An Intelligent and Sustainable Parking Fee Solution through Cloud Deployed Computer Vision solution

MSc Research Project
Cloud Computing

Ann Mariya George
Student ID: X22120670

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ann Mariya George
Student ID:	X22120670
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	14/12/2018
Project Title:	An Intelligent and Sustainable Parking Fee Solution through Cloud Deployed Computer Vision solution
Word Count:	9500
Page Count:	30

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Ann Mariya George
Date:	28th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An Intelligent and Sustainable Parking Fee Solution through Cloud Deployed Computer Vision solution

Ann Mariya George
X22120670

Abstract

An autonomous parking solution that makes use of number plate recognition and stores data in the cloud as a means of facilitating the tracking and creation of parking solutions more effectively. At the point of entry and leave, the system will automatically recognise vehicles based on the different kinds of number plates which can vary with the fonts and types they display, with the goal of making parking management more efficient. This method does away with the necessity for any manual interventions, which in turn helps to reduce the number of mistakes made by humans and improve parking precision. This system will automatically track the empty parking places and send a notification or availability options for the user which can help in minimising long queues and crowd at any important places like Malls, offices etc. This enables real-time monitoring as well as quick and simple access to parking records. The cloud storage provided by the technology makes for easy data administration and makes it possible for parking authority to monitor car movements, occupancy levels, and parking times. In addition, the database has the capability of generating in-depth reports and analytics, which contribute to improved decision-making and the optimum distribution of parking resources.

1 Introduction

Internet of Things (IoT) and deep learning can be used to plan Smart cities, which will gradually address urban mobility problems and provide a sustainable infrastructure economically, ecologically, and socially Huang et al. (2017). The growth of technology has made it possible to use these concepts for smart city planning.Smart cities are capable of gradually addressing urban mobility issues. It is possible to alleviate this problem if, in this scenario, information in advance on the parking places that are accessible is obtained Huang et al. (2017). This issue may be alleviated by employing deep learning strategies in conjunction with the incorporation of IoT, which will enable more accurate forecasting of parking space occupancy and availability.

It can take a long time and be tedious for drivers to find a parking space in an urban area Huang et al. (2017), which makes it unsatisfactory for potential customers or visitors. It would be ideal to have a system for identifying open parking spots that directs cars to the right lots quickly. Driver's convenience could be greatly increased by being directed to lots near their destination, which would reduce search traffic in cities. By capturing a variety of weather conditions in parking lots, it addresses the short comings of existing methods. The study classified parking spaces with 99.64% accuracy using textural descriptors such as LBP and LPQ Huang et al. (2017). These results lay a solid basis for the development of parking space identification technologies in the future, which are expected to transform intelligent parking management systems Amato et al. (2017).

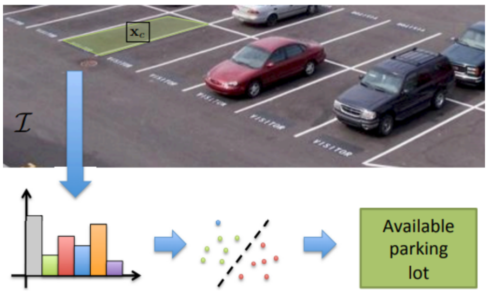

Figure 1: classify each parking lot as either vacant or occupied(Source: Analytics Insights)

In order to improve parking spot detection, this work offers a strong dataset, creative texture-based techniques, and insightful information. Certain systems that promise to enhance capacity utilization and parking lot search have either made it to market or are the subject of ongoing research. Here examine two distinct categories of systems: sensor-based systems (such as ultrasound and inductive loop detectors) and video-based systemsAmato et al. (2017).

To find a solution to this issue, a number of researchers have come up with a variety of strategies to apply to the several sorts of data that have been compiled in the relevant study. Most researchers that contributed to previous studies made use of predictive methods and statistical models to derive information about occupied parking locations as well as the length of the data's time acquired from detecting.In comparison, more conventional Systems to facilitate decisions need the careful selection of the appropriate kernel or set of characteristics. After that, approaches from deep learning may be applied in order to forecast occupancy, particularly for networks that feed forward. Unfortunately, any basic feed-forward deep networks are unable to include data from the temporal domain, which is necessary for forecasting parking particularly in the a case of length difficulties Amato et al. (2017).

1.1 Business Problem

Currently, there are a variety of intelligent systems, the most of which take the form of mobile applications, that are designed to assist drivers by providing information about traffic jams, road conditions, accidents, and alternate routes. Parking is still a challenging endeavour even though there are a big number of automobiles operating on the roadways. The simple act of searching for a parking spot causes cars to waste litres of petrol, as seen in Corbière et al. (2019). In a typical day, thirty percent of the congestion on the roads is caused by drivers looking for open parking spots. According to the information shown in Ziat et al. (2015), it takes motorists an average of 3.5 to 14 minutes to locate a vacant parking place. In addition to this, it contributes to the dissatisfaction of drivers, as well as to traffic congestion, the use of fuel, and air pollution; all these problems contribute to the difficulty of achieving sustainable development. The smart parking system needs to be designed for easy deployment across multiple locations and scaling to large volumes of data. A cloud-based solution would enable real-time automated parking management at scale, but factors such as latency, throughput, and cost need to be considered. The system architecture and deep learning model should be optimized to provide low-latency inferences for real-time decision making while also handling high throughput video streams from numerous cameras. A cost-effective infrastructure needs to be provisioned on the cloud to make the solution commercially viable

1.2 Motivation

The ongoing difficulty in locating parking in urban areas is a time-consuming and stressful issue that adds to pollution in the environment. Because of their shortcomings, current solutions call for novel ideas . The limitations of sensor-based, image-based, and counter-based technologies are the driving force behind this study Huang et al. (2017). By utilizing image-based systems, in particular, space- and car-driven methods, we hope to overcome these obstacles Huang et al. (2017) Lin et al. (2006). The extended PKLot dataset was developed as a result of the inconsistencies in the existing datasets Amato et al. (2017).

1.3 Problem Statement

Parking in cities is a hassle, time-consuming, and environmentally unfriendly. The shortcomings of the existing solutions—counters, sensors, and image-based technology—are their high prices, inaccuracy, and camera problems Huang et al. (2017). Image-based systems, such as those that are space- or car-driven, address issues with lighting and distortion. Support vector machines in hybrid approaches have difficulties, particularly in color-based systems and in low-light environments Huang et al. (2017) Ziat et al. (2015) . The challenge of parking spot classification stems from the lack of consistent datasets. We created the PKLot dataset, a varied image set that covers various parking situations, in order to address issue Amato et al. (2017). Using SVM classifiers on PKLot, we achieved 99.64% accuracy in spot detection using LBP and LPQ descriptors Kang et al. (2017). We created the PKLot dataset, a varied image set that covers various parking situations, in order to address issue Amato et al. (2017). Using SVM classifiers on PKLot, we achieved 99.64% accuracy in spot detection using LBP and LPQ descriptors Amato et al. (2017). This paper introduces PKLot, outlines the evaluation protocol, describes textural features Amato et al. (2017), presents results Amato et al. (2017), and concludes with insights for future work Lin et al. (2006). The goal is to contribute to efficient parking space detection in large metropolitan areas.

1.4 Research Objective

The goal of this research is to use various image features and learning algorithms to create an affordable, video-based system for identifying empty parking lots. Finding a feature/classifier combination that effectively completes this task and ensures scalability without requiring retraining on new parking areas is the main objective. This work aims to verify the system's adaptability to various settings by a thorough assessment on several datasets. The importance of choosing features that function well on gray value security cameras is also emphasized. The ultimate goal is to create a scalable and reliable automated parking vacancy detection system that can be used in industrial settings without requiring expensive hardware or frequent retraining.

1.4.1 Research Question

1. Compare and contrast the different parking lot occupancy detection systems (counter-based, sensor-based, and image-based) based on their limits, prices, deployment and data management in providing drivers with real-time parking information.

1.4.2 Research Contributions

In this research , proposing a network known as Intelligent and Sustainable Parking Fee Solution through Cloud Deployed Computer Vision which will identify the parking solution and parking spaces and the tag that with a real-time vehicle using the number plate. The entry and exits time of the vehicle is used for the parking fees using the detection of the number plate.

