Electrical and Computer Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Nairu Garcia-Acevedo and Rohan Bhatt

ENTITLED

AUTOMATED MIXED-DOSE PILL DISPENSER WITH IMAGE VERIFICATION

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

BACHELOR OF SCIENCE
IN
ELECTRICAL ENGINEERING

6/4/21

Thesis Advisor(s) date

Shoba Krishnan 6/9/2021

Department Chair(s) (use separate line for each chair) date

AUTOMATED MIXED-DOSE PILL DISPENSER WITH IMAGE VERIFICATION

By

Nairu Garcia-Acevedo and Rohan Bhatt

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Electrical and Computer Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Electrical Engineering

Santa Clara, California

Spring 2021

Abstract

With the unfathomable number of medications that are available today due to medicinal advancements, there is room for error in the distribution of such medication even within proper pharmaceutical settings. Our primary goal is to reduce the overall contribution of a wide range of human errors in the dispersion of medication which is found to be an extremely large industry in today's age while doing so in a cost effective manner. This is to be executed first using a robotic arm, to automate and create mobility, and second, including image verification, to classify and select pills based on certain characteristics. Through the incorporation of mixed-dose packaging, this product will generalize this service to larger markets such as hospitals and pharmacies that are otherwise only provided to individuals by third parties at an extra cost. Within our time frame, we have obtained successful dispensing for a preliminary set of pills based on color, size, and shape.

Acknowledgements

We would like to extend a special thanks to our academic faculty advisor, Dr. Andrew Wolfe, for all of his support, guidance, and constructive criticism throughout the duration of this project. His input has helped to shape our project from its inception into what it has successfully become. Additionally, we are extremely grateful to Santa Clara University School of Engineering faculty and staff for all of their encouragement as well as for providing us with financial support to retrieve the necessary resources to make this project feasible.

Table of Contents

List of Figures

Introduction and Motivations

1.1 Background

Thanks to the scientific advancements within the past few decades, humans have at their disposal tens of thousands of kinds of medications to better their health and prolong their lives. With many lives in the hands of pharmacists, doctors, and nurses, one can comprehend the importance of their job in being able to disperse the correct medication to the correct patient. While, in theory, the act of dispersing is quite mechanical and consists of checking patients' prescriptions, dosages, quantities of each pill, and packaging them into containers, the result is not always so clean cut.

As humans are an imperfect species, with that comes inevitable errors. These errors that are introduced due to the human involvement in the dispersion of pills consists of dispensing the wrong dosage, the wrong pill, other incorrect packaging, interruptions, distractions, and contaminations. For further clarification, interruptions and distractions are not errors themselves, but oftentimes do lead to potential dispersion errors. Interruptions and distractions consist of real-life situations where the pharmacist dispensing the patients' medication may need to attend to a patient, answer a colleague's questions or queries, or other circumstantial events related to the job. Any of these events can steer the pharmacist's focus away from the dispensing task and accidentally cause an error. For example, in a previous study, it was found that in a set of 5,072 prescriptions, there was an error rate of 3.23% where only about 80% of errors were actually detected. While this error rate seems really low, dispensing at such high volumes makes these errors amount to an alarming number, especially knowing that about 4.5 billion prescriptions are filled annually just within the United States. With so many prescriptions being filled, it seems beneficial to seek to improve our current dispensing process to help reduce these human contributions errors. Given that correct dispersion of medication directly impacts people'shealth, this is definitely a matter of priority.

However, we acknowledge that in the execution of our planned product, there are ethical considerations to be made. The introduction of an automated system for the dispersion of medication is accompanied by the possibility of harm which primarily comes in the form of wrong classification. This begs the question: what is the accuracy that should be reached to serve as a threshold to be considered safe? We have to find the balance between the idealistic 100% threshold and the practical threshold that is high enough yet achievable. In addition, against this product could be presented the argument that along with other automation technologies, it acts as a substitution for pharmacists. With that in mind, this product is really designed to serve as an addition to pharmacists on the job and optimize their time dedicated to other tasks, not to substitute the importance of proper pill dispersion or replace their jobs.

1.2 Our Novelty

It is important to acknowledge that there are various products that are currently on the market that have targeted fixing this problem. We are not reinventing the wheel. From what we have seen, there are three different types of products that are already being sold.

Among these existing products are automated machines that are priced a little below a million dollars that essentially achieve our goal, but are only used for large scale manufacturing purposes. Not only that, but they only perform unit-dose packaging. These industrial machines only automate the dispersion of pills from the manufacturer to the smaller industry local locations, but not the dispersion from these places directly to the consumer patient. On the other extreme, there are also many variations of at home pill dispensers that require the consumer to organize their medication prior so that it dispenses correctly at the necessary day and time. A third attempt at a solution is the creation of companies such as PillPack which offer this packaging service at an extra cost to consumers. They use an automated system to package unit doses to create an inventory where humans will retrieve the desired pill from. However, these third-party services require the individual to go out of their way to retrieve this service. As for this specific example, it still requires human involvement to gather the correct unit dose packaging to create the ultimate packaging that the customer receives.

Having done this research to see what is currently out there, we have managed to zero in on the aspects that make our idea and implementation different from the rest. We are aiming to provide this cost-effective method that can be used at a smaller scale like in hospitals and pharmacies. We hope to integrate this service and generalize it to these larger markets as opposed to individual clients to avoid having to go through third parties at an extra cost. Additionally, where we have only seen this done with unit doses, we are looking to incorporate mixed-dose packaging in the automation of the dispersion process.

Objectives

2.1 Project Goals

Our primary goal is to reduce human error in the dispersion of pills, specifically for prescriptions in the hospital and nursing home setting. We aim to reduce such human error contributions by providing a cost-effective method that can be integrated into these smaller scale industries while also adding an extra level of safety through the inclusion of image verification. The objective is that our product will be able to successfully perform mixed-dose packaging of a patient's daily prescription medication.

2.2 Project Requirements

In order for this project to be able to meet our set goals and be successful, there are a few factors that are required of the process. First, we will need to find an overall method that limits the amount of human involvement to a minimum, none if possible. This itself requires the usage of many distinct electronic devices, a camera and a robotic arm, being the two most important ones. The camera in conjunction with an image classification system should take care of replacing human's involvement with the verification of the pills in the classification process to a certain degree. The robotic arm will facilitate the movement of the individual pills from their storage locations to its final destination: a discard pile or a final container. In a big picture overview, these are the crucial requirements of our system to make meeting our goals feasible.

