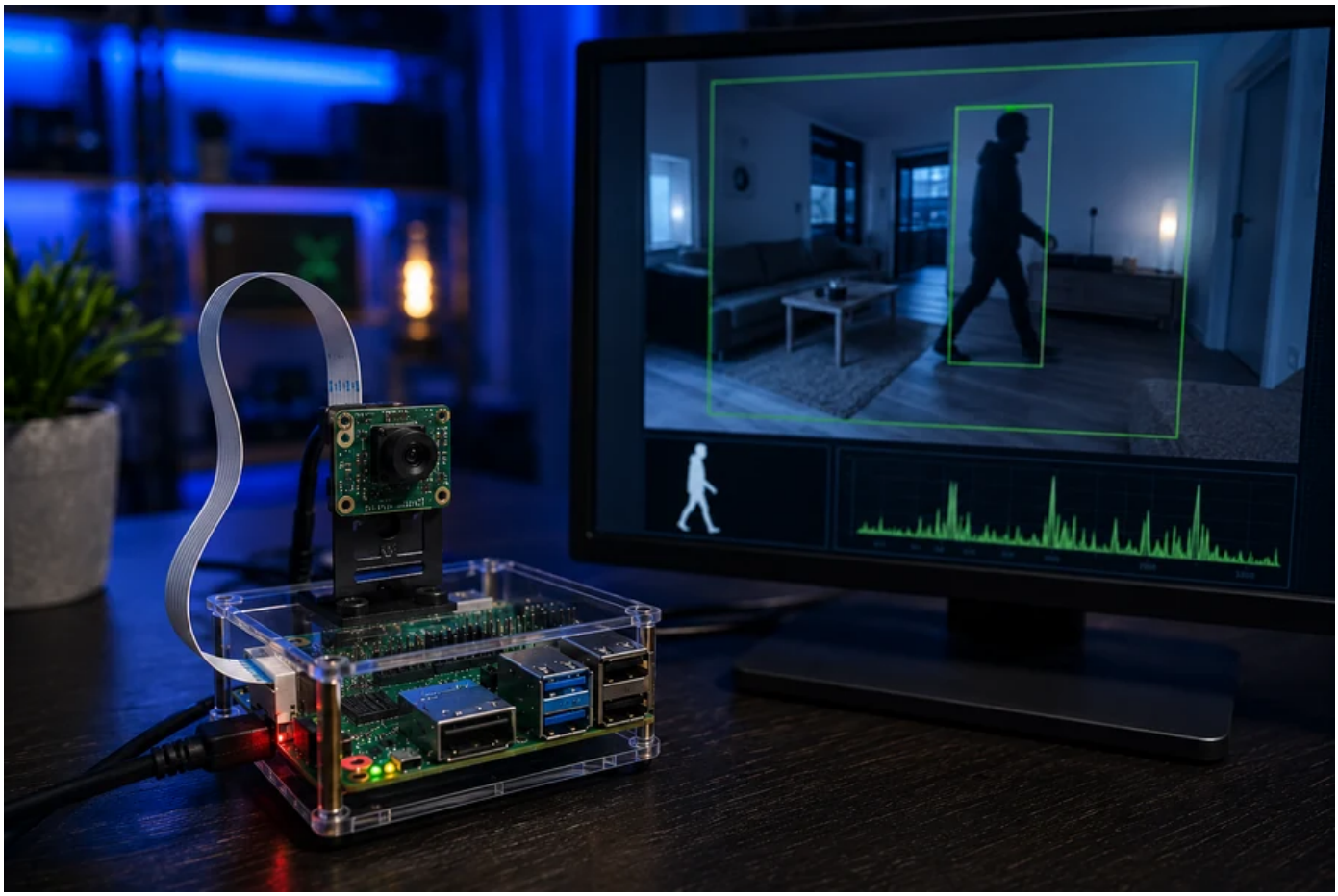
WolfieWeb Raspberry Pi OpenCV Motion Camera Lab Manual

Super detailed camera setup, OpenCV install, motion detection code, tuning, testing, and upgrade instructions.



What You Will Build

This project turns a Raspberry Pi into a motion-aware camera station. The Pi captures video frames, compares each frame against the next frame, highlights motion areas, and saves snapshots when movement crosses a threshold.

Important Build Rule

> Do not start by writing code. First confirm the camera works. A dead camera test will waste hours if you blame OpenCV for a cable, driver, permissions, or USB power problem.

