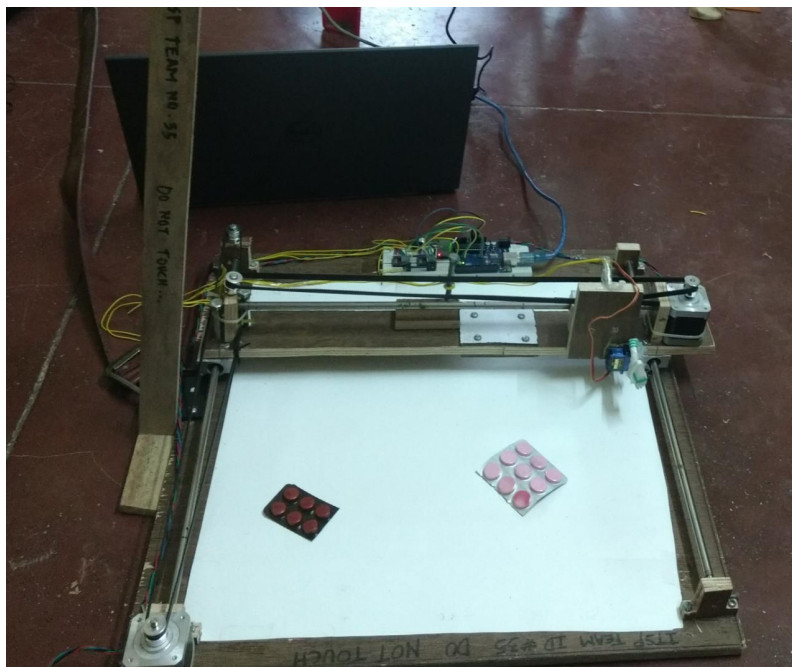


# Automated Dispensary

## Overview

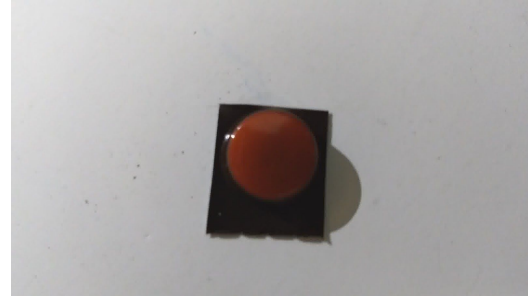
- We implemented an automated dispensary on a small scale.  
It is a mechanism that can identify strips of medicines kept on a table on the basis of their colour and shape, and pass them to a designated spot.
- There were two major components to the project -
  - Training a classifier and doing the related image processing for optimizing the process of getting the position of the object from the image
  - Building the CNC mechanism to transport the object.
- An additional UI was a mobile app to accept voice commands and to transmit commands to the Arduino through Bluetooth, and further to the main program.
- Also, this project has potential applications for bed-ridden patients.

The final image of the whole setup:



## Classifier and IP

- The classifier was built by retraining the final layer of the MobileNet v1.0 to suit the current problem. The classifier is inspired by a Google CodeLab for TensorFlow.<sup>[1]</sup>
- It was trained on a set of images obtained by frames extracted from a video using [FFmpeg](#).



Examples from our dataset

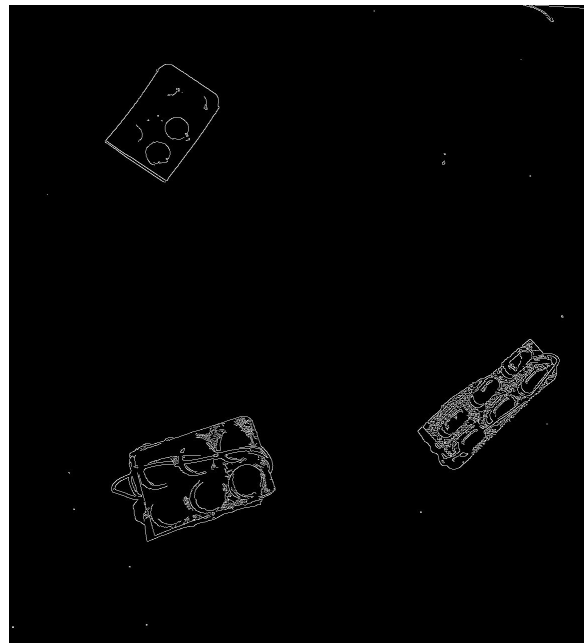
- Sliding window algorithm was implemented to get the coordinates of the pills in the actual image.
  - Problems faced: Misclassification was common especially when the window did not cover the entire pill/covers multiple pills.



