AUTOMATED PARKING SPACE DETECTION

A project report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

E.SIVANI (N180202)

Under the Esteem Guidance of

Dr.D.V.NAGARJUNA DEVI

Assistant Professor, CSE Dept.

Rajiv Gandhi University of Knowledge Technologies – Nuzvid
Department of Computer Science and Engineering
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Eluru Dist - 521202

Rajiv Gandhi University of Knowledge Technologies – Nuzvid
Department of Computer Science and Engineering
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Eluru Dist - 521202

CERTIFICATE OF COMPLETION

This is to certify that the project entitled, "**AUTOMATED PARKING SPACE DETECTION**" is work of "**E.SIVANI (N180202)** " is a recent record of bonafide work carried out by under our guidance and supervision for the partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session,February2023-July2023 at RGUKT – Nuzvid. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

-----------------------	------------------------
Dr.D.V.Nagarjuna Devi,	Examiner,
Project Supervisor,	Department of CSE,
Department of CSE,	RGUKT – Nuzvid.
RGUKT – Nuzvid.	

Rajiv Gandhi University of Knowledge Technologies – Nuzvid
Department of Computer Science and Engineering
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Eluru Dist - 521202

CERTIFICATE OF EXAMINATION

This is to certify that the work entitled,”**AUTOMATED PARKING SPACE DETECTION**” is work of “**E.SIVANI (N180202)**” and here by accord our approval of it as a study carried out and presented in a manner required for its acceptance in third year of Bachelor of Technology for which it has been submitted.This approval does not necessarily endorse or accept every statement made,opinion expressed or conclusion drawn, as recorded in this thesis.It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

Dr.D.V.Nagarjuna Devi,
Project Supervisor,
Department of CSE,
RGUKT – Nuzvid.

Examiner,
Department of CSE,
RGUKT – Nuzvid.

Rajiv Gandhi University of Knowledge Technologies – Nuzvid
Department of Computer Science and Engineering
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Eluru Dist - 521202

DECLARATION

I **" E.SIVANI (N180202) "** hereby declare that the project report entitled done by me under the guidance of **Dr.Nagarjuna Devi** is submitted for the partial fulfillment for the award of degree of Bachelor of Technology in Computer Science and Engineering during the academic session **February 2023- July 2023** at RGUKT-Nuzvid.

I also declare that this project is a result of my own effort and has not been copied or imitated from any source.Citations from any websites are mentioned in the references. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Date:16-08-2023 **E.SIVANI(N180202)**
Place:Nuzvid

ACKNOWLEDGEMENT

I would like to express my profound gratitude and regards to my guide “ **Dr.D.V. Nagrjuna Devi** “ for her exemplary guidance, monitoring and constant encouragement to us throughout the B.Tech course.I shall always cherish the time spent with her during the course of this work due to the invaluable knowledge gained in the field of Data Science.

I am extremely grateful for the confidence bestowed in me and entrusting my project entitled**“AUTOMATED PARKING SPACE DETECTION”**.I express gratitude to Dr.Sadu Chiranjeevi (HOD of CSE) and other faculty members for being a source of inspiration and constant encouragement which helped me in completing the project successfully.

Finally, yet importantly,I would like to express my heartfelt thanks to our beloved God and parents for their blessings,my friends for their help and wishes for the successful completion of this project.

CONTENTS

ABSTRACT

Automated Parking Space Detection in computer vision aims to improve parking efficiency in modern cities by using technology to optimize parking space utilization and management, thereby reducing congestion and enhancing urban mobility.Automated parking detection aims detect available space for parking,optimizing space usage and reducing traffic congestion in urban areas, ultimately enhancing user experience. The proposed methodology involves segmenting the parking lot into rectangular blocks to specify parking spots,acquiring live footage from a top-down view camera of the parking lot, and employing image processing techniques to count the number of foreground and background pixels to determine the occupancy status of each parking spot. To mitigate the influence of environmental factors, a threshold value of 900 pixels is used to differentiate between foreground and background pixels. If the count of foreground pixels in a block exceeds 900, the spot is considered occupied; otherwise, if the count of foreground pixels is less than900,the spot is considered vacant. This system displays the number of available parking spots and total parking spots on a screen.

