KNOWLEDGE INSTITUTE OF TECHNOLOGY

FUTURE READY TALENT

Microsoft internship

PROJECT REPORT

Submitted by

SANJIV KUMAR M

THAMARAIKANNAN V

VINOTHKUMAR R

SIVA SANJAY A

TABLE OF CONTENTS

Abstract

Street quality identification is a critical task in urban planning, transportation management, and infrastructure maintenance. This abstract presents an overview of street quality identification techniques and their applications. The quality of a street can impact various factors, including traffic congestion, safety, and the overall liability of a city. Traditional methods of street quality assessment often rely on manual inspections and subjective judgments, which are time-consuming and prone to bias.

In recent years, with the advent of advanced technologies such as remote sensing, machine learning, and computer vision, automated street quality identification approaches have emerged. These methods leverage various data sources, including street images, sensor data, and geographical information systems (GIS), to analyze and evaluate the condition of streets objectively and efficiently.

Machine learning algorithms, such as convolutional neural networks (CNNs) and support vector machines (SVMs), have been employed to classify street images based on visual features and texture analysis. Additionally, sensor data from vehicles or fixed sensors can provide valuable information about road roughness, potholes, and other defects. Integration of these diverse data sources enables a comprehensive understanding of street quality.

The applications of street quality identification are manifold. Urban planners can utilize the identified street quality information to prioritize maintenance efforts, allocate resources effectively, and plan future infrastructure projects. Transportation authorities can optimize route planning, reduce congestion, and enhance road safety. Furthermore, citizens can benefit from improved road conditions, leading to smoother rides, reduced vehicle wear and tear, and increased overall satisfaction with the transportation network.

However, there are challenges in implementing street quality identification systems at scale. These include data collection and processing, algorithm accuracy and robustness, privacy concerns, and cost considerations. Future research should focus on addressing these challenges and further refining the techniques to enable widespread adoption of street quality identification in real-world scenarios.

In conclusion, street quality identification plays a vital role in urban planning and transportation management. The combination of advanced technologies and data analysis techniques offers promising avenues for objective and efficient assessment of street quality. By leveraging these approaches, cities can make informed decisions, improve transportation infrastructure, and enhance the overall quality of life for their residents.

1. INTRODUCTION

1.1 Project Overview

Street quality identification is a crucial aspect of urban development and transportation management. The condition of streets directly affects various factors, including traffic flow, safety, and overall user experience. Traditionally, street quality assessment has relied on manual inspections, which are time-consuming, subjective, and often limited in scope. However, with advancements in technology, the development of automated street quality identification techniques has gained significant attention.

Automated street quality identification involves leveraging advanced technologies such as remote sensing, machine learning, and computer vision to objectively evaluate the condition of streets. These techniques utilize a range of data sources, including street images, sensor data, and geographical information systems (GIS), to analyze and assess street quality in a more efficient and accurate manner.

Visual analysis plays a crucial role in street quality identification. Machine learning algorithms, such as convolutional neural networks (CNNs), can be trained on large datasets of street images to recognize patterns and features associated with different levels of street quality. Texture analysis techniques can also be applied to extract valuable information from street images, aiding in the identification of cracks, potholes, or other surface defects.

In addition to visual data, sensor data collected from vehicles or fixed sensors can provide valuable insights into street conditions. This data includes measurements related to road roughness, vibrations, and other physical parameters that reflect the quality of the street surface. Integrating sensor data with other sources allows for a more comprehensive understanding of street quality, enabling better decision-making for maintenance and improvement efforts.

The applications of street quality identification are diverse and impactful. Urban planners can utilize the identified street quality information to prioritize maintenance projects, allocate resources efficiently, and plan future infrastructure developments. Transportation authorities can leverage this data to optimize route planning, reduce traffic congestion, and improve road safety. Additionally, citizens can benefit from improved road conditions, leading

to smoother rides, reduced vehicle maintenance costs, and increased satisfaction with the transportation network.

However, implementing street quality identification systems at scale presents several challenges. Data collection and processing, algorithm accuracy and robustness, privacy concerns, and cost considerations are some of the key factors that need to be addressed. Despite these challenges, ongoing research and technological advancements continue to drive progress in the field, making street quality identification an increasingly feasible and valuable tool for urban planning and transportation management.

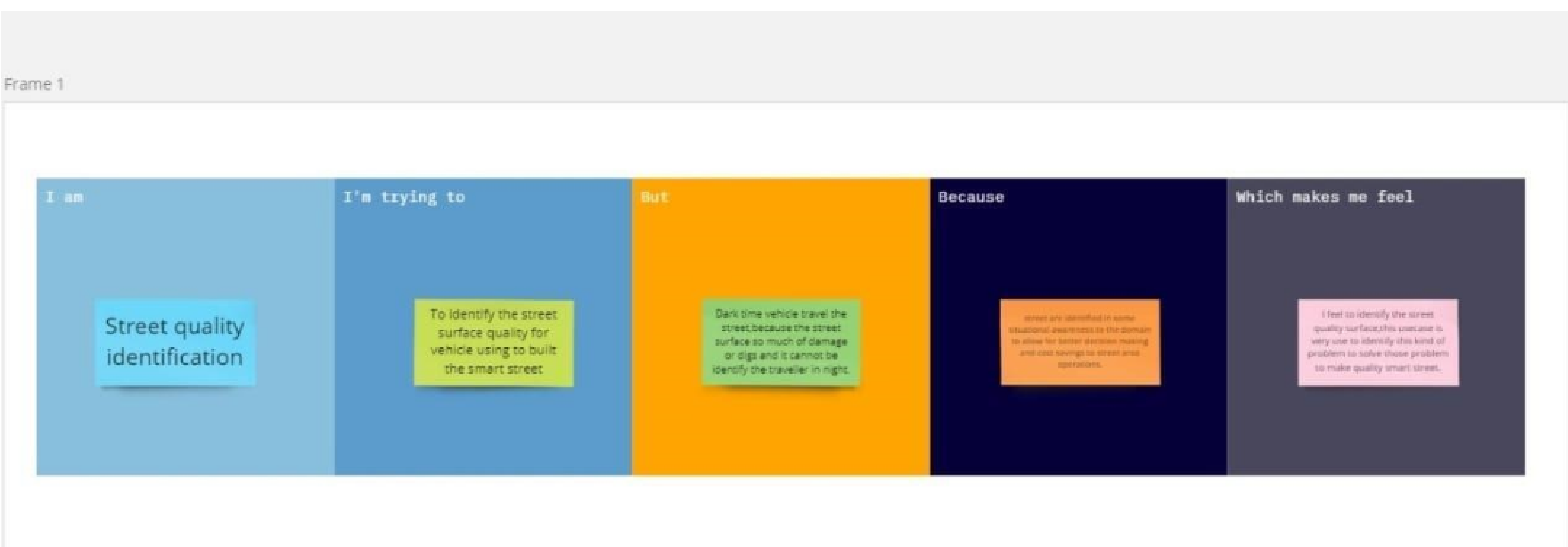
1.2 Purpose

Street quality identification helps local authorities and transportation departments prioritize their maintenance efforts effectively. By identifying streets with poor quality, such as those with potholes, cracks, or significant deterioration, resources can be allocated efficiently to address these areas first. This approach ensures that limited resources are utilized where they are most needed, improving the overall condition of the road network.

