AI ENABLE CAR PARKING USING OPENCV

A PROJECT REPORT

Submitted to the

LMS PORTAL

By

PATHAN AZMEEN	**(20KP1A1229)**
YENDLURI DHARANI	**(20KP1A1251)**
GUDALA PRASANNA LAKSHMI	**(20KP1A1211)**
VEMPATI PAWAN KUMAR	**(20KP1A4457)**
SHAIK NAZEER BASHA	**(20KP1A1242)**

NRI INSTITUTE OF TECHNOLOGY

VISADALA (M), GUNTUR(D), ANDHRA PRADESH

INTRODUCTION:

To finding parking availability for a specific time period is a very tedious job in urban areas. The Indian government now focusing on the smart city project, already they published city name for an upcoming project, already they published city name for an upcoming smart city project.

In smart city application , intelligent transportation system (ITS) plays an important role- in that finding parking place, specifically for thecar owner to avoid time computation, as well as congestion in traffic isgoing to be very important. In this article, we propose an intelligent car parking system for the smart city using Circle Hough Transform (CHT).

Keywords- Intelligent transportation system (ITS), Circle HoughTransform (CHT), Circle detection, Video-Image processing, smart city, parking system , OpenCV. For today's traffic monitoring and itsmanagement is a recent trend in research development. In this paper, weare focusing on parking component of the traffic parameter.

Traffic verycongested from last decade due to the increasing rise of automobilecompanies offers to a customer, privatization of that- mainly more andmore used in present day compared to last decade and it's also increasing inthe future may be with same or more speed. So now government thinkinghow to solve this problem in real time? Within specified time duration

1.1PROJECT OVERVIEW:

PURPOSE:

It allows car park operators and companies to track their facilities, vehicleentry, and real-time reporting of the availability of parking spots. Thishelps companies manage their parks in a central digital hub offered with parking software.

1.Superior Technology

Parking management systems are known for their integration withtechnology. Most of these systems are based on improved models andtechnological innovations, due to which they are suited to be used invarious car parks.

2.Better parking experience:

Better car park management means happier customers. A parkingmanagement system enhances the customer journey by providing themwith a unified procedure.

3.Increased Protection

Parking management systems have technologically advanced securityfeatures that enable you to prevent parking misuse and suspicious activityin your parking facility

4.Reduced traffic and pollution

Vehicles that keep circling an area in search of an empty parking spacecause most of the city traffic. Moreover, significantly driving around or waiting for a parking space to be vacant burns through a lot of fuel andreleases emissions daily.

5.Easy implementation and management

Another of the benefit of a parking management system is that it canefficiently be designed and implemented. These systems have a well-organised structure

6.Cost-effective

Another advantage that you obtain from installing a smart parkingmanagement system is the cost. It runs on a low workforce, so you cansave money and time

7.Uses integrated software and applications

Parking management solutions use software and applications that can becombined with another. Depending on your car park’s requirements, thereare lots of customisations available.

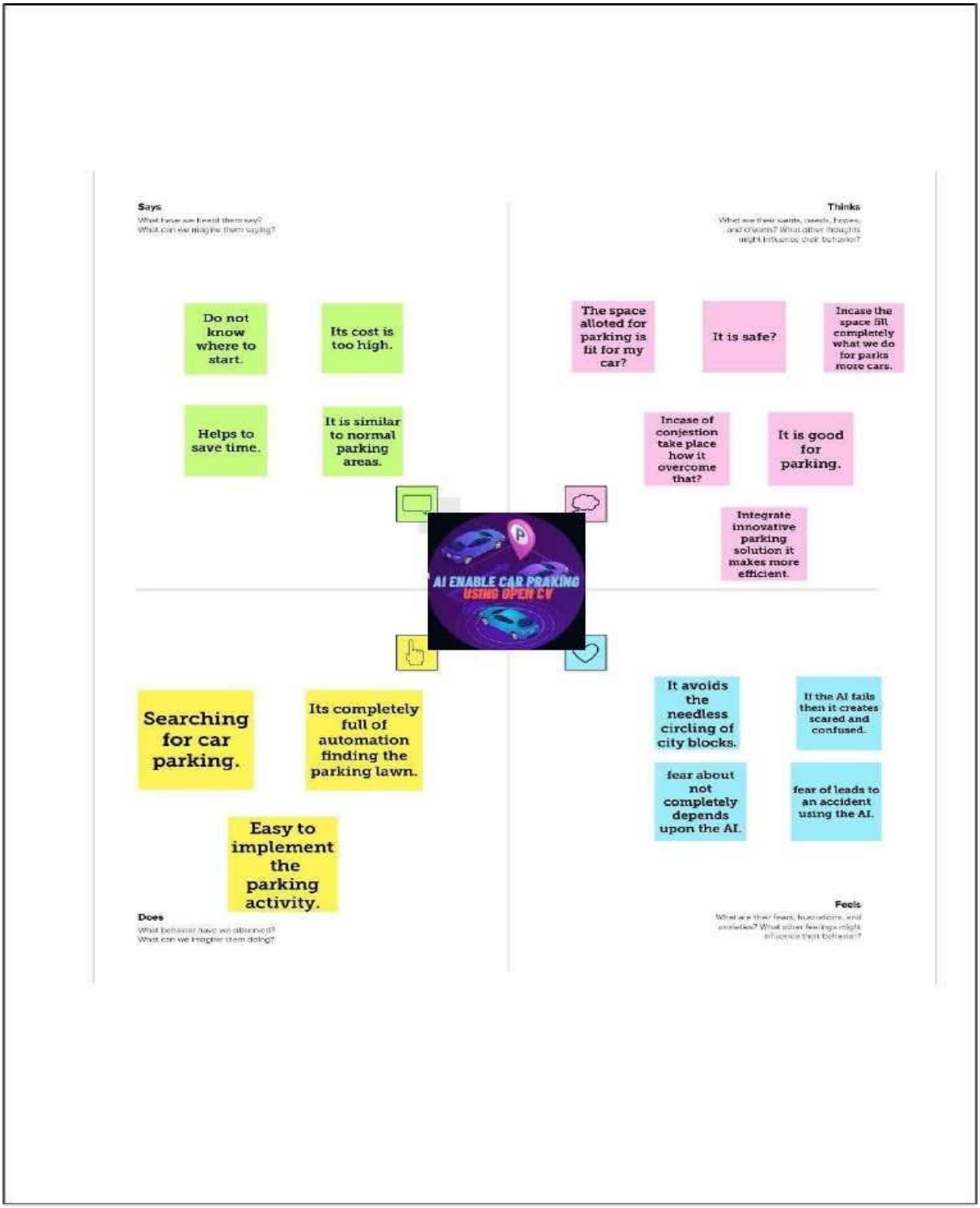
2.IDEATION & PROPOSED SOLUTION:

2.1 PROBLEM STATEMENT DEFINITION:

Problem Statement – AI Enable Car Parking Using Open CV:

By using ultrasonic sensors be able to keep a record of the number of cars parkedinside of a parking garage. Consequently, once a car enters a parking garagefollowed by a parking space, a ping ultrasonic sensor will then be able todetermine if a car is parked in the space or not

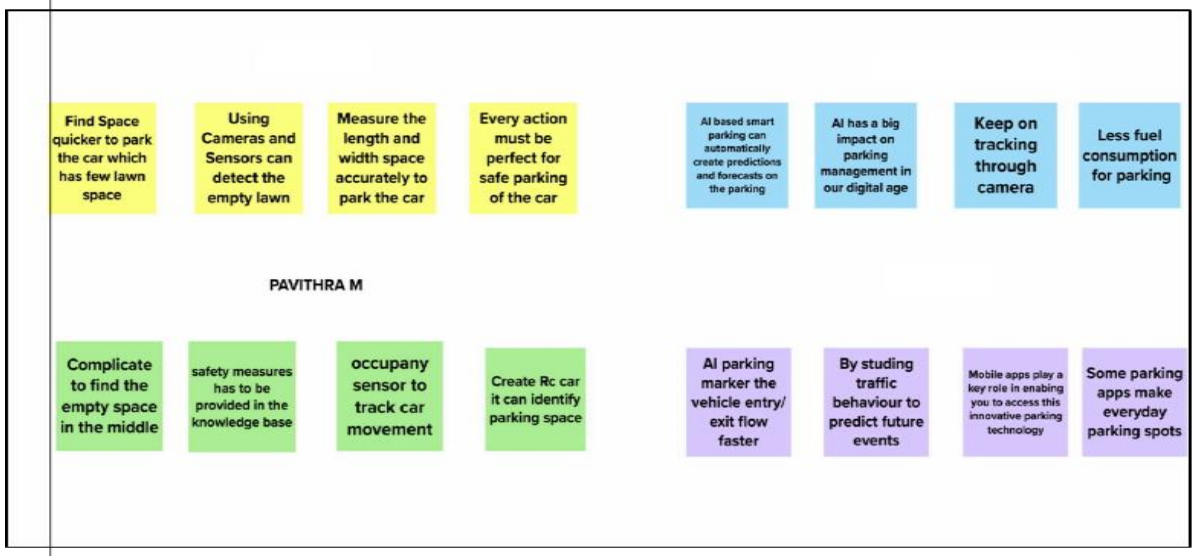
2.2 EMPATHY MAP CANVAS:



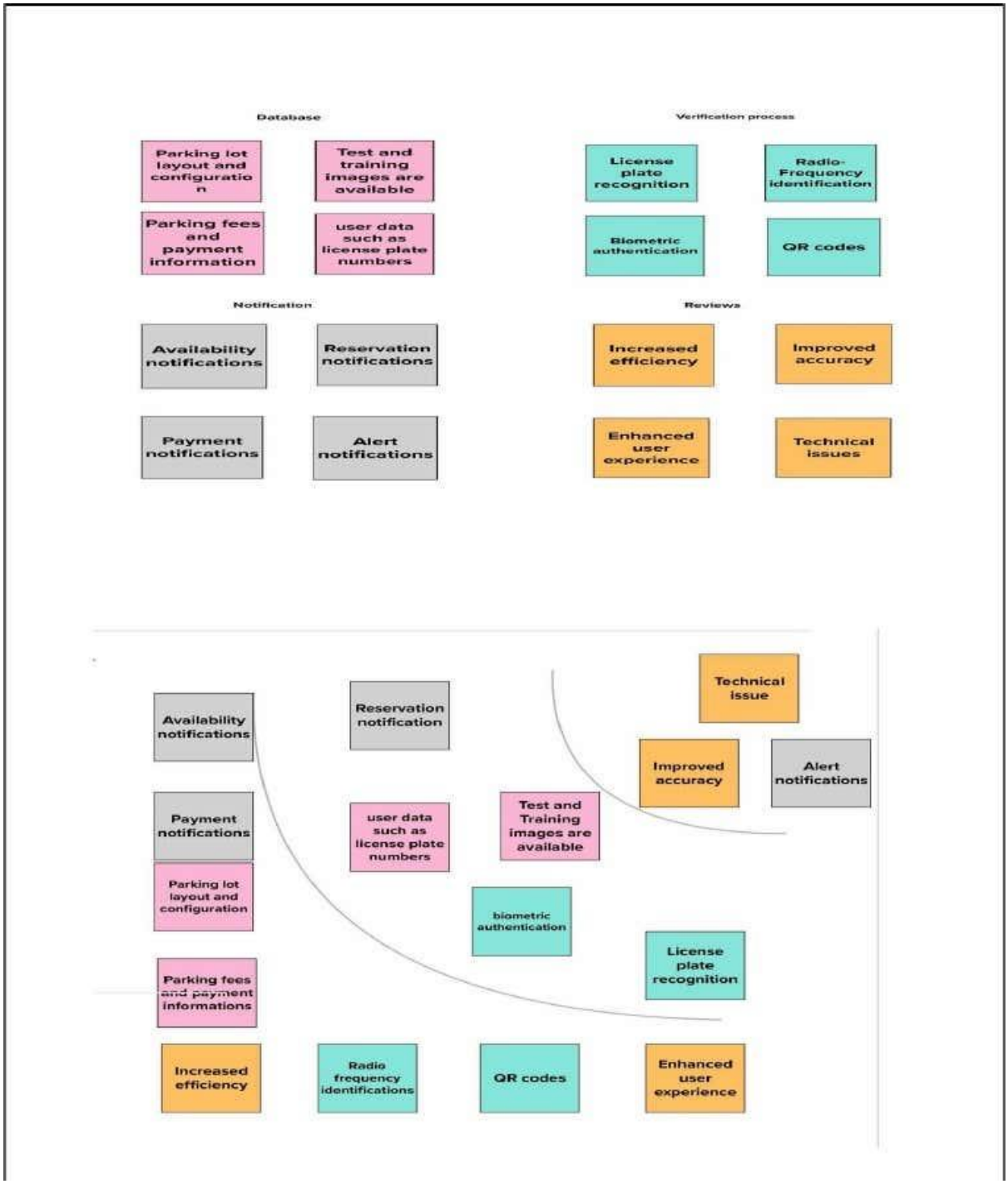
IDEATION AND BRAINSTORMING:

Brainstorm & Idea Prioritization Template: AI EnableCar Parking Using Open CV

Step 1: Brainstorm, Idea Listening and Grouping



Step 2: Idea Prioritization:



PROPOSED SYSTEM:

SNO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	To find the free parking slot in a minimum distance from a starting point
2	Idea/solution description	The idea of this project to find the difference between empty slot and occupied slot and given numbers for each slot in ascending order to find minimum distance unoccupied slot
3	Novelty/uniqueness	By above approach the parking slot is segregated as occupied and unoccupied slots

REQUIREMENT ANALYSIS:

FUNCTIONAL REQUIREMENT:

Following are the functional requirements of the proposed solution.

