**CAR PARKING SPACE DETECTION SYSTEM**

**Prasanna devi1,Thrikuda devi2,Sanjay3,Praveen4**

*Department of artificial intelligence and data science,anna university,india*.

***Abstract :***

***The global trend of increased urbanization makes space the***

***rare in city environments in general and for parking in the***

***particular. In addition, cars become bigger and often use of***

***more than one parking space. As a result neighbouring parking spaces can be affected by a parked car. So, a basically free parking space might be too narrow for an arriving car depending on the arriving car’s size. Therefore, means to detect car positions on parking spaces in a fine granular way are required to detect such situations and avoid inefficient parking space searches.Wireless sensor networks provide the possibility to sense the exact occupation of a parking space and potential influences on neighbouring parking spaces. However, current solutions focus only on the detection if a parking space is occupied or not. In our work, we present a sensor deployment and a machine learning-based approach able to provide the mentioned more fine-granular detection level. We have conducted an extensive real-world evaluation of our solution, in particular considering different characteristics of today’s car bodies. In our tests, our approach achieved an accuracy of more than 98%***

***Keywords:***

***Opencv,space detection,yolo,numpy***

# INTRODUCTION

Parking space management is a major problem in today's metropolitan surroundings, where common problems include traffic jams, inefficiency, and driver annoyance. Automated vehicle parking detection systems have surfaced as a viable remedy for these issues, providing real-time parking space optimization and monitoring. In order to effectively manage parking spaces, the goal of this project is to construct a Car Parking Detection System utilizing Python and OpenCV (Open Source Computer Vision Library).By automating the process of identifying open parking spaces, the Car Parking Detection System seeks to completely transform conventional parking management. The system is able to identify occupied and vacant parking spaces in real-time by analyzing video feeds from cameras deployed in parking lots through the use of machine learning algorithms and advanced image processing techniques.

# RELATED WORKS

This research presents a machine learning and computer vision integrated smart parking system. It makes use of machine learning methods to forecast the availability of parking spaces using learning methods to forecast the availability of parking spaces using OpenCV for vehicle identification and categorization. Real-time parking guidance and management features are provided by the system. These connected papers demonstrate the many strategies and tactics used in the creation of computer vision and Python-based car parking detection systems. Through the utilization of OpenCV and other cutting-edge technologies, scholars and programmers persist in their innovative efforts to effectively tackle the obstacles related to parking management.

# TECHNIQUES USED

* Image Preprocessing
* Object Detection
* Feature Extraction and Description
* Machine Learning and Deep Learning
* Post-processing
* Graphical User Interface

## IMAGE PREPROCESSING

Image Thresholding: The parking lot image can be divided and objects of interest (cars, markings, etc.) are able to differentiate from the background. utilizing methods such as binary thresholding, adaptive thresholding, or Otsu's thresholding.

Noise reduction: To remove noise and smoothen the image, techniques like Gaussian blurring or morphological operations (erosion and dilation) are used. This improves the accuracy of future processing steps.

## OBJECT DETECTION

Mapping contours in an image can be used to find the boundaries of objects, such as automobiles and parking space signs. OpenCV provides techniques for this reason. Prior-trained Haar Cascade classifiers can be applied to identify automobiles in the parking lot picture.

## FEATURES EXTRACTION AND DESCRIPTION

Vehicle identification and classification can be helped by the use of Histogram of Oriented Gradients (HOG) descriptors, which can be computed to extract features from vehicle images.   
Local Binary Patterns (LBP): To help distinguish between parking spaces and other items, LBP features can be used to characterize textures and patterns inside the parking lot image.

## MACHINE LEARNING AND DEEP LEARNING

Support Vector Machines (SVM): Using extracted features, SVM classifiers can be trained to recognize if parking spaces are occupied or vacant.