Road Safety Improvement: Street quality directly affects road safety. Roads with potholes, uneven surfaces, or other defects can lead to accidents, vehicle damage, and discomfort for drivers and passengers. By identifying and addressing these issues promptly, the risk of accidents and injuries can be reduced, making the streets safer for all us users.

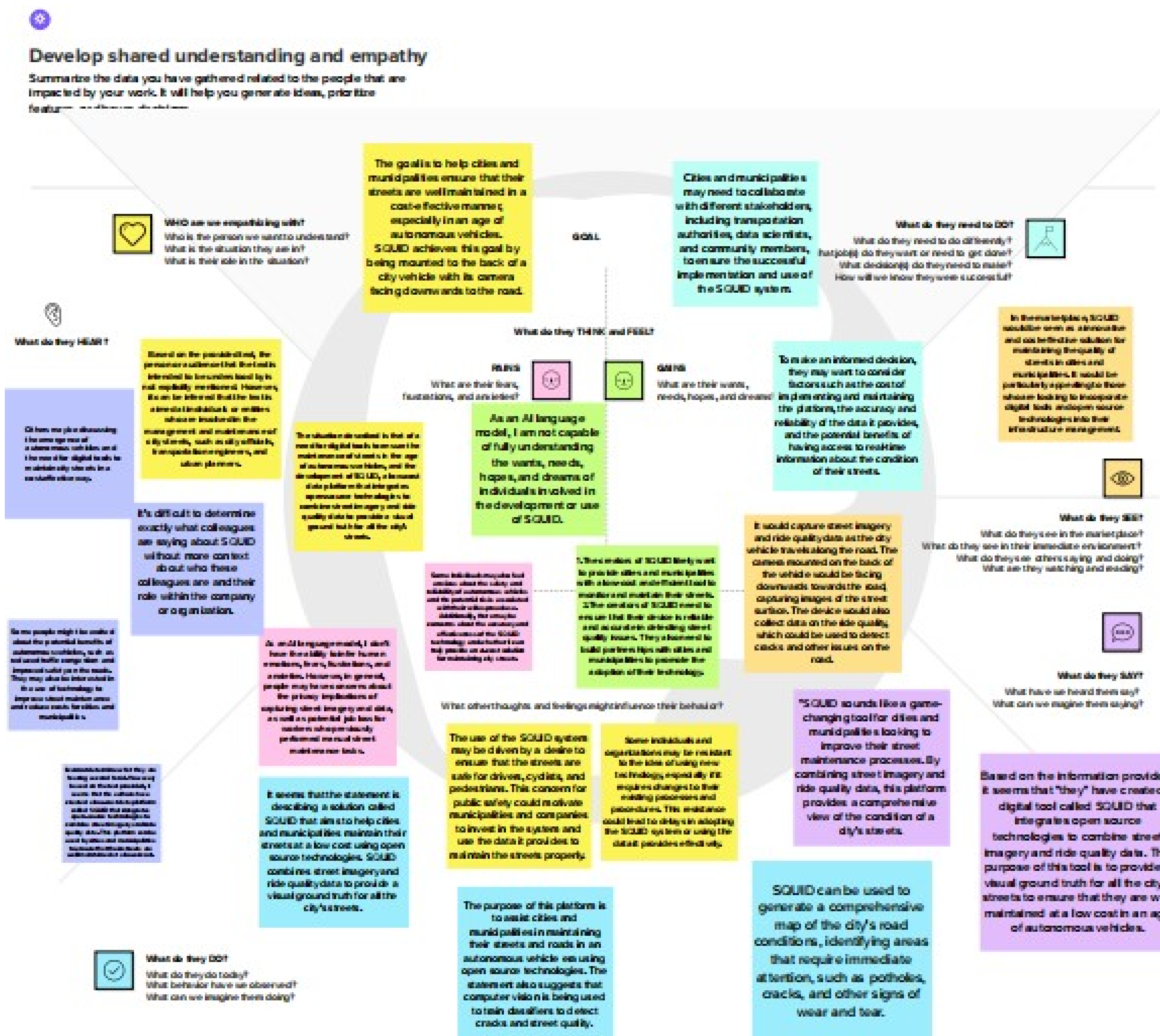
2.1 Problem statement definition

Street quality identification is critical challenge that needs to be addressed in order to ensure safe and efficient transportation infrastructure. The current problem revolves around the lack of an accurate, efficient, and comprehensive system to identify and assess the quality of streets.Street quality issues can cause disruptions in traffic flow, leading to congestion and delays. Potholes, for instance, can force vehicles to slow down or change lanes abruptly, affecting the overall efficiency of transportation networks.Lack of Accountability: In the absence of a reliable street quality identification system, it becomes difficult to hold responsible parties accountable for maintaining and repairing the streets.



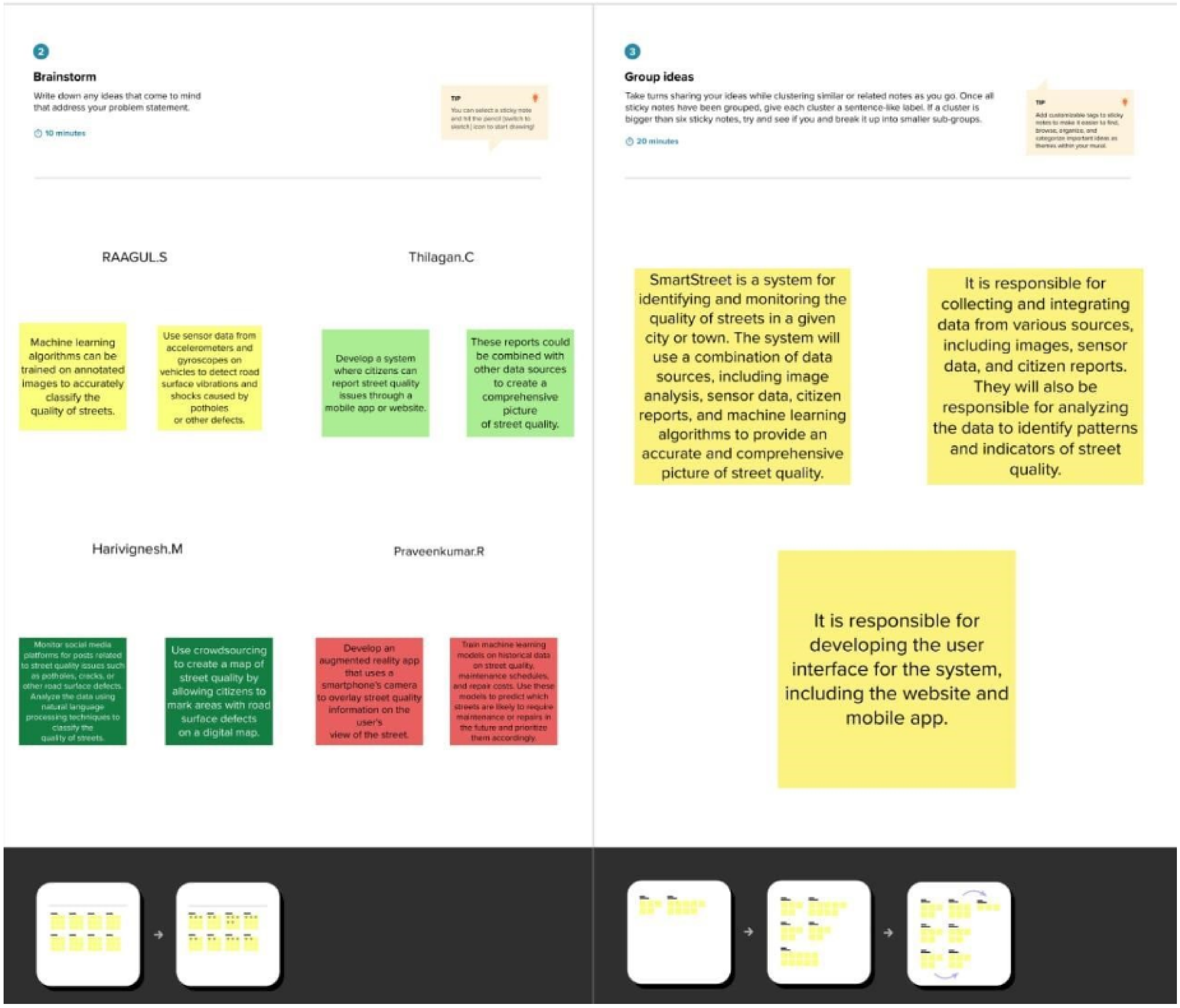
2.2 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community. Have the team members speak about the sticky notes as they place them on the empathy map. Ask questions to reach deeper insights so that they can be elaborated for the rest of the team. To help bring the user to life, you may even wish to sketch out the characteristics this person may have on the center of the face.