1.Introduction

Automated parking systems have become increasingly popular in recent years, as they offer a convenient and efficient solution to the growing problem of parking space scarcity in urban areas. One of the critical components of an automated parking system is the detection of parking spots and the determination of their occupancy status.This project aims to provide an automated parking detection system using OpenCV, a popular computer vision library that provides tools for image processing, object detection, and machine learning.The proposed system uses a bird-eye view camera to capture real-time video footage of the parking area,and the OpenCV library is used to perform various image processing techniques to identify and track vehicles in the parking lot.

1.1 Purpose

The purpose of this project is to use computer vision for real-time detection of parking spot occupancy. This helps optimize parking space use, reduce traffic congestion, enhance user experience, and showcase the potential of technology in addressing urban challenges.The system's core goal is to enhance urban mobility by minimizing traffic congestion and reducing the time drivers spend searching for available parking spots, ultimately contributing to improved traffic flow, reduced environmental impact, and increased convenience for individuals navigating busy urban environments.

1.2 Overview

The project involves building an automated parking detection system using computer vision. By analyzing real-time video data from a bird's-eye view camera, the system tracks vehicle presence in parking spots. The primary goal is to optimize parking space utilization, ease traffic congestion, and demonstrate technology's role in urban problem-solving. The project aims to enhance urban mobility by offering an efficient and practical parking solution.

1.3 Scope

The scope of this project is to create a functional prototype that demonstrates the system's effectiveness in optimizing parking space utilization and reducing traffic congestion, potentially leading to broader applications in urban mobility solutions.

2 Analysis of Existing and Proposed Systems

2.1 Problem Statement

Parking space scarcity in urban areas remains a persistent challenge, causing congestion and inefficiency.The project aims to develop an automated parking detection system using computer vision methods, capable of accurately identifying and tracking the occupancy status of parking spots in real time.The system's primary objective is to optimize parking space utilization, alleviate traffic congestion, and enhance the overall urban mobility experience by providing drivers with timely information about available parking spots. Through technological innovation and data-driven solutions, the project seeks to contribute to more efficient and sustainable urban transportation systems.

2.2 Existed System

The existing parking management systems typically rely on manual methods or simple sensors to monitor parking occupancy.These systems often lack real-time data accuracy and can be labor-intensive to maintain.Additionally,conventional payment methods and limited parking guidance contribute to inefficiencies and user dissatisfaction. The proposed automated parking detection system aims to overcome these limitations by utilizing computer vision to provide accurate real-time information on parking availability, ultimately improving space utilization and the overall parking experience.

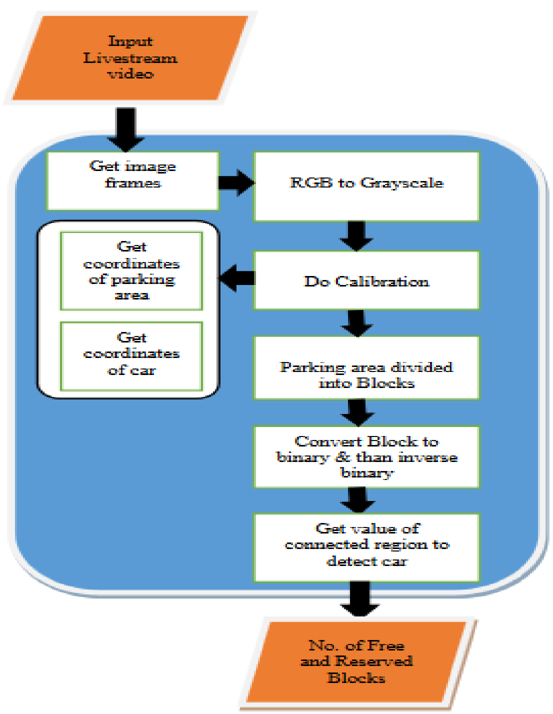
2.2.1 Disadvantages of Existed System

- <u>Limited accuracy</u>
  May provide inaccurate occupancy information,leading to incorrect parking guidance.
- <u>Lack of Real-Time Information</u>
  Traditional systems lack the ability to offer driver up-to-date parking availability data.
- <u>Lack of Data Insights</u>
  Existing systems cannot gather valuable data on parking trends for future planning.
- <u>User Dissatisfaction</u>
  Cumulative issues result in a frustrating parking experience, affecting user perception of the facility.