Experimental Methodology

3.1 Project Breakdown & Flow

As previously explained, creating a process that successfully automates the movement and classification of pills is the essence of our project. With having to encompass a process that has various moving parts, it was crucial to our execution to break it down into three distinct manageable sections: the physical setup environment, the image classification system, and the robotic arm movement system.

Before diving into these individual sections, it is important to gain a general understanding of the functional flow of our product. With the assumption that all storage and final destinations are open-lid containers, our program, written in Python, will extract information from a patient prescription database containing their name, the pill names, its dosage, and quantity at the very minimum. By following along in Figure 1, one will see how based on the extracted information, the robotic arm will move the selected pills over from their corresponding storage locations to the image classification platform where they will be examined based on color, size, and shape. The results of the classification system will be compared to a pill database containing the exact characteristics of each pill. After classifications have been compared, the robotic arm will move the pills from the platform to either a final container or a discard pile depending on whether it matched on all three accounts thus achieving mixed-dose packaging of pills.

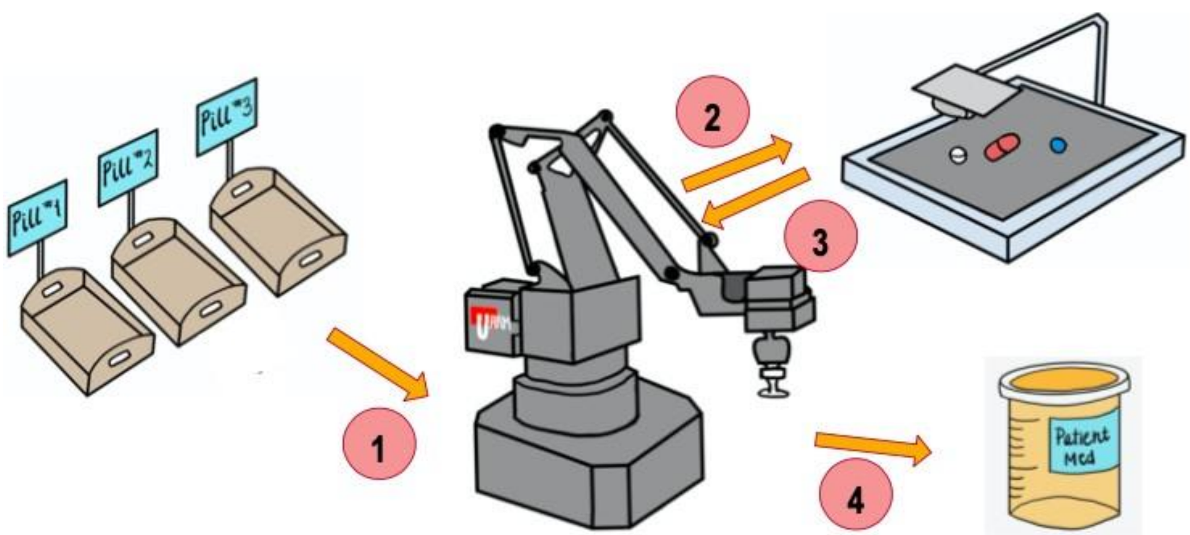
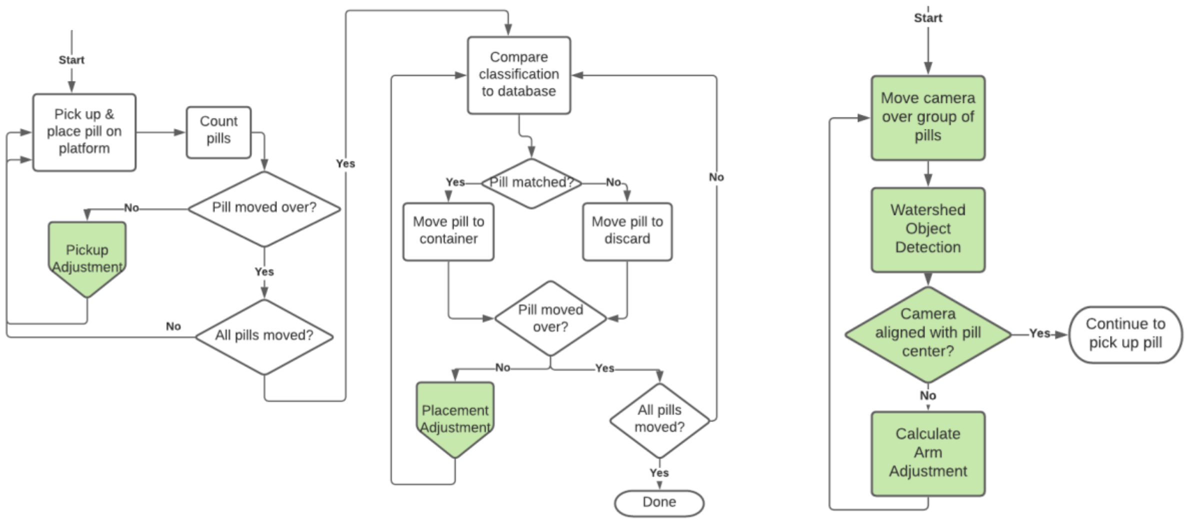


Figure 1: Drawing of General Movement Across Setup Environment

While Figure 1 provides a visual of the overall movement, Figures 2a and 2b provide more details with respect to the specific questions our system needs to answer to make distinctive decisions. These flow charts particularly provide more insight to the adjustment algorithms in place to deal with inaccuracies and limitations accompanying our robotic arm.



(a) (b)

Figure 2: (a) Flow Chart of Overall Automated Process. (b) Flow Chart of Adjustment Algorithm.

3.2 Physical Setup Environment

To be able to see if our system works, we needed a setup environment that would help produce repeatable results. This setup includes a robotic arm mount, a light source and mount, storage containers, final and discard containers, a camera mount, and platform space for classification to occur.

Our entire setup is supported by a wooden plank where brackets were drilled and secured to serve as a mount for our robotic arm. The robotic arm has extended branches on three of its sides with openings where the brackets, shown in brown in Figure 3, would be able to hook onto and anchor it down. This created a leveled platform for the robotic arm to maintain stability and have it start at a fixed location, but still allow the arm to be portable as it is easily removable.