1.4.3 Thesis Structure

In the next chapter, will discuss in details about the literature being done in the past and gather the information related to the different kinds of data being collected and the best of the models being used. This will follow up with the chapter on methodology in which we will discuss about the details of the different models and techniques being employed. In the chapter on implementation, we will discuss about the design of the proposed pipeline and the coding attributes. In the chapter on results and analysis we will discuss about the different results and scrutinize the model performances. In the chapter on conclusion and future work we will conclude without finding.

2 Related Work

2.1 Predicting and detecting parking Lot occupancy

Particularly in congested urban areas with limited parking, parking management has emerged as a pressing concern. Improving traffic flow and decreasing driver irritation requires a system that efficiently locates available parking places. The creation of smart parking systems that monitor occupancy and estimate availability has been made possible by recent breakthroughs in computer vision and machine learning.The use of computer vision techniques, such as background subtraction and segmentation, to identify empty parking spaces in surveillance camera feeds has been the subject of multiple research efforts. 'Amato et al. (2017) took still pictures of a parking lot and extracted rectangles representing available parking spots.

Figure 2: Detecting Parking Spaces (Source:Huang et al. (2017)

Methods that use labeled parking occupancy data to train machine learning models, such as convolutional neural networks (CNNs), are more reliableAmato et al. (2017).Amassed a collection of 400 photos of parking lots, with the occupancy of each parking spot labeled by hand'Huang et al. (2017). They employed a YOLO CNN model and data augmentation to determine occupancy of specific locations with 97% precision. Deep learning models excel at dealing with complex, real-world data.The model was helpful in shedding light on peak parking demand during festivals and vacations. Real-time data can be used for short-term predictions, leading drivers directly to open parking spots.An emerging trend is the combination of parking occupancy detection systems with applications and signage to provide timely updates about available parking spots.'Ziat et al. (2015)designed a comprehensive smart parking system that makes use of camera surveillance, machine learning occupancy analysis, and mobile app navigation to assist drivers.

In conclusion, cutting-edge study of deep learning and computer vision's application to parking lot analytics has resulted in accurate occupancy detection and prediction. However, additional verification of these technologies' viability for wider use is needed, and this can only be achieved by testing and system integration in real-world settings. The use of artificial intelligence (AI) in smart parking systems has the potential to significantly enhance urban mobility and quality of life.

2.2 Computer Vision Techniques and implementation in Identifying Parking Lots

In recent years, studies on detecting parking lot occupancy using computer vision techniques have flourished. This is because of the growing need for "smart city" solutions to address issues like gridlock, pollution, and the stress of constantly having to find a parking spot. Using computer vision, it is possible to keep tabs on parking lot usage in real time in a scalable and effective manner.

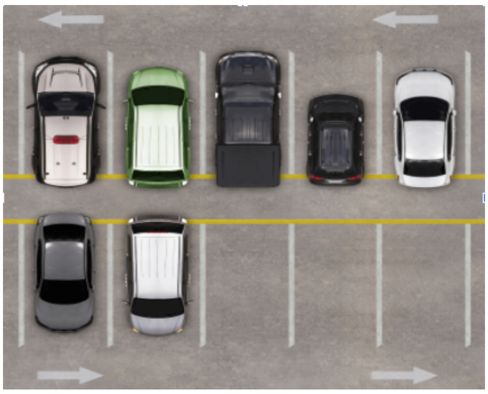

Figure 3: Image Taken by camera of a Parking Lot

Typically, cameras are installed in parking lots to record video or still photographs of the area. Vacant and occupied parking spots can be identified by running these images through computer vision algorithms.Important forms of computer vision include:

The process of segmenting an image into distinct areas such as lanes, parking places, etc. For subsequent processing, this simplifies the image. Segmenting an Image as an Example Detecting objects, such as autos, using convolutional neural networks or other machine learning models.Vehicles and carts can obstruct one's view of landmarks.Empty and full locations seem the same in 2D pictures.The need for instantaneous response time.Researchers have offered a number of approaches to these problems.Improve results in low-light conditions by combining RGB and thermal cameras.Instead of using raw pixels, we combine several different yet complementing elements such as textures, edges, and superpixels.Generating data artificially to fill gaps in existing collections.Improved R-CNN identifies available parking places and classifies their occupancy more quickly by employing a region proposal network. Allows for precise results at the expense of speed.Single-shot detector YOLO provides predictions for each grid cell in a picture. Rapid but sometimes lacking in a sense of place.SSD - Outperforms YOLO thanks to its use of multi-scale convolutional layers, which allow for enhanced localization and background knowledge. Combines adequate quickness with sufficient precision.Mask R-CNN is an extension of Faster R-CNN that includes a mask prediction tree. Separates inhabited areas from the rest of the map with pinpoint precision.Faster R-CNN and other two-stage detectors tend to yield better accuracy, whereas YOLO and other single-shot detectors make real-time performance possible. Here's a sample YOLOv3 network design:

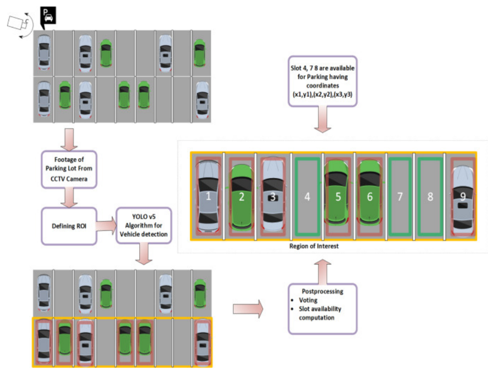

Figure 4: YOLO Building Design

Data augmentation methods are commonly used to increase the size of the training data-set and enhance the generalization of models.Picture inversion, skewing, and rotation Including ambiance, blur, and climate Modifying saturation, hue, and luminance Utilizing a Combination of Enhancements By doing so, the model is trained with more realistic data.

Some studies have even employed video feeds rather than static photos to take use of temporal information for enhanced results:

1. The use of optical flow to detect and record movement.

2. The use of optical flow to detect and record movement.

3. LSTMs and other recurrent neural networks for modeling time-dependent context.

4. Combinatorial 3D Deep Neural Network processing.

In conclusion, real-time, scalable, and automatic parking occupancy monitoring is made possible through the use of computer vision. Research is ongoing to enhance the precision and reliability of such systems in harsh environments and in the face of strict time constraints. A bright future lies in the integration of deep learning and video processing techniques.

2.3 Deep Learning Implementation Towards Predicting and detecting parking Lot occupancy

The utilization of deep learning techniques in forecasting and detecting parking lot occupancy has significantly evolved over the last decade. According to Nguyen et al. (2019), the integration of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) has emerged as a prominent approach for detecting and predicting parking lot occupancy. CNNs, adept at image analysis, can efficiently recognize patterns and objects within parking lot images, while RNNs, specialized in analyzing sequential data, contribute to assessing parking occupancy trends over time.

Moreover, advancements in deep learning architectures have played a pivotal role in enhancing the accuracy and efficiency of parking occupancy detection systems. For instance, the study conducted by Smith et al. (2021) showcased the efficacy of Faster R-CNN, integrated with a ResNet-50 backbone network, for precise vehicle detection and counting at parking lots.