FR NO.FunctionalRequirement (Epic)Sub requirement(story / Sub-Task)

FR-1User RegistrationRegistration through FormRegistration through familiarize with the systemRegistration through mobile appFR-2User ConfirmationConfirmation via EmailConfirmation via approval of parking passFR-3Object DetectionThe system should be able to detect the presence of a car in a parking spot.FR-4Parking monitoringThe system should be able to monitor the parked cars and detect any illegal activities,such as double parking or parking in a handicapspot.

FR-5Real-time updatesThe system should provide real-time updates on parking availability and other relevantinformation to drivers and parking lot staff.FR-6User-friendly interfaceThe system should have a user-friendlyinterface that is easy to use and understand, toensure a smooth and hassle-free parkingexperience for drivers.

Non – Functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.Non-FunctionalRequirementDescription

NFR-1UsabilityThe system should be user-friendly and intuitive,with a simple and easy-to-use interface that isaccessible to all users. NFR-2SecurityThe system should be designed with robustsecurity features, to ensure the privacy and safetyof drivers and their vehicles, and to preventunauthorized access and data breaches.

NFR-3ReliabilityThe system should be reliable and stable, with highavailability and minimal downtime. NFR-4PerformanceThe system should be able to process data quicklyand accurately, with minimal delay and highefficiency. NFR-5AvailabilityThe system should be highly available, withminimal downtime and interruption to the parkingservice NFR-6ScalabilityThe system should be able to handle a largenumber of parking spots and users, and be easilyscalable as the demand increases.

PROJECT DESIGN:

4.1 DATA FLOW DIAGRAM:

SOLUTION & TECHNICAL ARCHITECTURE:

TECHNOLOGICAL ARCHITECTURE:

Table-1: Components & Technologies:S. NoComponentDescriptionTechnology

1.User InterfaceUser Interface is used byuser in mobileapplication or In Build incar display itself HTML, CSS,JavaScript / Angular JS / React JS etc.2.User Logic-1Framework used fordesign the softwarePython , python-flask

3.User Logic-2Access the software inthe car by the driver todetect spotPython, Open CV

4.Application Logic-1Open CV is an open-source platform for providing real timecomputer visiontechnologyOpen CV

5.Data BaseContains images andvideo frames stores indata baseMySQL, NoSQL, etc.

6.Cloud Data BaseData Base Service oncloudIBM DB2, IBMCloud etc.

7.File StorageFile storage requirementsIBM Block Storage or Other Storage Serviceor local File system

8.External API-1They make it easy fordevelopers to storemanage and deploycontainer imagesContainer registry

10.Machine LearningModelUses test and trained dataimages and video tolearn the environmentObjectrecognitionmodels, etc.

11.Infrastructure (server /cloud)ApplicationDevelopment on Localsystem / cloudLocal, cloud Foundry, python-flask, etc.

USER STORIES:

PURPOSE

Itallows car park operators and companies to track their facilities, vehicleentry, and real-time reporting of the availability of parking spots.

Thishelps companies manage their parks in a central digital hub offered with parking software.

1.Superior Technology

Parking management systems are known for their integration withtechnology. Most of these systems are based on improved models andtechnological innovations, due to which they are suited to be used invarious car parks.

2.Better parking experience

Better car park management means happier customers. A parkingmanagement system enhances the customer journey by providing themwith a unified procedure.

3.Increased Protection

Parking management systems have technologically advanced securityfeatures that enable you to prevent parking misuse and suspicious activityin your parking facility

4.Reduced traffic and pollution

Vehicles that keep circling an area in search of an empty parking spacecause most of the city traffic.

Moreover, significantly driving around or waiting for a parking space to be vacant burns through a lot of fuel andreleases emissions daily.

5.Easy implementation and management

Another of the benefit of a parking management system is that it canefficiently be designed and implemented. These systems have a well-organised structure

6.Cost-effective

Another advantage that you obtain from installing a smart parkingmanagement system is the cost. It runs on a low workforce, so you cansave money and time

7.Uses integrated software and applications

Parking management solutions use software and applications that can becombined with another. Depending on your car park's requirements, thereare lots of customisations available.

IDEATION & PROPOSED SOLUTION:

PROBLEM STATEMENT DEFINITION:

Problem Statement – AI Enable Car Parking Using Open CV:

By using ultrasonic sensors be able to keep a record of the number of cars parkedinside of a parking garage.

Consequently, once a car enters a parking garagefollowed by a parking space, a ping ultrasonic sensor will then be able todetermine if a car is parked in the space or not.

Says

What have we heard them say?
What can we imagine them saying?

Do not know where to start.

Its cost is too high.

Helps to save time.

It is similar to normal parking areas.

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

The space alloted for parking is fit for my car?

It is safe?

Incase the space fill completely what we do for parks more cars.

Incase of conjestion take place how it overcome that?

It is good for parking.

Integrate innovative parking solution it makes more efficient.

Searching for car parking.

Its completely full of automation finding the parking lawn.

Easy to implement the parking activity.

Does

What behavior have we observed?
What can we imagine them doing?

It avoids the needless circling of city blocks.

If the AI fails then it creates scared and confused.

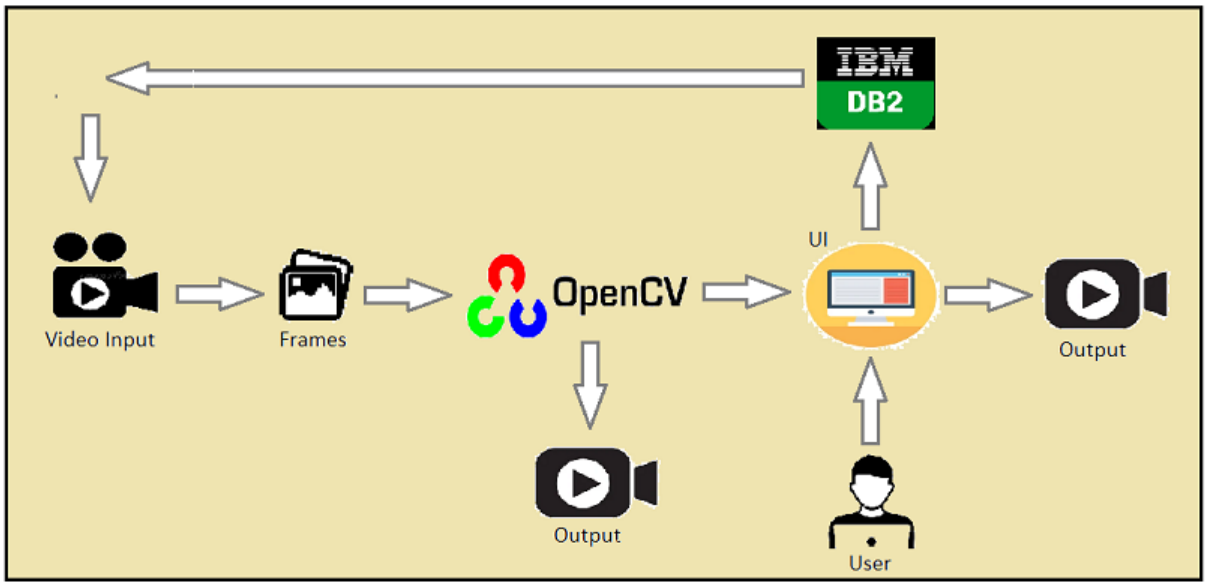
fear about not completely depends upon the AI.