The incorporation of the light source was very key not only to the overall physical set up, but in obtaining repeatable results for our image classification system. This took high priority when it came down to building the setup. While the initial intention was to mount the USB-powered ring light tripod onto the slab of wood, we found that this did not provide the necessary height or stability of light that we sought. By removing only the ring light component of the device, we were able to determine the appropriate height for the ring light which wasclose enough to maintain brightness, but far enough to avoid interference with the robotic arm's movement at its maximum possible height. The ring light is placed between two thin pieces of wood that are also mounted onto the big wooden surface. This new light mount design provided the ability to make any necessary height adjustments as well as sustain the weight of the ring light so that it stays in place above the pill tray.

As mentioned previously, critical to the scope of our project, we had to assume that all pill containers were open-lid containers. We hardcoded the locations of the pill containers for all storage, discard, and final containers. In order to stay fixed to such hardcoded locations, all storage containers were glued to their spots, while the final and discard container locations were placed at marked locations. This was done to facilitate the removal of discarded pills and the replacement of the final container for a new patient.

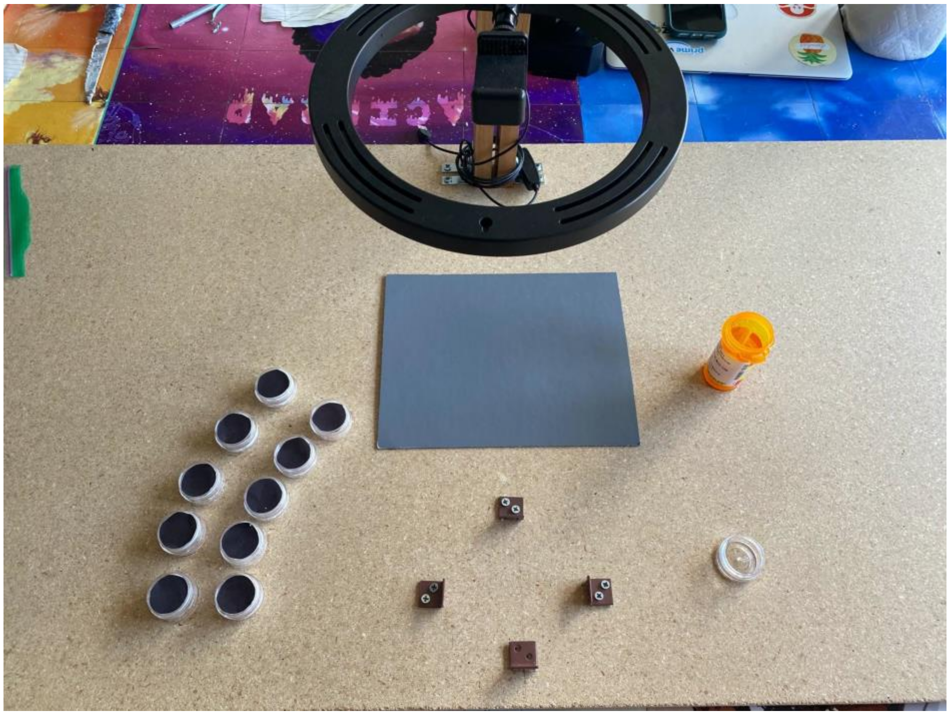
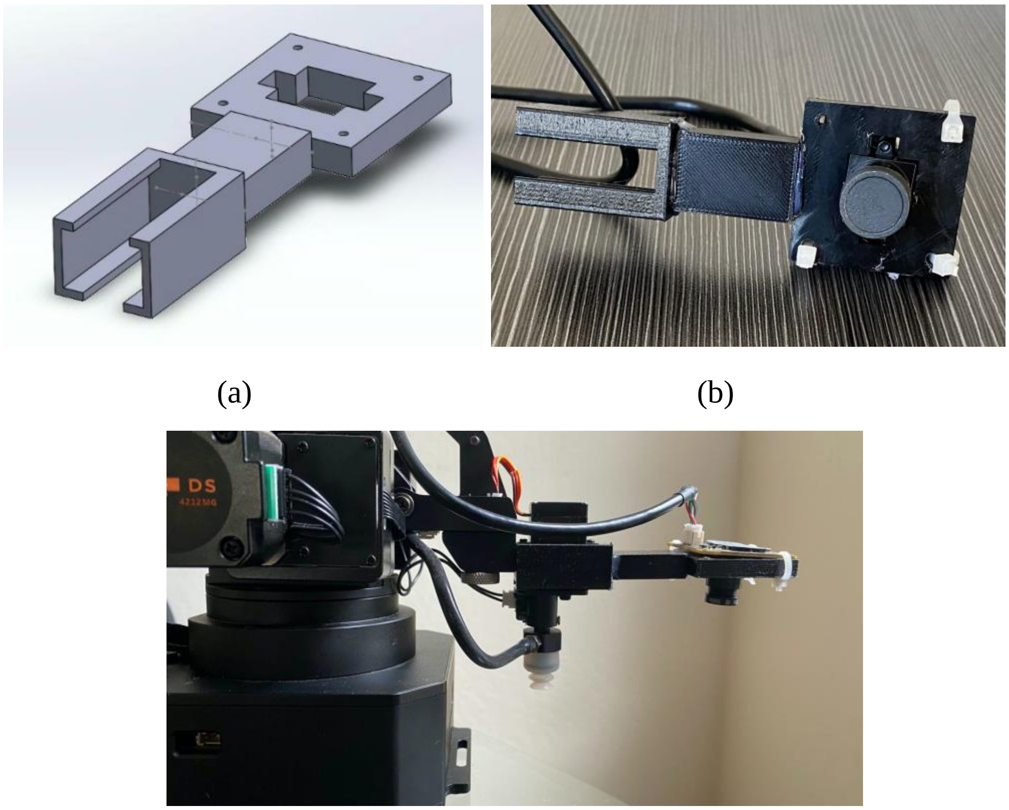


Figure 3: Physical Setup and Testing Environment

The last component of the physical environment is related to the camera mount. The original concept was that there would be multiple cameras located over each section of the platform and that one camera would be mounted onto the phone holder component of the ring light to cover the image classification platform (shown in gray in Figure 3). This presented issues of the robotic arm movement blocking the camera's view of the pills lined up to be classified and not being close enough to produce a good enough picture to perform such classification. Instead, we designed our own 3-D printed camera mount (Figure 4) that would attach to the robotic arm by sliding into and hooking onto the base of the end effector. In addition to providing the ability to adaptively move the camera in all directions, our design also helped to reduce our budget costs, thus contributing to one of our goals of providing a cost-effective product.



