

## INF2009\_Setup

**Q1.** What is the purpose of the `fswebcam` command in the Raspberry Pi setup?

**A1.** Captures images from a webcam. `fswebcam -r 1280x720 --no-banner image.jpg` takes a 1280x720 image without a banner.

---

**Q2.** How can you set up a static IP address on a Raspberry Pi?

**A2.** Edit `/etc/dhcpd.conf`, add `static ip_address`, `routers`, and `domain_name_servers`, then reboot.

---

**Q3.** Why is enabling VNC useful on the Raspberry Pi?

**A3.** It allows remote desktop access to control the Pi's GUI from another device.

---

## INF2009\_SoundAnalytics

**Q4.** How do you test if a USB microphone works on Raspberry Pi?

**A4.** Use `arecord --duration=10 test.wav` and `aplay test.wav`.

---

**Q5.** Why is a virtual environment recommended in this project?

**A5.** It isolates dependencies and avoids package conflicts.

---

**Q6.** Name three Python libraries used for sound processing.

**A6.** `pyaudio`, `sounddevice`, `scipy`.

---

**Q7.** What does the Fourier Transform do in sound analysis?

**A7.** Converts time-domain audio into frequency components.

---

**Q8.** Why would you use a bandpass filter in audio analytics?

**A8.** To isolate specific frequency ranges and remove noise.

---

**Q9.** What is `librosa` used for?

**A9.** Audio feature extraction, e.g., spectrograms, MFCC.

---

## INF2009\_ImageAnalytics

**Q10.** What is Edge Computer Vision (ECV)?

**A10.** Performing image analytics locally on edge devices.

---

**Q11.** Why is ECV beneficial?

**A11.** Enables real-time insights, improves privacy, and reduces reliance on cloud.

---

**Q12.** How does Raspberry Pi interface with a USB webcam?

**A12.** Using tools like `fswebcam` to capture image data for processing.

---

**Q13.** What is OpenCV's role in this project?

**A13.** Loading images, applying filters, edge detection, and visualization.

---

**Q14.** What is Canny edge detection used for?

**A14.** To detect object outlines in images.

---

**Q15.** How are images visualized on the Raspberry Pi?

**A15.** Using `matplotlib` or `cv2.imshow()` from OpenCV.

---

## INF2009\_VideoAnalytics

**Q16.** What does `cv2.VideoCapture(0)` do?

**A16.** Opens the webcam and starts video capture.

---

**Q17.** How does frame differencing detect motion?

**A17.** By comparing current and previous frames for pixel changes.

---

**Q18.** Why are only keyframes processed in edge video analytics?

**A18.** To reduce bandwidth and computation.

---

**Q19.** What are typical use cases for video analytics on the edge?

**A19.** Real-time surveillance, traffic detection, gesture-based interfaces.

---

**Q20.** What challenge does edge video analytics face?

**A20.** Limited processing power, need for real-time inference.

---

## **INF2009\_DLonEdge**

**Q21.** What is model quantization?

**A21.** Reducing model weights/activations to lower precision (e.g., 32-bit → 8-bit) for faster inference.

---

**Q22.** Why is quantization useful for edge deployment?

**A22.** Reduces model size and speeds up processing.

---

**Q23.** What is pruning in neural networks?

**A23.** Removing unnecessary weights to make models more efficient.

---

**Q24.** What is TensorFlow Lite (TFLite)?

**A24.** A lightweight ML runtime optimized for mobile and edge devices.

---

**Q25.** What is a limitation of running DL on Raspberry Pi?

**A25.** Limited RAM/CPU/GPU, requiring model optimization.

---

## **INF2009\_MQTT**

**Q26.** What is a topic in MQTT?

**A26.** A hierarchical string used to categorize published/subscribed messages (e.g., `sensor/audio`).

---

**Q27.** What are MQTT QoS levels?

**A27.**

- 0: At most once
  - 1: At least once
  - 2: Exactly once
- 

**Q28.** What does the `publish_message` function do?

**A28.** Connects to MQTT broker and sends messages to a topic.

---

## **INF2009\_AWSIoTCore**

**Q29.** What files are required to connect a device to AWS IoT Core?

**A29.** Certificate, private key, and Amazon Root CA.

---

**Q30.** What is a device shadow in AWS IoT?

**A30.** A JSON doc storing the device's last known and desired state.