3. Proposed System

Considering the issues in the existing system our proposed method for the Automated Parking Space Detection project involves acquiring live videos from a top-view camera, segments parking spaces, and extracts keyframes for efficiency.It generates virtual parking lines based on vehicle size, assigns numeric labels, and processes real-time videos.

- Capture top-view live video of parking area
- Segment frames and extract keyframes for efficiency.
- Generate virtual parking lines, assign numeric labels.
- Process live video: grayscale, binary transformation.
- Use dilation for pixel classification and accuracy.
- Count white pixels in blocks, set occupancy threshold.
- Analyze blocks in rows/columns for occupancy status.
- Continuously update real-time parking availability.
- Adjust occupancy based on vehicle entry/exit.



3.1 Advantages of Proposed System

- Real-time detection guides users to available parking spaces, minimizing search time and congestion.
- User satisfaction and reduces frustration.
- Reduces the need for human intervention in monitoring.

4. Requirements

4.1 Hardware Requirements

Operating System : 64 bit windows
RAM : 4 GB(approx)
Disk : 500GB
Camera : Web Camera 10 Megapixels
Processor : I5

4.2 Software Requirements

Operating System : 64 bit windows
Code Editor : Jupyter Notebook, Python IDE,Visual studio code
Programming Language : Python

4.3 Technologies

4.3.1 Python

It is Commonly used for developing websites and software,task automation,data analysis and data visualization. In python we used different kinds of libraries.

4.3.1.1 Numpy

It is very useful for fundamental scientific computations in Machine Learning.NumPy is a very popular python library with the help of a large collection of high-level mathematical functions.

4.3.1.2 Cvzone

MediaPipe library is a versatile tool used for computer vision and machine learning tasks. It offers components for gesture recognition, facial recognition and tracking, object detection and tracking, augmented reality applications, body tracking, and audio processing. Developers can utilize MediaPipe's functionalities to build real-time multimedia processing pipelines for various applications.

4.3.1.3 Pickle

Pycaw library is a python interface that allows developers to interact with the core audio API in windows. It provides tools for controlling and managing audio devices including volume control, playback and device information retrieval. PYCAW simplifies audio related operations in windows applications by offering convenient access to the core audio functionality.

4.3.1.4 Opencv

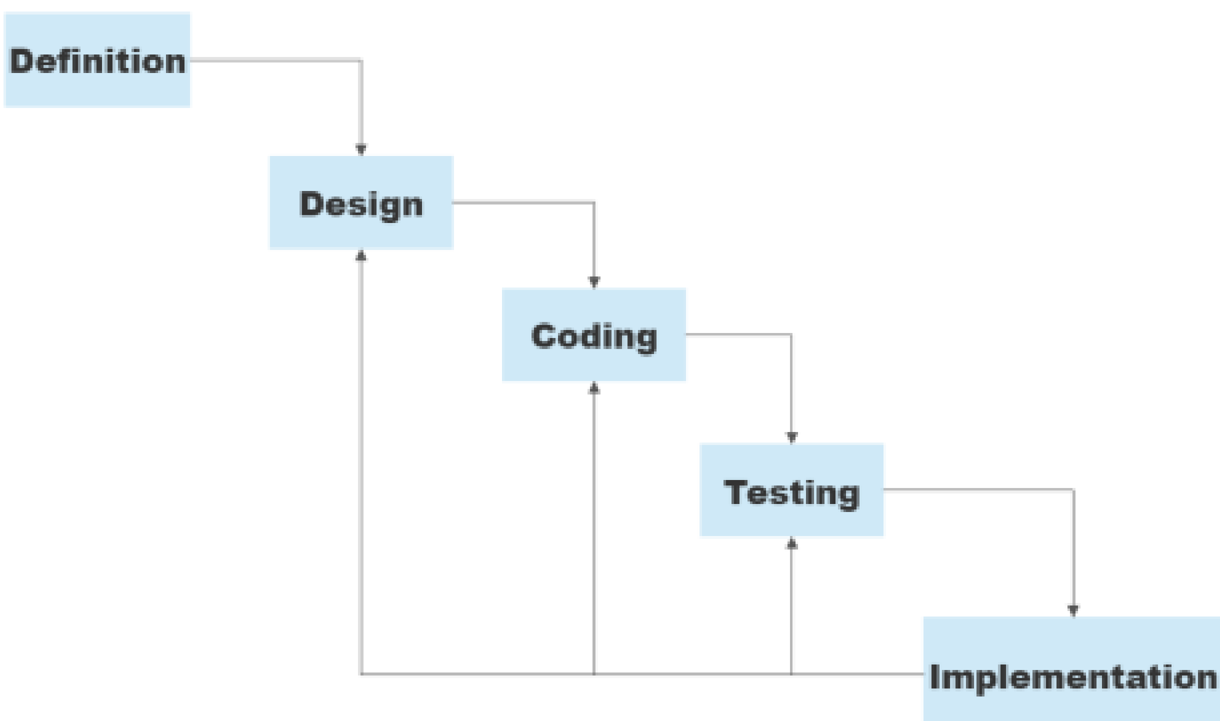
OpenCV is a Python library that allows you to perform image processing and computer vision tasks. It provides a wide range of features, including object detection, face recognition, and tracking. In this OpenCV Tutorial in Python, we'll be learning more about the library.