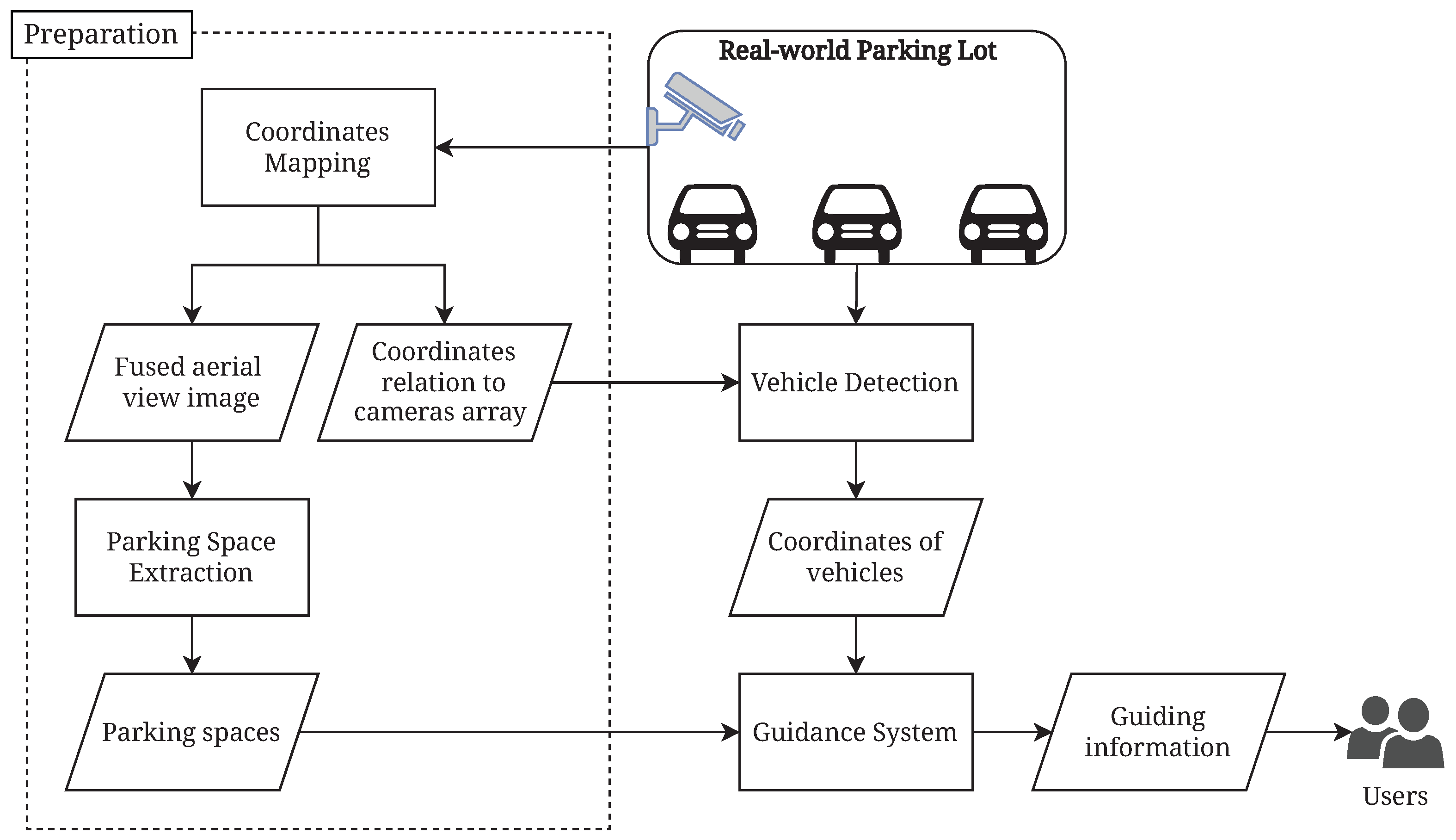
Convolutional Neural Networks (CNNs ) : Real-time object identification, including the detection of vehicles and spots to park, can be realized by using CNN architectures such as YOLO (You Only Look Once) or SSD (Single Shot MultiBox Detector).

## POST PROCESSING:

Bounding Box Refinement: During object detection, redundant detections can be reduced and localization accuracy improved by filtering bounding boxes around vehicles and parking spots using methods like non-maximum suppression (NMS).   
Measuring Occupancy: To estimate occupancy and predict the availability of parking spots, additional analysis could involve tracking noted vehicles over time.

## GRAPHICAL USER INTERFERANCE:

kinter or PyQt: A user-friendly GUI for showing the parking lot image, determined items, and parking space status in real-time can be created with Python libraries like Tkinter or PyQt.   
By using all of these techniques, an efficient Car Parking Detection System with great effectiveness and dependability may be created using Python and OpenCV. This system can safely and accurately detect and monitor parking places.



**4.REAL TIME DETECTION**:

The ability of a system to process and evaluate data in real-time, usually with little delay or latency, is referred to as real-time detection. Real-time detection in car parking detection systems that use Python and OpenCV entails continually analyzing video feeds from cameras placed in parking lots promptly accepting both filled and vacant parking places as cars come and go.   
  