Our original  
implementation of  
sliding windows

- Proposed solution: Retraining the model on a dataset having multiple configurations of pills, and increasing the probability threshold of acceptance.

- There were two major problems with the sliding window-
  - The size of the image captured was encapsulated by over 3000 sliding windows, leading to huge processing time.
  - Our final aim was to get the x and y coordinates of the entire pill strip, and in a naive sliding window implementation, this would be done by computing the centroid of all the sliding windows which gave a positive result for the pill. The problem arose if there were two distinct pill strips, in which case the centroid would be at a wrong location. One workaround for this was to use some clustering algorithm, but this had its own problems.
- The solution - we decided to apply contouring to our image, and after some filters on the size and aspect ratio of the contours, we would run sliding windows only on the smaller sub-images. This also had some problems, as often one pill-strip was broken down into multiple contours (due to lighting differences etc.). To overcome this, we decided to merge adjacent contours together.

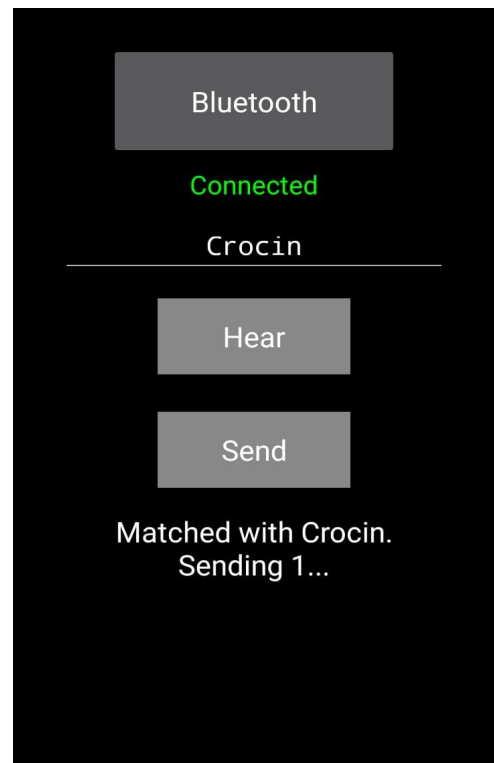


## Arduino communication

- PySerial library was used to establish communication between Arduino and laptop using a Python program.
  - Problems faced: Serial.read() function in Arduino could only accept a single byte as input, but Python does not have inbuilt single byte classes. Also, writing an int value (in Python code) did not work well.
  - Solution: encode() function in Python can be used to convert string object into a *bytes* object, which is compatible with the write() function in *serial*.