5.Methodology

Methodology is about the study of methods that have been used in order to complete a task or project. Thus, in this section, the project methodology will be discussed in two aspects of methodology, which is the system overview or the flow of the sub tasks and the development tools that have been used for developing the emotion aware music system. In the system overview, it would be more into the lifecycle model that had been identified as appropriate for the project while in the development tools section; it will give a brief explanation on what development.

5.1 Procedure Identification

As for this project, Iterative waterfall methodology has been considered to be applied as the methodology. An Iterative waterfall methodology structures a project into distinct phases with defined deliverables from each phase. The phases are definition, design, coding, testing and implementation stages. Below is the figure of the Iterative waterfall model.



5.1.1 Definition Stage

The first phase of this project is to try to capture what the system will do (its requirements) based on milestones. This is to ensure the system will be on track. Here, comprehensive research has been done to study the basic concept of the system itself. During this stage, the problem definition and problem statement of the system have been started.

5.1.2 Design Stage

The second stage determines how it will be designed. All the functionalities required are designed in this phase.

5.1.3 Coding Stage

In the middle of the stage the actual programming started. Here, the appropriate software and hardware tools as mentioned below are being applied in order to facilitate the project development process.

5.1.4 Testing Phase

The fourth phase is the full system testing where the stages of testing like unit testing, integration testing, system testing and user acceptance testing are involved in order to ensure quality of the system.

5.1.5 Implementation Stage

The final phase is focused on implementation tasks such as go-live, training, and documentation. Here, the documentation is being prepared to conclude on the overall research and experiment, which is basically to know which method can be applied.

6. Main Functionality

Automated Parking Space Detection is a python-based application, which helps the user or driver to show available parking space.Automated Parking Space Detection reduces traffic in urban areas and saves user’s time.

6.1 Video Feed Processing

This section likely involves acquiring live video feed from the camera installed in the parking area.Video feed processing includes capturing frames from the video feed and preparing them for further analysis.Pre-processing steps, such as grayscale conversion, Gaussian blur, and thresholding, might be performed to enhance image quality and isolate relevant features.

6.2 Loading Parking Space Positions

This section involves loading predefined parking space positions into the system.These positions serve as reference points for identifying individual parking spaces.The positions are typically defined as coordinates or rectangles in the image frame.

6.3 Image Processing

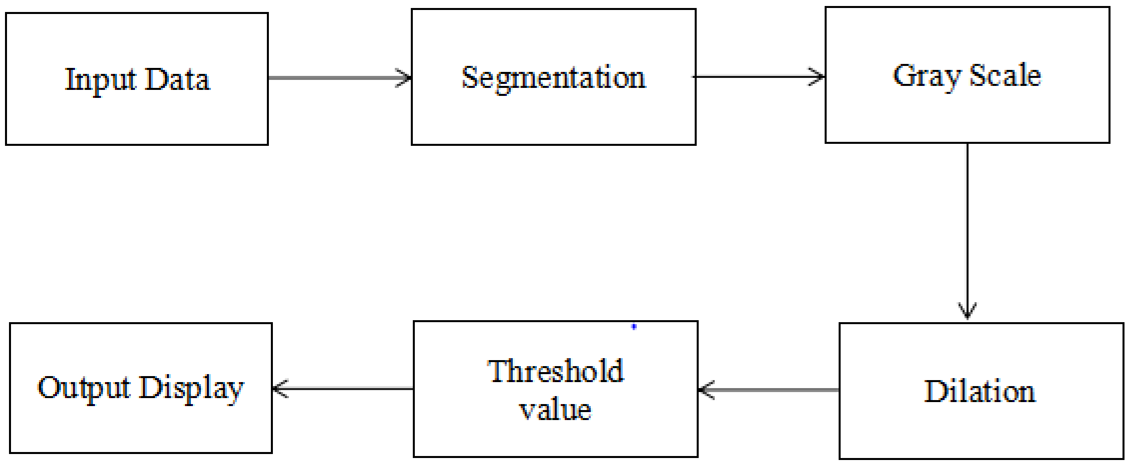
.Image processing is a crucial step where the pre-processed frames are further manipulated to detect and analyze parking spaces.Techniques like edge detection, dilation, and erosion might be employed to enhance features and segment the image.This step is aimed at creating a processed image that simplifies the detection of parking space occupancy.