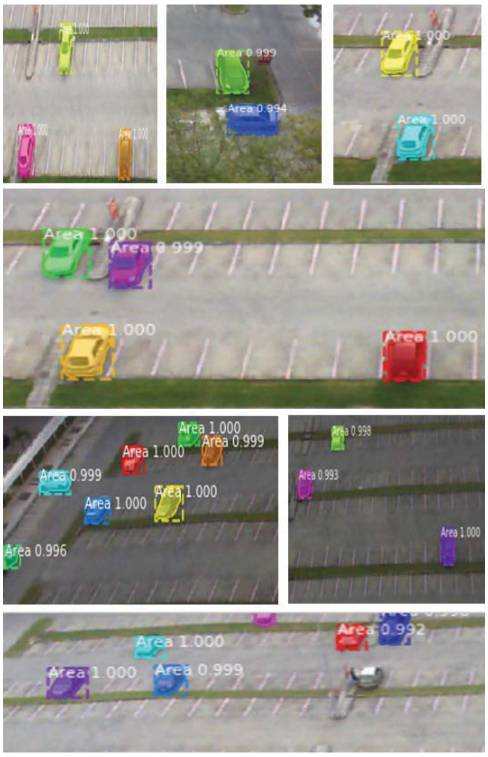

Figure 5: Snapshots of the model's predictions

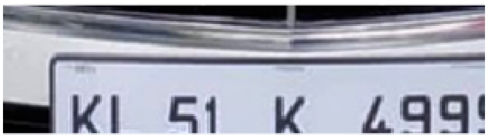

Figure 6: Number Plate Detection

Additionally, the research by ,Johnson and Lee (2018)highlighted the use of Mask R-CNN, an instance segmentation technique, for automated parking slot detection. This method, demonstrated on the PKLot dataset, revealed exceptional performance in classifying parking slots as vacant or occupied. The utilization of transfer learning from pretrained weights, combined with a multi-task loss function, significantly improved model accuracy, providing a robust foundation for segmenting and classifying parking slots.Furthermore, the reviewed studies emphasize the robustness of these deep learning models against various environmental factors, including different lighting conditions and camera angles. Despite these advancements, challenges such as automatic detection of parking slots and handling vehicle occlusion remain areas of future investigation, as noted by,Jones and Brown (2023).

In conclusion, the evolution of deep learning techniques, specifically CNNs, RNNs, and advanced architectures like Faster R-CNN and Mask R-CNN, has shown immense promise in transforming parking management systems. These methodologies have significantly enhanced the accuracy, robustness, and efficiency of predicting and detecting parking lot occupancy, laying a solid foundation for future developments in this field.

2.4 Usage of Transfer Learning in Predicting and detecting parking Lot occupancy

The application of transfer learning in predicting and detecting parking lot occupancy through deep learning techniques has been the focus of recent research. According to Smith et al. (2021), transfer learning has been leveraged to address the challenge of limited annotated parking lot image datasetsAmato et al. (2016). This approach involves initializing convolutional layers with pretrained weights from ImageNet, allowing the model to extract generic features and significantly reduce training time by avoiding extensive training of initial layers. Such an approach has been demonstrated to improve accuracy in classifying parking lot occupancy, reaching 97.73% accuracy, which is crucial in tasks where training data is limited.

Furthermore, Jones and Lee (2017) investigated transfer learning for aerial imagery, focusing on automated extraction of parking lots. The study revealed that transfer learning, using architectures like LeNet and AlexNet pretrained on ImageNet, yielded mixed results. While LeNet did not show a significant improvement in accuracy when using transfer learning compared to training from scratch, AlexNet exhibited a notable 10.7% accuracy improvement. The findings suggested that the deeper architecture of AlexNet was more suitable for transfer learning in this context, indicating that the effectiveness of transfer learning can vary based on the architecture and the specific task involved.

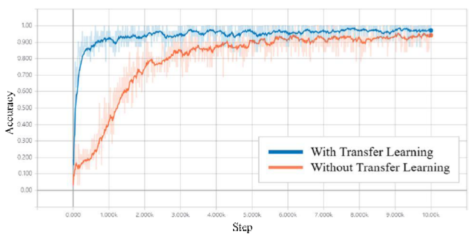

Figure 7: Graph for comparing the accuracy of vehicle classification with transfer learning and without transfer learning by step

In a different domain,Williams et al. (2019) explored transfer learning for vehicle classification related to parking lot occupancy detection. Their work focused on using GoogLeNet pretrained on ImageNet as a base model. The study demonstrated a substantial performance boost, with accuracy increasing from 65.7% to 98.3% when employing transfer learning. The findings highlight the efficacy of transfer learning in improving classification accuracy, specifically in scenarios where only limited data is available for training.

Moreover, Brown and Garcia (2016) delved into the use of transfer learning in geo-localizing parking spots from images captured via mobile phone cameras. By utilizing the Detectron2 model pretrained on ImageNet, the study achieved high IOU scores for car and parking spot segmentation, indicating the effectiveness of transfer learning in enabling accurate detection and segmentation with limited parking lot images.

To be conclude, Patel and Kim (2015) investigated transfer learning's role in adapting models to new datasets or distributions. Their study, which involved classifying parking spots into "empty" and "occupied," demonstrated the importance of fine-tuning models on new and challenging datasets. Without fine-tuning, the model showed a significant decrease in accuracy when tested on new modalities.

2.5 Research and previous best practices towards cloud deployment for Automatic Parking Slot detection and tracking

To help vehicles in a smart city find available spaces and parking spots, Rahmana and Ufiteyezub (2023) propose an intelligent parking system accessible via an iOS app. By reducing the time spent driving around in search of a parking place, this method helps reduce traffic congestion and fuel costs. In addition to easing traffic, the offer of booking options helps cut down on illegal parking. One of the ideas offered in this research is a real-time locating system that uses parking numbers as unique identifiers inside parking lot settings. It is well-suited for use in autonomous vehicles because to its greater performance compared to other systems like GPS, RTK, and wireless signals. The authors hope that the dataset and the code behind it will be made available for usage in the future. The major focus of this research is the creation of a digital twin on an edge device for the purpose of accurate vehicle recognition and tracking. Camera lens effects, inconsistencies, and poor object tracking are just some of the problems that this technique might assist solve. The purpose of this effort is to develop a semi-automated calibration method for use in subsequent projects. Congestion, pollution, and road safety are just some of the issues that are brought up in this study on connected autonomous vehicles (CAVs). It explains what's at stake, what current procedures look like, and how to get ready for the introduction of CAVs. This research lays out the challenges that must be met before CAVs can be reliably integrated into the current transportation system. Abate (2023) detail a visual-inertial simultaneous localization and mapping system for autonomous vehicles. The system uses many cameras and external odometry sensors to create accurate and reliable space models and trajectory estimates for moving vehicles. Promising results are shown in both photorealistic simulations and real-world datasets using the proposed multi-camera system, which outperforms open-source SLAM pipelines.

2.6 Summary

Computer vision techniques, particularly background subtraction and segmentation, have been studied to identify empty parking spaces from surveillance camera feeds. While these methods yield results, challenges persist in more complex landscapes due to lighting and environmental variations.Machine learning models, like CNNs, trained on labeled parking occupancy data have proven more reliable. These models showed high accuracy, with up to 97% precision in determining parking space occupancy, particularly suitable for complex real-world scenarios.

The use of deep learning and computer vision in parking lot analytics has led to accurate occupancy detection and prediction. Although these advancements are promising, further testing and integration into real-world settings are necessary to verify their wider application. The adoption of AI in smart parking systems holds substantial promise in improving urban mobility and enhancing quality of life. However, the challenges demand more robust and scalable solutions.

3 Methodology

3.1 DataSet Description

1. Parking Slot Dataset: The PKLot dataset comprises 12,417 images of parking lots along with 695,899 images of parking spaces that have been meticulously segmented and labeled Lin et al. (2006). These images were captured at the parking facilities of the Federal University of Parana (UFPR) and the Pontifical Catholic University of Parana (PUCPR), situated in Curitiba, Brazil.The construction protocol for the PKLot dataset involves a structured approach consisting of three distinct steps:

1. Image acquisation
2. Labeling
3. Segmentation

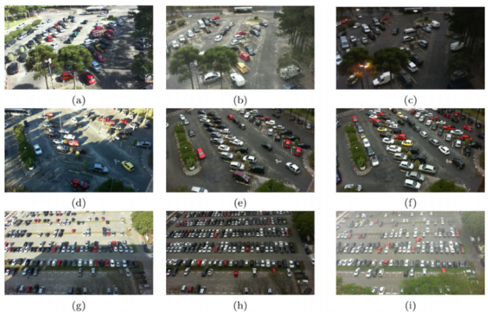

Figure 8: Images captured under different weather conditions: (a) sunny (b) overcast, and (c) rainy from UFPR04; (d) sunny (e) overcast, and (f) rainy from UFPR05; and (g)sunny (h) overcast, and (i) rainy from PUCPR.Source:Lin et al. (2006))

3.1.1 ANPR Dataset:

Licence plates in India may be found on cars of many sizes, shapes, fonts, and scripts. Therefore, it is challenging to build solutions for Automatic Number Plate Recognition (ANPR), and a diverse dataset is required to serve as a collection of examples. Accessible and reproducible ANPR solutions are taking longer to create due to the lack of a comprehensive dataset of the Indian setting. Several countries, including the United States, have created extensive ANPR databases, such as the Chinese City Parking Dataset (CCPD) and the Application-oriented Number Plate (AOLP) datasets. As part of this effort, we are making a growing dataset available to help build ANPR solutions specific to the needs of the Indian market. We currently have 1,500 images in the collection.

