

# Gesture Controlled Audio Software

## Capstone Final Presentation

Anusha Bhattacharyya, Troy Watts, Maxx  
Smith, and Winston Phillips

# Introduction

## Problem:

- Smart Devices
- Live Music
- Interactive Presentations
- Theater Productions

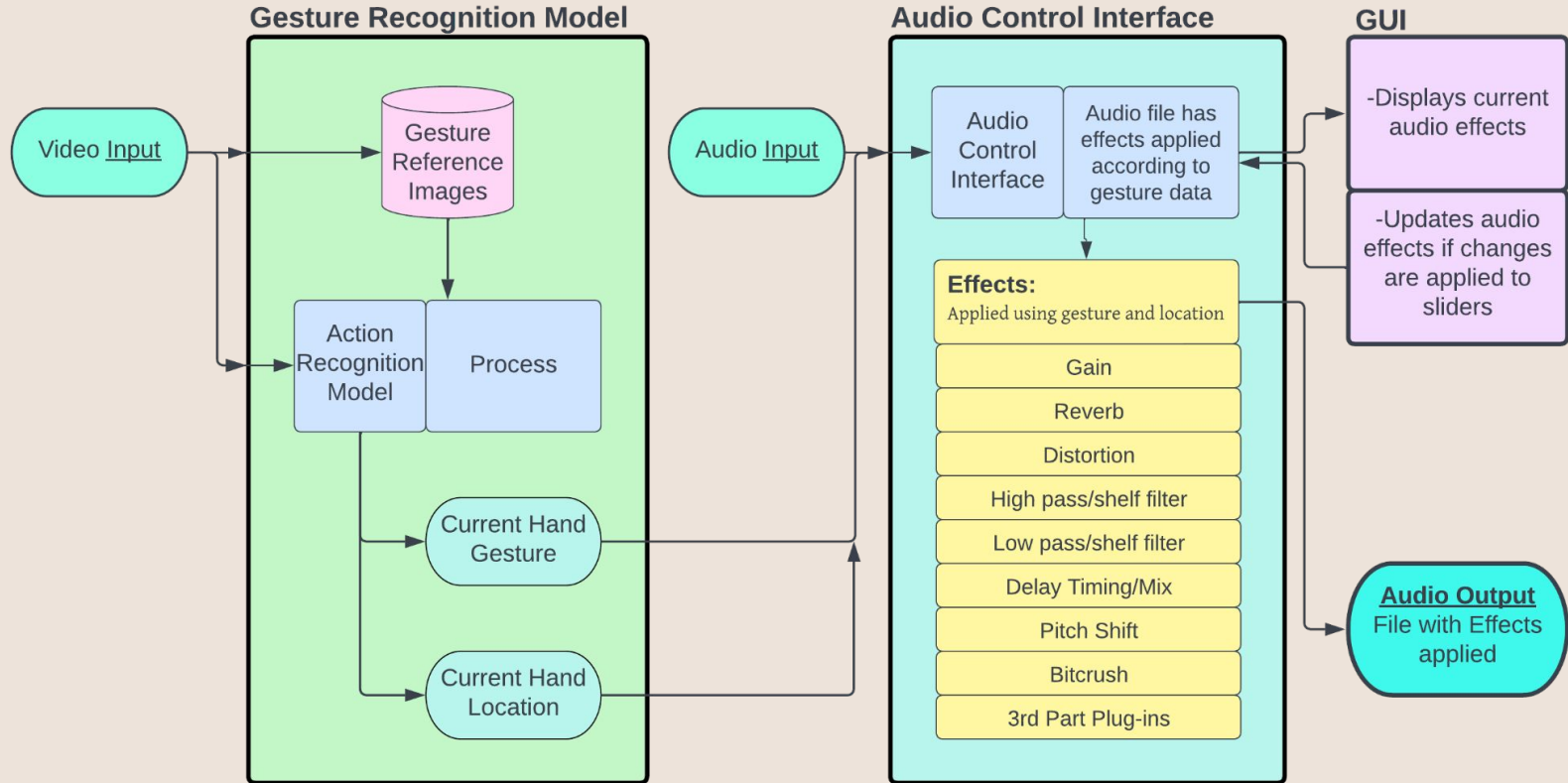
## Objective:

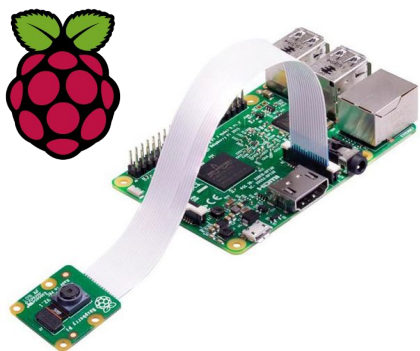
The objective of this project is to design, prototype, and build a program that can take in video input and turn that into audio output depending on what gestures are being made using an AI generated algorithm.

## Keywords:

- Key-Point Inference Algorithm/Paradigms
- Audio Control Interface (Pedalboard)
- DAW (Digital Audio Workstation)

# Project Architecture





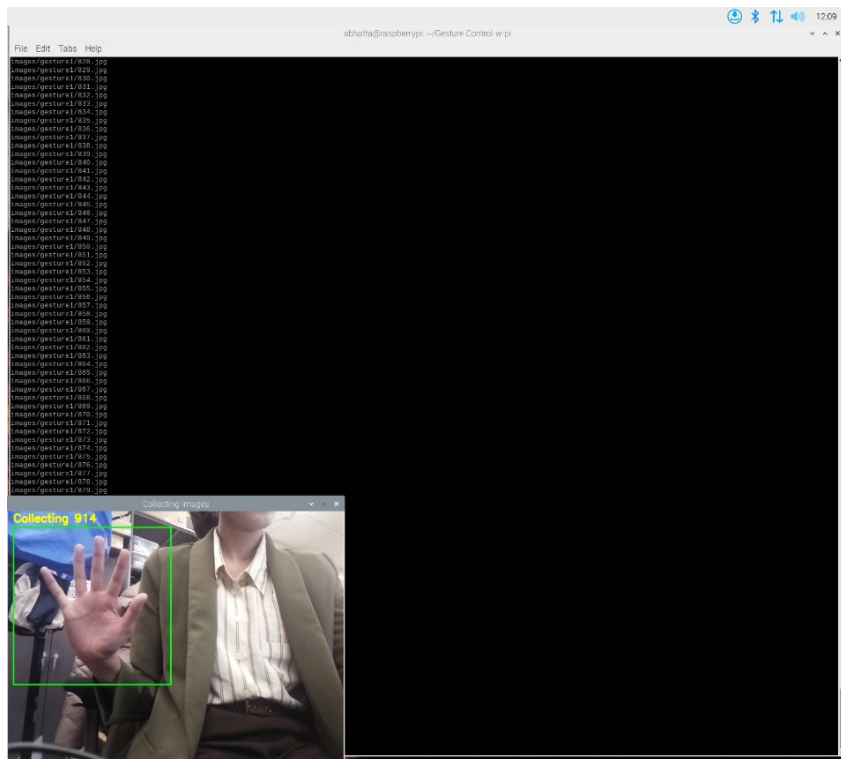
# Action Recognition Model

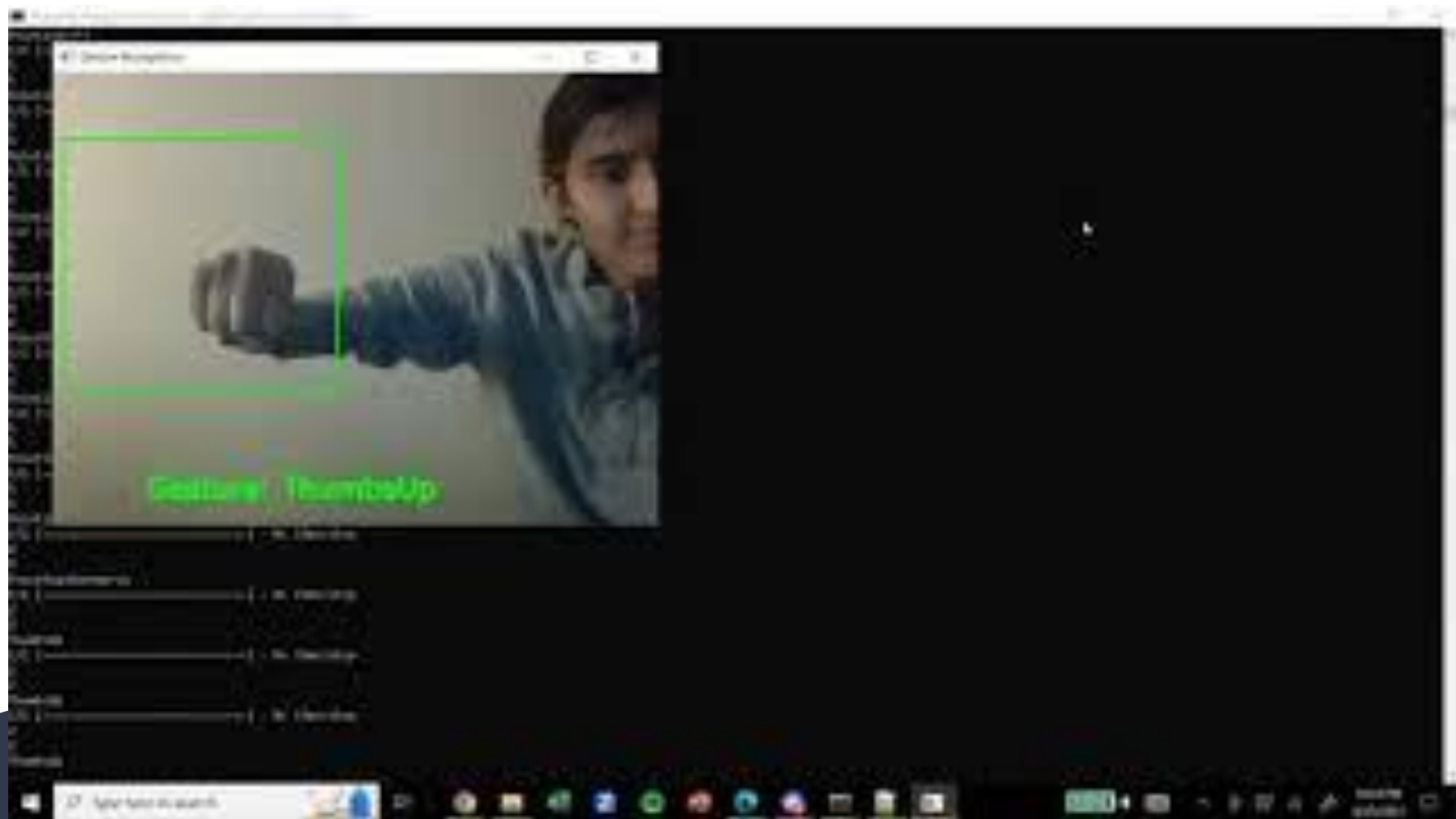


```

1 import cv2
2 import numpy as np
3 from keras.squeezenet import SqueezeNet
4 from keras.optimizers import Adam
5 from keras.utils import plot_model
6 from keras.layers import Activation, Dropout, Convolution2D, GlobalAveragePooling2D
7 import tensorflow as tf
8 import os
9