In these kinds of systems, real-time detection involves many significant steps:  
  
  
1. \*\*Data Acquisition\*\*: The parking lot cameras supply video feeds to the system. These cameras record footage of the spots for parking continuously, giving the detecting process's input data.   
  
2. \*\*Preprocessing\*\*: To improve quality and lower noise, incoming video frames undergo preprocessing. methods like color space conversion, image resizing.

3.Object Detection: Within the video frames, items of interest (cars) are found using computer vision techniques. For accurate and efficient identification, this step usually uses methods based on deep learning, like CNN s (Convolutional Neural Networks), HOG (Histogram of Oriented Gradients), or Haar cascades algorithms.   
  
   
  
4.categorizing: Observation items are divided as either vehicles or backdrop components (such signs or impediments). In order to improve accuracy and refine the detection results, this phase might need further processing.  
  
  
5.Space Occupancy Determination: The system determines each parking space's occupancy state based on the vehicles that are observed. The system differentiates between occupied and vacant parking spaces by examining the size and position of the vehicles in relation to the boundaries that have been established.   
  
6.Output Generation: A real-time output displaying the parking spaces' state is generated by the system. Users are able to get instant feedback on parking availability by using GUIs, which are.   
  
For successful parking management, real-time detection is crucial due to the fact that it gives drivers and parking operators the flexibility to make quick choices based on the most recent information affecting the availability of parking spaces. Real-time detection systems optimize parking space use, reduce traffic, and boost overall parking operations efficiency by continuously tracking parking spaces and offering real-time updates.

**5.OPEN CV**

Originally created by Intel in 1999, OpenCV (Open Source Computer Vision Library) is an open-source software library for computer vision and machine learning. For processing photos and videos, it offers a broad range of tools and algorithms for tasks including object detection, tracking, segmentation, and recognition. Since OpenCV is built in C++ and includes bindings for Python, Java, and MATLAB, an extensive variety other mechanisms and programming languages can use it.

**5.1 APPLICATION OF OPEN CV:**

Due to its excellent characteristics and ability to adapt, OpenCV (Open Source Computer Vision Library) has a wide range of applications. Here are a few common applications for OpenCV:

Video Analysis and Processing: OpenCV makes it easier to do tasks like motion estimation, optical flow, background subtraction, and video stabilization. Applications like augmented reality, activity recognition, video presenting, and video surveillance all make use of it.   
  
Tools for stereo vision, depth estimation, camera calibration, and three-dimensional in nature reconstruction are included in OpenCV. Applications such as autonomous navigation systems, robots, augmented reality, and 3D scanning make advantage of thesecapabilithy.

Machine Learning Integration: Scikit-learn, PyTorch, TensorFlow, and other machine learning libraries can be easily incorporated with OpenCV. It is employed in the creation of machine learning models for applications like as identification of patterns, object detection, picture segmentation, and image classification. By automating the process of identifying open parking spaces, the Car Parking Detection System seeks to completely transform conventional parking management. The system is able to identify occupied and vacant parking spaces in real-time by analyzing video feeds from cameras.

**6.IMPLEMENTATION:**

**PYTHON:**

Python: Suitable for web development, computer vision, and machine learning, Python is a popular and responsive programming language with a large collection and structures.

**OpenCV**:

With a wide range of tools for processing images and videos, object detection, feature extraction, and other tasks, OpenCV is a strong open-source computer vision toolkit. Because it has Python bindings, linking it with Python-based apps is simple.

**DJANGO:**

Django are popular python web development frameworks that can be used to build web based interfaces for the parking space detection system.these frame works provides tools for handling http requests,rendering dynamic web pages and managing user authentication and sessions.

**HTML, CSS, and JavaScript** :

required technologies for developing interactive and visually attractive user interfaces when developing a web-based application. To make the development process go more quickly, you can use frontend frameworks like React or Bootstrap.

You may create a reliable and effective Python automobile parking space detection system that can accurately identify and manage parking spaces in real-time with the help of this technological stack.