(a) (b)

(c)

Figure 4: (a) Solid Works Assembly of Camera Mount Design. (b) Front View of 3-D Printed Design with Camera Attached. (c) Side View of Camera Mount Attached to Robotic Arm

3.3 Image Classification System

3.3.1 Image Capture

The success of our image classification system highly relied on the camera that was chosen to take all the images that were to be processed. Based on some of our physical requirements, we performed the calculations seen in Figure 5 to determine the target resolution of the camera we hoped to buy. Considering that we wanted to obtain detailed images of the pills, we estimated the camera to have about a three-inch square field of view. We also wanted to have at least 100 pixels of the image (F_p) encapsulating the diameter of the pill. Knowing from our pill database that the smallest pill we would be classifying is 6mm, we were able to find the spatial resolution (R_s) which in conjunction with the field of view, provided us our target resolution of 1270x1270 pixels. We ultimately purchased the ELP HD USB Camera providing 1280x1024 pixel resolution which we found to produce a high enough quality image to perform our image classifications accurately at a cost that fits within our budget.

$$Field\ of\ View\ (FOV) = 76.2mm\ x\ 76.2mm$$
$$F_p = 100\ pixels$$
$$R_f = 6\ mm$$
$$R_s = {}^{R_f}/_{F_p} = 0.06\ {}^{mm}/_{pixel}$$
$$R_{ih} \&\ R_{iv} = {}^{FOV}/_{R_s} = 1270\ pixels$$
$$\boldsymbol{Target\ Resolution} = 1270\ x\ 1270\ pixels$$

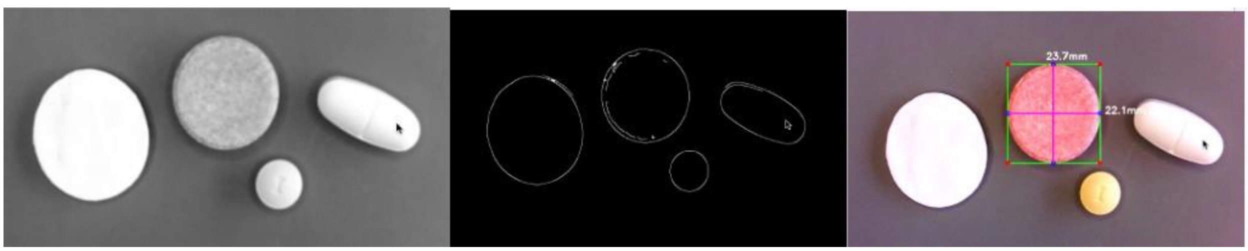
Figure 5: Camera Target Resolution Calculations

Figure 6: ELP HD USB Camera

3.3.2 Size and Shape Detection

We used OpenCV libraries to perform size and shape detection, which were approached hand in hand within our program. Size detection was the most important of the two which is why it was the first classification program we attempted. We started off by taking the captured image of the pills to be classified and converting to grayscale. Further filters needed to be applied, such as gaussian blur and canny-edge detection (both functions are included in OpenCV libraries) to prevent any discontinuities along the edges of the individual pills. The output of all of these filters is shown in Figure 7a below. We performed further edge detection by applying dilation and erosion which, in simple terms, compress and expand each individual pixel. These forms of edge detection helped accentuate the edges enough to create the smooth contours around the pills that are shown in Figure 7b. These filters are necessary since the program has to detect the edges between the object and the background and reduce noise in the image. Based on these contours, we created a bounding box depicted by the red dots and green lines. This allowed us to find midpoints of these lines to find the center coordinate in pixels which is used for other calculations as well. Using a quarter as a reference value, we calculate the realistic lengths and widths of these purple midpoint lines.

We realized the pills should be as close to the center of the image as possible to obtain accurate readings, which is why we no longer sorted them in a straight horizontal line, as the pills on the edges tend to get distorted. From the set of images below, one can see this was fixed by creating a zig-zig placement formation. Once this was done, the size detection program worked within one and a half millimeters of accuracy. Some limitations of this detection algorithm are that it needs the quarter present as the leftmost object as a reference for accurate sizing.



(a) (b) (c)

Figure 7: Intermediate Results for Size Detection. (a) Output Image of Gaussian Blur & Canny-Edge Detection Filters. (b) Output Image of Dilation and Erosion Edge Filtering. (c) Image Displaying Bounding Box

From this point on is where shape detection comes into play. Shape detection solely depends on the results from the previous size detection algorithm just described. The ratio of the length and width of each pill is compared against an eccentricity threshold to determine whether thepill is round or oval. With a given room of error, if the ratio is above this threshold, the pill is considered oval, otherwise, it is round. The main limitation of this algorithm is that we are only able to detect circles and ovals, which is acceptable for our project since a majority of known physical pills fit either one of those descriptions.

3.3.3 Color Detection

Although color detection was the next algorithm approached, it definitely was the most complex of the three. For this, we converted a three-channel, 0 to 255 valued RGB-encoded image to a three-channel, 0 to 360 valued HSV image (hue, saturation, value). This resulted in the program automatically dividing the hue value by 2 to fit the encoding size, so any sample hue values would need to be halved as well. We also had to manually convert these RGB baseline values to HSV. HSV is more versatile since it reduces lighting factor by weighting hue over saturation and value. This was a necessary design change in our algorithm to meet the realistic light variability presented in our physical setup environment. We used a gray card background to place the pills on, as this would reduce reflection and shadows in the captured image and increase the accuracy of the color detection program. We stored the average color baselines for all possible colors in the database, which for our scope included white, yellow, red, and purple.

From there, the distance of the pill's center pixels HSV to average color baselines can be calculated. Keeping in mind the hue will be weighted more, the baseline with the shortest distance will be the predicted color for that pill. The issue we ran into at this stage was distinguishing white from other colors. We found that white depends only on its saturation value, whereas other colors depend only on hue. Thus, we used a threshold for saturation to determine if a pill is white, and if it is over that threshold, the program would proceed to check the other colors.

