Smart Pet Feeder with Facial Recognition Control
Device Configuration

1. **Set up Raspberry Pi:**
    - Install the Raspberry Pi OS. (Raspberry Pi OS (Legacy) with desktop and recommended software: Debian version: 11 (bullseye) in this case)
        - Flash Raspberry Pi OS onto a microSD card.
        - Insert the microSD card into the Raspberry Pi.
    - Enable SSH and configure Wi-Fi for remote access.
    - Update and upgrade the system:
      sudo apt update && sudo apt upgrade -y
      sudo raspi-config
        - Interface Options > Camera (Legacy) > Enable.

2. **Install Dependencies:**
    - Python and required libraries (if not already installed):
      sudo apt install python3-pip
      pip3 install boto3 numpy
      pip3 install RPi.GPIO
      pip3 install AWSIoTPythonSDK
      sudo apt install libcamera-tools -y

3. **Connect Hardware**
    - **Connect Camera**
        - Attach the camera module to the CSI port on the Raspberry Pi and enable it via
          sudo raspi-config
        - Select Interface Options > Camera > Enable. (If not already done)
        - Follow Arducam software configuration. (Refer https://docs.arducam.com/Raspberry-Pi-Camera/Native-camera/8MP-IMX219/)

          sudo nano /boot/config.txt
          #Find the line: camera_auto_detect=1, update it to:
          camera_auto_detect=0
          #Find the line: [all], add the following item under it:
          dtoverlay=imx219
          #Save and reboot.

    - **Connect Servo Motor**
        - Wire the servo motor:
            1. *Power (Red) → 5V pin on Raspberry Pi.*
            2. *Ground (Brown/Black) → GND pin.*
            3. *Signal (Orange/Yellow) → GPIO 17*

4. **Set Up the Camera:**
    - Test your camera connection:
      libcamera-jpeg -o test.jpg

- Ensure the camera is enabled in raspi-config

5. **Configure MQTT Client:**
   - Use the AWS IoT Core MQTT endpoint and certificates for secure communication.
   - Copy your AWS IoT Core certificates (root-CA.crt, device.pem.key, device.pem.crt) into raspberry pi.
   - Update the MQTT configuration in your device code (e.g., endpoint, topic, and certificate paths).

6. **Run the Code:**
   - Ensure your code is executable and uses correct paths and can communicate with the cloud:
     python3 deviceCode.py

7. **Setup Automation:**
   - To run your code on boot, add it to the rc.local file.
     sudo nano /etc/rc.local
     - Add commands to execute the python program deviceCode.py
       sudo python3 deviceCode.py &