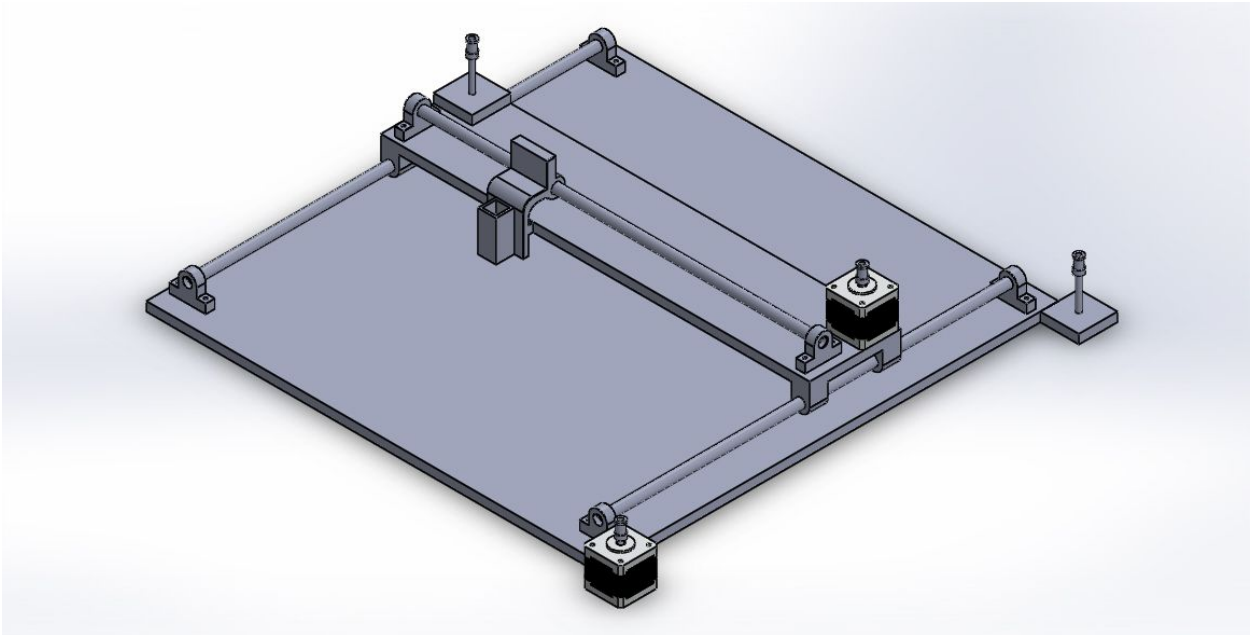
## Mobile Application

- A mobile application was made using [MIT App Inventor 2](#) to accept voice commands from the user. Speech recognition is performed by Google's speech recognition engine which comes inbuilt in Android. The app then matches the text with known medicines and send a corresponding digit to Arduino.
- Bluetooth communication between the mobile application and Arduino was achieved by using an HC-05 module and the SoftwareSerial library for Arduino.
  - Problems faced: Instead of the desired digits, garbage values were transmitted to Arduino.
  - Solution: Using the baud rate of the HC-05 module (9600 in our case) solved this.



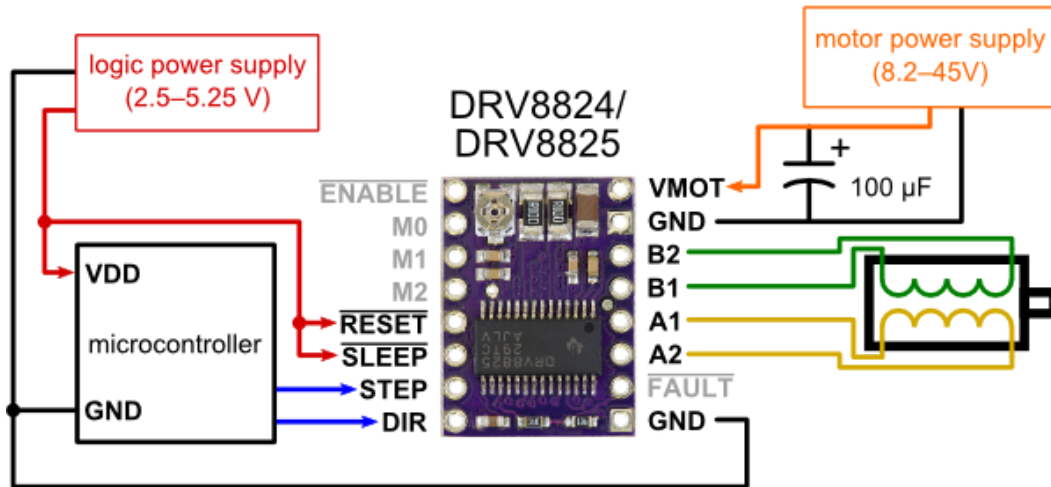
## CNC Arm

- In the first week itself, we built a SolidWorks model of the CNC arm. We used timing belts and timing pulleys to achieve translational motion driven by the stepper motors. Other popular mechanisms are the screw and the string mechanism, but we rejected these based on the size of our arm (50cm x 50cm).



- We built most of our structure with wood since it was easy to work with. However, there are three essential components to this mechanism, which cannot be chimped on- timing belts<sup>[4]</sup>, timing pulleys<sup>[2]</sup>, and linear bearings<sup>[3]</sup>. One major problem we faced in the manufacturing was that the motion of the arm was obstructed due to the rod bending and the linear bearing not being smooth enough. These can be eliminated by aligning the rod supports properly and by being careful while inserting the rod into the bearing.
- The arm was moved by 2 stepper motors (specs ahead), driven by 2 DRV8825 drivers. Some precautions to be followed while handling these-
  - Before using the driver, set the current limit to around 1-1.5 A (depending on your motor). A [tutorial](#) to do this.
  - Ensure that a capacitor of at least 100 $\mu$ F is connected across the motor power supply.
  - Do not disconnect the motor while the driver is powered.

- Monitor the temperature of the heat sink on the driver. If it gets too hot, disconnect the power source.
- If the motors are just vibrating instead of rotating, check the wiring of the motor coils, and check the battery's voltage.
- The wiring for DRV8825 stepper motor drivers:



- Once the coordinates of the pill were determined, we encoded the path as a series of letters ('r','u','l' and 'd' for right, up etc.) and passed these to the Arduino one by one (because the Arduino serial buffer is only 64 bytes long).