# **Codes:**

## **Serial Communication**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**I. Capturing a Single Image & Mounting it onto our trained Model**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cam\_port = 0

cam = cv2.VideoCapture(cam\_port, cv2.CAP\_DSHOW)

# Check if the camera is opened successfully

if not cam.isOpened():

print("Error: Could not open camera.")

else:

# Reading the input using the camera

result, image = cam.read()

# Check if the read operation was successful

if result:

cv2.imwrite("position.jpg", image)

print("Image captured and saved as position.jpg.")

else:

print("Error: Could not read from the camera.")

# Release the camera

cam.release()

# It's also a good practice to close any OpenCV windows that may be open

cv2.destroyAllWindows()

new\_pred=new\_model.predict(tf.expand\_dims(image, axis=0))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**II. Establishing Serial Communication**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import serial

import time

SerialObj = serial.Serial('COM3')

SerialObj.baudrate = 9600

SerialObj.bytesize = 8

SerialObj.parity ='N'

SerialObj.stopbits = 1

time.sleep(3)

if new\_pred\_name == 'pos\_1':

BytesWritten = SerialObj.write(b'1')

elif new\_pred\_name == 'pos\_2':

BytesWritten = SerialObj.write(b'2')

elif new\_pred\_name == 'pos\_3':

BytesWritten = SerialObj.write(b'3')

elif new\_pred\_name == 'pos\_4':

BytesWritten = SerialObj.write(b'4')

elif new\_pred\_name == 'pos\_5':

BytesWritten = SerialObj.write(b'5')

elif new\_pred\_name == 'pos\_6':

BytesWritten = SerialObj.write(b'6')

elif new\_pred\_name == 'pos\_7':

BytesWritten = SerialObj.write(b'7')

elif new\_pred\_name == 'pos\_8':

BytesWritten = SerialObj.write(b'8')

elif new\_pred\_name == 'pos\_9':

BytesWritten = SerialObj.write(b'9')

print('BytesWritten = ', BytesWritten)

SerialObj.close()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**III. Initializing Servo for Each Part of the Arm**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <Servo.h>

Servo bassservo; // create servo object to control a servo

Servo sholservo; // create servo object to control a servo

Servo elservo; // create servo object to control a servo

void setup() {

Serial.begin(9600); // Initialize serial communication

bassservo.attach(9); // attaches the servo on pin 9 to the servo object

bassservo.write(9);

delay(1000);

sholservo.attach(10); // attaches the servo on pin 9 to the servo object

sholservo.write(60);

delay(1000);

elservo.attach(6); // attaches the servo on pin 9 to the servo object

elservo.write(85);

delay(1000);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**IV. Defining Servo Motion for Each Position**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void pos1() {

// Code for the pos1 function

Serial.println("Executing pos1");

for (int bass = 9; bass <= 151; bass += 1) { // goes from 0 degrees to 180 degrees

bassservo.write(bass); // tell servo to go to position in variable 'pos'

Serial.println(bass);

delay(150); // waits 15ms for the servo to reach the position

}

delay(2000); // pause for a second

for (int shol = 60; shol <= 74; shol += 1) { // goes from 0 degrees to 180 degrees

sholservo.write(shol); // tell servo to go to position in variable 'pos'

Serial.println(shol);

delay(150); // waits 15ms for the servo to reach the position

}

delay(2000); // pause for a second

for (int el = 85; el <= 100; el += 1) { // goes from 0 degrees to 180 degrees

elservo.write(el); // tell servo to go to position in variable 'pos'

Serial.println(el);

delay(150); // waits 15ms for the servo to reach the position

}

delay(2000); // pause for a second

for (int bass = 151; bass >= 9; bass -= 1) { // goes from 180 degrees to 0 degrees

bassservo.write(bass); // tell servo to go to position in variable 'pos'

Serial.println(bass);

delay(150); // waits 15ms for the servo to reach the position

}

delay(2000); // pause for a second

for (int shol = 74; shol >= 60; shol -= 1) { // goes from 0 degrees to 180 degrees

sholservo.write(shol); // tell servo to go to position in variable 'pos'

Serial.println(shol);

delay(150); // waits 15ms for the servo to reach the position

}

delay(2000); // pause for a second

for (int el = 100; el >= 85; el -= 1) { // goes from 0 degrees to 180 degrees

elservo.write(el); // tell servo to go to position in variable 'pos'

Serial.println(el);

delay(150); // waits 15ms for the servo to reach the position

}

}

In a similar fashion, for 8 other positions, 8 different functions have been created with necessary commands.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**V. Running the Positional Function Based on Input from Python Script**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void loop() {

if (Serial.available() > 0) {

// If there is data available in the serial buffer

char input = Serial.read(); // Read the incoming character

// Check the value of the incoming character

switch (input) {

case '1':

pos1(); // Call pos1 function if '1' is received

break;

case '2':

pos2(); // Call pos2 function if '2' is received

break;

case '3':

pos3(); // Call pos1 function if '1' is received

break;

case '4':

pos4(); // Call pos1 function if '1' is received

break;

case '5':

pos5(); // Call pos1 function if '1' is received

break;

case '6':

pos6(); // Call pos1 function if '1' is received

break;

case '7':

pos7(); // Call pos1 function if '1' is received

break;

case '8':

pos8(); // Call pos1 function if '1' is received

break;

case '9':

pos9(); // Call pos1 function if '1' is received

break;

default:

// Handle other cases if needed

break;}}}