One of our limitations is that our design is more accurate under consistent lighting conditions, hence the purchase of the ring light that was mounted overhead. Taking into account natural light fluctuations, we worked on the project around the same time of day. Our product may not produce the same degree of accuracy with respect to color detection at other times of the day and other lighting conditions.

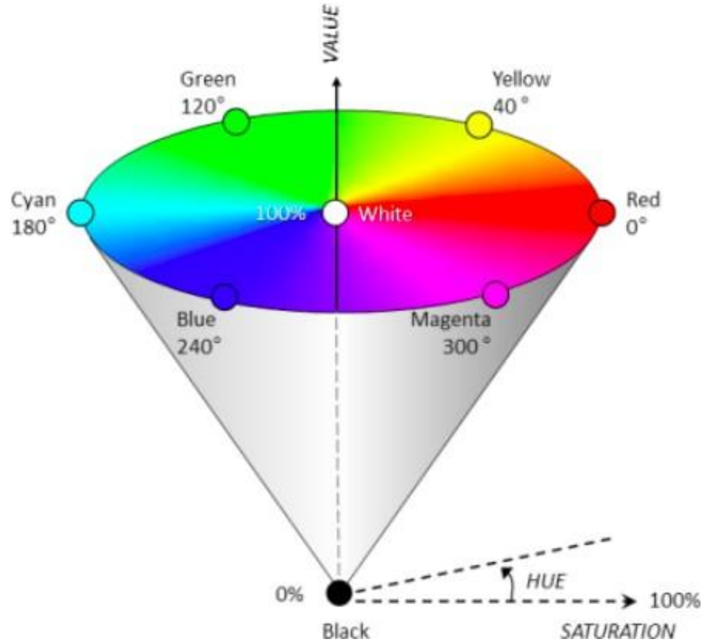


Figure 8: HSV Color Space Visualization

3.4 Robotic Arm System

3.4.1 Robotic Arm

The third significant section of our project is the robotic arm. We researched at the start of our project the necessary specifications to meet our design goals, which helped us to ultimately choose the UArm Swift Pro with 4 DoF, since it met our desired requirements. One of the important specifications that this model met was that it is Python programmable making it easier to merge with the image capture and classification components. Next, it has 2 mm of precision, which is important for picking up pills around 10 mm in diameter. The suction cup end effector included in the robotic arm kit was valuable, as a gripper was too large, bulky, and did not have the most refined interface potential with pills to be considered accurate. We substituted the larger suction cup with a smaller one available, as this allowed the arm to pick up the smallest pills possible. Initially, we had tried using a washer and funnel to reduce the suction cup size, but this decreased the effectiveness of suction itself. One last feature of this robotic arm that was useful is that it can be programmed using a coordinate system that includes positional (x,y,z) and polar (stretch, rotation, height) coordinates, which later allows us to convert pixels to millimeters.

We found that the robotic arm was our main source of limitations to our overall project. An 8 mm in diameter pill was the smallest pill the robotic arm was able to pick up reliably. It could physically pick up pills as small as 6mm if handled manually (UArm software), but with our algorithm, the 2mm precision became a limitation. In addition, the pill concavity mattered, as it was harder for the arm to pick up minimum sized flat pills. Smaller pills with larger concavity had a higher chance of being picked up.

Figure 9: UArm Swift Pro with 4 DoF and Suction Cup

3.4.2 Watershed Detection

A realistic circumstance that we had to encounter while designing and testing with the robotic arm was that sometimes it would not be able to pick up the pill initially. We had to incorporate two adjustment algorithms to accommodate this problem: pick-up and placement adjustment. The main difference between these two adjustment algorithms is where the robotic arm is moving pills from. Pick-up adjustment covers the movement from the storage locations to the image classification platform, while the placement adjustment algorithm manages the movement from this platform to either the discard or final container.

Both algorithms are triggered if the pill does not make it to its next destination either because it was never initially picked up or because it was dropped along the way. When it is triggered, the robotic arm will go back to the location the pill was meant to be picked up from and look for it. To do this, the camera will be positioned over the pill container. Then, we employ the watershed algorithm which references the image as a topographic landscape with peaks and valleys. The background of the image is treated as "sea level". The furthest point from the background, which is the center of each object, is treated as a peak, shown as a blue dot in Figure 10 below. This algorithm allows us to differentiate between pills even if two or more are touching each other. We set a threshold for the distance in between the peaks to be the diameter of the pill which allows for this. This also filters out noise that was picked up on the background and edges of the container. The contours differentiating the individual pills are seen in green in Figure 11. Finally, we find the closest pill to the center of the image and calculate its distance to the center of the image (in pixels). This pixel distance is stored to use during the conversion process which will be explained in further detail next.

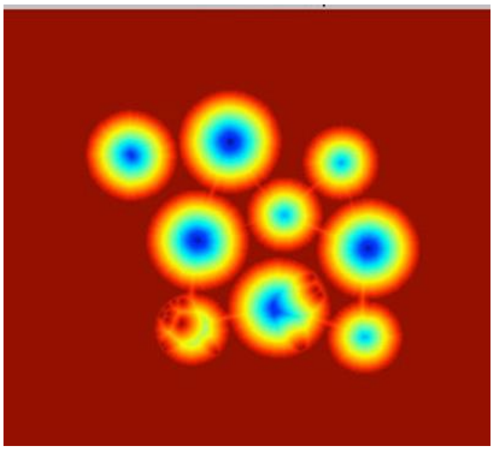


Figure 10: Watershed Elevation Visualization

Figure 11: Post-Watershed Image Result

3.4.3 Image to Coordinate System

Once we find the pixel distance of the pill's center to the image center, as just described in the previous section, we can convert these pixel offsets to physical coordinates that the robotic arm can understand. We used our understanding of the robotic arm's positional and polar coordinates as can be visualized in Figure 12, to make these conversions.

We programmed these adjustments using polar coordinates. The robotic arm's stretch adjustment is found using the pixel displacement in the vertical direction divided by the pixels per mm of the vertical component of the image. The new rotation value is calculated using the pixel displacement in the horizontal direction and the physical distance from the base of the robotic arm to the center of the image area. Both the formulas used in the Python program are shown in Figure 13 below. When the camera is adjusted and centered over the pill, the program applies a fixed offset which is predetermined as the distance between the camera and suction, so the suction will now be centered over the pill to pick it up.