fear of leads to an accident using the AI.

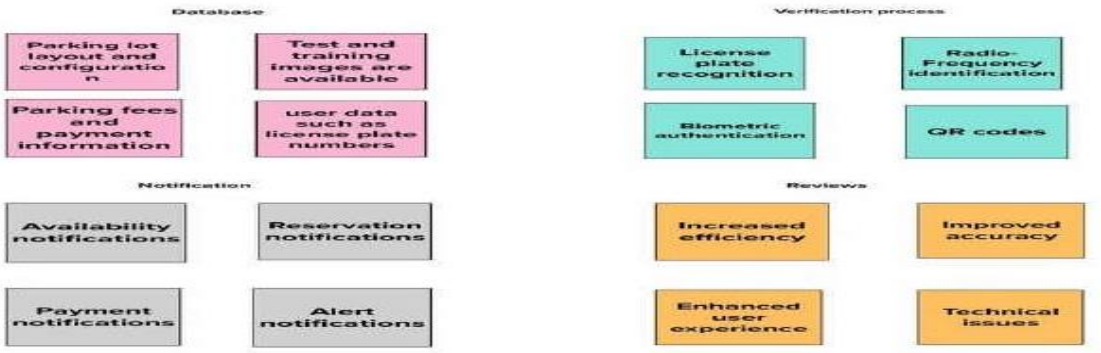
Feels

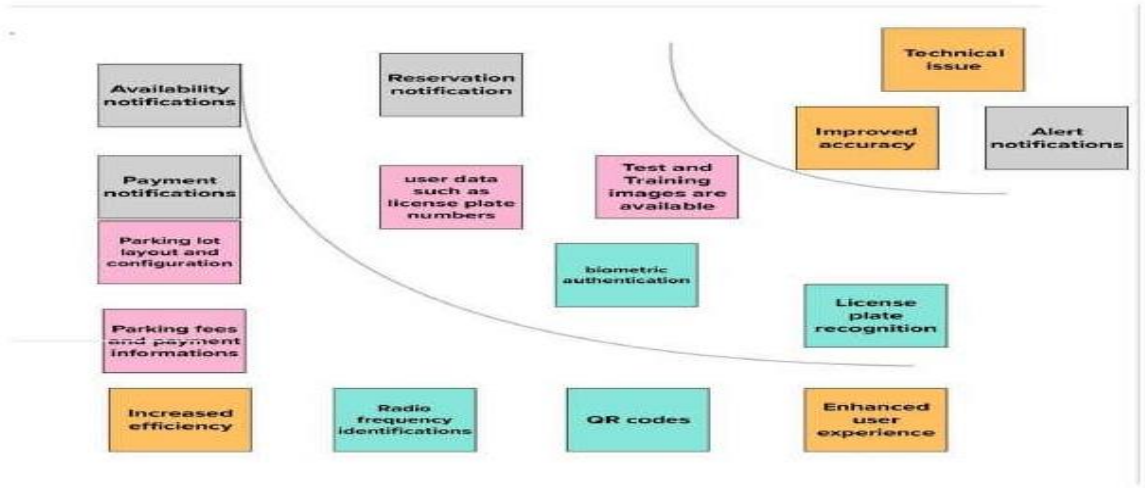
What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

AI Enable Car Parking Using Open CV



Step 1: Brainstorm, Idea Listening and Grouping.





PROPOSED SYSTEM:

1.Problem Statement (Problem to be solved)To find the free parking slot in aminimum distance from a starting point.

2.Idea / Solution descriptionThe idea of this project to find thedifference between empty slot andoccupied slot and given numbers for each slot in ascending order to findminimum distance unoccupied slot.

3. Novelty / UniquenessBy above approach the parking slot issegregated as occupied and unoccupiedslots

4.Social Impact / Customer SatisfactionThis technique will reduce the timetaken park their car and therebyimproving the customer satisfaction.

5.Business Model (RevenueModel)The result of this project could beimplemented in public places and theywill be able to achieve the accuracy.

6.Scalability of the SolutionThe outcome of this project will bevery helpful in parking managementsystem.

REQUIREMENT ANALYSIS:

FUNCTIONAL REQUIREMENT:

Following are the functional requirements of the proposed solution.

FR NO.FunctionalRequirement (Epic)Sub requirement(story / Sub-Task)

FR-1User RegistrationRegistration through FormRegistration through familiarize with the systemRegistration through mobile appFR-2User ConfirmationConfirmation via EmailConfirmation via approval of parking passFR-3Object DetectionThe system should be able to detect the presence of a car in a parking spot.FR-4Parking monitoringThe system should be able to monitor the parked cars and detect any illegal activities,such as double parking or parking in a handicapspot.

FR-5Real-time updatesThe system should provide real-time updates on parking availability and other relevantinformation to drivers and parking lot staff.

FR-6User-friendly interface The system should have a user-friendlyinterface that is easy to use and understand, to ensure a smooth and hassle-free parking experience for drivers.

Non – Functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.Non-FunctionalRequirementDescription

NFR-1UsabilityThe system should be user-friendly and intuitive,with a simple and easy-to-use interface that isaccessible to all users.

NFR-2SecurityThe system should be designed with robustsecurity features, to ensure the privacy and safetyof drivers and their vehicles, and to preventunauthorized access and data breaches.

NFR-3ReliabilityThe system should be reliable and stable, with highavailability and minimal downtime.

NFR-4PerformanceThe system should be able to process data quicklyand accurately, with minimal delay and highefficiency.

NFR-5AvailabilityThe system should be highly available, withminimal downtime and interruption to the parkingservice

NFR-6ScalabilityThe system should be able to handle a largenumber of parking spots and users, and be easilyscalable as the demand increases.