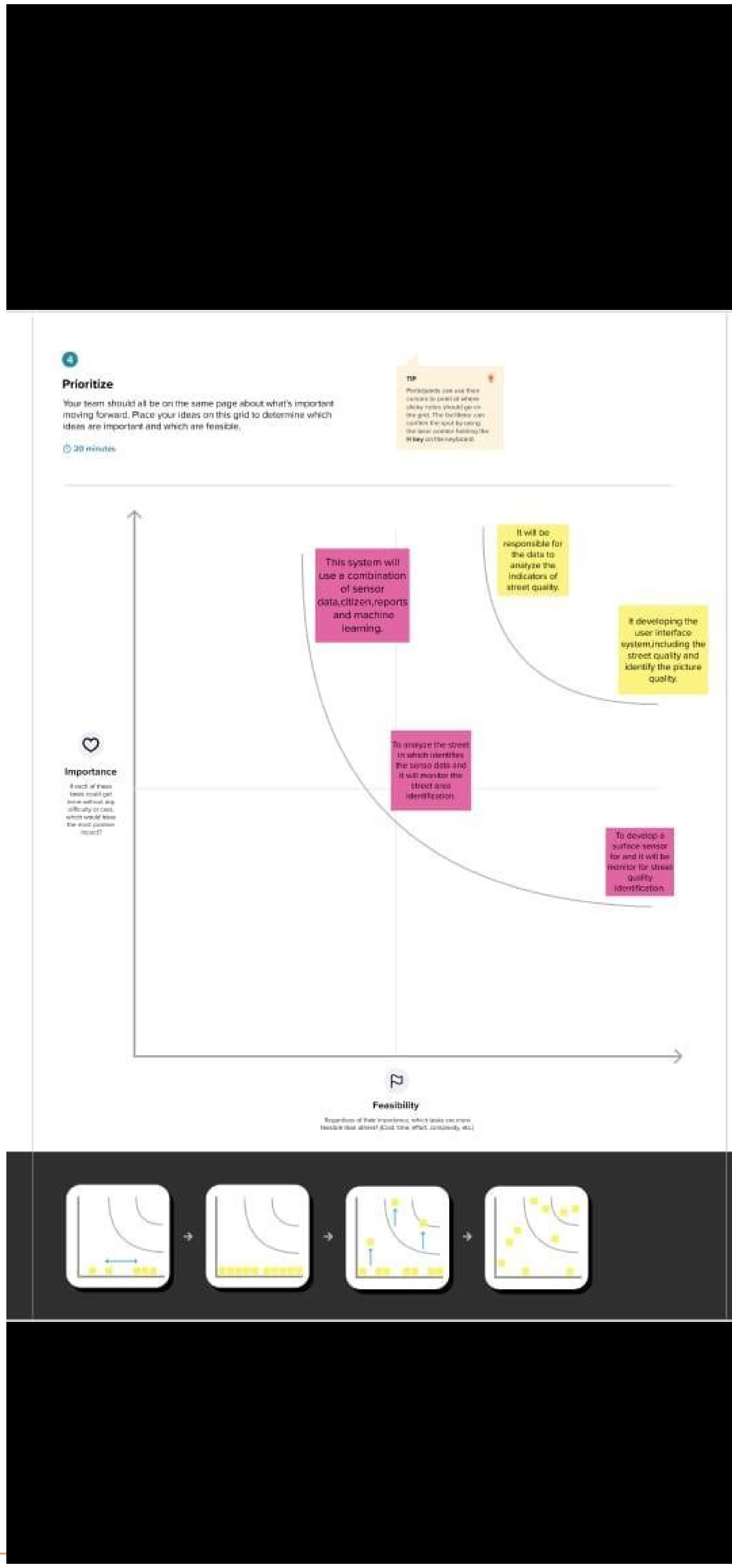


2.3 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!
RAAGUL.S
Machine learning algorithms can be trained on annotated images to accurately classify the quality of streets.
Use sensor data from accelerometers and gyroscopes on vehicles to detect road surface vibrations and shocks caused by potholes or other defects.
Thilagan.C
Develop a system where citizens can report street quality issues through a mobile app or website.
These reports could be combined with other data sources to create a comprehensive picture of street quality.
Harivignesh.M
Monitor social media platforms for posts related to street quality issues such as potholes, cracks, or other road surface defects. Analyze the data using natural language processing techniques to classify the quality of streets.
Use crowdsourcing to create a map of street quality by allowing citizens to mark areas with road surface defects on a digital map.
Praveenkumar.R
Develop an augmented reality app that uses a smartphone's camera to overlay street quality information on the user's view of the street.
Train machine learning models on historical data on street quality, maintenance schedules, and repair costs. Use these models to predict which streets are likely to require maintenance or repairs in the future and prioritize them accordingly.
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
SmartStreet is a system for identifying and monitoring the quality of streets in a given city or town. The system will use a combination of data sources, including image analysis, sensor data, citizen reports, and machine learning algorithms to provide an accurate and comprehensive picture of street quality.
It is responsible for collecting and integrating data from various sources, including images, sensor data, and citizen reports. They will also be responsible for analyzing the data to identify patterns and indicators of street quality.
It is responsible for developing the user interface for the system, including the website and mobile app.

SQUID – Street quality identification



2.4 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	The SQUID device is mounted to the back of a city vehicle with its camera facing downwards to the road. The device uses computer vision to detect cracks and street quality, and different classifiers can be trained for this purpose. The goal of SQUID is to provide a cost-effective and efficient way for cities and municipalities to maintain their streets, ensuring that they are safe for both autonomous and non-autonomous vehicles.
2.	Idea / Solution description	SQUID is a low-cost data platform that utilizes open source technologies to help cities and municipalities maintain their streets at an optimal level. The solution involves mounting a device to the back of a city vehicle with its camera facing downwards to the road. The device captures street imagery and ride quality data, which is used to provide a visual ground truth for all the city's streets.To detect cracks and street quality, SQUID utilizes computer vision techniques.