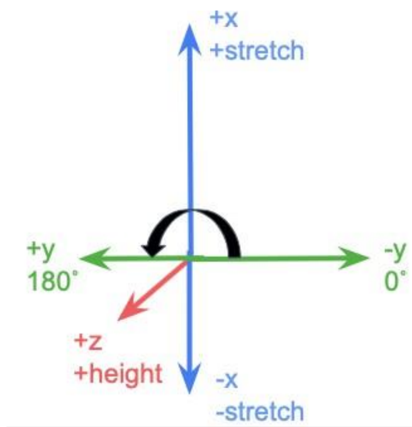


Figure 12: Robotic Arm Polar & Positional Coordinate System

$$\textit{\textbf{new stretch}} = -\frac{PdistY}{\frac{1024}{34}}$$

$$\textit{\textbf{new rotation}} = \frac{180}{\pi}\text{atan}\,\frac{\left(\frac{-PdistX}{1280/_{51}}\right)}{\text{pill\_loc[pill\_num]}}$$

Figure 13: Pixel to Coordinate Calculations

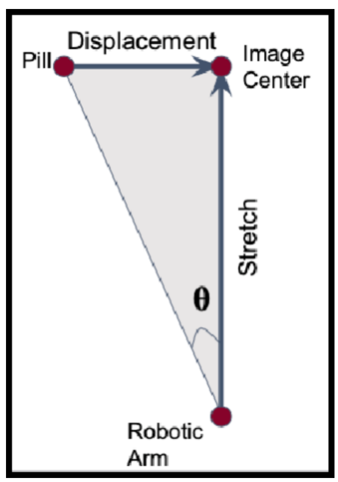


Figure 14: Rotation Angle Visualization

Results and Discussion

4.1 Project Outcomes

For our outcomes, we were able to complete this entire process of integrating the image capture, image classification, and robotic arm systems to work together. This was possible because all programs were run in the same python environment, making the program flow and cross-communication easier. We also found a way to adaptively find and pick up pills instead of relying on fixed locations. This added a layer of intelligent adaptation and complexity by locating a pill within the region of a picture and picking up the pill corresponding to its location on the image. The image classification objective also works reliably to verify pills are of correct size, shape, and color using the respective classification program. The entire process starts with a sample prescription and results in mixed-dose packaging of the patient's correct pills, fulfilling our main objective.

We found that the full process of our project worked in 9/10 trials, and the image classification portion works with a 100% success rate. Most of the missing 10% is from robotic arm movement failures, such as failing multiple times to pick up or drop off a pill on certain iterations. The robotic arm sometimes needed multiple attempts to pick up a pill, but it usually does pick up the correct pill within two iterations. For further clarification, even if the pill were to be picked up after 2+ iterations, it would be considered a successful process. We were able to test and obtain reliable results by striving to keep a fixed setup environment as discussed earlier. Overall, the goals we set out with for the project were achieved, as we got a demonstrable robotic arm setup that reliably completes our target tasks.

4.2 Project Limitations

It would be naive to believe that our design is perfect and ready to be deployed on the market. As alluded to throughout various points thus far, there are still many limitations that need to be addressed to improve and broaden the capabilities of our current design. First, in relation to our physical setup and color image classification, we are limited in the colors that can currently be detected as well as by the lighting variability. With respect to size and shape classifications, we are limited to only two shapes and the necessity of having a reference object, a quarter in our case. Together our whole classification system limits us to a selection of pills that are different enough to be differentiated by solely these three physical characteristics. The most problematic limitation is the robotic arm's precision preventing overall reliable pickup.