**Numpy:**

Numpy is a fundamental packages for numerical computing in python.It provides support for large,multi dimensional arrays and matrices,along with a collection of mathematical functions to operate on these array efficiently

**Pickle:**

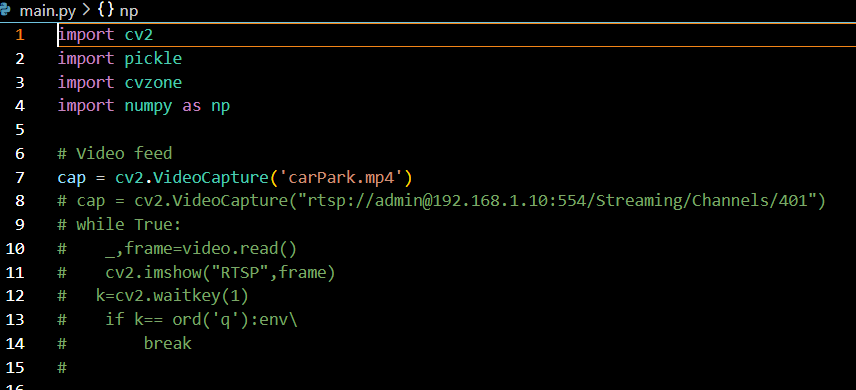
The serialization module pickle, which is used to encode and deserialize Python objects, is known as "pickle" in Python.

Fig.1 input picture

Figure.1 shows that our project input . In this we imported the cv2,pickle,cvzone,numpy.cv is know as the computer vision it

Is used to detect the space for parking.pickle is used for making

Our model as folder and numpy we know it is a numerical python.

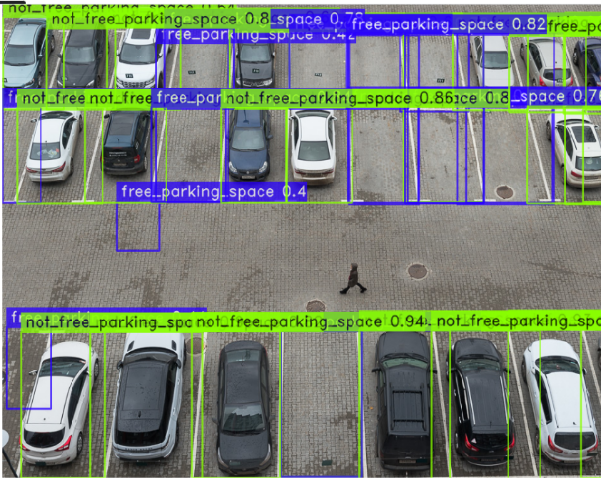
****

Fig.2 the output picture

Figure.2 shows the the green colour box shows the free

Space for parking and the blue colour shows the not free

Space for parking and we easily utilize the the place for

Parking.

**EXISTING PROJECT:**

Develop a Python parking management system project. Parking spaces may be handled successfully and effectively with the help of the Python parking management system project. The management team's workload is reduced because it automates the parking management process. The admin and users can access real-time updates through such simple to use system. Also, it speeds up the parking process and simplifies the whole parking experience.

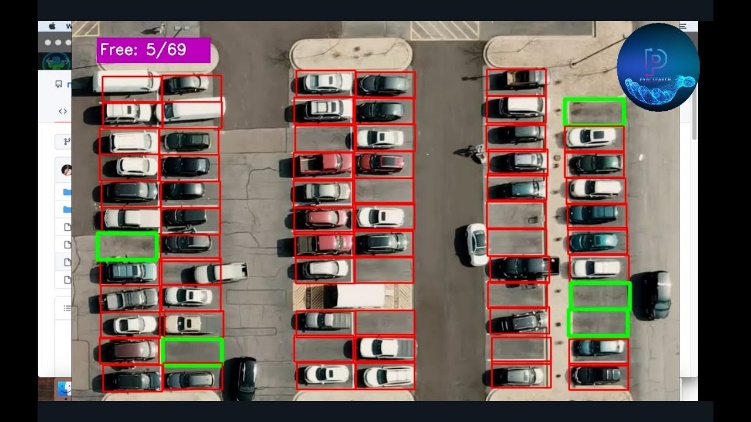
****

**PROPOSED PROJECT:**

One of the most annoying things happening in our era is looking for a parking spot and not being able to find one, especially when you had to be somewhere five minutes ago and you are currently looking for a parking spot for 20 minutes now.However, any problem must have a solution, or more solutions, based on complexity and efficiency. There are many solutions for smart parkingsystems out there, including deep learning implementations, weight sensors, lightsensors and all ofthat science fiction stuff that surrounds the world these days.This article is focusing on guiding you trough one of the simplest approaches to a smart parking system using only a webcam and few lines of code**.**

Problem Definition finding out the empty parking spaces in a car parking lot automatically from a surveillance camera. solution Extracting the parking lot coordinates from the image by car\_park\_coordinate\_generator.py script. Then use these coordinates to process every car parking space individually. Implementing digital image processing techniques to find out empty and occupied parking spaces. drawing the results into the image.