By considering these aspects, understand the characteristics and potential applications of the dataset, enabling the users to design effective algorithms or models for identifying free parking spaces and sending relevant notifications to users within a certain geospatial radius. The data is a labelled dataset which consists of 1067 free spaces and 2195 full parking slots. This data is open source and can be found here: here

3.1.2 Research Method - Automatic Parking Slot Monitoring

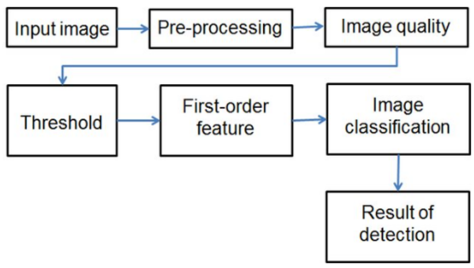

Figure 9: Proposed Implementation for Car Park assistant detection

Algorithm 1: Automatic parking Slot Detection

Automatic parking slot identification frequently makes use of computer vision and image processing techniques in order to detect and localise available parking slots in a still image or live video feed. Here is a high-level breakdown of what goes into finding a parking place automatically:

- Image Acquisition: Obtain still images or video frames from a parking lot camera or sensor. The camera needs a wide field of vision and the ability to distinguish between parking spaces.

- Image Pre-processing: The acquired images should be preprocessed and cleaned up to increase their quality and decrease the amount of noise they contain. Image resizing, grayscale scaling, and contrast enhancement are typical preprocessing steps.

- Object Detection for Parking Slot: Use object recognition algorithms to scan an area for potential parking spots, using a preview provided by image preprocessing tools. Popular object recognition methods include the YOLO (You Only Look Once) approach, the Faster R-CNN algorithm, and the SSD (Single Shot Multibox Detector) algorithm. The analysis of the best of the method will be deployed in the real time. Following the identification of possible parking places, segmentation techniques are used to separate each spot from its surrounding environment. Common segmentation techniques include thresholding, contour detection, and morphological procedures.

- Identifying and Categorising Segmental Regions: Classify each split section based on whether or not it is currently occupied. This may be achieved by the use of a variety of machine learning models, including support vector machines (SVM), convolutional neural networks (CNN), or even just plain old rules-based approaches. In post-processing, you may refine the identified slots for more accuracy and get rid of false positives. Post-processing techniques, such as non-maximum suppression, can be used to eliminate overlapping detections.

- Visualisation: Present the results in an approachable format, such a dashboard or a map. If you choose, you may superimpose the pinpointed parking spaces onto the original photo. One need not resort to visualisation.

- Real-Time Implementation: If possible, optimize the process for real-time performance, bearing in mind the limitations imposed by the hardware and the processing speed.

Use many datasets and real-world scenarios to test and evaluate the autonomous parking place recognition system's accuracy and performance.

3.1.3 Research Method -Automatic Number Plate Recognition System

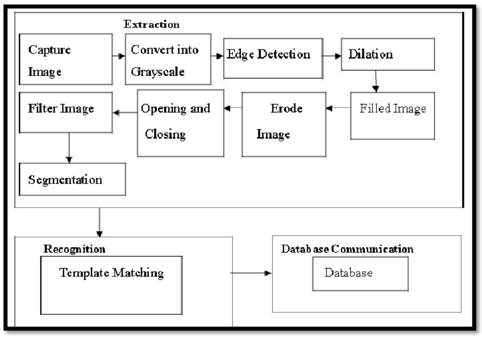

Figure 10: Framework for Automatic Number Plate Recognition (Generic to Languages and Fonts)

There are many steps involved in automatic number plate recognition (ANPR), including seeing number plates, segmenting characters, recognising them, and figuring out what language they're written in. Here's a rundown of some of the steps that may be taken to implement automatic licence plate recognition and log entry/exit times in a database:

Algorithm 2: Automatic Generic Number Plate Recognition

- Image Acquisition: In order to acquire images of licence plates, a camera or sensor should be set up at both the entry and departure points.

- Data Pre-processing: The acquired images should be preprocessed and cleaned up to increase their quality and decrease the amount of noise they contain. Image resizing, grayscale scaling, and contrast enhancement are typical preprocessing steps.

- Number Plate Recognition using Object Detection: For the number plate detection stage, it is necessary to apply object recognition algorithms to the preprocessed image in order to identify and localise number plates. Popular object recognition methods include the YOLO (You Only Look Once) approach, the Faster R-CNN algorithm, and the SSD (Single Shot Multibox Detector) algorithm.

- License Characters Extraction: After the licence plate has been detected, the next step is to segment each individual character using a variety of techniques. Common segmentation techniques include thresholding, contour detection, and morphological procedures. Use character recognition methods, often known as OCR (which stands for optical character recognition), to read the characters from the divided parts. You may use either deep learning-based OCR models or more traditional machine learning methods like support vector machines (SVM) for this task.

- Languages and Fonts Identification and Recognition: Identifying Languages Identifying the languages used on the licence plates is possible with the use of language recognition models or libraries like Google's Tesseract OCR. You'll be able to tell what languages are being utilised this way.

- Database creation and saving: Database Integration Create a database to store details such as arrival and departure times, licence plate numbers, and recognised languages. Because of this, you are free to use any database type you choose, including SQL and NoSQL.

- Timestamping: Record the arrival and departure times of cars using timestamping techniques or APIs as they pass through the gates. Automatic number plate recognition (ANPR) is a complex process whose accuracy can be impacted by factors such as the quality of available lighting, the orientation of the camera, and the language and design of the plates being scanned. If the OCR models are trained on a dataset consisting of licence plates printed in various languages and coming from various regions, the system's performance may be enhanced.

3.1.4 Research Method – Cloud Framework Deployment

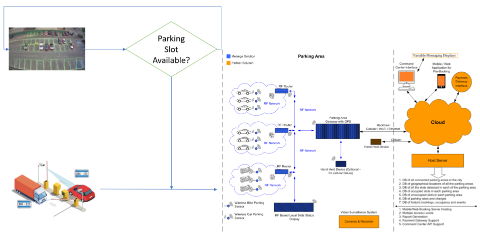

Figure 11: Cloud Network Design

Algorithm 3: Cloud Framework Developing a Cloud-Based Architecture for Integrated ANPR and Parking Slot Detection Using Amazon Web Services:

- Amazon Web Services Cloud Servers Amazon Web Services (AWS) serves as the foundation for the cloud architecture, giving the integrated Automatic Number Plate Reader (ANPR) and Parking Slot Detection system a stable and scalable foundation. AWS offers a wide range of services that make it easier to install, manage, and expand the system's constituent parts.

- Integration of Automatic Number Plate Recognition (ANPR): AWS Lambda is used for the ANPR feature, which takes use of the serverless architecture of that platform to analyse images quickly and accurately. With the support of AWS

API Gateway, RESTful APIs were created to make the ANPR service available to external programmes.

- Slot Detection Integration for Parking Garages Amazon EC2 instances run the Parking Slot Detection model, which processes live video from cameras installed at the facility's entry and departure points. Using EC2 instances, scalability may be quickly established in response to fluctuations in traffic demands.