PROJECT DESIGN:

Line Detection

To detect the parking spots, I knew I could take advantage of the lines demarking the boundaries.

The Hough Transform is a popular feature extraction technique for detecting lines. OpenCV encapsulates the math of the Hough Transform into HoughLines(). Further abstraction in captured in HoughLinesP(), which is the probabilistic model of creating lines with the points that HoughLines() returns. For more info, check out the OpenCV Hough Lines tutorial.

The following is a walkthrough to prepare an image to detect lines with the Hough Transform. Links point to OpenCV documentation for each function. Arguments for each function are given as keyword args for clarity.

Reading in this image:

```
img = cv2.imread(filename='examples/hough_lines/p_lots.jpg')
```

I converted it to gray scale to reduce the info in the photo:

```
gray = cv2.cvtColor(src=img, code=cv2.COLOR_BGR2GRAY)
```

Gave it a good Gaussian blur to remove even more unnecessary noise:

```
blur_gray = cv2.GaussianBlur(src=gray, ksize=(5, 5), sigmaX=0)
```

Detected the edges with Canny:

```
edges = cv2.Canny(image=blur_gray, threshold1=50, threshold1=150, apertureSize=3)
```

And then, a few behind-the-scenes rhos and thetas later, we have our Hough Line results.

```
lines = cv2.HoughLinesP(image=edges, rho=1, theta=np.pi/180, threshold=80, minLineLength=15, maxLineGap=5)
for x1,y1,x2,y2 in lines[0]:
    cv2.line(img,(x1,y1),(x2,y2),(0,255,0),2)
```

Well that wasn't quite what I expected.

I experimented a bit with the hough line, but toggling the parameters kept getting me the same one line.

A bit of digging and I found a promising post on StackOverflow

After following the directions of the top answer, I got this:

Which gave me more lines, but I still had to figure out which lines were part of the parking space and which weren't. Then, I would also need to detect when a car moved from a spot.

I was running into a challenge; with this approach, I needed an empty parking lot to overlay with an image of a non-empty lot. Which would also call for a mask to cover unimportant information (trees, light posts, etc.)

Given my scope for the weekend, it was time to find another approach.

Drawing Rectangles

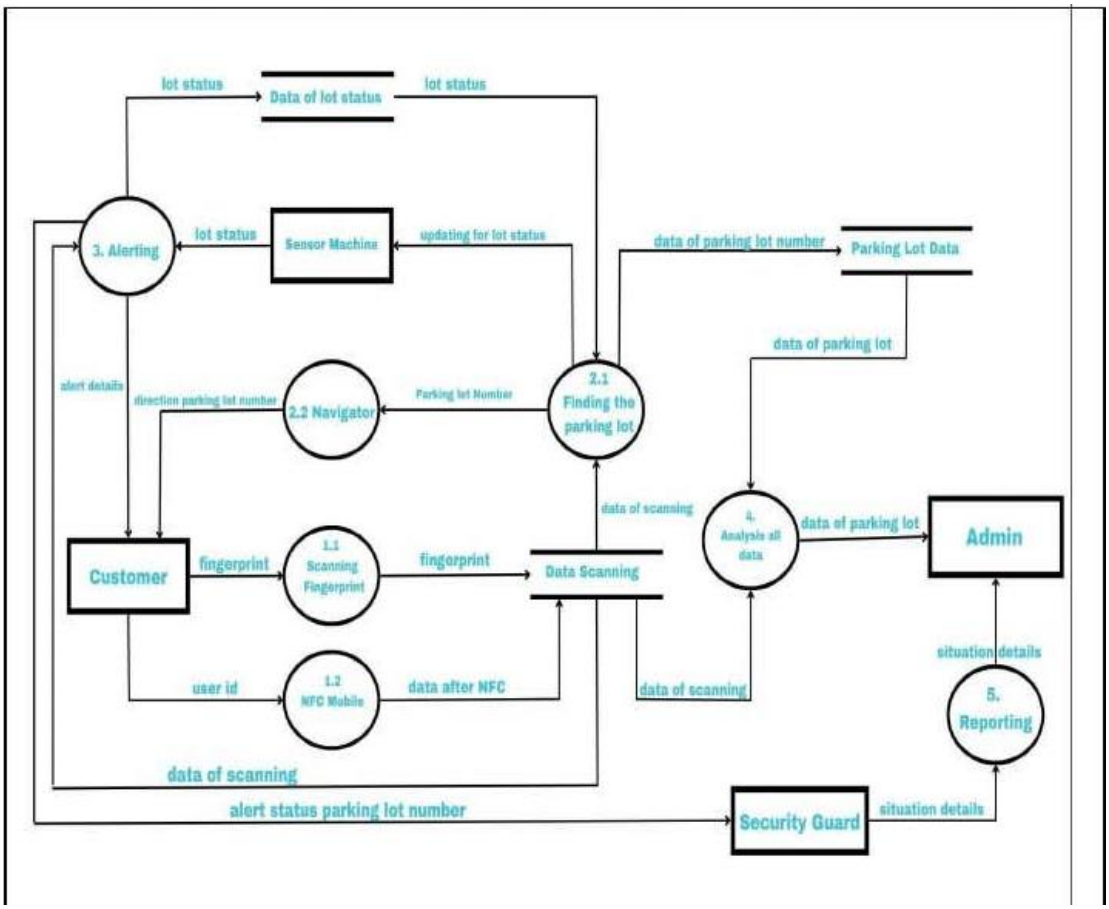
If my program wasn't able to detect parking spots on it's own, maybe it was reasonable to expect that the user give positions for each of the parking spots.

Now, the goal was to find a way to click on the parking lot image and to store the 4 points that made up a parking space for all of the spaces in the lot.