10 IMG_SAVE_PATH = 'images'
11
12 CLASS_MAP = {
13     'gesture1': 0,
14     'nothing': 1
15 }
16
17 NUM_CLASSES = len(CLASS_MAP)
18
19 def mapper(val):
20     return CLASS_MAP[val]
21
22 def get_model():
23     model = Sequential([
24         SqueezeNet(input_shape=(227, 227, 3), include_top=False),
25         Dropout(0.5),
26         Convolution2D(NUM_CLASSES, (1, 1), padding='valid'),
27         Activation('relu'),
28         GlobalAveragePooling2D(),
29         Activation('softmax')
30     ])
31     return model
32
33 # Load images from the directory
34 dataset = []
35 for directory in os.listdir(IMG_SAVE_PATH):
36     path = os.path.join(IMG_SAVE_PATH, directory)
37     if not os.path.isdir(path):
38         continue
39     for item in os.listdir(path):
40         # file size is bigger than 1000 get in our way
41         if not os.stat(path).st_size > 1000:
42             continue
43         img = cv2.imread(os.path.join(path, item))
44         img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
45         img = cv2.resize(img, (227, 227))
46         dataset.append((img, directory))
47
48 data, labels = zip(*dataset)
49 labels = list(map(mapper, labels))
50
51 # we list create the labels
52 labels = np_utils.to_categorical(labels)
53
54 # define the model
55 model = get_model()
56 model.compile(
57     optimizer=Adam(lr=0.0001),
58     loss='categorical_crossentropy',
59     metrics=['accuracy'])
60
61 # start training
62 model.fit(np.array(data), np.array(labels), epochs=15)
63
64 # save the model for later use
65 model.save('model.h5')

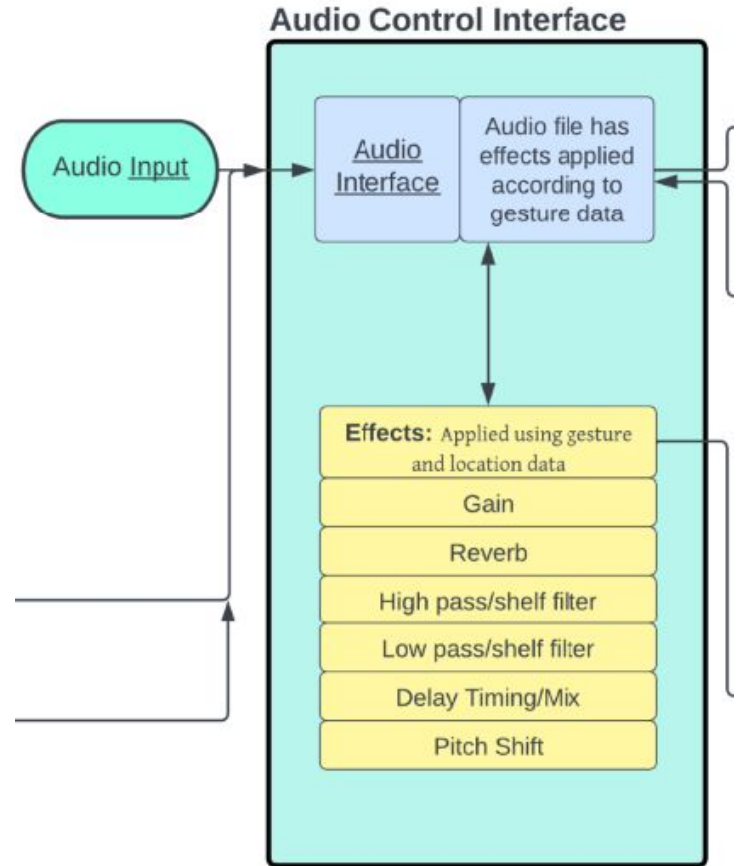
```