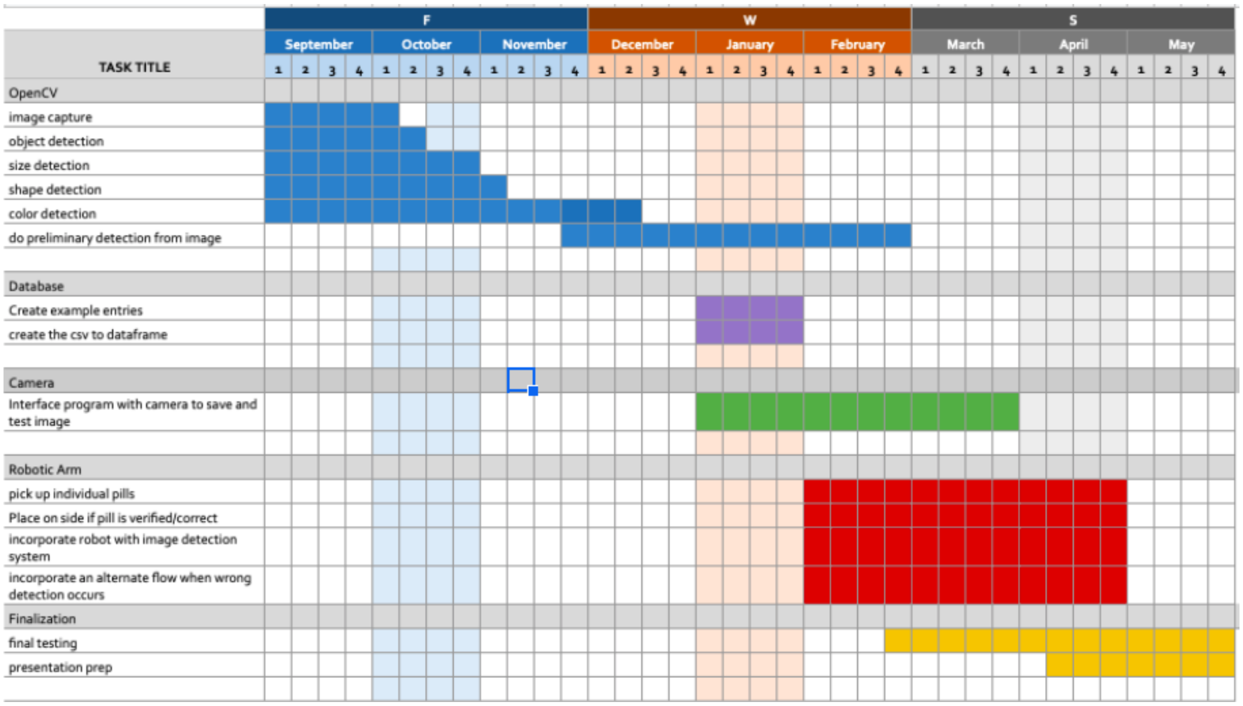
Final Design

5.1 Design Analysis

Given the limited size of the pill container and our thorough calculations for the robotic arm, the reliability is not perfect, likely due to the fact that the robotic arm has an inherent 2 mm precision. This is accurate enough for our project and budget, but we found on some iterations that being off by a few millimeters could result in a vacuum not being created between the suction cup and pill, thus not picking it up that iteration. We would need more time to deploy our product to consumers in order to account for an even larger variety of pills and an even more robust accuracy. One way to improve accuracy is to include text detection, which would allow for a larger variety of sample pills and adding an extra level of verification, but this would need a lot more time than we had to work with. Our current 1280 x 1024 resolution USB camera captures acceptable quality pictures, but using an upgraded camera that has autofocus would capture higher resolution images, thus increasing our classification reliability.

Overall, the flow of our project starts from a sample patient prescription and ends with the patient's container being filled with the correct pills. The program components, such as size, color, and watershed pickup adjustment algorithms all work reliably. The robotic arm also is able to locate and pick up the closest pill to the center of the image, and move the pill to a discard container if the classification results were not as expected. The start-to-finish time to complete a patient's prescription was around 2 minutes. This is due to running the robotic arm at a slow-to-normal speed, since we prioritized accuracy and debugging over a faster run time. The image classification programs also displayed their results as an on-screen image each step, which consumed more time. If we were to deploy the system, the robotic arm could be sped up on most segments and there wouldn't be a need to display all the steps for image classification.

5.2 Project Timeline



Professional Issues and Constraints

It was important for ethical issues to be considered during all stages of our project. We wanted to create a safe project that would yield long-term value. The goal of our project is to design an automated solution that can help mediate the human error contributions to pill dispensation, and that can be implemented in the hospital, pharmacy, and nursing home industry in the long run. This entails accepting responsibility for the consequences of our design. Our project, if deployed to the real world, should not include claims that it is accurate without fault, since exaggerating the performance could result in the pharmacist not using the product as intended. Therefore, we must add a disclaimer that the robotic arm simply adds an additional level of verification on top of the pharmacist's role, but doesn't take away from their responsibility.

Since our project is focused on automation, we understand that with that comes possibly displacing people's traditional roles. However, to counter this we can focus on how these people's roles can be adjusted. Our project doesn't completely remove humans from the pill-dispensing process, but enhances their role by adding an extra level of safety. Fewer incorrect pills given to users means less waste and increased safety for the user, thus improving environmental and business outlooks. Our project mostly deals with programming, thus we did not have as many sustainability issues to consider.

Conclusions and Future Work

7.1 Summary

Overall, our project managed to achieve the goals that we set out initially, as we got the image classification programs to work reliably for our use case and integrated it with the robotic arm. We believe our project was successful for a few reasons. We set the scope of the project to what was achievable in a one-year timeframe, and excluded some aspects that were not critical to the basic functionality. We also started researching and using the image classification libraries as early as the summer, which gave us enough time to deal with setbacks and adjustments. The initial planning stage helped tremendously later on, as we were able to justify our problem statement and were confident about how we would implement it. Although our product is not quite ready for deployment, we successfully reached our objectives stage that serves as a basic platform for implementing and designing this product.

7.2 Future Work

For future groups looking to expand upon our project, one major project is including text detection. After researching, we found that the available text detection algorithms would not be sufficient for our project. The reason is that most pills have engraved text, which is the same color as the rest of the pill. Without significant computing power, an accessible text recognition program is not able to differentiate between the shadows of the text and the pill which is the same color. Text recognition would allow a greater variety of pills in our database and would be a challenging project on its own.

Another future expansion of our project could be to adaptively find pill storage locations. This could be done with a barcode verification system that we were unable to implement due to lack of time and not fitting within the scope. Other methods could still be explored.

Lastly, an important focus to expand our product on would be to attempt to reduce the size limitation of the pills being picked up and detected. This could range from improving the accuracy of the individual classification programs to improving or changing the robotic adjustment algorithms to working with other robotic arm models. There is a lot of room for improvement that future groups can take on, if interested.

References

[1] Cheung, Ka-Chun, et al. "Medication Errors: the Importance of Safe Dispensing." *British Journal of Clinical Pharmacology*, Blackwell Science Inc, June 2009, www.ncbi.nlm.nih.gov/pmc/articles/PMC2723208/.

[2] Bose, Gitika & Pabari, Tejit. (2019). Pill Detection & Prescription Analysis. May 2019, 10.13140/RG.2.2.24583.34728.
https://www.researchgate.net/publication/340528010_Pill_Detection_Prescription_Analysis

[3] Cheung, Ka-Chun, et al. "Medication Errors: the Importance of Safe Dispensing." *British Journal of Clinical Pharmacology*, Blackwell Science Inc, June 2009, www.ncbi.nlm.nih.gov/pmc/articles/PMC2723208/.

[4] Flynn, Elizabeth Allan, et al. "Impact of Interruptions and Distractions on Dispensing Errors in an Ambulatory Care Pharmacy." *OUP Academic*, Oxford University Press, 1 July 1999, academic.oup.com/ajhp/article-abstract/56/13/1319/5159269.

[5] Chamberlain, Michael. "Prescription Filling Process, How Prescriptions Are Filled." *Prescription Hope*, Prescription Hope, Inc., 8 July 2020, prescriptionhope.com/blog-prescription-filling-process-how-prescriptions-are-filled/.

[6] "Uarm Swift Pro Standard 4 DoF Metal Robotic Arm Bluetooth Suction Cup." *RobotShop.com*, www.robotshop.com/en/uarm-swift-pro-standard-4-dof-metal-robotic-arm-bluetooth-suction-cup.html.

[7] Chouinard, Jon. "Imaging Basics: Calculating Resolution for Machine Vision." *1stVision Inc. - Machine Vision Articles*, 6 May 2019, www.1stvision.com/machine-vision-solutions/2015/07/imaging-basics-calculating-resolution.html.

