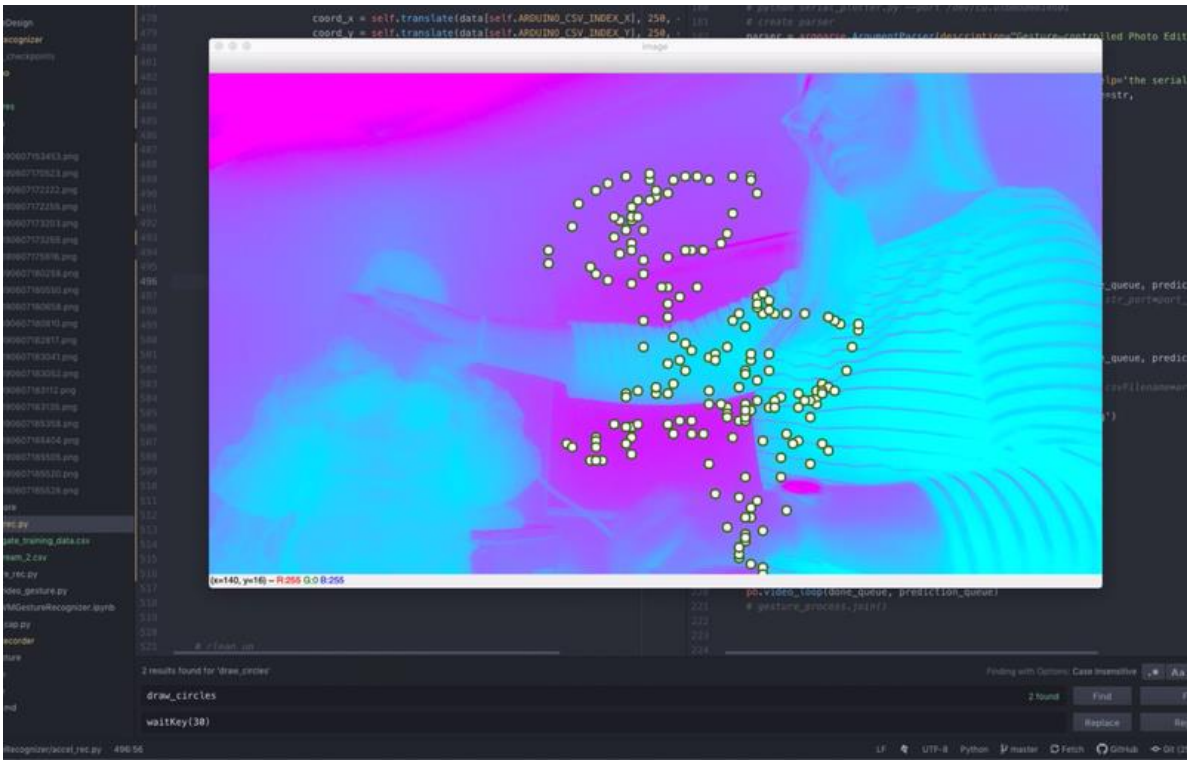
PROJECT	Gesture based tool for sterile browsing images
TEAM ID	PNT2022TMID39067

Gesture Based Photo Booth Tool (images generator class)

Introduction:

Gesture Based Photo Booth Tool



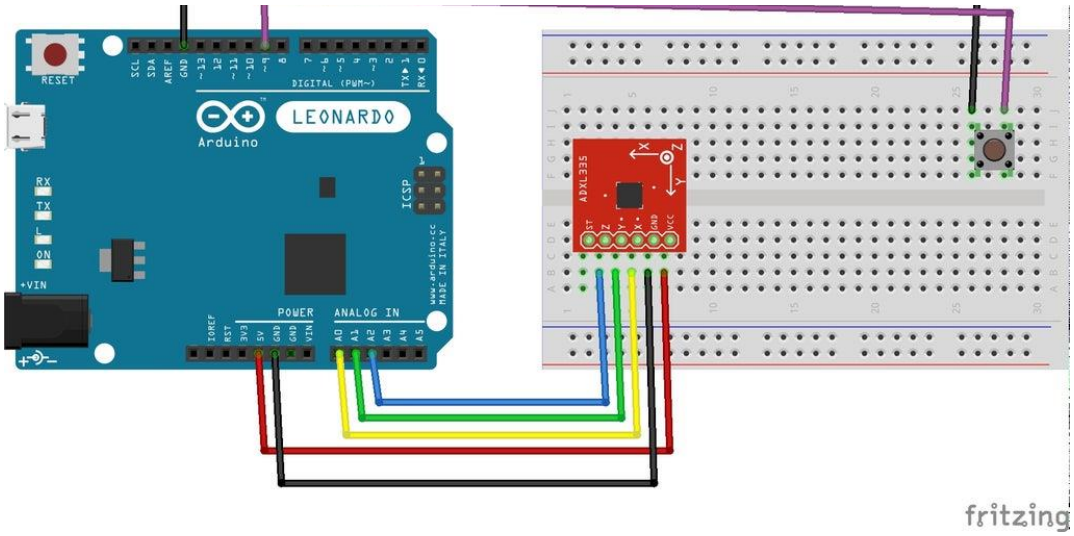
In this project, we built a gesture based photo booth type tool. Through a variety a gestures, the user can control draw on or filter their webcam image. With our tool, you can swipe through filters including greyscale and edge detection among others. You can also push a button on our controller to draw abstract art on top of your image. Finally, you can either save your image or undo all changes through circle and X gestures.