3.	Novelty / Uniqueness	SQUID's uniqueness lies in its combination of low cost data collection and open source technologies to provide a comprehensive and visual ground truth of a city's streets.Finally, by using computer vision techniques to detect cracks and other road defects, SQUID can help identify areas that require maintenance, allowing for proactive repairs that can save money and prevent accidents.
4.	Social Impact / Customer Satisfaction	SQUID can identify areas that need maintenance and repair more quickly and accurately, allowing for more efficient allocation of resources and reducing the likelihood of accidents and injuries caused by poor road conditions. Secondly, SQUID can contribute to improving the overall quality of life for people living in urban areas. Poor road conditions can lead to increased traffic congestion, longer commute times, and higher vehicle maintenance costs.

5.	Business Model (Revenue Model)	The device is mounted on the back of a city vehicle and captures street imagery and ride quality data. The data is then processed using computer vision techniques to detect cracks and other issues with street quality.SQUID is likely a product-based business model. The company creates and sells the SQUID device to cities and municipalities, who then use the device to gather data on street quality and maintenance. The company may also offer maintenance and support services for the device.
6.	Scalability of the Solution	SQUID has the potential for scalability due to its low-cost design and use of open-source technologies. The device can be easily mounted onto any city vehicle, allowing for efficient data collection across all areas of the city.The scalability of SQUID ultimately provides a sustainable and scalable solution for maintaining city streets in an age of autonomous vehicles.

3. Requirement Analysis

Functional Requirements

Following are the functional requirements of the proposed solution.

Fr No.	Functional Requirement (Epic)	Sub Requirement(Story/Sub-Task)
FR-1	Image Capture	SQUID should be able to capture images of the road surface while the vehicle is in motion.
FR-2	Image processing	The images captured by SQUID should be processed using computer vision algorithms to identify cracks, potholes, and other road surface defects.
FR-3	Data Integration	The data collected by SQUID should be integrated with other data sources, such as GPS and weather data, to provide a comprehensive view of street conditions.
FR-4	Data analysis	SQUID should be able to analyze the data collected and provide insights on the condition of the city's streets.

FR-5	User Interface	SQUID should have a user-friendly interface that allows users to easily access and analyze the data collected.
FR-6	Maintenance	SQUID should be easy to maintain, with regular calibration and cleaning to ensure accurate and reliable data collection.

4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NF R-1	Usability	The system should be able to accurately identify different types of street defects and their severity levels. The accuracy should be high enough to enable effective decision-making by the city officials.
NF R-2	Security	The system should be able to quickly process large amounts of data in real-time to provide upto-date information on the street conditions
NF R-3	Reliability	The system should be able to handle large volumes of data as the city grows, without compromising its performance.
NF R-4	Performance	The system should be able to quickly process large amounts of data in real-time to provide up-to-date

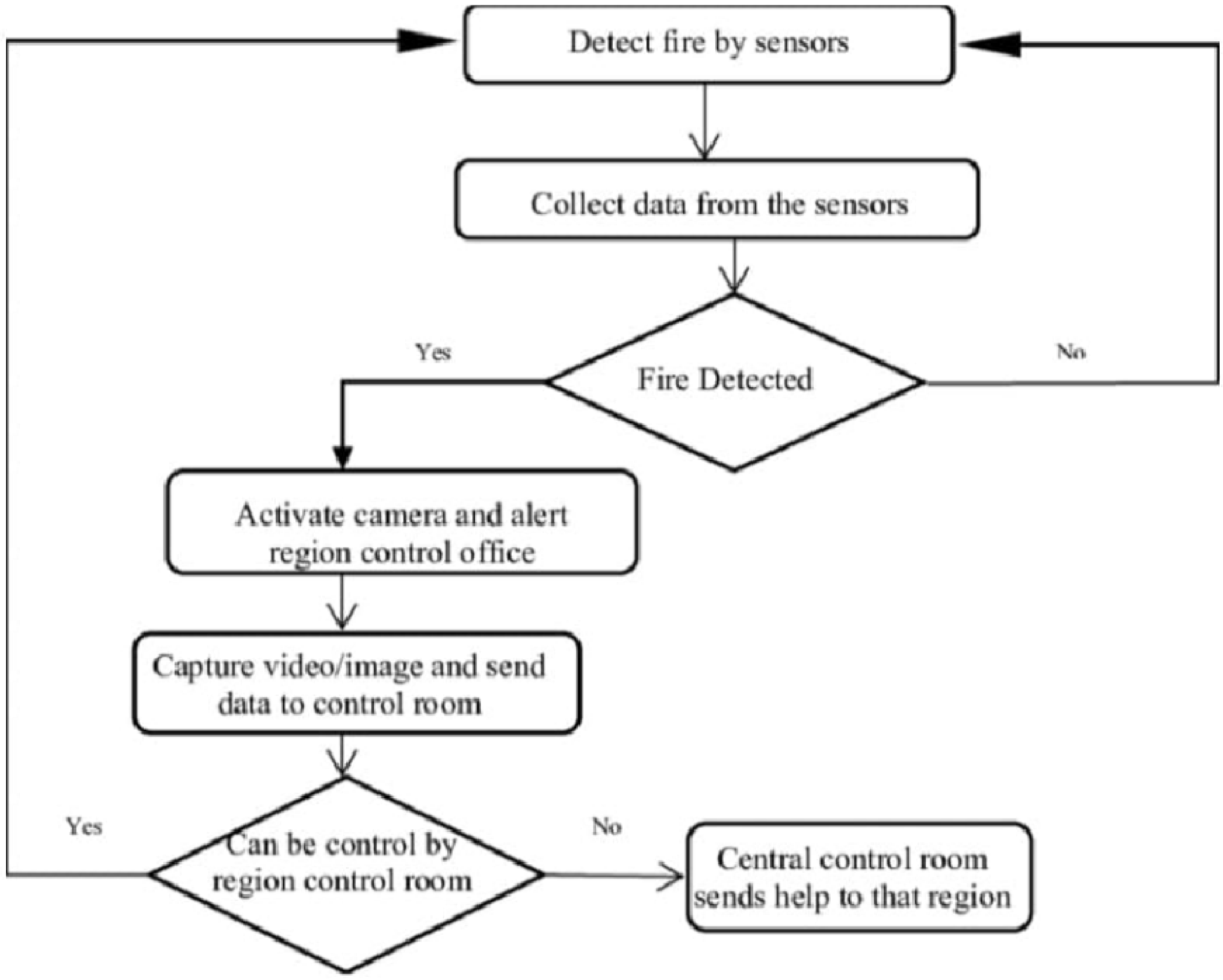
		information on the street conditions.
NH R-5	Availability	The system should be user-friendly and easy to use, with a clear and intuitive interface that can be accessed by both technical and non-technical users.
NF R-6	Scalability	Scalability systems should be designed with scalable security mechanisms to protect against increased threats and vulnerabilities associated with growth.

