Lab Instruction Links:

No.	Link
1	https://github.com/drfuzzi/INF2009_Setup/tree/main
2	https://github.com/drfuzzi/INF2009_SoundAnalytics
3	https://github.com/drfuzzi/INF2009_ImageAnalytics
4	https://github.com/drfuzzi/INF2009_VideoAnalytics
5	https://github.com/drfuzzi/INF2009_DLonEdge
6	https://github.com/drfuzzi/INF2009_MQTT
7	https://github.com/drfuzzi/INF2009_AWSIoTCore

INF2009_Setup: This repository provides instructions for configuring a Raspberry Pi 400 with a webcam, including steps for installing the Raspberry Pi OS, enabling remote access via SSH and VNC, and setting up a static IP address. It also guides users on installing software to capture images and record videos using the webcam. GitHub

Libraries Used: The setup involves using command-line tools like `fswebcam` for capturing images and `ffmpeg` for recording videos. Additionally, `vlc` is recommended for video playback.GitHub

INF2009_SoundAnalytics: This repository focuses on setting up a USB microphone with the Raspberry Pi to capture audio and perform basic sound analytics. It covers installing necessary Python libraries, capturing audio using Python, and visualizing sound waves and their frequency components. GitHub

Libraries Used: The project utilizes Python libraries such as `pyaudio` or `sounddevice` for audio capture, and `scipy` and `matplotlib` for audio analysis and visualization.GitHub

INF2009_ImageAnalytics: This repository guides users on setting up a webcam with the Raspberry Pi to capture images and perform both basic and advanced image analytics. It includes instructions for installing necessary Python libraries and conducting image processing tasks. GitHub

Libraries Used: The project employs Python libraries like `opencv` for image processing tasks.

INF2009_VideoAnalytics: This repository provides guidance on setting up a webcam with the Raspberry Pi to capture video streams and perform video analytics. It covers installing necessary Python libraries and conducting video processing tasks.

Libraries Used: The project utilizes Python libraries such as `opencv` for video processing tasks.

INF2009_DLonEdge: This repository focuses on running deep learning models on edge devices like the Raspberry Pi. It discusses model optimization techniques, including quantization, to enable efficient execution of complex models on hardware-constrained devices.

Libraries Used: The project involves deep learning frameworks such as TensorFlow or PyTorch, along with tools for model optimization and quantization.

INF2009_MQTT: This repository introduces the MQTT protocol for IoT communications. It guides users on installing and configuring an MQTT broker on the Raspberry Pi, and creating publisher and subscriber clients to test communication between devices.

Libraries Used: The project utilizes the `paho-mqtt` Python library for implementing MQTT clients.

INF2009_AWSIoTCore: This repository provides an introduction to AWS IoT Core, demonstrating how to create an IoT device (e.g., Raspberry Pi), send real-time data securely to AWS IoT Core over the MQTT protocol, and ingest device data into AWS DynamoDB via IoT rules.

Libraries Used: The project employs the `AWS IoT SDK` for Python to facilitate communication with AWS services.

🔧 1. INF2009_Setup (Raspberry Pi Setup)

- **Commands**:
  - Set static IP: `sudo nano /etc/dhcpcd.conf`
  - Enable SSH: `sudo raspi-config`
- **Camera Commands**:
  - Capture image: `fswebcam image.jpg`
  - Record video: `ffmpeg -f v4l2 -i /dev/video0 output.mp4`

🎤 2. INF2009_SoundAnalytics

- **Python Libraries**: `sounddevice`, `scipy`, `numpy`, `matplotlib`

Code Snippets:

python
CopyEdit

```
import sounddevice as sd
audio = sd.rec(int(5 * 44100), samplerate=44100, channels=1)
sd.wait()
```

-

Plotting waveform:

python
CopyEdit

```
from scipy.io.wavfile import write
write("output.wav", 44100, audio)
```

📸 3. INF2009_ImageAnalytics

- **Library**: `opencv-python`

Image Operations:

python
CopyEdit

```
import cv2
img = cv2.imread('image.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(img, 100, 200)
cv2.imshow('Edges', edges)
cv2.waitKey(0)
```

🎥 4. INF2009_VideoAnalytics

Capture Video:

python
CopyEdit

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    cv2.imshow('Frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

-
- **Motion Detection Tip**: Compare pixel differences between frames

---

🤖 5. INF2009_DLonEdge

- **Concepts**: Quantization, TensorFlow Lite, running models on edge
- **Workflow**:

    1. Train model

Convert to `.tflite` using:

python
CopyEdit

```
converter = tf.lite.TFLiteConverter.from_saved_model(saved_model_dir)
tflite_model = converter.convert()
```

2.
3. Run with TFLite Interpreter on Pi

📡 6. INF2009_MQTT

- **MQTT Basics**:
  - Broker = middleman (e.g., Mosquitto)
  - `Publisher` sends → `Topic` → `Subscriber` receives
- **Common Commands**:
  - Install broker: `sudo apt install mosquitto`

Python MQTT (paho-mqtt):

python
CopyEdit

```
import paho.mqtt.client as mqtt
client = mqtt.Client()
client.connect("broker.hivemq.com", 1883, 60)
client.publish("topic/test", "Hello")
```

☁️ 7. INF2009_AWSIoTCore

- **Key Concepts**:
  - Certificates & Policies
  - MQTT topics for communication
  - IoT Rules → DynamoDB / Lambda
- **Connection Tip**:
  - Use AWS IoT Device SDK for Python
  - Use `.pem` certificates for secure connection

Publish Message:

python
CopyEdit

```
myAWSIoTMQTTClient.publish("topic", json.dumps(payload), 1)
```

📘 INF2009 Lab Quiz Glossary

🔊 Sound/Image/Video Terms

- Sampling Rate – Number of samples per second in audio (e.g., 44100 Hz).
- Waveform – A visual representation of the sound signal.
- Grayscale – Image with only shades of gray (no color).
- Canny Edge Detection – An algorithm to find edges in an image.
- Frame – A single still image from a video.
- FPS (Frames Per Second) – Number of frames shown per second in a video.

🤖 Deep Learning on Edge

- Inference – Running a trained model to make predictions.
- Quantization – Reducing model size by converting weights (e.g., from float32 to int8).
- TensorFlow Lite (TFLite) – Lightweight version of TensorFlow for mobile/edge devices.
- Interpreter – Executes `.tflite` models on devices like Raspberry Pi.

📡 MQTT and IoT

- Broker – The server that manages messages between devices (e.g., Mosquitto).
- Client – A device or app that connects to the MQTT broker.
- Topic – A message channel (like a folder); clients publish/subscribe to topics.
- QoS (Quality of Service) – MQTT setting for how messages are delivered (0, 1, 2).
- Payload – The actual data/message being sent over MQTT.
- Publish – Send data to a topic.
- Subscribe – Listen for data on a topic.

☁️ AWS IoT

- Thing – A registered device in AWS IoT.
- Certificate – Used to authenticate the device securely.
- Policy – Sets permissions for what a device can do.
- Rule – A condition in AWS IoT to forward data to other services (e.g., DynamoDB).
- Shadow – A virtual representation of the current state of a device.

🛠️ Raspberry Pi Setup

- SSH – Secure remote terminal access to the Pi.
- VNC – Remote desktop access to the Pi.
- Static IP – A fixed IP address for your Pi so it doesn’t change on reboot.