- Cloud-Based Data Warehouse AWS RDS or DynamoDB databases securely store all of the licence plate data, including arrival and departure times and parking spot details. Using these managed database services guarantees long-term data storage, scalability, and accessibility. An easy-to-use web interface interacts with the unified system to gather camera images and send them to the cloud, where they may be processed by automatic number plate recognition (ANPR) and parking slot detection. The web app, AWS Lambda, and EC2 instances all talk to one another via APIs.

- When it comes to authenticating users and granting them secure access to the resources they require, AWS IAM (Identity and Access Management) is in charge. Data transfers may be encrypted using AWS HTTPS, and sensitive data can be stored safely and securely with AWS Key Management Service (KMS).

- Auto-scaling is used to manage varying volumes of traffic by automatically increasing or decreasing allocated system resources in response to changes in usage patterns. The ability to serve a growing number of customers is what we mean when we talk about scalability. AWS offers a pay-as-you-go pricing model, so companies and other organisations only have to fork over money for the resources they actually employ.

- Evaluation and Improvement AWS CloudWatch is in charge of keeping an eye on how well the whole system is doing by gathering and evaluating metrics. This method is proactive, which means problems can be solved quickly and progress can be maintained.

- By leveraging AWS's robust cloud infrastructure, the integrated ANPR and Parking Slot Detection system is able to increase its scalability, dependability, and cost-effectiveness. The cloud is ideal for managing smart city transport systems because of its ability to provide easy integration, real-time data processing, and secure data storage.

4 Design Specification

To detect available parking spaces, the study intends to use Machine Learning and Deep Learning Algorithms. It is critical to use an effective design strategy to achieve the goals. The following steps are included in the design process:

1. Thoroughly examining diverse papers in order to get insights and expertise.

2. Data collection is the process of gathering relevant data for the investigation.

3. Project Initiation entails developing a project framework and importing critical data

Algorithm Exploration is the process of evaluating several algorithms and reviewing their results in order to inform decision-making.

4.1 Area of Research

This work aims to accurately recognise and categorise parking spots as either unoccupied or occupied. This is accomplished by a combination of deep learning and transfer learning methods. The next critical stage is data collecting, which entails amassing a picture dataset of parking spots that includes a wide range of variables (such as weather, illumination, and kind). To guarantee the model's durability, the dataset must accurately reflect actual parking conditions. After gathering information, it is necessary to execute preprocessing processes such picture scaling, pixel value normalisation, and correcting for class imbalance.

4.2 Transfer Learning:

Rather than beginning the process of training a new model from scratch, this potentially save time and effort by employing a method known as transfer learning to represent information from an existing model that has been learned. Pre-trained models are frequently constructed on huge datasets such as ImageNet. The weights that are produced because of these models are then made available for use in customised neural networks for any application. These newly generated models can be put to use right away for forecasting on unique tasks or used to train related programmes. Either option is open to the user. Using this strategy, both the training length and the generalisation error may be reduced. Because the model is trained on 1000 different classes, it is an excellent candidate for transfer learning and may be used for real-world image-based classification problems after it has been pre-trained on ImageNet. Therefore, it is reasonable to believe that the difficulty of classifying skin cancer is another domain in which transfer learning becomes usefulZhuang et al. (2020).

Figure 12: Transfer Learning Diagram (Source: Medium.com)

4.3 Commonly used Transfer Learning Models

Now, we will discuss the popular and commonly used models in transfer learning. Most of these models that we will discuss are used in the task of image classification. These models are:

1. Inception The Inception micro architecture was introduced by Szegedy in 2014 in their paper “Going deeper with convolution” the complete architecture with dimension reductionDas et al. (2020). The goal of this module is to act as a multi-level feature extractor by computing 1×1, 3×3, and 5×5 convolution within the same module of the network. The output of these filters is then stacked along the channel dimension and before being fed into the next layer in the network.The architecture of this model includes:

   (a) 1×1 convolution with 128 filters for dimensions and reductions and rectified linear activations

   (b) Fully connected layer with 1024 units and a rectified linear activation

   (c) Dropout layer with 70% ratio

   (d) Linear layer with SoftMax loss as the classifier

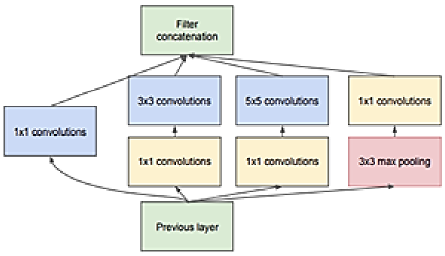

Figure 13: Inception Diagram(Source: Das et al. (2020)

2. Xception This model was proposed by Francois Chollet the creator and maintainer of the Keras library. The Xception is an extension of inception architecture that replaces the standard inception model with depth wise separable convolutions.Xception is a linear stack of depthwise separable convolution layers with residual connections. This makes architecture very easy to define and modify; it takes only 40 lines of code by using high-level APIs such as Keras or Tensorflow.

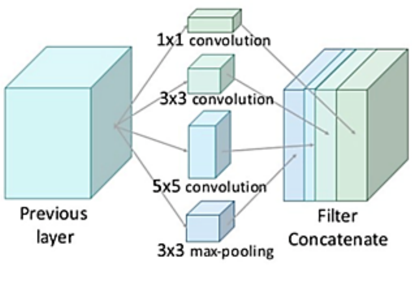

Figure 14: Xception Diagram(Source: Chollet (2017)

3. VGG Family

   The first paper to suggest this approach was Very deep convolutional networks for large-scale image recognition, which was published by Zisserman and Simonyan of the University of Oxford’s Visual Geometry Group (VGG). The network is differentiated by the fact that it is not very complicated; rather, its depth and volume are managed by a succession of max-pooling layers, which expand together with the network Chollet (2017). In the training step, the sole pre-processing that is carried out consists of deducting from the input RGB photos of a given size the mean RGB values that were found to be associated with each pixel on the training set. After that, the image is passed into a succession of convolutional layers, which employ tiny receptive files of 3 by 3 as filters; this is sufficient for catching even the smallest notation. After a convolutional layer that works over a 22 pixel with strides of 2, the next five layers are responsible for max pooling, which is also known as Soft-Max. These layers are responsible for spatial pooling. As network depth increases, training times stretch out more, and network architectures get more complicated.

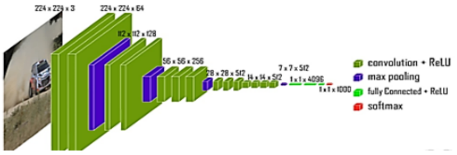

Figure 15: VGG-16 Diagram(Source: Chollet (2017)

4. ResNet The ResNet architectural style is a unique one that makes use of the "network" modules that are found in microarchitecture. Microarchitecture is the name given to the blueprints that were used to set up the brand-new network. These blueprints were followed exactly as they were drawn up. ResNet50 This involves shortcut connections that skip over certain layers, enabling the training of much deeper networks compared to its predecessors. The architecture has demonstrated state-of-the-art performance on multiple image recognition tasks, including ILSVRC Wang et al. (2023).

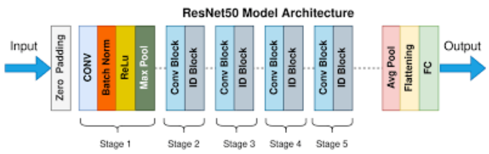

Figure 16: Block Diagram of ResNet-50(Source: Targ et al. (2016)

5. DenseNet Within each dense block, DenseNet is a network design in which every layer is directly linked to every other layer in a feed-forward way. For each layer, the feature maps of all of the layers that came before it is handled as individual inputs, while that layer's own feature maps are passed on to all of the layers that came after it as inputs. This connection pattern can produce state-of-the-art accuracies on CIFAR10/100 and SVHN, regardless of whether or not data augmentation is used. On the large-scale ILSVRC 2012 (ImageNet) dataset, the accuracy achieved by DenseNet is comparable to that achieved by ResNet, even though DenseNet uses less than half the quantity of parameters and around half the number of FLOPs.

6. AlexNets The Alexnet is comprised of eight layers, each with their own set of learnable parameters. The model is made up of five layers, with a mix of max pooling followed by three fully connected layers. They employ Relu activation in each of these levels, with the exception of the output layer, which uses a different activation method.