4. Project Design

4.1 Data Flow Diagrams

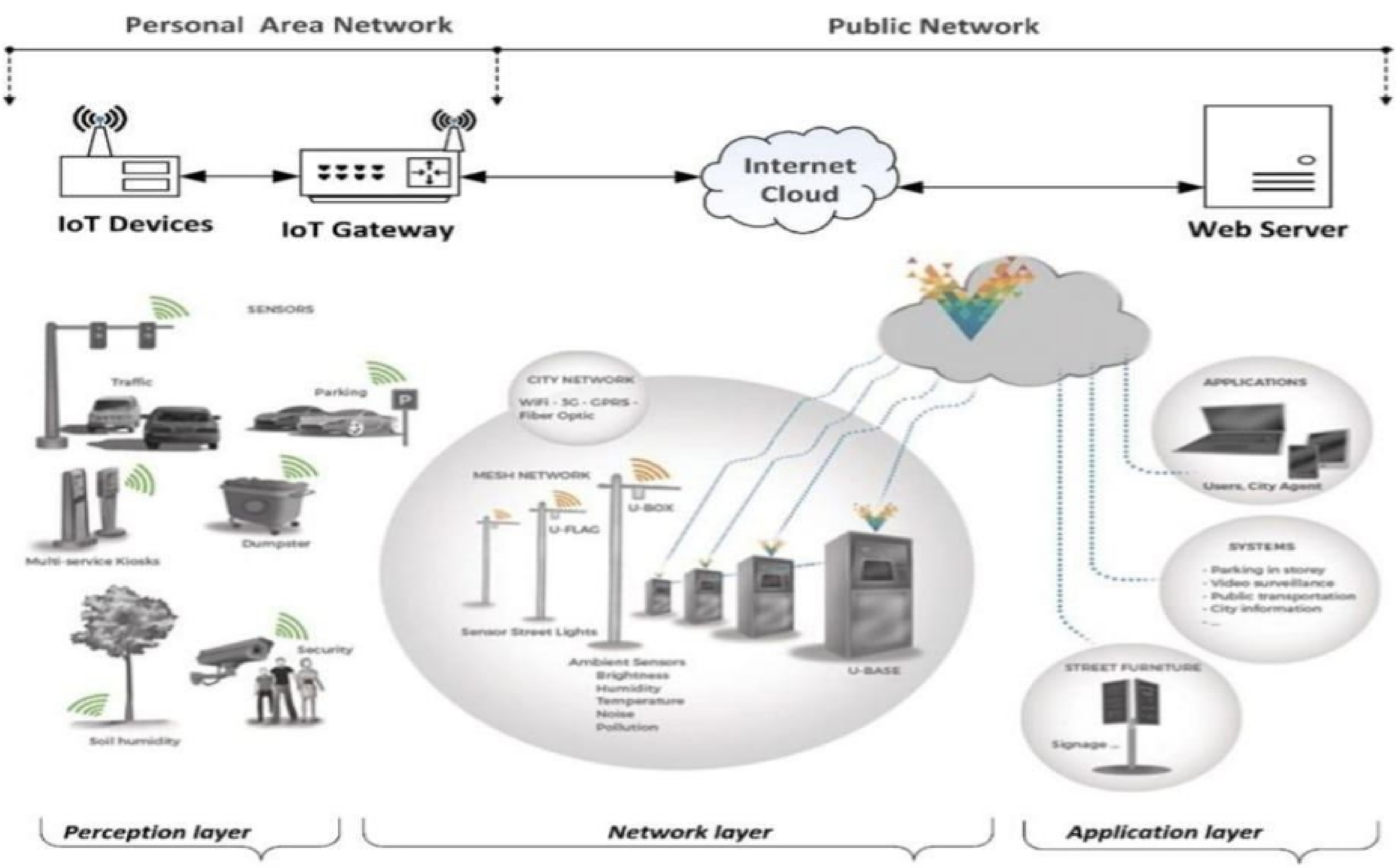
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored.

4.2 Solution and Technical Architecture



In the case of SQUID, the solution architecture involves integrating various open-source technologies to create a low-cost data platform that combines street imagery and ride quality data to provide a visual ground truth for a city's streets.The architecture includes hardware components such as the device that is mounted to the

back of a city vehicle with its camera facing downwards, as well as software components such as computer vision classifiers that are trained to detect cracks and assess street quality.The architecture also includes data storage and processing components to handle the large amounts of data collected by the device. Overall, the solution architecture is designed to provide a cost-effective and efficient way for cities and municipalities to maintain their streets and ensure safe and smooth travel for autonomous vehicles.



Technical Architecture:

S.No	Component	Description	Technology
1.	Street imaginery	SQUID uses street imagery captured by its camera to identify cracks, potholes, and other defects on the road surface.	High-Resolution Cameras,LiDAR,Satellite Imagery,Mobile Mapping Systems,Computer Vision
2.	Computer vision classifiers	SQUID uses computer vision algorithms to analyze the street imagery and identify different types of defects on the road surface.	Convolutional Neural Networks,Support Vector Machines
3.	Data integration	By combining these two data sources, SQUID can identify areas that may require maintenance before they become major issues, and enable cities to Prioritize maintenance activities more effectively.	cracks, potholes

4.	Ride quality Data	SQUID uses ride quality data to identify areas of the street that may have issues with uneven pavement, roughness, or other ride quality problems. This data is captured using sensors mounted on the city vehicle, and provides a quantitative measure of the condition of the street.	Accelerometers,GPS

4.3 User Stories

User Type	Functional Requirement	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
City street	Accessibility,safety,Infrastructure	USN-1	As a city street with a device is mounted on the back of a city vehicle and captures street imagery and ride quality data. The data is then processed using computer vision techniques	The device is mounted on the back of a city vehicle and captures street imagery and ride quality data. The data is then processed using computer vision	High	Sprint-1

			cracks and other issues with street quality.	techniques to detect cracks and other issues with street quality.		

Cracks	Image capture,Image processing,Crack detection,Data Storage,Integration	USN-2	As a Cracks with the system should be able to provide real-time feedback to city officials on the quality of the roads, including alerts for areas that require immediate attention or repairs.	The system should be able to provide real- time feedback to city officials on the quality of the roads, including alerts for areas that require immediate attention or repairs.	Medium	Sprint-2

Potholes	Image recognition,Location tracking,low cost,Data storage and analysis	USN-3	As a Potholes to the device should be able to store the collected data and transmit it to a central database for analysis and visualization. This would allow city officials to track road conditions over time, identify patterns and trends, and make datadriven decisions about road	This would allow city officials to track road conditions over time, identify patterns and trends, and make data-driven decisions about road maintenance and repair.	Low	Sprint-3

			maintenance and repair.			

Ride quality Data	Accuracy,Precision	USN- 4	As a street ride quality data collected by SQUID should be integrated with other data sources, such as weather and traffic data, to provide a comprehensive view of street conditions	The ride quality data collected by SQUID should be integrated with other data sources, such as weather and traffic data, to provide a comprehensive view of street conditions	Medium	Sprint-

Vehicle Sensor	Imaging capability, GPS tracking,Accelerometer,Data Storage,Wireless Connectivity	USN-5	As a sensor should have wireless connectivity capabilities to allow for the transfer of data to a central server or cloud-based storage	The sensor should have wireless connectivity capabilities to allow for the transfer of data to a central Server or cloud- based storage	Low	Sprint-

4

4

Parking IOT attendant	Real time monitoring,Data collection,Security,Maintenance	USN-6	As a parking lot attendant, I want to be able to easily identify and locate the vehicles parked in the lot, so that I can quickly and efficiently retrieve them when needed.	The system should be able to integrate with existing parking lot	Medium	Sprint-

5.CODING & SOLUTIONING

The project is intended to develop a website application for fuel consumption prediction. The application provides users with accurate prediction along with few features that users can make use of with great ease.