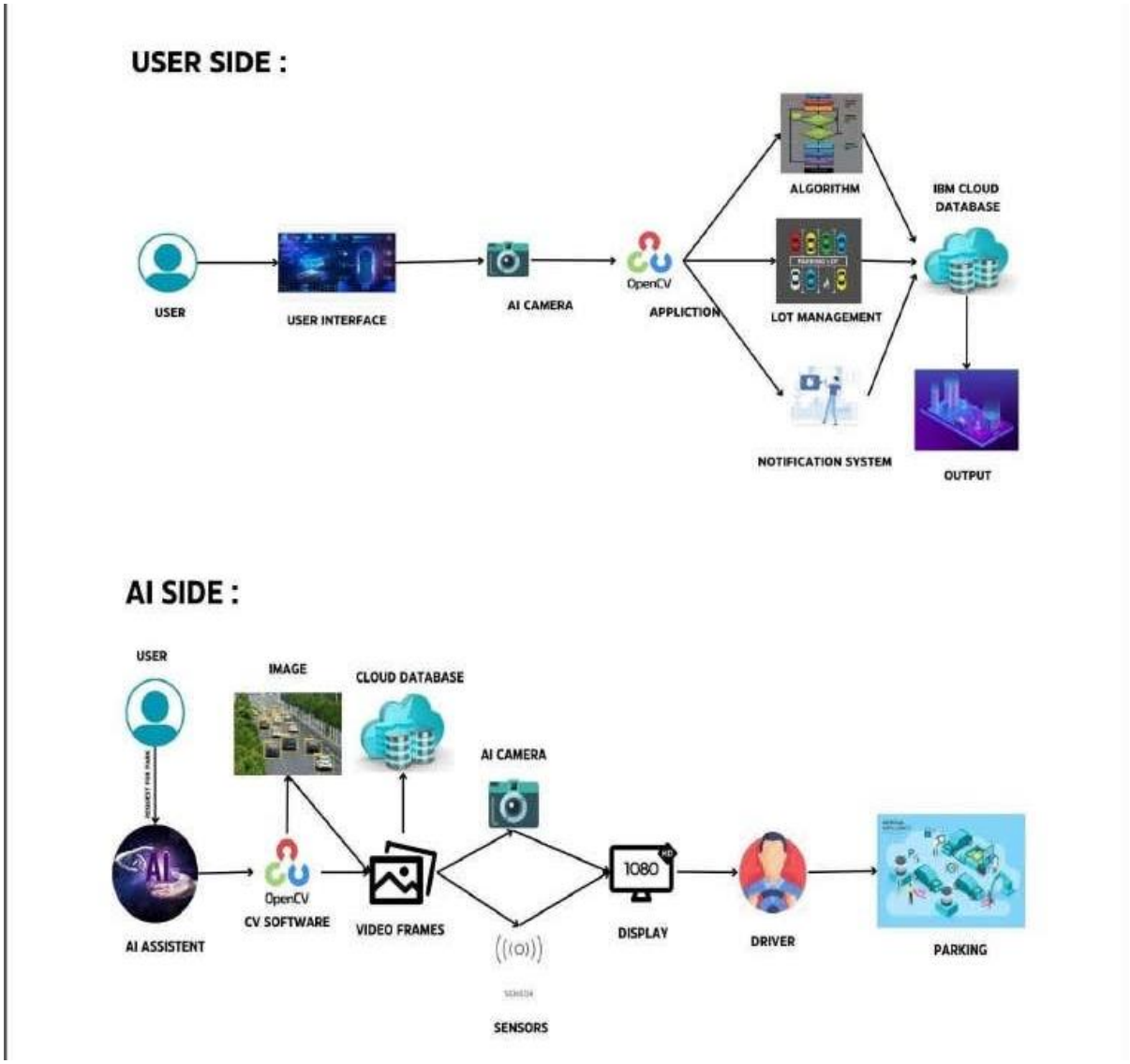
I discovered that I could do this using a mouse as a "paintbrush"

After some calculations for the center of the rectangle (to label each space), I got this:

DATA FLOW DIAGRAM:



SOLUTION & TECHNICAL ARCHITECTURE:

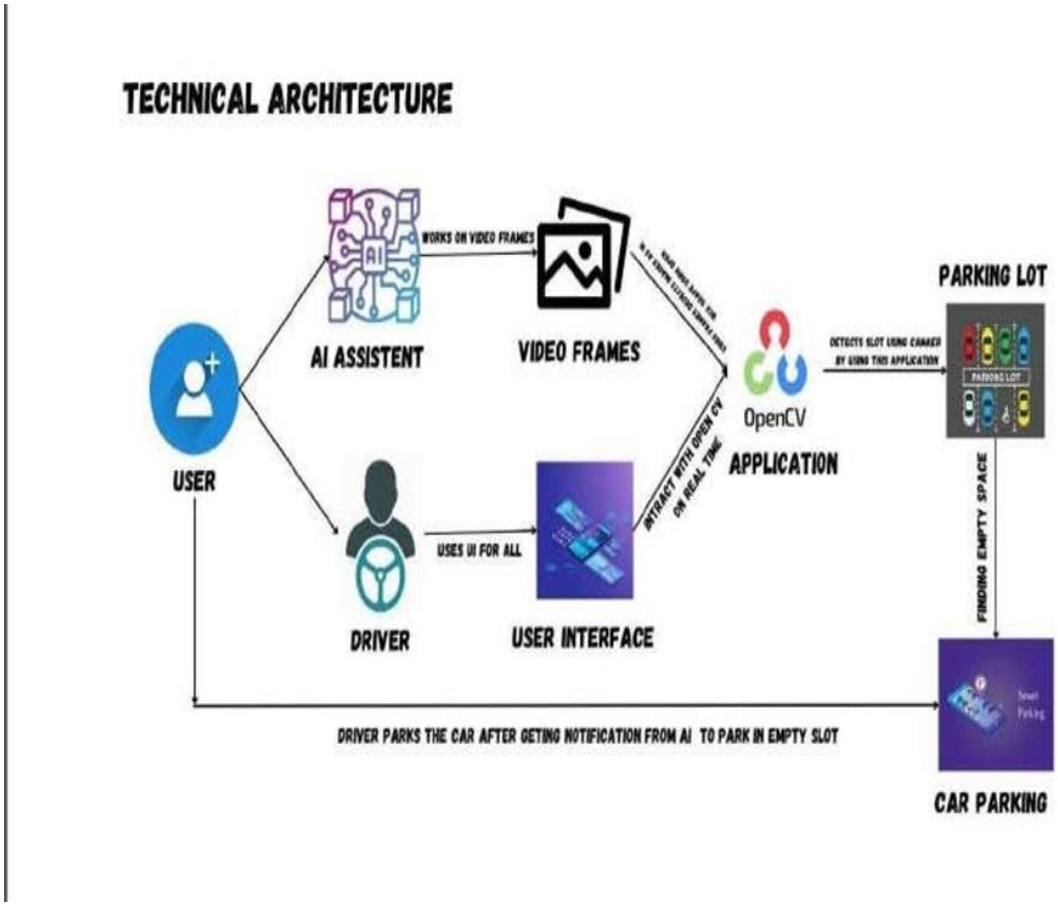


TECHNOLOGICAL ARCHITECTURE:

Table-1: Components & Technologies:S. NoComponentDescriptionTechnology

1.User InterfaceUser Interface is used byuser in mobileapplication or In Build incar display itself HTML, CSS,JavaScript / Angular JS / React JS etc.

2.User Logic-1Framework used fordesign the softwarePython , python-flask



3.User Logic-2Access the software inthe car by the driver todetect spotPython, Open CV

4.Application Logic-1Open CV is an open-source platform for providing real timecomputer visiontechnologyOpen CV

5.Data BaseContains images andvideo frames stores indata baseMySQL, NoSQL, etc.