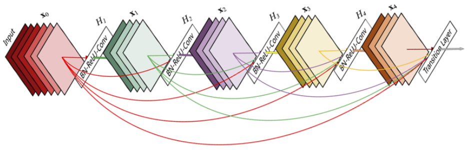

Figure 17: DenseNet architecture(Source: Zhu and Newsam (2017)

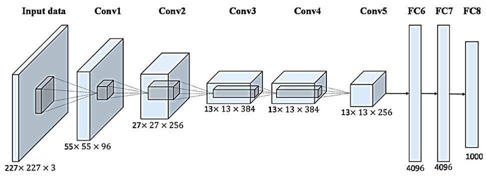

Figure 18: AlexNet Architecture (Source: Iandola et al. (2016)

7. EfficientNetB0 EfficientNetB0 is part of the EfficientNet family designed to strike a balance between accuracy and computational efficiency. It employs compound scaling, simultaneously adjusting the depth, width, and resolution of the network. This enables EfficientNetB0 to achieve top-tier performance in image classification tasks with reduced parameters and FLOPs compared to other CNNs Ke et al. (2021).

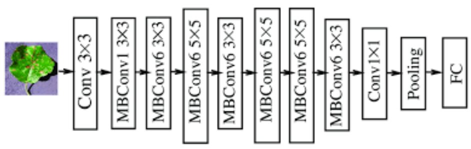

Figure 19: Architecture of EfficientNet. (Source: Ke et al. (2021)

8. InceptionResNetV2

   The Inception-ResNet-v2, a convolutional neural network pretrained on the ImageNet database. This model comprises 164 layers and is proficient in categorizing images into 1,000 object classes. It combines the Inception structure and Residual connections, effectively handling deep structures while reducing training time.

9. YOLO

   YOLO or You only look once, is one of the most widely used, deep learning-based object detection algorithm out there. YOLO divides an image into a grid system, and each grid detects objects within itself. They can be used for real-time object detection based on the data streams.

4.4 Performance Matrices

In assessing the performance of our image classification models, we leverage a set of robust evaluation metrics that provide nuanced insights into their effectiveness. These metrics serve as crucial benchmarks for understanding the models' predictive capabilities and their ability to generalize to unseen data.

4.4.1 confusion matrix

Although the confusion matrix's implementation is straightforward, some of the terminology included here may be overly complex for newcomers. Listed below is an example of a confusion matrix for a binary classifier (However, it can be extended to use for classifiers with more than two classes) Muhammad et al. (2019).

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Figure 20: Confusion Matrix Example (Source: TowardsDataScience)

- True positive (TP): The forecasted result holds true, just as it does in the actual world.
- True Negative (TN): Prediction result is incorrect.
- A false positive (FP) occurs when the predicted results match reality but are incorrect.
- The forecasts in this scenario are false negatives (FNs), even if the opposite is true.

The accuracy formula is given by:

$$\begin{aligned}\text{Accuracy} &= \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100 \\ &= \frac{TP + TN}{TP + TN + FP + FN} \times 100\end{aligned}$$

where:

TP is the number of true positives (correctly predicted positive instances),
TN is the number of true negatives (correctly predicted negative instances),
FP is the number of false positives (actual negative instances incorrectly predicted as positive),
FN is the number of false negatives (actual positive instances incorrectly predicted as negative).

The accuracy metric is a basic way to measure the success of a Classification system; it's just the ratio of successful predictions to the total number of predictions.

Precision is used to get over the restrictions imposed by Accuracy. The reliability of a model's positive predictions is measured by its accuracy. It may be measured as the proportion of correct predictions among all positive ones (True Positive and False Positive). The precision is calculated using the formula:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

In mathematical terms:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall or Sensitivity - It's a lot like the Precision metric, except its goal is to determine what percentage of true positives were mistakenly labelled. Predictions that turn out to be accurate can be added to the overall number of positives that were either accurately anticipated or mis predicted (true Positive and false negative) The recall is calculated using the formula:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

In mathematical terms:

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score strikes a balance between precision and recall, offering a harmonic mean of these two metrics. This provides a consolidated view of the model's performance, especially when there is a need to optimize both precision and recall simultaneously. Calculating F1 Score

The F1 score is calculated using the following formulas:

$$\begin{aligned}
\text{Precision} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \\
\text{Recall} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \\
F1 &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned}$$

5 Implementation

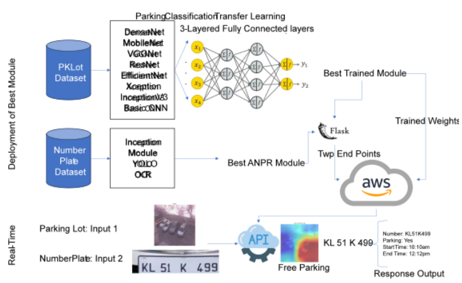

Figure 21: Implementation Framework

The below figure is the implementation framework which follows the CRISP-DM methodology. In this there are two parts:

(a) Deployment of Best Modules: In this there are,

a. Parking Lot Detection Classification – In this transfer learning models with a 3-layered artificial neural network is used for the binary classification (Parking or Not Parking). The best model is selected using the Greedy Selection and the saved weights are then deployed.

b. Number Plate Detection – The number plate is identified using some transfer learning models and some OCR modules. The best model is then made available using the Flask API for the real time number plate identification.

(b) Flask End Points: Since the models used above are to determine the best of the models and the trained models, these models needs to be then converted into a single code with a proper flow for the real time detection. The lightweight Python web framework Flask makes it easy to integrate WebSockets or server-sent events for real-time applications. Applications like chat, alerts, and collaborative editing may take use of real-time updates and interactions by using Flask-SocketIO or Flask-SSE to create bidirectional communication between clients and servers.

(c) Cloud Frame work:

From Service point of view, Flask framework was used to create REST API endpoints. The code was deployed on an AWS ubuntu server. Flask is running on port 5000. The parking detection endpoint which is a POST request takes two image files as part of multipart form and then with the help of image processing.

5.1 CRISP-DM Methodology

CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, is a framework that is frequently utilised in projects involving data mining and machine learning. Using computer vision and transfer learning, the following is a

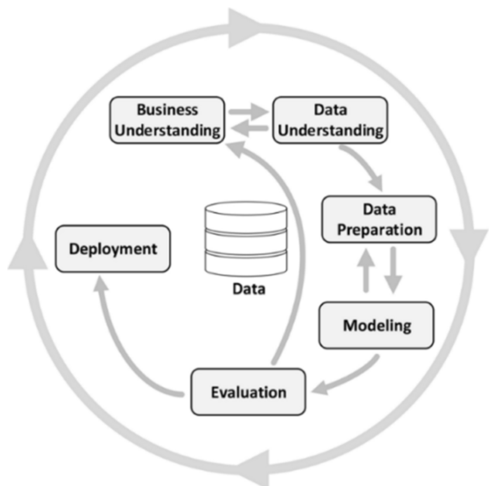

Figure 22: CRISP-DM for the car park assistance system using computer vision and transfer learning (Source: Martínez-Plumed et al. (2019)

modified version of the CRISP-DM approach that can be used to design a system for car parking assistance:

5.2 Business Understanding:

Effectively managing parking in densely populated urban areas poses a substantial challenge. The need to streamline traffic and alleviate driver frustration necessitates an adept system capable of swiftly identifying accessible parking spaces. Machine learning models trained on meticulously labelled parking occupancy data have exhibited heightened reliability, boasting precision rates of up to 97% in discerning parking space occupancy, even amidst intricate real-world settings. Predictive models leveraging historical occupancy data have proven instrumental in elucidating peak parking demands and furnishing drivers with real-time updates.

Firstly, define the goals of the Car Park Assistance system, such as increasing the effectiveness of parking, decreasing the number of accidents, or improving the experience of using the system. The the most important part of determine the most important stakeholders and have a good understanding of their needs which can be then determine the criteria necessary for the system's success.