[8] Steve, Gadget, et al. "Measuring Size of Objects in an Image with OpenCV." *PyImageSearch*, 16 Apr. 2021, www.pyimagesearch.com/2016/03/28/measuring-size-of-objects-in-an-image-with-opencv/.

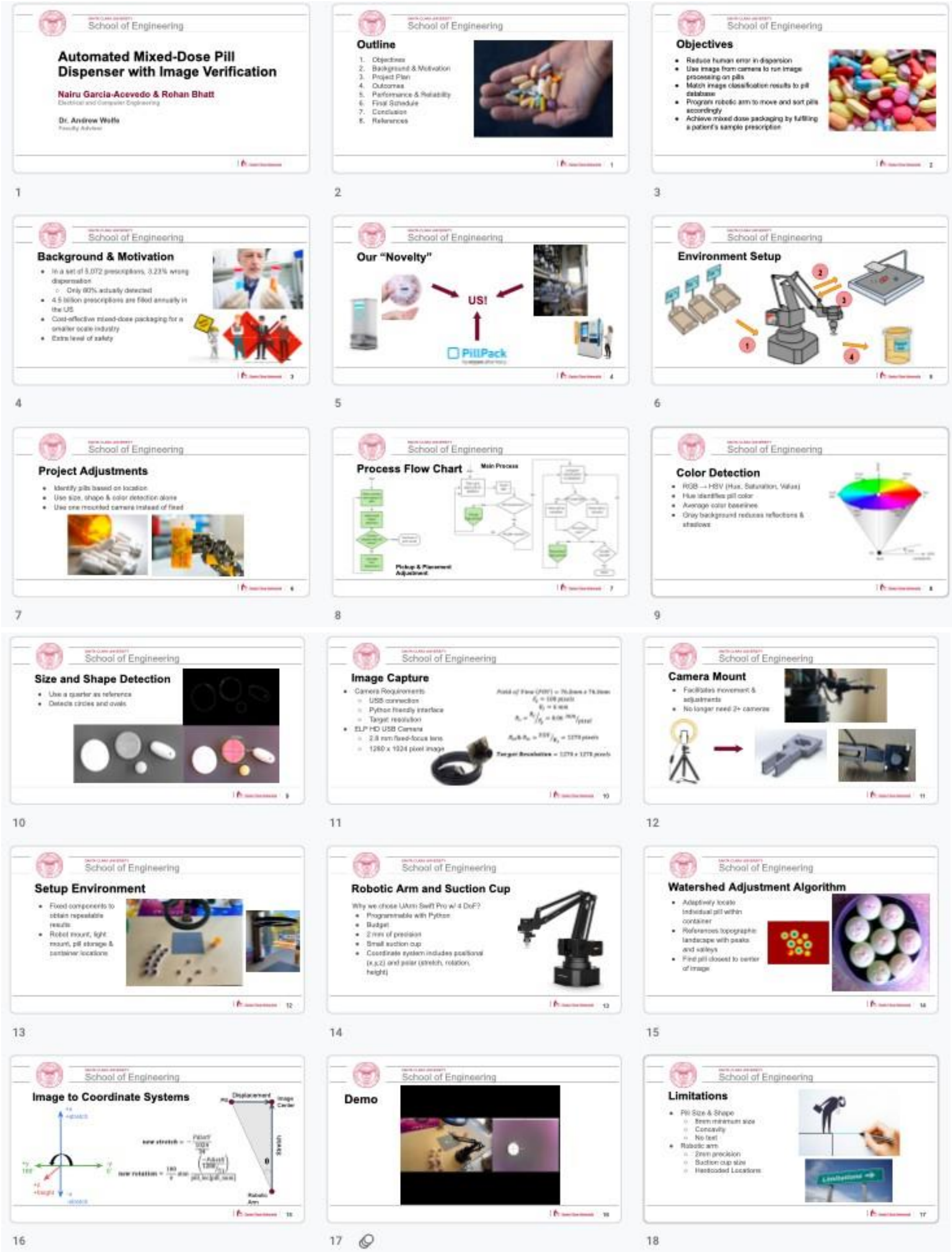
[9] Rosebrock, Adrian, et al. "OpenCV and Python Color Detection." *PyImageSearch*, 16 Apr. 2021, www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/.

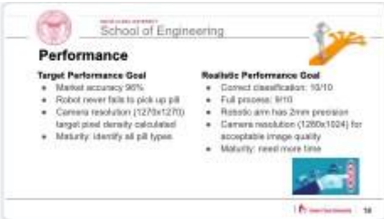
[10] Rosebrock, Adrian, et al. "Watershed OpenCV." *PyImageSearch*, 17 Apr. 2021, www.pyimagesearch.com/2015/11/02/watershed-opencv/.

[11] UArm-Developer. "UArm-Developer/UArm-Python-SDK." *GitHub*, github.com/uArm-Developer/uArm-Python-SDK.

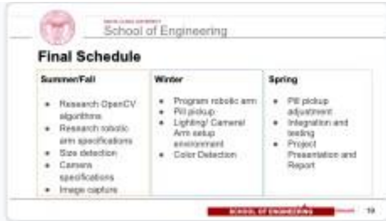
[12] UArm-Developer. "UArm-SDK: Swift." *GitHub*, github.com/uArm-Developer/uArm-SDK/blob/master/doc/swift_api.md.

Appendix A: Senior Design Slides as Presented



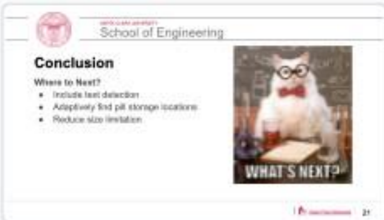


19

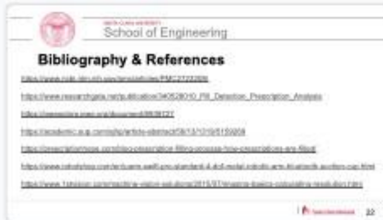


20

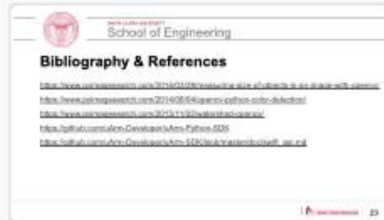
21



22



23



24

25