Project Title:ESP32-CAM Microcontroller Hand Written Detection

Hardware Requirements:

- ESP32-CAM module
- Arduino IDE or PlatformIO extension for VS Code
- Python
- Tesseract OCR
- OpenCV library for Python

Python Code

The Python script reads the live stream from the ESP32-CAM, performs OCR using Tesseract, and displays the stream with extracted text using OpenCV. Snippet:

```
# Python Code Snippet

# (Refer to the full Python code in main.py)

Import cv2

Import urllib.request

Import numpy as np

Import pytesseract


# ...


While True:

    # Read live stream from ESP32-CAM

    Img_resp = urllib.request.urlopen(url)
```

```
    Imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)

    Frame = cv2.imdecode(imgnp, -1)


    # Extract text using Tesseract

    Text = pytesseract.image_to_string(frame, config='—psm 6')


    # ...


    Key = cv2.waitKey(1) & 0xFF

    If key == ord('q'):

        Break

cv2.destroyAllWindows()
```

Applications:

Document Digitization & OCR:

Real-time document scanning: Capture images of documents and instantly process them for text extraction.

Business card readers: Automatically extract information from scanned business cards.

Receipt processing: Automate the process of extracting data from receipts.

Text Recognition: Implement Optical Character Recognition (OCR) to convert handwritten text into digital format.

Text-to-speech applications: Convert handwritten text into spoken words.

Industrial Applications:

Serial number reading: Detect and read serial numbers on products.

Product label verification: Verify the authenticity of product labels.

Quality control: Automate visual inspection tasks.

Educational Tools:

Interactive learning systems: Create interactive learning experiences where students can interact with handwritten content.

Language learning aids: Help language learners practice writing and recognition.

Text-to-speech applications: Convert handwritten text into spoken words for educational purposes.

Project Title: QR Code detection Using ESP 32 Cam

Requirements:- ESP32-CAM Module, USB-to-TTL Programmer, Arduino IDE

Python Code:- #include <esp_camera.h>

#include <ESP32QRCodeReader.h>

// Camera configuration

#define PWDN_GPIO_NUM -1

#define RESET_GPIO_NUM -1

#define XCLK_GPIO_NUM 0

#define SIOD_GPIO_NUM 26

#define SIOC_GPIO_NUM 27

```
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

ESP32QRCodeReader reader(CAMERA_MODEL_AI_THINKER);

void setup() {
  Serial.begin(115200);

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
```

```
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

config.frame_size = FRAMESIZE_QVGA;
config.jpeg_quality = 12;
config.fb_count = 1;

// Initialize camera
```

```
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  reader.setup();
}

void loop() {
  reader.read();

  if (reader.hasPayload()) {
    Serial.println("QR Code Detected:");
    Serial.println(reader.getPayload().c_str());
  }

  delay(200);
}
```

Applications:- 1. Access Control / Smart Door Lock

- Scan a QR code to unlock a door or gate.
- Only valid QR codes (authorized users) trigger relays or servos.

2. Attendance System

- Students or employees scan a QR code as proof of presence.
- ESP32 sends data to a server or logs via SD card.

3. Inventory Management

- Scan QR codes on products to update or verify stock.
- ESP32 can log items to the cloud or send via Wi-Fi.

4. Wireless QR Scanner

- Use the ESP32-CAM as a portable QR scanner.
- Detected data can be sent via Bluetooth, Wi-Fi, or saved.

5. Payment Terminals / Vending Machines

- Detect payment or identity QR codes to approve transactions.
- Compact, low-power setup for standalone machines.

6. IoT Device Authentication

- Scan a QR to securely identify and authorize other IoT devices or users.

7. Library / Book Checkout

- Scan book QR codes to manage borrowing.
- ESP32 can link with a cloud-based database.

8. Event Ticket Validation

- Validate event tickets or passes using QR codes at the entrance.
- Standalone system without need for PC or large hardware.