5.1 Feature 1

To make users convenient to the website an user dashboard has been created where the progress of user will be managed. The user needs to register/login to the website in order to access the dashboard. This dashboard consists of user profile, prediction tab, prediction history with a simple interface.

```
# importing necessary libraries
import numpy as np import cv2
from matplotlib import pyplot as
plt

import time import sys
import
ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device
Credentials organization = "umqu0q"
deviceType = "Praveen18" deviceId =
"Praveen8743" authMethod = "token"
authToken = "PraveenKumar"


def ibmstart(x):
   def myCommandCallback(cmd):
      print("Command received: %s" % cmd.data['command'])
print(cmd)
    try:
     deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
```

```
"auth-method": authMethod, "auth-token": authToken}
deviceCli = ibmiotf.device.Client(deviceOptions)      #...........
................................
```

5.2 Feature 2

Visual features involve analyzing street images or videos to extract information about street quality. These features are often extracted using computer vision techniques and machine learning algorithms. Some common visual features include:

Cracks and potholes: Detecting and quantifying the presence and severity of cracks and potholes in the street surface.

Surface texture: Analyze the texture of the street surface to identify rough or uneven areas that may impact ride quality.

Pavement condition: Assessing the overall condition of the pavement, including signs of deterioration, erosion, or damage.

Markings and signage: Examining the visibility and legibility of road markings, signs, and signals, which are essential for navigation and safety.

```
    except Exception as e:

      print("Caught exception connecting device: %s" % str(e))

sys.exit()
```

```
   deviceCli.connect()
lat=random.randint(9,37)
long=random.randint(68,97)
   data = { 'latitude' : lat, 'longitude': long ,'Status': x}
   #data = { 'Status' : x}
#print data     def
myOnPublishCallback():
      print ("Published Status = %s" % x, "to IBM Watson")

   success = deviceCli.publishEvent("SQUID", "json", data,
qos=0, on_publish=myOnPublishCallback)     if not success:
      print("Not connected to IoTF")


   deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()


# read a cracked sample image img =
cv2.imread('input set/Cracked_04.jpg')
flag=0
```

```
# Convert into gray scale

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Image processing ( smoothing )

# Averaging blur =

cv2.blur(gray,(3,3))


# Apply logarithmic transform

img_log = (np.log(blur+1)/(np.log(1+np.max(blur))))*255

# Specify the data type img_log =

np.array(img_log,dtype=np.uint8)


# Image smoothing: bilateral filter bilateral =

cv2.bilateralFilter(img_log, 5, 75, 75)


# Canny Edge Detection edges =

cv2.Canny(bilateral,100,200)
```

```
# Morphological Closing Operator

kernel = np.ones((5,5),np.uint8)

closing = cv2.morphologyEx(edges, cv2.MORPH_CLOSE, kernel)


# Create feature detecting method
# sift =

cv2.xfeatures2d.SIFT_create() # surf

= cv2.xfeatures2d.SURF_create() orb

= cv2.ORB_create(nfeatures=1500)
```

5.3 Feature 3

By combining and analyzing these different features, street quality identification systems can provide a comprehensive and multi-dimensional assessment of the condition of streets. This information enables urban planners, transportation authorities, and maintenance crews to make informed decisions, prioritize resources, and implement effective strategies for improving and maintaining the quality of urban road networks.

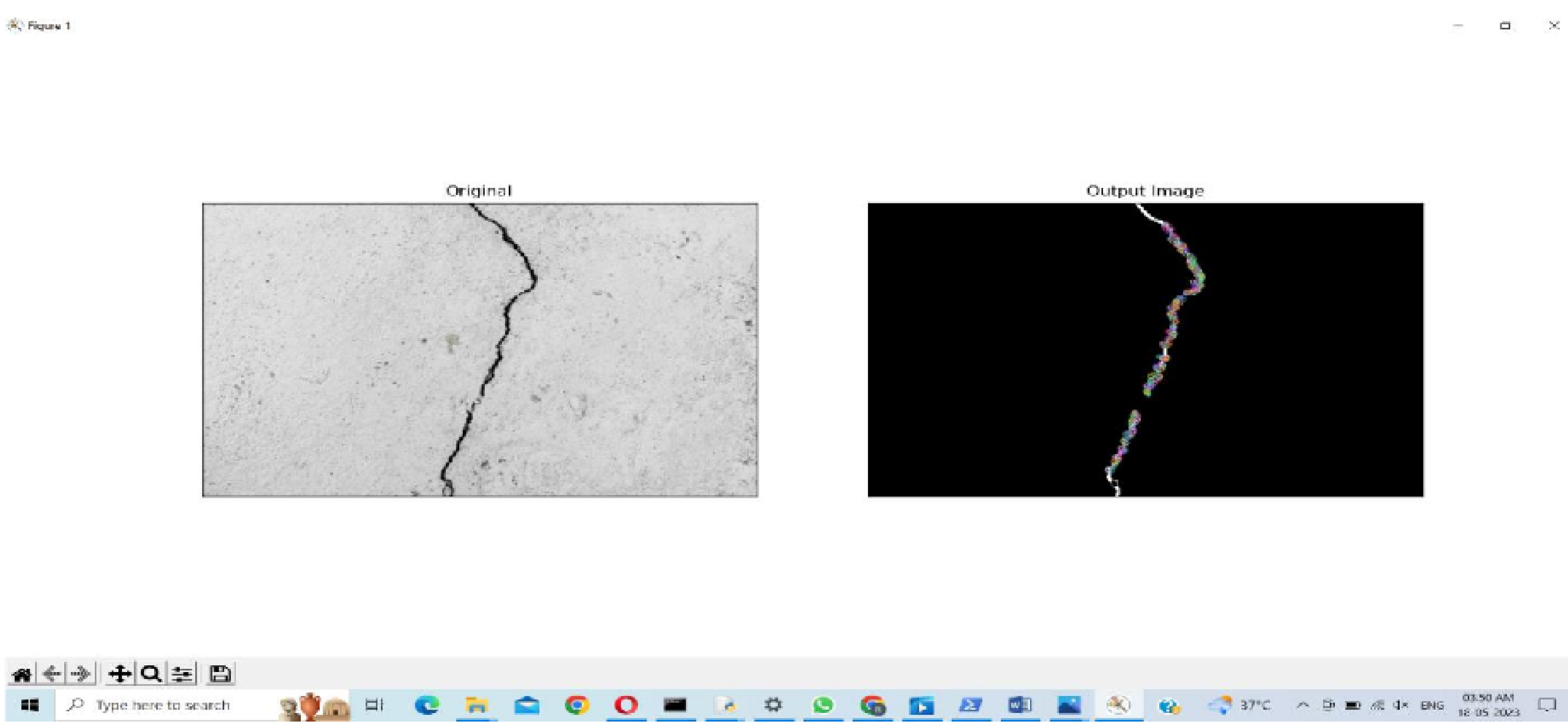
```
# Make featured Image keypoints, descriptors =
orb.detectAndCompute(closing, None) featuredImg =
cv2.drawKeypoints(closing, keypoints, None)

# Create an output image cv2.imwrite('Output
Set/CrackDetected-7.jpg', featuredImg) flag=1

# Use plot to show original and output image
plt.subplot(121),plt.imshow(img)
```

plt.title('Original'),plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(featuredImg,cmap='gr
ay') plt.title('Output Image'),plt.xticks([]),
plt.yticks([]) print(flag) ibmstart(flag) plt.show()

