P. Arul Rino Fernando

950321104012@gracecoe.org

Python script for the IoT devices to send real-time traffic data to the traffic information platform:

1.IoT Device (Traffic Data Simulator):

```
# IoT Device (Traffic Data Simulator) script
import paho.mqtt.client as mqtt
import json
import time
import random

# Define your MQTT broker information
broker_address = "mqtt.yourbroker.com"  # Update with your broker's address
port = 1883  # Default MQTT port
topic = "traffic_data"  # MQTT topic to publish data

# Create an MQTT client
client = mqtt.Client("TrafficSimulator")

# Callback when the client connects to the broker
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to MQTT broker")
    else:
        print(f"Connection failed with code {rc}")

# Set the callback function
client.on_connect = on_connect

# Connect to the broker
client.connect(broker_address, port, 60)

# Simulate and send traffic data
```

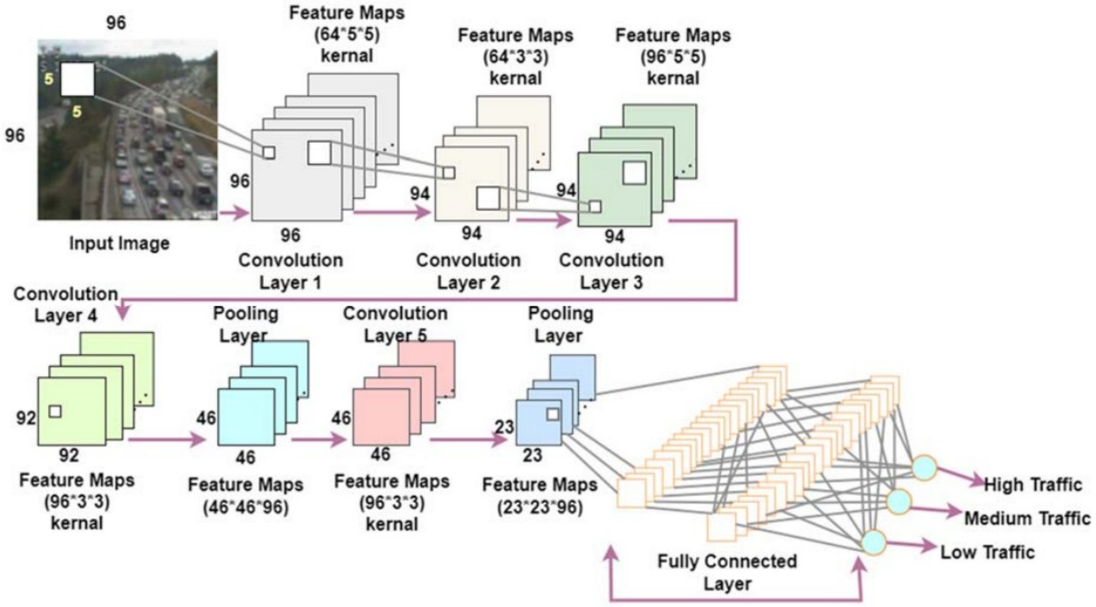
```
while True:
    # Generate synthetic traffic data
    traffic_data = {
        "location": "Intersection A",
        "vehicle_count": random.randint(0, 100),  # Simulated vehicle count
        "average_speed": random.uniform(20.0, 60.0),  # Simulated average speed in km/h
        "timestamp": int(time.time())
    }

    # Convert traffic data to JSON
    traffic_data_json = json.dumps(traffic_data)

    # Publish traffic data to the MQTT topic
    client.publish(topic, traffic_data_json)

    # Print a confirmation message
    print(f"Published traffic data: {traffic_data_json}")

    # Wait for some time (e.g., 10 seconds) before sending the next data
    time.sleep(10)

# Keep the script running
client.loop_forever()
```

2. Traffic Information Platform (Data Receiver):

```
# Traffic Information Platform (Data Receiver) script
import paho.mqtt.client as mqtt
import json

# Define your MQTT broker information
broker_address = "mqtt.yourbroker.com"  # Update with your broker's address
port = 1883  # Default MQTT port
topic = "traffic_data"  # MQTT topic to subscribe

# Create an MQTT client
```

```
client = mqtt.Client("TrafficDataReceiver")

# Callback when the client connects to the broker
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to MQTT broker")
        # Subscribe to the traffic data topic
        client.subscribe(topic)
    else:
        print(f"Connection failed with code {rc}")

# Callback when a message is received
def on_message(client, userdata, msg):
    # Process the received traffic data
    traffic_data = json.loads(msg.payload.decode())
    print("Received traffic data:")
    print(json.dumps(traffic_data, indent=4))

# Set the callback functions
client.on_connect = on_connect
client.on_message = on_message

# Connect to the broker
client.connect(broker_address, port, 60)

# Keep the script running to receive data
client.loop_forever()
```

Example:


96
96
5
5
Input Image
Feature Maps
(64*5*5)
kernal
96
96
Convolution
Layer 1
Feature Maps
(64*3*3)
kernal
94
94
Convolution
Layer 2
Feature Maps
(96*5*5)
kernal
94
94
Convolution
Layer 3
Convolution
Layer 4
92
92
Feature Maps
(96*3*3)
kernal
Pooling
Layer
46
46
Feature Maps
(46*46*96)
Convolution
Layer 5
46
46
Feature Maps
(96*3*3)
kernal
Pooling
Layer
23
23
Feature Maps
(23*23*96)
Fully Connected
Layer
High Traffic
Medium Traffic
Low Traffic