Parts Needed

Part	Purpose	Notes
Raspberry Pi 4 or Raspberry Pi 5	Runs OpenCV and captures video	Pi 5 or Pi 4 with good cooling is best.
Pi Camera Module or USB webcam	Video input	USB webcam is easiest; Pi Camera is cleaner for embedded builds.

Camera mount or tripod	Keeps view stable	A shaky camera causes false motion alerts.
MicroSD card	Stores OS and snapshots	Use a quality card if saving many images.
Power supply	Keeps Pi stable	Undervoltage can break camera capture.

Camera Placement and Cable Layout

Placement Rules

Area	Best Practice	Why
Camera angle	Aim at one stable scene	Less background change means fewer false positives.
Lighting	Avoid flickering lights and windows with fast shadows	OpenCV sees lighting changes as motion.
Mount	Use a rigid stand or bracket	Small camera movement creates false motion.
Distance	Keep the target area within clear focus	Blurry footage makes contours unreliable.
Storage	Save snapshots into a dated folder	Prevents a messy capture folder.

The ZIP includes an editable SVG placement guide at images/opencv_motion_camera_placement.svg.

Install OpenCV

Update the Pi and install the basic packages:

```
sudo apt update
sudo apt install python3-pip python3-opencv -y
python3 -c "import cv2; print(cv2.__version__)"
```

If you use a USB webcam, test the camera device:

```
ls /dev/video*
v4l2-ctl --list-devices
```

If v4l2-ctl is missing, install it with sudo apt install v4l-utils -y.

If you use the Raspberry Pi Camera Module, check that the ribbon cable is seated evenly and test the camera before using OpenCV.

Pi Camera Quick Test

```
libcamera-hello
libcamera-still -o camera_test.jpg
```

Motion Detection Python Script

```
import cv2
import datetime
import os

camera = cv2.VideoCapture(0)
os.makedirs("motion_captures", exist_ok=True)

ret, frame1 = camera.read()
ret, frame2 = camera.read()

while camera.isOpened():
    difference = cv2.absdiff(frame1, frame2)
    gray = cv2.cvtColor(difference, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (5, 5), 0)
    _, threshold = cv2.threshold(blur, 25, 255, cv2.THRESH_BINARY)
    dilated = cv2.dilate(threshold, None, iterations=3)
    contours, _ = cv2.findContours(dilated, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    motion_found = False

    for contour in contours:
        if cv2.contourArea(contour) < 1800:
            continue

        motion_found = True
        x, y, w, h = cv2.boundingRect(contour)
        cv2.rectangle(frame1, (x, y), (x+w, y+h), (0, 255, 0), 2)

    if motion_found:
        filename = datetime.datetime.now().strftime("motion_captures/%Y-%m-%d_%H-%M-%S.jpg")
        cv2.imwrite(filename, frame1)
        print("Motion saved:", filename)

    cv2.imshow("WolfieWeb Motion Camera", frame1)

    frame1 = frame2
    ret, frame2 = camera.read()

    if cv2.waitKey(10) == 27:
        break

camera.release()
cv2.destroyAllWindows()
```

Save this as opencv_motion_camera.py. Run it with python3 opencv_motion_camera.py. Press Escape to close the camera window.

How the Code Works

Step	What Happens	Why
Capture frames	The script reads two frames from the camera	Motion is found by comparing frame differences.
Convert to grayscale	Color is removed	This reduces processing load.
Blur image	Small noise is softened	Prevents tiny pixel changes from triggering alerts.
Threshold	Bright difference areas are isolated	This marks changed areas.
Find contours	OpenCV detects moving blobs	Large blobs become motion events.
Save snapshot	A JPG file is saved when motion is found	You get evidence without recording all day.

Tuning and Troubleshooting

Problem	Likely Cause	Fix
Too many false alerts	Threshold or contour area too low	Raise cv2.threshold value or contourArea minimum.
No motion detected	Contour area too high or camera not reading	Lower contourArea and print camera read status.
Camera opens black	Wrong camera index or blocked camera	Try VideoCapture(1) or test /dev/video devices.
Program is slow	Resolution too high	Lower camera resolution or use Pi 5 cooling.
Files pile up quickly	Every motion frame is being saved	Add cooldown timing between saved snapshots.

Recommended Starting Values

Threshold: 25. Contour area: 1800. Delay: 10 ms. These are safe starting points, not magic numbers. Tune them for the room, camera quality, lighting, and distance.

Upgrade Path

After the camera works, connect it to the rest of the advanced Raspberry Pi page. Send motion alerts over MQTT, save motion events into SQLite, show the latest capture on the Flask dashboard, or move the Servo Pan-Tilt mount toward the detected motion area.