In this tutorial, we will go through the steps to build an arduino based controller, design an enclosure for the controller, record gesture data, train an SVM on the recorded data, and implement the photo booth-like interface.

Supplies

- Arduino Leonardo
- Perf Board
- ADXL335 3-axis accelerometer
- Tactile push switch

Step 1: Arduino Circuit Design



Our circuit diagram is pictured here. We ended up soldering the accelerometer to a small perf board and soldering wires to the tactile push switch to allow us to raise the button to the top of the controller enclosure.

Step 2: Enclosure Design

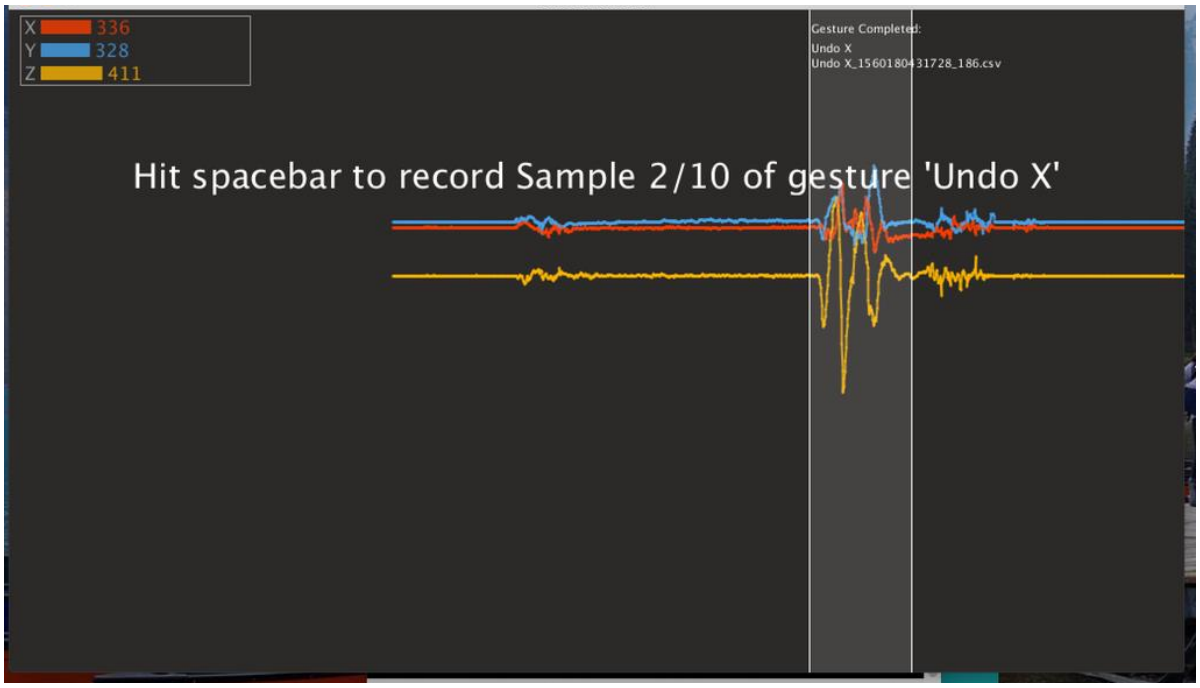
`

We designed our enclosure in Autodesk Fusion 360.. Our enclosure has finger grips, a USB slot, a hole to fit the tactile push switch, and a snap lid.