6.4 Parking Space Detection

This involves using computer vision techniques to analyze the processed image and determine the occupancy status of each parking space.By comparing the pixel count or certain features within each parking space, the system can classify them as either occupied or vacant.The results of this analysis can be visualized on a display or communicated to users in real time.

7. Architecture

7.1 Architecture diagram



7.1.1 Camera Input

The methodology proposed is firstly,live footage is acquired from a top-down view camera of the parking lot.The camera is positioned such that it provides a clear view of all the parking spots in the lot.The captured video footage is subjected to pre-processing steps aimed at eliminating noise and artifacts that could potentially interfere with the accuracy of the subsequent detection algorithm.

By effectively removing noise and refining the input data, the methodology sets the foundation for precise parking space detection, enhancing the reliability and effectiveness of the algorithm used for this purpose.

7.1.2 Segmentation:

The parking lot is segmented into rectangular blocks to specify parking spots.The size of the blocks can be customized based on the size of the parking lot and the size of the vehicles that will be parked in the lot. The blocks are arranged in rows and columns to cover the entire parking lot.

7.1.3 Gray Scale:

The pre-processed footage is transformed into grayscale to convert into monotones. Thus a binary product is obtained.One important reason for gray scaling is that by converting to grayscale, it separates the luminance plane from the chrominance planes. Luminance is also more critical for distinguishing visual features in an image. This prevents any ambiguity in detection due to any presence of shadows. Thresholded to segment the foreground and background pixels. A threshold value of 900 is used to differentiate between foreground and background pixels. If the count of foreground pixels in a block exceeds 900, the spot is considered occupied; otherwise, if the count of foreground pixels is less than 900, the site is considered vacant.

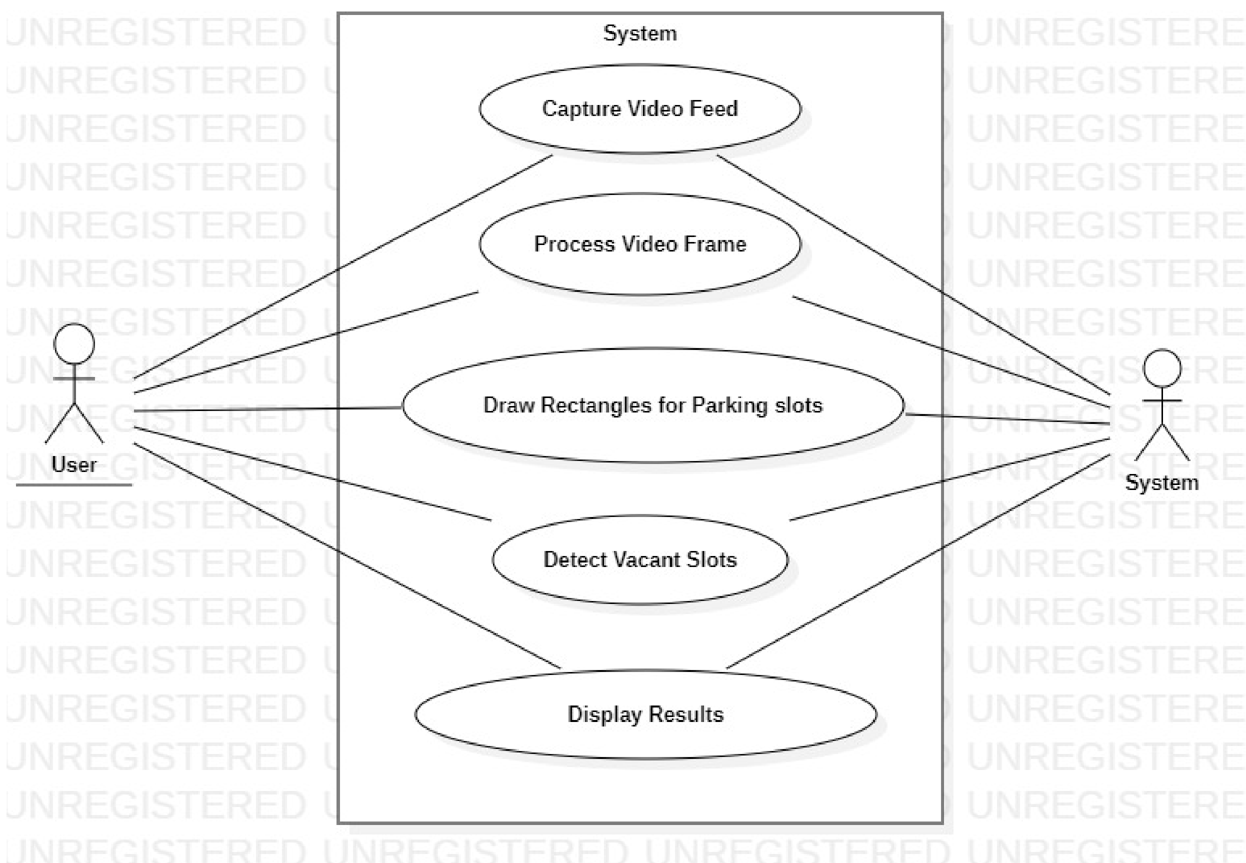
7.1.4 Dilation:

Dilation of an image is the process by which the object area in the image is increased and emphasizes their features. This process is used to accentuate features in the image. It increases the white region in the image or the size of the foreground object increases. This process helps in the classification of pixels black and white pixels.which is integral to the accurate identification of parking spaces and vehicle presence.

7.1.5 Threshold Value:

The hand gesture recognized is a pinch gesture, where the thumb tip and index finger tip come close together within a certain distance. A threshold is set to segment the foreground and background pixels.A threshold value of 900 pixels is efficiently set to differentiate between foreground and background pixels. If the count of foreground pixels in a block exceeds 900 pixels,the spot is considered to be occupied; otherwise,if the count of foreground pixels is less than 900,the site is considered vacant and is available for parking. After detecting the occupancy status of the parking lot.It displays the number of available parking spots to the total parking spots on a screen. Available spaces are marked in green and the occupied ones in red.

7.2 Use case diagram



8.Implementation code:

```python
# import image of parking space
# mark every parking space as Region Of Interest
# works with fixed camera
import cv2
import pickle

# store clicked rectangles


width, height = 107, 48

try:
    with open('CarParkPos', 'rb') as f:
        posList = pickle.load(f)
except:
    posList = []

def mouseClick(events, x, y, flags, params):
    if events == cv2.EVENT_LBUTTONDOWN:
        posList.append((x, y))
    if events == cv2.EVENT_RBUTTONDOWN:
        for i, pos in enumerate(posList):
            x1,y1 = pos
            if x1<x<x1+width and y1<y<y+height:
                posList.pop(i)
```