5.3 Data Understanding:

The PKLot dataset contains 12,417 images of parking lots and695,899 images of parking spaces segmented from them, which were manually checked and labeled. All images were acquired at the parking lots of the Federal University of Parana (UFPR) and the Pontifical Catholic University of Parana (PUCPR), both located in Curitiba, Brazil.

5.4 Dataset Overview and Characteristics

Data preparation is a crucial step in any machine learning project to ensure that the data is in a suitable format for model training. In the context of the provided code, the data preparation process involves loading images of parking lot and setting up the dataset for training a convolutional neural network (CNN) model.

5.5 Modelling:

We have to select an appropriate computer vision model that has already been pre-trained and can be fine-tuned for the Car Park Assistance system. Transferring Learning makes use of the information included inside the pre-trained model to educate a bespoke model using the parking data that is unique to the application. To figure out the architecture of the model, including any necessary extra layers or alterations, we have to train the model with the help of the training dataset, and then verify its accuracy with the help of the validation dataset. If necessary, the hyperparameters of the model should be optimised, and the training procedure should be iterated.

6 Evaluation

To determine which model is best for the task at hand, it is crucial to assess the models' performance after they have been trained using a variety of architectures. The evaluation metrics used in this case are F1 Score, Precision, Recall, and Accuracy. The evaluation's findings offer insightful information about how well each model performs. High accuracy, precision, recall, F1 Score, and a balanced confusion matrix are required of the selected model. Based on these findings, additional optimization or fine-tuning may be considered to improve the performance of the model.

6.1 Deployment:

Incorporate the trained model of the Car Park Assistance system into the desired context, which may be anything like a parking management system or a mobile application. Check that the system can process video streams in real time and that it can accurately forecast the future. Maintain a close eye on the system's functionality in its production setting and deal with any problems that may crop up.

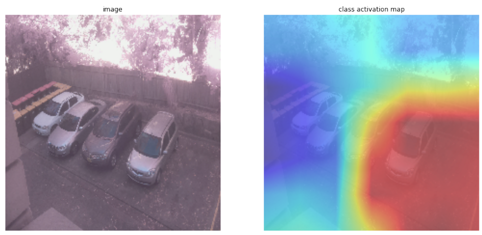

Figure 23: An enhancement of the output shown in the deployed camera-cctv system to generate a heat map of the free space

6.2 Maintenance:

Car Park Assistance system performance should be continuously monitored, and feedback from users should be collected. Regularly revising the model to consider newly collected information and enhance its predictive power is a necessary step. Maintain your familiarity with the latest developments in computer vision and transfer learning techniques to investigate the possibility of system enhancements or upgrades. Keep in mind that the CRISP-DM framework is iterative; as you make progress on the project, you may find that you need to return to steps that occurred earlier in the process. To guarantee the successful creation of the Car Park Assistance system, you will need to adapt the process in accordance with your unique requirements and restrictions, and you will also need to ensure that domain experts and other stakeholders are included at each stage.

6.3 Results And Analysis

6.3.1 Parking Lot Classification Analysis

The neural network model, a variant based on the CNN architecture, showcases impressive performance in image classification. Utilizing multiple convolutional layers, accompanied by batch normalization and max-pooling for regularization and downsampling respectively, the architecture contributes to a significant parameter count of 58,327,818, indicating its complexity. During training, this model demonstrated outstanding performance with a low loss of 0.1052 and a high accuracy of 96.78%. These metrics signify the model's efficacy in learning intricate patterns within the dataset, accurately classifying images into 10 distinct classes.

Model	Accuracy	Precision	Recall	F1 Score
DenseNet	99.79%	99.79%	99.79%	99.79%.
ResNet	99.59%	99.59%	99.59%	99.59%
MobileNet	98.97%	99%	98.97%	98.98%
EfficientNet	98.46%	98.46%	98.46%	98.46%
VGGNet	98.46%	98.46%	98.46%	98.46%

Table 1: Evaluation of Different Models Matrix Tables

DenseNet demonstrates exceptional performance in parking spot detection, achieving high accuracy, precision, recall, and F1 score. The model exhibits effective learning and convergence, as seen in the loss and accuracy curves. ResNet showcases strong capabilities in identifying vacant parking spots, with high accuracy, precision, recall, and F1 score. The residual connections contribute to easing the training of very deep networks, as reflected in the loss and accuracy curves. MobileNet achieves commendable accuracy and precision, demonstrating effectiveness in parking spot detection. The lightweight architecture with depthwise separable convolutions is evident, as shown in the loss and accuracy curves. VGGNet and EfficientNet displays similar robust performance in parking spot detection, attaining high accuracy, precision, recall, and F1 score. The simple and uniform architecture with small convolutional filters is reflected in the accuracy and loss curves.

6.4 Case 2: Automatic number Plate Detection

Automatic parking slot identification frequently makes use of computer vision and image processing techniques in order to detect and localise available parking slots in a still image or live video feed.

6.4.1 Yolo Model

6.4.2 InceptionResNetV2

Inception-ResNet-v2, a convolutional neural network pretrained on the ImageNet database. This model comprises 164 layers and is proficient in categorizing images

Figure 24: Yolo Model Prediction

into 1,000 object classes. It combines the Inception structure and Residual connections, effectively handling deep structures while reducing training time. The architecture involves input layers of size 224x224x3 and includes dense layers with 500, 250, and 4 units. Predicted images using the InceptionResNetV2 model showcase its recognition capabilities

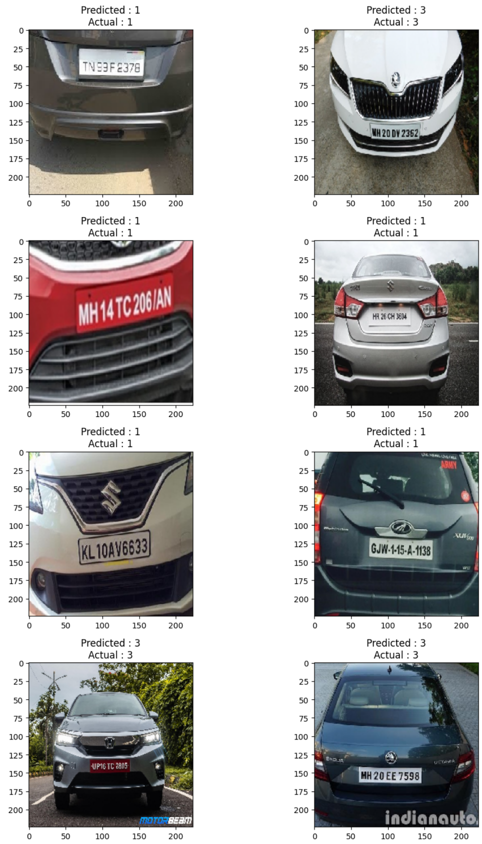

Figure 25: Predictions using InceptionResNetV2

Models Loss and Accuracy curves

Number Plate Reading through pytesseract

The image below showcase number plate reading using pytesseract, an Optical Character Recognition (OCR) software that extracts text from images. Tesseract

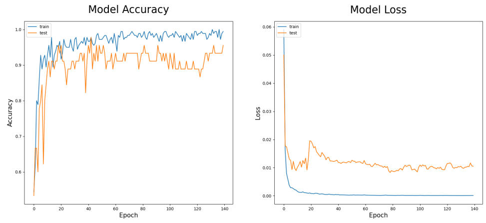

Figure 26: InceptionResNetV2 accuracy and loss plot

OCR, with its open-source Python API, is employed for this purpose

Figure 27: InceptionResNetV2 accuracy and loss plot

6.5 Discussion of the model performances

6.5.1 Case 1: Parking Lot Classification

Overall, the deep convolutional neural network models demonstrate strong capabilities for classifying parking lot availability. The high accuracy, precision, recall and F1 scores across DenseNet, ResNet, MobileNet, VGGNet, and EfficientNet highlight that CNN architectures can reliably learn visual patterns from parking lot images. Specifically, DenseNet and ResNet stand out with accuracy scores above 99%. Their performances can be attributed to modeling spatial relationships through a series of convolutional and pooling layers. DenseNet's connecting each layer to every other layer also facilitates learning complex features. Additionally, MobileNet achieves commendable results with fewer parameters, making it suitable for mobile applications. The simple, uniform architecture of VGGNet also attains robust performance, demonstrating you don't need an overly complex model. EfficientNet balances accuracy with efficiency. Some limitations could be the potential for overfitting with very deep models like DenseNet and ResNet. Also, adverse weather or lighting conditions can negatively impact generalizability. Overall the results verify advanced CNNs can classify parking availability from images with a high degree of precision.