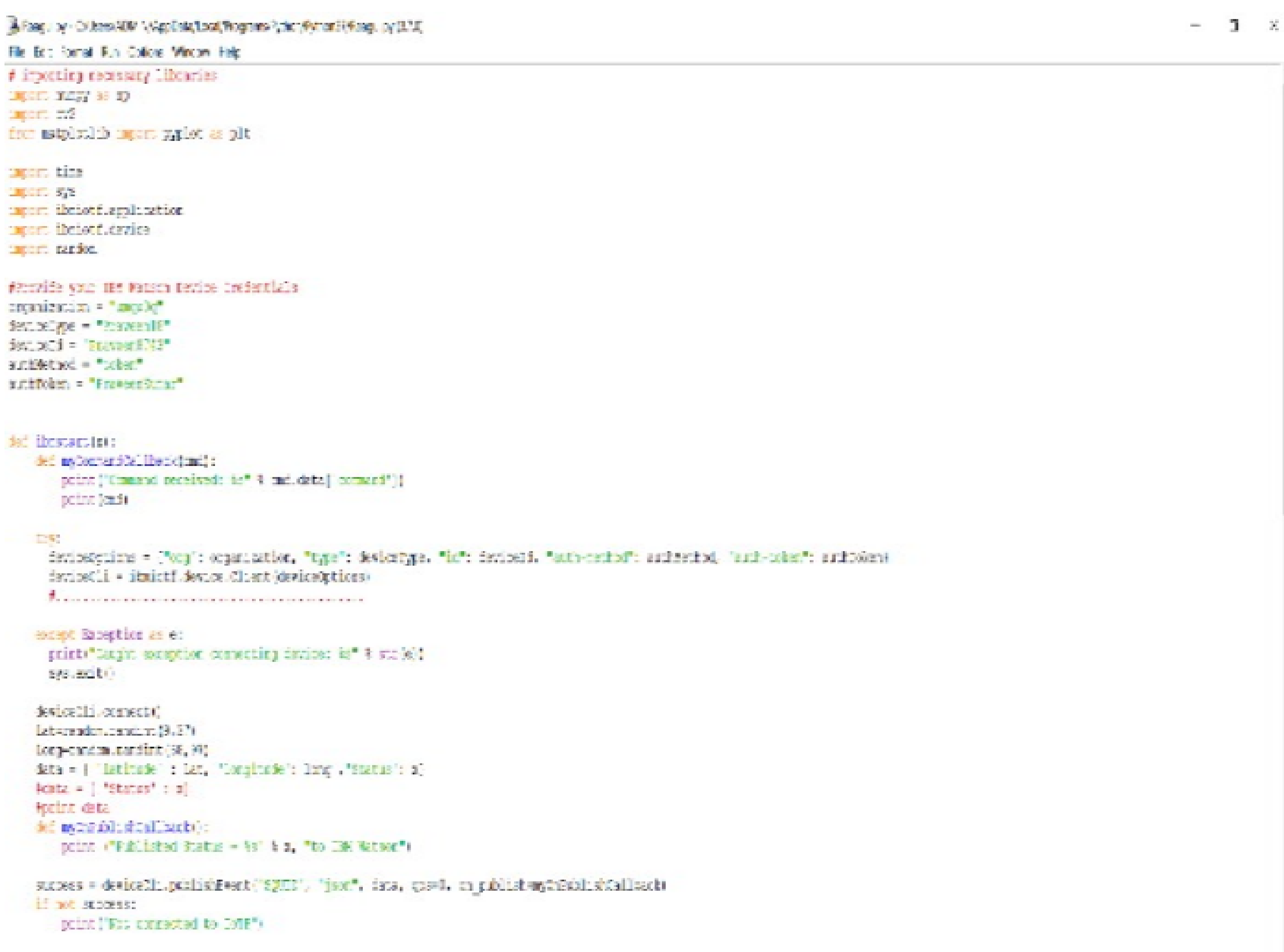
Solution



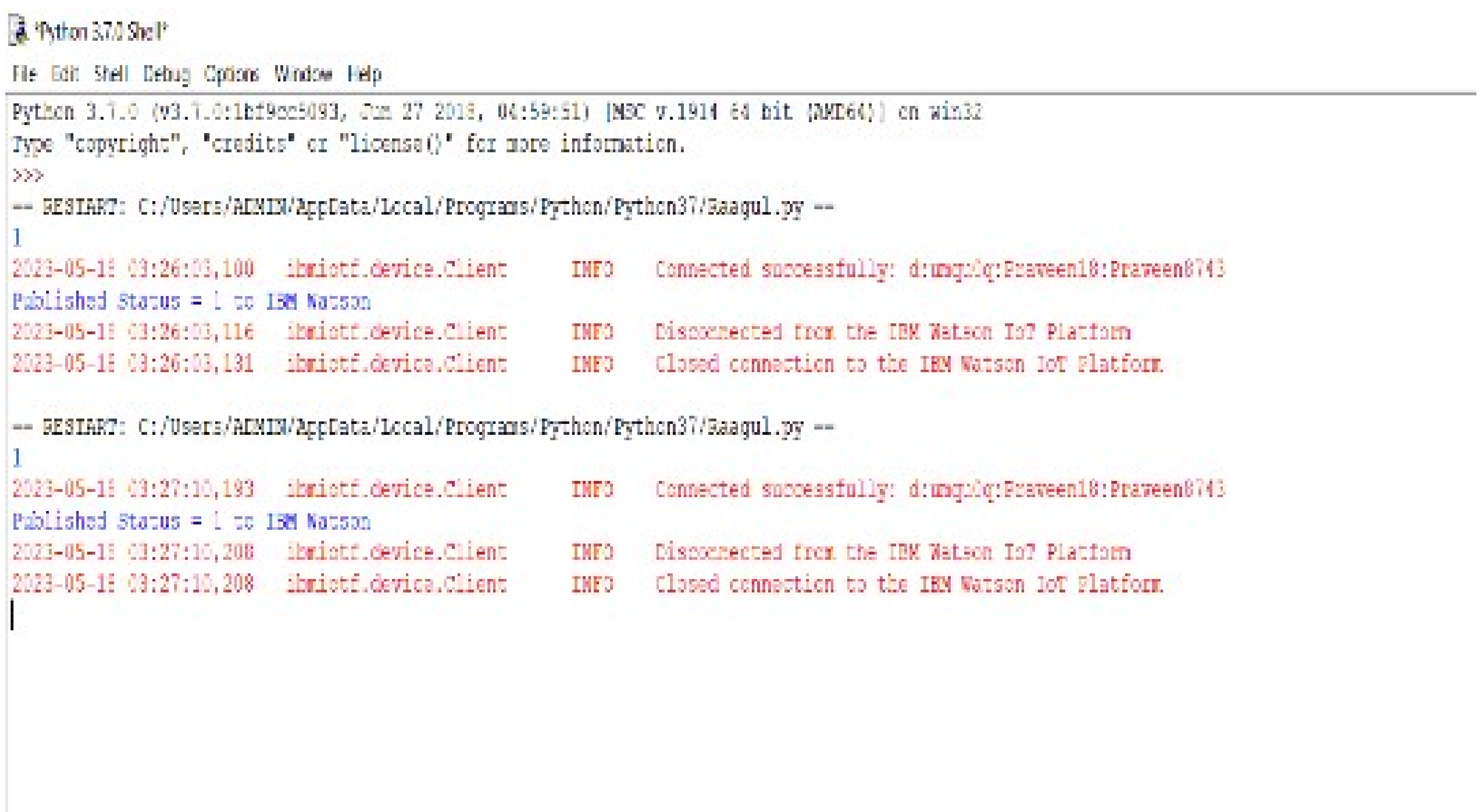
6. RESULTS

6.1 Performance metrics

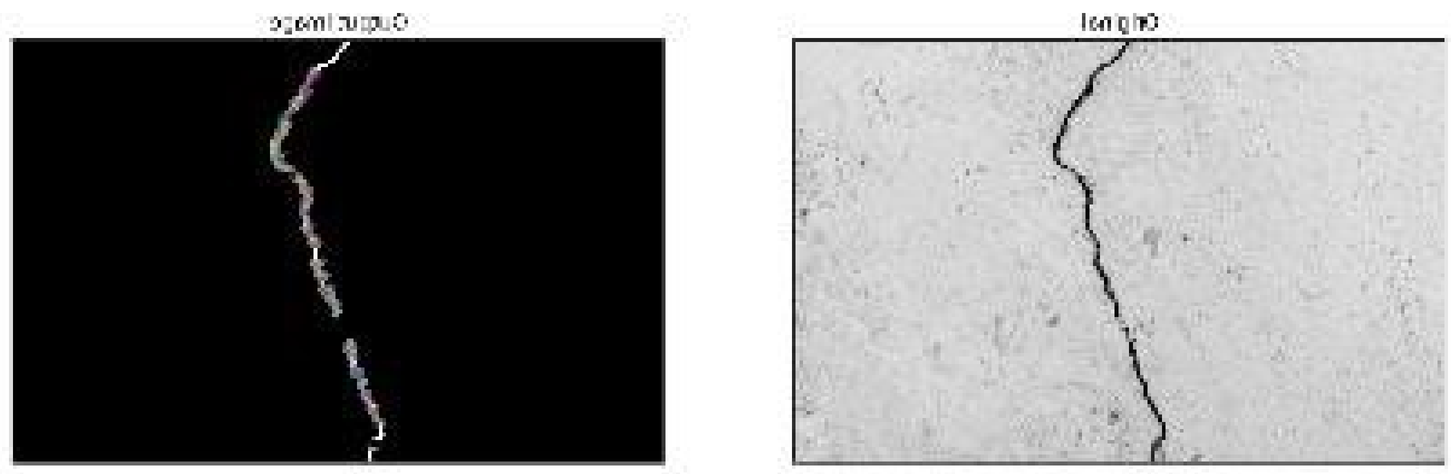
S.No.	Parameter	Values
1.	Metrics	Python accuracy of prediction and output screenshot



Performance Metrics



Outputs:



7. ADVANTAGES & DISADVANTAGES

1. Advantages

- It’s ability to improve infrastructure maintenance and enhance safety to promote the efficient mobility.
- It increase public satisfaction, support economic growth, and facilitate the data-driven decision making in urban planning and transportation management.

2.Disadvantages

- It requires investment in technology, equipment, and trained personnel to collect and analyze data.
- Monitoring and capturing images of streets and surrounding areas can inadvertently capture personal information or infringe upon privacy rights. It is essential to implement appropriate measures to address these concerns and protect individuals' privacy.

8. CONCLUSION

The street quality identification provides significant advantages in terms of infrastructure maintenance, safety, mobility, public satisfaction, economic benefits, and data-driven decision making. By accurately assessing the condition of streets, authorities can allocate resources more efficiently, prioritize maintenance work, and enhance overall transportation systems. However, there are also potential disadvantages to consider, including the cost and resource allocation, subjectivity and accuracy challenges, limited coverage and privacy concerns, technological limitations, bias and equity issues, and the risk

overemphasizing infrastructure. It is essential to address these concerns through careful
implementation, addressing biases, ensuring privacy protection, and adopting a holistic approach to transportation planning. With proper consideration and management, street quality identification can contribute to well-maintained and efficient road networks, promoting safer and more liveable communities.

9.FUTURE SCOPE

It's Emerging technologies such as artificial intelligence (AI), machine learning, and computer vision can enhance street quality identification. AI algorithms can analyze vast amounts of data collected from various sources, including sensors, cameras, and crowd-sourced information, to provide more accurate and real-time assessments of street conditions. Computer vision techniques can improve the identification of specific defects or hazards on roads. IOT-enabled sensors can be deployed across road networks to gather real-time data on street conditions, including temperature, moisture, traffic flow, and structural integrity.By leveraging these advancements, street quality identification can contribute to the development of smart, sustainable, and safe cities, enhancing the overall quality of transportation infrastructure and the well-being of communities.