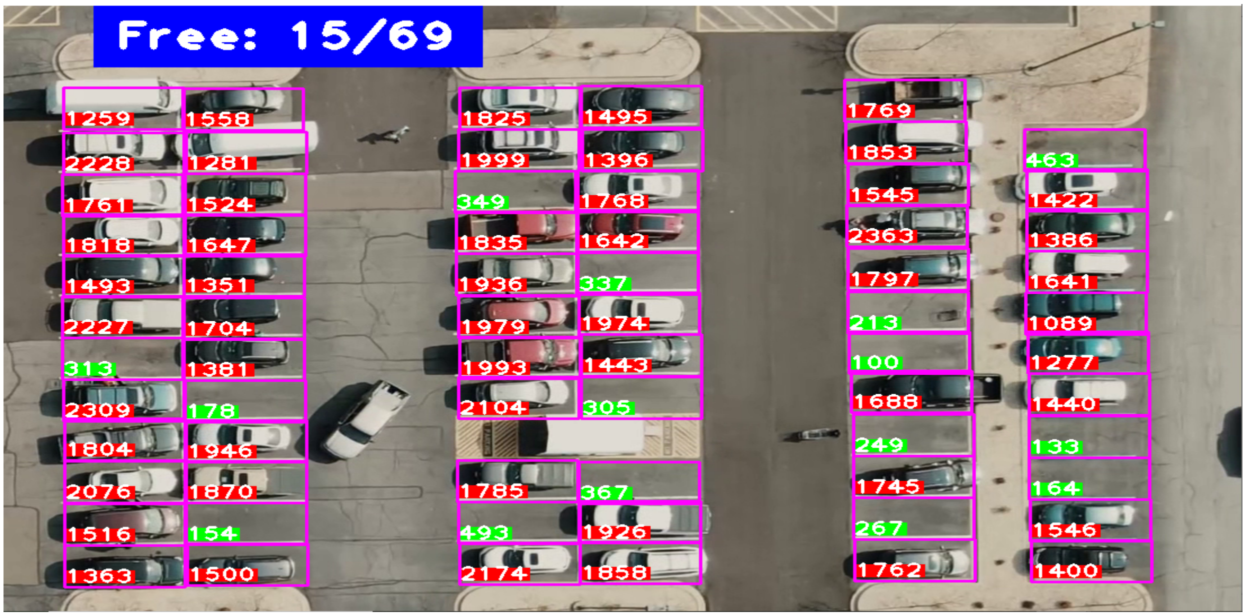
```python
    with open('CarParkPos', 'wb') as f:
        pickle.dump(posList, f)


while True:
    # create rectangle in exact positions
    # cv2.rectangle(img,(50,192),(157,240),(255,0,255),2)

    img = cv2.imread("./assets/carParkImg.png")
    for pos in posList:
        cv2.rectangle(img, pos, (pos[0]+width, pos[1]+height),(255,0,255),2)

    cv2.imshow("image",img)
    cv2.setMouseCallback("image", mouseClick)
    cv2.waitKey(1)

print("code executed")
```