We printed using an Ultimaker 3 printer with a 0.2 mm layer height and infill at 20%. We also printed supports with the body of the case. Because we used a more coarse layer height,

we allowed for some more wiggle room in our design (i.e. a slightly larger USB slot and a larger hole for the push switch)

Step 3: Gesture Recording



Using the controller constructed in steps 1 and 2, we recorded training data using simple Arduino code and Processing. We programmed our Arduino to send the Arduino time stamp and the x, y, and z values for the accelerometer over the Serial port. For gesture recording, we ignore the button portion of our controller.

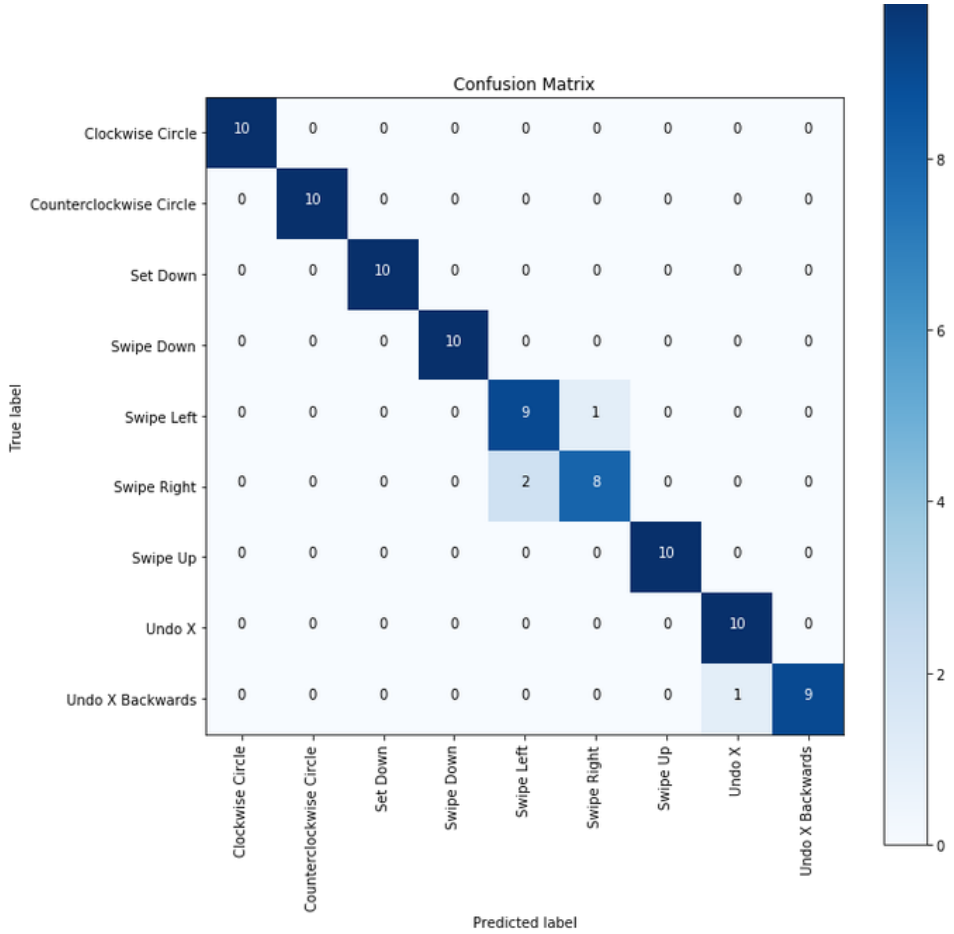
The Processing program records 10 samples of each gesture and writes the Processing time stamp, Arduino time stamp, and x, y, and z accelerometer values to a file. It creates one large full stream CSV file and individual gesture files for each sample of each gesture. The gestures are segmented using cues from the space bar; that is, the user presses the space bar to both begin and end a gesture recording.