# Audio Interface

- Pedalboard
- Gesture control changes variables for FX
- Audio workstation file applies changes to audio file
- Audio file is saved as a new output file

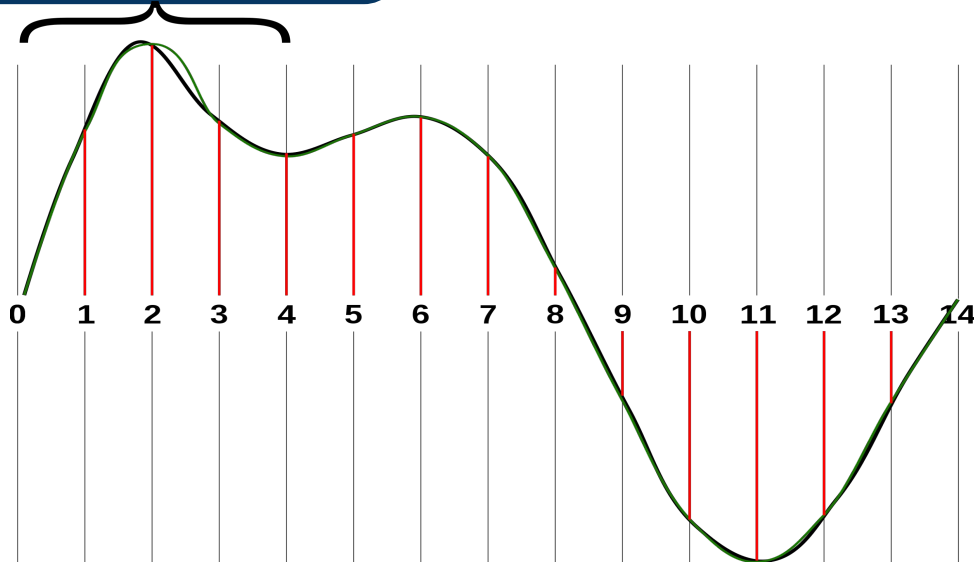


# Audio Effects - Sample Array

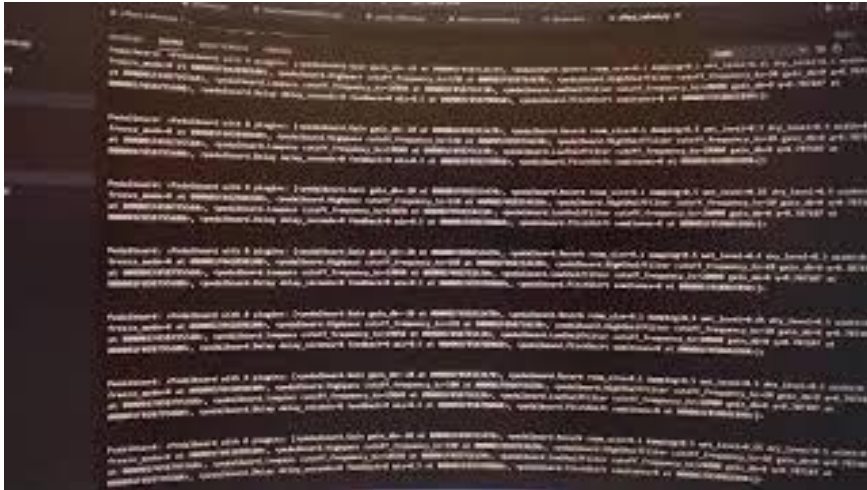
```
samples[  
  [0.001, 0.103, 0.105, 0.107, 0.109],  
  [0.012, 0.023, 0.029, 0.033, 0.043]  
]
```



```
samplesFX[  
  [0.1211, 0.1531, 0.1652, 0.1674, 0.1993],  
  [0.0124, 0.0231, 0.0291, 0.0334, 0.0433]  
]
```



# GUI and Audio Control Demo



Example audio changes



# Potential Future Changes

- AI model needs about 4000 pictures for each gesture
- Some gestures are more difficult to recognize than others
- Implement two hand recognition
- Third-party plugin support

# Gesture Controlled Audio Software

## Capstone Final Presentation

Anusha Bhattacharyya, Troy Watts, Maxx  
Smith, and Winston Phillips