Finding the parking spots is the first stage in a parking space detecting system. There are multiple techniques to do this. Such as locating the parking lines in a space to identify the parking slots. The edge detectors that OpenCV offers can be used for this. However, the issue here is that these previously determined limits are missing from every parking spot.  
  
Believing that the cars in question are parked in their spots is another strategy we could use. Put differently, legal parking spots are just spots with parked cars. However, this also doesn't seem to be safe. Both true negatives and false positives could result from it.  
  
When automation fails to be honest, what should we do? We carry it out by hand. We just need to mark out parking lot boundaries and surrounding road areas once in order to configure our system for a new parking facility, in opposition to space-based techniques that call for labeling and training for each unique parking facility.   
  
Here, we will recognize the parking areas by marking a frame from our video or broadcast that shows the parking site. The PolygonSelector technique is provided by the matplotlib Python library. Here, it satisfies our needs to the letter. It offers the ability to choose what regions of the polygon to select.  
To identify the polygon regions on one of the first frames of the input video, I created a basic Python script. It uses the video's journey as support for its claims, and

****

Recognize that designing a system that can identify parking spaces in cars can be challenging and need skills in software development, machine learning, and computer vision. When applying such technologies in real-life situations, take regulatory compliance, privacy, and data security into thought as well.

**CONCLUSION:**

In conclusion, there are a lot of advantages and chances for effective parking area management when creating a car parking space registration system with Python and computer vision (CV) approaches. This sys maximizes parking space utilization while offering users convenience byeffectively detecting parking space availability in real-time through the use of machine learning models and image processing techniques.   
  
With the help of Python libraries like TensorFlow and OpenCV, we can build a scalable and reliable system that can manage a variety of the parking situations. A Parking lot managers can effectively the manage parking resources and keep an eye on occupancy levels thanks to the inclusion of CV algorithms, which provideextremely precise understanding of both occupied and vacant parking spaces.

**REFERENCE:**

[1] Björn Lundell,Simon Butler,Thomas Fischer,Jonas Gamalielsson,Christoffer Brax. .Effective Strategies for Using Open Source Software and Open Standards in Organizational Contexts.IEEE access, DOI: 10.109/MS.2021.3059036

[2] Ivan B. Djordjevic, the Hybrid of CV -DV Quantum Communications and Quantum Published in: IEEE Access Volume: 10). DOI: 10.1109/ACCESS.2022.3154468.

[3] Christine Dewi;Rung-Ching Chen;Yan-Ting Liu;Xiaoyi Jiang;Kristoko Dwi Hartomo; Yolo V4 for Advanced Traffic Sign Recognition With Synthetic Training Data Generated by Various GAN. Publisher: IEEE. Volume: 9;

[4] Daniel Padilla Carrasco;Hatem A. Rashwan;Miguel Ángel García;Domènec Puig; .T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks; Publisher: IEEE; Volume: 11;

[5] Abubakar Sharif;Jinhao Guo;Jun Ouyang;Sheng Sun;Kamran Arshad;Muhammad Ali Imran;Qammer H. Abbasi; Compact Base Station Antenna Based on Image Theory for UWB/5G RTLS Embraced Smart Parking of Driverless Cars; Publisher: IEEE; Volume: 7

[6] Syrus Gomari;Rohith Domakuntla;Christoph knothConstantinos Antoniou; .Development of a Data-Driven On-Street Parking Information System Using Enhanced Parking Features; Publisher: IEEE; Volume: 4;

[7] Gijs A. G. M. Hendriks;Gert Weijers;Chuan Chen;Madeleine Hertel;Chi-Yin Lee;Peter M. Dueppenbecker;Marcus Radicke;Andy Milkowski;Hendrik H. G. Hansen;Chris L. de Korte; Comprehensive Comparison of Image Quality Aspects Between Conventional and Plane-Wave Imaging Methods on a Commercial Scanner; Publisher: IEEE; Volume: 69, Issue: 6;

[8] Matthias Joachim Ehrhardt;Simon R. Arridge; Vector-Valued Image Processing by Parallel Level Sets; Publisher: IEEE; Volume: 23, Issue: 1