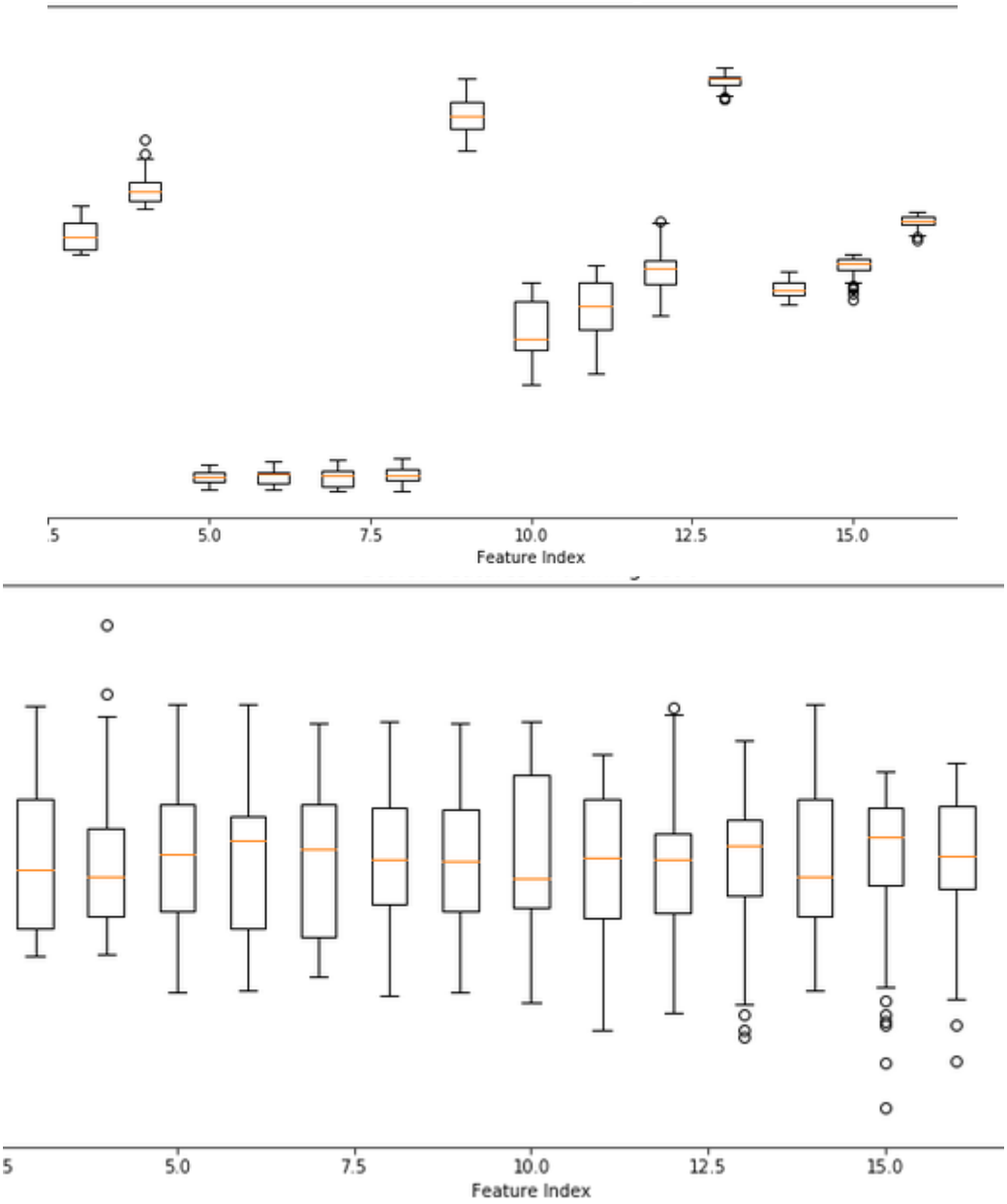
The gestures we allow in our tool are:

- swipe right, to swipe through filters

- clockwise or counter-clockwise circle, to capture and save an image
- mid-air X, to undo all changes and revert to the original image

Step 4: Training the SVM





We chose to use Python to implement the remainder of our photo booth tool. We leveraged several python libraries to simplify training and testing a model for gesture classification.

We selected 19 features based on our recorded gesture data for training and classification, and trained on a gesture set including 10 trials for each gesture. We did some processing on the data including detrending and smoothing to allow for less noise in our model. We calculated the magnitude for each signal when loading in the gesture CSV files. We also trained a scaler to scale the features in our model. Once our model was trained, we saved the

Model and feature scaler to a file for use in our main photo booth program.

The features we selected include:

- number of zero crossings
- number of peaks
- average peak height
- max x, y, z, and mag values
- min x, y, z, and mag values
- mean x, y, z, and mag values
- standard deviation of x, y, z, and mag values

Step 5: Real Time Classification