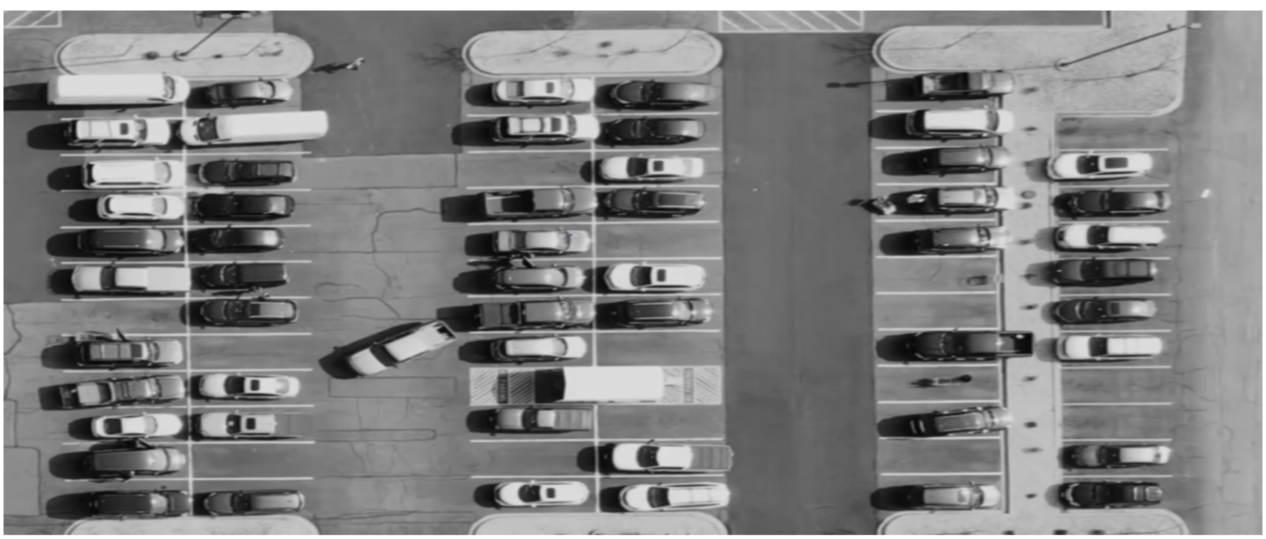
```python
import cv2
import pickle
import cvzone
import numpy as np

# video feed
cap = cv2.VideoCapture('./assets/carPark.mp4')

with open('CarParkPos', 'rb') as f:
    posList = pickle.load(f)

width, height = 107, 48


def checkParkingSpace(imgPro):

    spaceCounter = 0;

    for pos in posList:
        x, y = pos

        # cropping the images
        imgCrop = imgPro[y:y+height,x:x+width]
        # cv2.imshow(str(x*y), imgCrop)

        # count the pixels in each parking place
        count = cv2.countNonZero(imgCrop)
```

```python
        # put count of pixels of each place in rectangles
        cvzone.putTextRect(img,str(count),(x,y+height-5), scale=1.5, thickness=2, offset=0, colorR=(0,0,255))

        if count < 900:
            cvzone.putTextRect(img,str(count),(x,y+height-5), scale=1.5, thickness=2, offset=0,colorR=(0,255,0))
            thickness = 5
            spaceCounter += 1

        else:
            cvzone.putTextRect(img,str(count),(x,y+height-5), scale=1.5, thickness=2, offset=0,colorR=(0,0,255))
            thickness = 2
    # put count of pixels of each place in rectangles
    cv2.rectangle(img, pos, (pos[0]+width, pos[1]+height),(0,0,255), thickness)
    cvzone.putTextRect(img, f'Free: {spaceCounter}/{len(posList)}' ,(100, 50), scale=3, thickness=5, offset=20,colorR=(255,0,0))

while True:

    # infinite loop video
    if cap.get(cv2.CAP_PROP_POS_FRAMES) == cap.get(cv2.CAP_PROP_FRAME_COUNT):
        cap.set(cv2.CAP_PROP_POS_FRAMES,0)

    success, img = cap.read()
```

```python
imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
imgBlur = cv2.GaussianBlur(imgGray, (3,3), 1)

# converting image to binary (white lines on black bg)
imgThreshold = cv2.adaptiveThreshold(imgBlur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV,25,16)

# clear out the "noise" pixels
imgMedian = cv2.medianBlur(imgThreshold, 5)

kernel = np.ones((3,3), np.uint8)
imgDilate = cv2.dilate(imgMedian, kernel, iterations=1)

checkParkingSpace(imgDilate)

for pos in posList:
    cv2.rectangle(img, pos, (pos[0]+width, pos[1]+height),(255,0,255),2)

cv2.imshow("Image", img)
cv2.imshow("ImageBlur", imgBlur)
cv2.imshow("ImageThresh", imgMedian)

# slows down the video
cv2.waitKey(10)
```

9.Output


Free: 15/69
1259
2228
1761
1818
1493
2227
313
2309
1804
2076
1516
1363
1558
1281
1524
1647
1351
1704
1381
178
1946
1870
154
1500
1825
1999
349
1835
1936
1979
1993
2104
1785
493
2174
1495
1396
1768
1642
337
1974
1443
305
367
1926
1858
1769
1853
1545
2363
1797
213
100
1688
249
1745
267
1762
463
1422
1386
1641
1089
1277
1440
133
164
1546
1400



Results comparison with other approaches:

Traditional Sensor-based Systems:

Conventional systems relying on ultrasonic or magnetic sensors may be prone to interference and environmental factors,leading to inaccuracies.The proposed visual-based approach offers better accuracy

Deep Learning Methods:

Deep learning techniques, such as convolutional neural networks (CNNs), can achieve high accuracy but might require extensive training data and computational resources. The proposed methodology offers real-time performance without necessitating a massive training dataset..

10. Conclusion

Automated parking analysis using OpenCV is an efficient and accurate method for detecting the occupancy status of parking spots in a parking lot. The proposed methodology involves segmenting the parking lot into rectangular blocks, acquiring live footage from a top-down view camera, and applying image processing techniques to count the number of foreground and background pixels to determine the occupancy status of each parking spot. The use of a threshold value to differentiate between foreground and background pixels mitigates the influence of environmental factors, resulting in an accurate detection algorithm. The developed app provides real-time updates of the occupancy status of each parking spot,enabling users to find available parking spots quickly and easily.

11.References

1.Car Parking Space Detection using opencv.

(https://easychair.org/publications/preprint_open/hVnT)

2.Automated Parking Space Detection

(https://ijirt.org/master/publishedpaper/IJIRT159290_PAPER.pdf)