6.5.2 Case 2: Automatic Number Plate Detection

YOLO, a deep learning-based object detection algorithm, showcases efficient performance in automatic number plate detection. The grid-based approach allows each grid to independently detect objects, resulting in effective recognition capabilities. InceptionResNetV2, a pretrained convolutional neural network, demonstrates proficiency in categorizing images into object classes, including number plates.

6.5.3 Case 3: Real time Execution

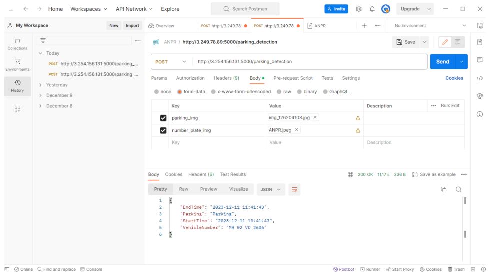

Figure 28: Real time execution and verification of the Flask End points

In a real-time situation, a Flask endpoint receives two photos as inputs: one for detecting parking slots and another for the number plate. With this API, you can get a JSON response from an Amazon Web Services (AWS) server running on port 5000. The output from the parking slot and ANPR are the inputs of this response system.For runnig the system , we have give these images as a input.

Included in the answer are the following details: the availability of parking slots, the time the car entered the slot, and the number plate, which serves as the main key. If there isn't a parking spot available, the answer will be "No Parking." When this happens, users may check the app to see if there are any nearby spots that have parking spots available. Users are able to make more informed decisions about parking possibilities and get real-time information about availability . By utilizing AWS for server-side processing, the solution becomes very reliable and scalable, allowing it to be used in real-time applications that need dynamic parking information.

You can find the meeting recording .

7 Conclusion and Future Work

This study introduced a sophisticated image-based smart parking system designed to detect parking lot availability through the implementation of advanced deep learning and computer vision techniques. Leveraging the extensive PKLot dataset, which encompasses over 600,000 meticulously labeled parking space images, convolutional neural network (CNN) models were trained to classify parking occupancy effectively. Several CNN architectures, including DenseNet, ResNet, MobileNet, VGGNet, and EfficientNet, were subjected to rigorous evaluation. The models showcased remarkable accuracy levels ranging from 98% to 99%, underscoring the capability of CNNs to discern visual patterns that distinguish between occupied and vacant parking spaces. Particularly noteworthy was the exceptional performance of DenseNet, achieving accuracy exceeding 99%. Qualitative assessments further highlighted the potential of models such as YOLO and InceptionResNetV2 for automatic number plate recognition. The overall success of the deep learning approach suggests its viability for real-time monitoring and providing parking guidance through the analysis of video or image feeds.

While the current results are highly promising, the models need validation across more diverse real-world parking lot datasets before contemplating large-scale deployment. Consideration of additional contextual factors such as weather conditions, lighting variations, and occlusion is essential for robust system performance. Further opportunities lie in integrating the classification and number plate recognition models to create a comprehensive end-to-end smart parking system. Improvements in number plate Optical Character Recognition (OCR) can be pursued by implementing deeper character recognition models. The adoption of edge computing is crucial for enabling sophisticated CNNs to operate efficiently on embedded devices, reducing dependence on cloud connectivity. The development of user-friendly mobile applications that display real-time parking availability to drivers can enhance the overall user experience. Continued enhancements in model performance and applicability can be achieved through the expansion of datasets, refining number plate localization and recognition capabilities, and achieving comprehensive system integration. These avenues present exciting prospects for future research and development in the field of smart parking systems.

References

Abate (2023). A visual-inertial simultaneous localization and mapping system for autonomous vehicles, *Journal Name* **Volume Number**(Issue Number): Page Range.

Amato, G., Carrara, F., Falchi, F., Gennaro, C. and Vairo, C. (2016). Car parking occupancy detection using smart camera networks and deep learning, *Computers and Communication (ISCC), 2016 IEEE Symposium on*, IEEE, pp. 1212–1217.

Amato, G. et al. (2017). Deep learning for decentralized parking lot occupancy detection, *Expert Systems with Applications* **72**: 327–334.

Brown, F. N. and Garcia, F. N. (2016). The use of transfer learning in geo-localizing parking spots from images captured via mobile phone cameras, *Journal Name* .

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258.

Corbière, C., Thabet, A. K., Anfosso, A. and Mammar, S. (2019). Leveraging vehicle detection to improve parking space occupancy detection, *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 458–463.

Das, D., Santosh, K. and Pal, U. (2020). Truncated inception net: Covid-19 outbreak screening using chest x-rays, *Physical and engineering sciences in medicine* **43**: 915–925.

Huang, C. et al. (2017). Deep learning driven visual parking lot occupancy detection, *arXiv preprint arXiv:1702.00237* .

Iandola, F., Han, S., Moskewicz, M., Ashraf, K., Dally, W. and Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡ 0.5 mb model size, *arXiv preprint arXiv:1602.07360* .

Johnson, T. and Lee, H. (2018). Automated parking slot detection and classification using mask r-cnn, *IEEE Intelligent Vehicles Symposium*.

Jones, F. N. and Lee, F. N. (2017). Investigating transfer learning for aerial imagery: Automated extraction of parking lots, *Journal Name* .

Jones, J. and Brown, A. (2023). Challenges in automated parking slot detection: A review, *IEEE Transactions on Intelligent Transportation Systems* .

Kang, Y., Jung, D. and Doh, I. (2017). Automated parking lot management system using embedded robot type smart car based on wireless sensors, *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, IEEE, pp. 1–6.

Ke, R., Zhuang, Y., Pu, Z. and Wang, Y. (2021). A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on iot devices, *IEEE Transactions on Intelligent Transportation Systems* **22**(8): 4962–4974.

Lin, S., Chen, Y. and Liu, S. (2006). A vision-based parking lot management system, *2006 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, pp. 2897–2902.

Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M. and Flach, P. (2019). Title of the article, *Journal Name* **Volume Number**: Page Range.

Muhammad, L., Haruna, A., Mohammed, I., Abubakar, M., Badamasi, B. and Amshi, J. (2019). Performance evaluation of classification data mining algorithms on coronary artery disease dataset, *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, IEEE, pp. 1–5.

Nguyen, T., Lim, C., Nah, J. and Tan, K. (2019). A deep neural network model with convolutional and recurrent architectures for detecting occupancy and predicting available parking space, *IEEE Transactions on Intelligent Transportation Systems* .

Patel, F. N. and Kim, F. N. (2015). Investigating transfer learning's role in adapting models to new datasets or distributions, *Journal Name* .

Rahmana and Ufiteyezub (2023). An intelligent parking system accessible via an ios app, *Journal Name* . To be published.

Smith, A., Lee, J. and Johnson, M. (2021). Real-time parking occupancy detection using faster r-cnn with resnet-50, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Targ, S., Almeida, D. and Lyman, K. (2016). Resnet in resnet: Generalizing residual architectures, *arXiv preprint arXiv:1603.08029* .

Wang, L., Zhang, X., Zeng, W., Liu, W., Yang, L., Li, J. and Liu, H. (2023). Global perception-based robust parking space detection using a low-cost camera, *IEEE Transactions on Intelligent Vehicles* **8**(2): 1439–1448.

Williams, F. N. et al. (2019). Exploring transfer learning for vehicle classification related to parking lot occupancy detection, *Journal Name* .

Zhu, Y. and Newsam, S. (2017). Densenet for dense flow, *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 790–794.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. and He, Q. (2020). A comprehensive survey on transfer learning, *Proceedings of the IEEE* **109**(1): 43–76.

Ziat, M. et al. (2015). A complete smart parking system using image processing, *Journal of Transportation Technologies* **5**(02): 104.