10. APPENDIX

10.1 Source Code

```
# importing necessary libraries
import numpy as np import cv2
from matplotlib import pyplot as
plt

import time import sys
import
ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device
Credentials organization = "umqu0q"
deviceType = "Praveen18" deviceId =
"Praveen8743" authMethod = "token"
authToken = "PraveenKumar"


def ibmstart(x):     def
myCommandCallback(cmd):
      print("Command received: %s" % cmd.data['command'])
print(cmd)
      try:          deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
     deviceCli = ibmiotf.device.Client(deviceOptions)
```

```
#............................................

except Exception as e:
  print("Caught exception connecting device: %s" % str(e))
sys.exit()

deviceCli.connect()
lat=random.randint(9,37)
long=random.randint(68,97)     data = {
'latitude' : lat, 'longitude': long ,'Status': x}
#data = { 'Status' : x}
#print data     def
myOnPublishCallback():
    print ("Published Status = %s" % x, "to IBM Watson")

success     =     deviceCli.publishEvent("SQUID",     "json",      data,  qos=0,
on_publish=myOnPublishCallback)     if not success:
    print("Not connected to IoTF")


deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```

read a cracked sample image img = cv2.
imread('input set/Cracked_04.jpg') flag=0
Convert into gray scale gray = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)

Image processing (smoothing)
Averaging blur =
cv2.blur(gray,(3,3)) #
Apply logarithmic
transform img_log = (np.
log(blur+1)/(np.log(1+n
p.max(blur))))*255

# Specify the data type img_log =
np.array(img_log,dtype=np.uint8)

# Image smoothing: bilateral filter bilateral =
cv2.bilateralFilter(img_log, 5, 75, 75)

# Canny Edge Detection edges =
cv2.Canny(bilateral,100,200)

Morphological Closing Operator kernel =
np.ones((5,5),np.uint8) closing = cv2.morphologyEx(edges,
cv2.MORPH_CLOSE, kernel)

```
# Create feature detecting method
# sift =
cv2.xfeatures2d.SIFT_create() # surf
= cv2.xfeatures2d.SURF_create() orb
= cv2.ORB_create(nfeatures=1500)

# Make featured Image keypoints, descriptors =
orb.detectAndCompute(closing, None) featuredImg =
cv2.drawKeypoints(closing, keypoints, None)

# Create an output image cv2.imwrite('Output
Set/CrackDetected-7.jpg', featuredImg) flag=1

# Use plot to show original and output image
plt.subplot(121),plt.imshow(img)
plt.title('Original'),plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(featuredImg,cmap='gray')
plt.title('Output Image'),plt.xticks([]), plt.yticks([])
print(flag)

ibmstart(flag)
plt.show()
```

10.2 Github & Project Video Demo Link

Github - https://github.com/IBM-EPBL/IBM-Project-45492-1660730462

Video Demo link - https://drive.google.com/file/d/1bF-VhHB3uagiOIzXY8QSHevK6dD3hi74/view?usp=share_link