6.Cloud Data BaseData Base Service oncloudIBM DB2, IBMCloud etc.

7.File StorageFile storage requirementsIBM Block Storage or Other Storage Service or local File system

8.External API-1They make it easy for developers to store manage and deploy container images Container registry

10.Machine LearningModelUses test and trained dataimages and video tolearn the environmentObjectrecognitionmodels, etc.

11.Infrastructure (server /cloud)ApplicationDevelopment on Localsystem / cloudLocal, cloud Foundry, python-flask, etc.

USER STORIES:

the below template to list all the user stories for the product

User Type Functional Requirement(Epic)User Story Number User Story/ Task Acceptance criteria Priority Team Member

Customer (Mobileuser)RegistrationUSN-1As a user,I canregister for the application by enteringmy email, password,and confirmingmy password.

I can access my account /dashboard HighSprint -1USN-2As a user, I will receive confirmation email once I have registeredfor the application I can receiveconfirmationemail &click confirm High Sprint -1USN-3As a user, I can register for the applicationthroughFace book I can register & access the dashboard with Face book Login Low Sprint -2USN-4As a user, I can register for the application through G mail I can register the app withe mail account Medium Sprint -1LoginUSN-5As a user, I can log into the I can register &access user High Sprint -1

application by enteringemail & password profile/account withGmailaccountRequesting/conferrer USN-6As aconferrer Ican requestvacant parkingspace to park my car I can getinformationabout parking ratesHighSprint -2Customer(Web user) profileUSN-7As a user Ican seeregistration page ,login page andChabot for I can check availabilityof parkingspots in realtimeI can loginthroughemailand socialmediaaccountfor registrationMediumSprint -2Customer CareExecutiveHelp desk /usersupportUSN-8As acustomercareexecutive Ican solvethe queriesof the usersI can replytotheir queries andsolvetheir related problemsHighSprint -3Administrator RegistrationUSN-9Asanadministrator,Icanviewthedatabase

oftheregisteredusersI can checkand verifythe personswho are theregisteredtheir mailid’s andinformation’

MediumSprint -4Dash boardUSN-10Asanadministrator ,I canview howmanyI can checkthe numberof requirementsandmonitor LowSprint -4

membersrequestedforwhattroubleoccursin parkingavehicletheavailabilitychatbot UserinterfaceUSN-11Inadditiontothecustomercareexecutivelcansolveallthequeriesofthecustomeras well astheconferrer I can reply toall thequestionswhich areasked by theusers that arerelated to theservice we providedMediumSprint -4

```
def main():

global bikes,cars,bicyclestry:

while True:

print(" ")

print("\t\tParking print(" ")

print("1.Vehicle Entry")

print("2.Remove Entry" )

 print("3.View Parked Vehicle")

print("4.View Left ParkingSpace ")

print("5.AmountDetails ") print("6.Bill")

print("7.Close Programme ")
```

CODING & SOLUTION:

FEATURE 1:

1.Add Vechicle Records

In this code block, we are importing thetimemodule to implement its methodsand function in the project.

We have initialized the variables vehicle number,vehicle type, vehicle name, owner name date, and time to some default value.

As well as bikes, cars, and bicycles with some initial value.

2.Create a while loop block to display the options inVehicle Parking Management Project

#Impor t TimeimporttimeVehicle_Number=['XXXX-XX-XXXX'] Vehicle_Type=['Bike']vehicle_Name=['Intruder']Owner_Name=['Unknown']Dat e=['22-22-3636']Time=['22:22:22']

ifch==1:no=True typee=Truewhile typee==True:

Vtype=str(input("\tEnter vehicletype(Bicycle=A/Bike=B/Car=C):")).

lower() if Vtype=="":

print("###### Enter Vehicle Type######") elif Vtype=="a":

Vehicle_Type.append("Bicycle") bicycles-=1typee=notTrue elif print("+ +") ch=int(input("\tSelectIn this code block,

we have initialized the bikes, cars, and bicycles as globalvariables. They are accessible through the entire main block. Here we are providing the options to choose the service options from the list, for the vehicle parking management system.

3.Code for vehicle number entry

Ch is for choice, Once we select the ch option as 1 which is for vehicle entrynumber, then we provide the while loop. while the number(no is True). We willstore the vehicle number in Vno.

If the vno is empty i.e vno==“”.The user asks to enter the vehicle number, else If the vno entered is already present in thevehicle number then it prints the vehicle number already exists. Else iflen(vno)==12, It will ask to append the info to the vehicle number variable.

5.Code to enter the vehicle type

name=Truewhile name==True:

vname=input("\tEnter vehiclename - ") if vname=="":

print("########Please Enter Vehicle Name########") else:

vehicle_Name.append(vname)o=Truewhile o==True:

OName=input("\t Enter owner- " ==”" bikes-=1typee=notTrueelif Vtype=="c":

Vehicle_Type.append("Car") cars-=1typee=not Here

we have to initialize the type variable to true. While the condition is True,the system asks to enter the vehicle type i.e a,b, or c which will accept the input in the lower case. Here A is for bicycle, B is for Bike and C is for Car.

Any vehicle type you enter is stored in the variable V type. If the V type==””(empty).It will ask to enter the vehicle type. According to the type of variable you enter the vehicle type will be stored in the variable and type variable is set to not True.

6.Code to enter the vehicle name

Here we have set the name== True. While the name == True i.e until we enter the name.v name store the value i.e. vehicle name. if the v name is empty system asks to enter the vehicle name, else it will store the name using the append function to the vehicle name variable The name variable is initialized to not True.

Code to enter the owners name

Prerequisites:

1. Python (version 3.x)

2. OpenCV (install using `pip install opencv-python`)

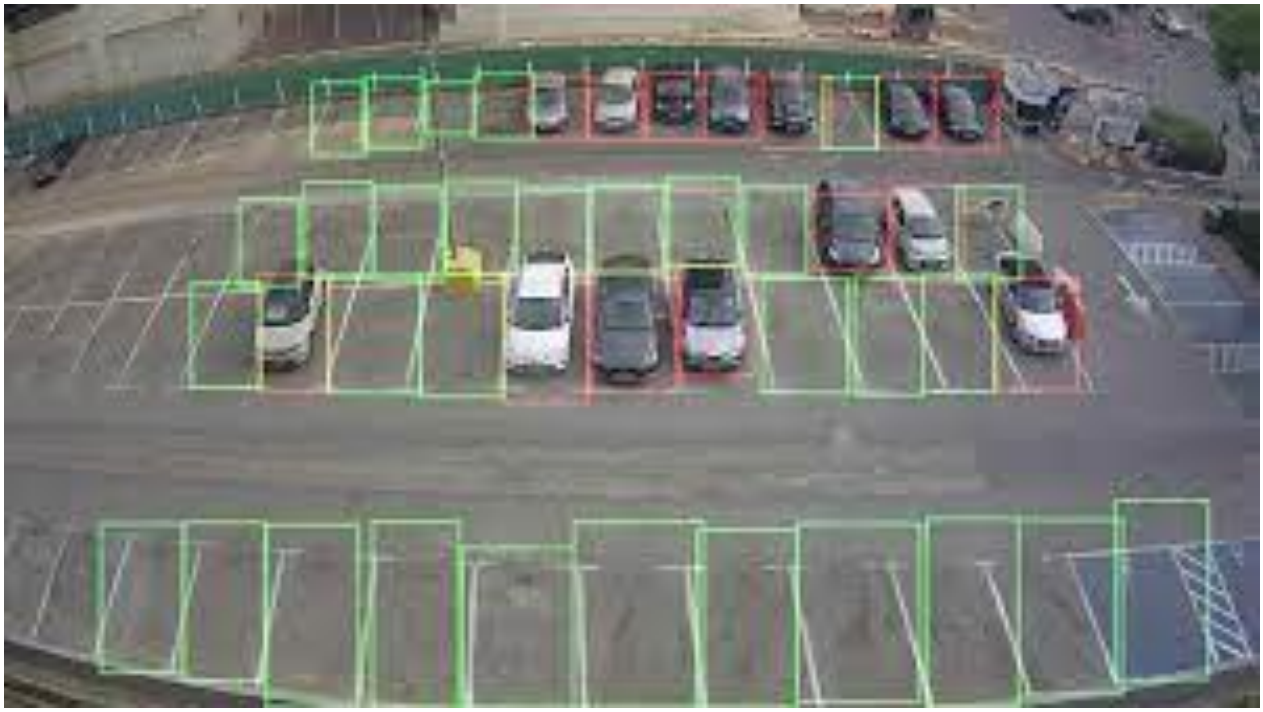
3. A video file or a live camera feed (you can use your webcam) for testing the system.

```python
import cv2


def detect_cars(video_path):
    # Load YOLO pre-trained model
    net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
    classes = []
    with open("coco.names", "r") as f:
        classes = [line.strip() for line in f.readlines()]

    # Get output layer names
```

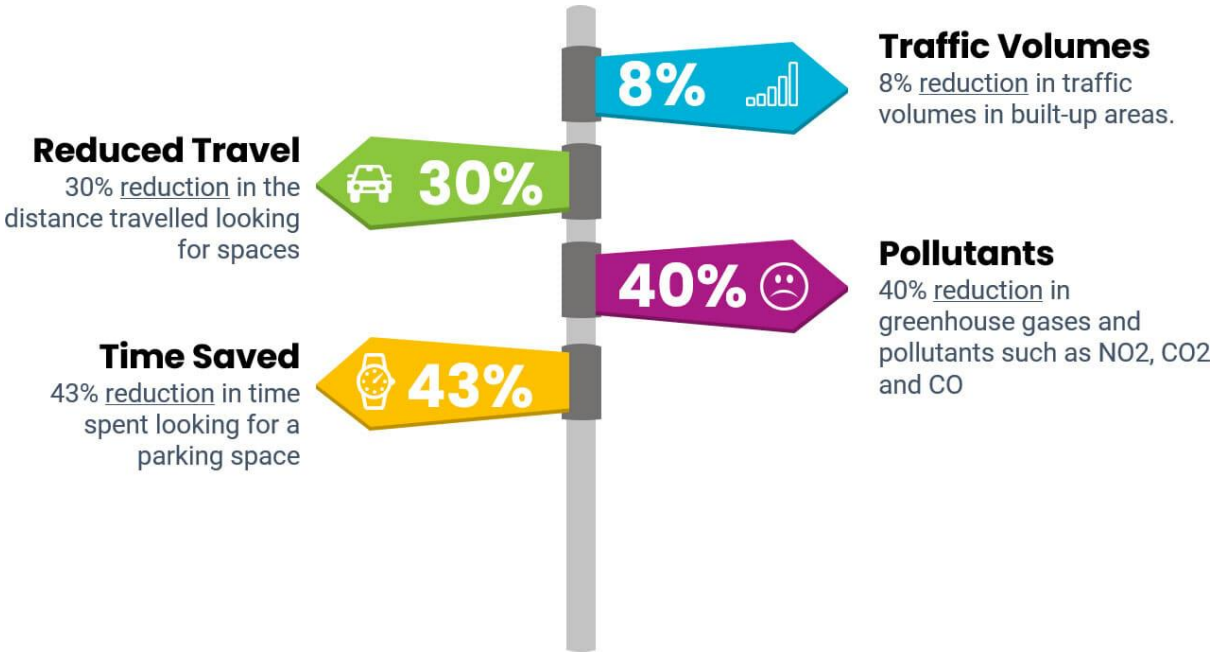
```
layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
# Read the video file or use live camera feed
cap = cv2.VideoCapture(video_path)
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    # Detect cars in the frame
    height, width, channels = frame.shape
    blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(output_layers)
    # Process detections
    class_ids = []
    confidences = []
    boxes = []
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5 and classes[class_id] == "car":
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)
                w = int(detection[2] * width)
                h = int(detection[3] * height)
                x = int(center_x - w / 2)
                y = int(center_y - h / 2)
                class_ids.append(class_id)
                confidences.append(float(confidence))
```

```
                boxes.append([x, y, w, h])

        # Non-maximum suppression to remove duplicate detections
        indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

        # Draw bounding boxes on cars and display available parking spaces
        for i in range(len(boxes)):
            if i in indexes:
                x, y, w, h = boxes[i]
                label = "Car"
                color = (0, 255, 0)
                cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
                cv2.putText(frame, label, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

                # To show available parking spaces, you can add more logic here
                # Calculate the distance of each car bounding box from reference points

        cv2.imshow("AI-Enabled Car Parking", frame)
        if cv2.waitKey(1) & 0xFF == 27:  # Press 'Esc' key to exit
            break

    cap.release()
    cv2.destroyAllWindows()

if __name__ == "__main__":
    video_path = "path_to_video_file.mp4"  # Replace with your video file or use 0 for live camera feed
    detect_cars(video_path)
```

RESULTS:



ADVANTAGES:



CONCLUSION:

As conclusion, the objectives of this project have been achieved. Thehassle in searching for available parking slots has been completely eliminated. The designed system could be applied everywhere due to its ease of usage and effectiveness. It facilitates the problems of urban availability, transportation mobility.

Environment sustainability,theInternet of Things integrates the hardware, software and network connectivity that enable objects to be sensed and remotely controlled across existing network .Such integration allows users to monitor availableand unavailable parking spots that lead to improved efficiency, accuracyand economic benefit.

FUTURE SCOPE: