Three Level Deep Learning based Approach for Automated Smart Parking

Shaunak Gupta*, Abhinav Aggarwal †, Pushkar Garg‡, Gaurav Goyal§ and Kanu Goel¶
Department of Computer Science and Engineering
Punjab Engineering College (Deemed to be University)
Chandigarh, India
Email: *shaunakgupta.bt21cse@pec.edu.in, †abhinavaggarwal.bt21cse@pec.edu.in, ‡pushkargarg.bt21cse@pec.edu.in,
§gauravgoyal.bt21cse@pec.edu.in, ¶kanugoel@pec.edu.in

***Abstract*—Urbanization and increasing vehicular density have amplified challenges in conventional parking systems, necessitating innovative solutions for effective parking management. This research paper identifies and addresses critical issues in contemporary parking infrastructure. Common challenges include the absence of real-time information on available parking slots, limited data analysis tools for space utilization, reliance on manual processes, and the inadequate incorporation of automation at entry and exit of vehicles. The proposed smart parking project endeavors to mitigate these challenges by integrating advanced technologies to enable efficient parking slot detection, comprehensive data analysis, and seamless automation. By offering insights into the problems plaguing current parking systems and proposing a technologically-driven approach, this research contributes to the advancement of smart and responsive parking management solutions. Building upon the identified challenges in current parking systems, this research introduces a set of innovative techniques for smart parking management. The proposed system integrates automation at entry and exit points, streamlining the parking process. Leveraging advanced computer vision and sensor technologies, the system provides real-time identification of empty parking slots, enhancing overall space utilization. Furthermore, the implementation of robust data analytics tools enables continuous monitoring and analysis of the number of available parking slots, contributing to dynamic pricing. The system also incorporates accident tracking mechanisms to enhance safety within parking facilities. By amalgamating these techniques, the research aims to revolutionize traditional parking infrastructure, fostering a more efficient, secure, and automated smart parking ecosystem. The research paper presents a novel accident detection model attaining a commendable accuracy of 94.5% (Other 2 models are image processing models rather than machine learning models) .**

***Index Terms*—Smart Parking, Number Plate Recognition, parking slots availability , accident detection, real-time monitoring, artificial intelligence**

I. Introduction

The burgeoning challenges of urbanization in contemporary cities necessitate innovative solutions to address the multifaceted issues surrounding urban mobility. Among these challenges, the efficient utilization of parking spaces emerges as a critical concern, demanding a paradigm shift in traditional parking management systems. In response to this pressing need, our research introduces a groundbreaking Smart Parking Management System that not only tackles immediate congestion issues but also lays the groundwork for a sustainable and technologically advanced parking solution. By seamlessly integrating the latest technologies and adhering to thoughtful design principles, this system envisions a future where private parking areas transform into hubs of efficiency, setting a new standard for parking solutions in the face of the evolving demands of modern urban environments.

The rapid urbanization of cities worldwide has ushered in unprecedented challenges, particularly in the realm of urban mobility. As cityscapes evolve, the strain on parking infrastructure becomes increasingly evident. This research responds to the growing need for an intelligent and adaptive parking management system that goes beyond conventional approaches. The proposed Smart Parking System (SPS) stands out as a revolutionary solution tailored for marked parking slots, harnessing the power of advanced image processing techniques, most notably Convolutional Neural Networks (CNNs). At the heart of the Smart Parking System lies its ability to revolutionize parking space detection through the application of CNNs. Inspired by the visual processing mechanisms of the human brain, CNNs serve as the linchpin of the image processing capabilities, extracting intricate features from parking lot images. This technological innovation enables real-time analysis and identification of available parking spaces with unparalleled accuracy, guiding users to the nearest parking slots. By mitigating congestion and optimizing the parking experience, the system addresses a crucial aspect of urban mobility, ushering in a new era of efficient parking space utilization.

The integration of Automatic Number Plate Recognition (ANPR) technology further augments the capabilities of the Smart Parking System. Working seamlessly with the image processing prowess of CNNs, ANPR automates ticketing through contactless vehicle identification. This not only streamlines the entry and exit processes but also contributes to the intelligence and efficiency of the parking management system. The synergy between CNNs and ANPR marks a pivotal advancement in parking technology, enhancing user convenience and expediting parking transactions. Acknowledging the paramount importance of safety within parking facilities, our research extends the functionality of the Smart Parking

System to include real-time accident detection. Achieved through a network of strategically placed sensors and cameras, this dual-faceted approach works in tandem with the image processing capabilities of CNNs. Beyond enhancing parking logistics, this feature fortifies the security and safety aspects of parking infrastructure, ensuring a comprehensive approach to smart parking management.

As we delve into the intricate interplay between image processing, CNNs, ANPR, and accident detection in the subsequent sections, the aim is to illuminate the transformative potential of the Smart Parking System. By harnessing the power of cutting-edge technologies, this research endeavors to redefine urban parking, setting new benchmarks for efficiency, user convenience, and safety in metropolitan environments. The ensuing sections will delve deeper into the technical underpinnings and real-world applications of these innovative parking management techniques, offering a comprehensive understanding of their transformative impact on urban mobility.

The evolution of the Smart Parking System extends beyond conventional boundaries, embracing technological frontiers to redefine the urban mobility landscape. In the upcoming sections, we will explore in-depth the intricate details of how image processing, CNNs, ANPR, and accident detection seamlessly converge to create a holistic and intelligent parking solution. By delving into the technical intricacies and real-world applications of these innovative techniques, this research endeavors to showcase not just the theoretical underpinnings but also the practical implications for urban environments grappling with the challenges of parking management. The journey into the future of smart parking beckons, promising a transformative shift in the way we perceive, manage, and experience parking in contemporary cities.

A. *Contributions*

Our smart parking system, featuring ANPR, parking spot detection, and accident detection to address several challenges inherent in traditional parking systems. The unique contributions of our model to society can be summarized as follows:

- Optimized Space Utilization: Traditional parking systems often suffer from inefficiencies due to inadequate space utilization. Our model dynamically detects and allocates parking spaces, optimizing the usage of available areas. This contributes to reduced congestion and efficient space utilization, ultimately minimizing the environmental impact of vehicular traffic.
- Enhanced User Experience: By incorporating ANPR technology, our system streamlines the entry and exit processes. Users experience a seamless parking experience as the system automatically recognizes their vehicle, eliminating the need for manual ticketing or identification. This not only saves time but also enhances the overall convenience for users.
- Accident Detection for Safety: The inclusion of accident detection is a crucial safety feature. Traditional parking systems lack the ability to monitor and respond to accidents in real-time. Our model detects accidents promptly, triggering immediate responses such as alerting emergency services. This contributes significantly to the safety of both pedestrians and vehicle occupants within the parking facility.
- Real-time Monitoring and Management: Our smart parking system offers real-time monitoring of parking spaces and the overall facility. This feature allows for efficient management of parking resources, quick identification of issues, and proactive response to changing conditions. Traditional parking systems often rely on manual monitoring, making them less responsive to dynamic situations.
- Environmental Impact Reduction: Through the optimization of parking space usage and the reduction of traffic congestion within the parking facility, our model contributes to a decrease in vehicular emissions. This aligns with societal goals for sustainability and environmental conservation, promoting a greener and healthier urban environment. In conclusion, our smart parking system goes beyond the capabilities of traditional systems by combining advanced technologies to enhance efficiency, safety, and user experience. Its unique contributions to society address the limitations of conventional parking solutions and pave the way for a more intelligent and sustainable urban mobility ecosystem.

II. Literature Review

Dasari *et al.* have developed a smart parking system that integrate sensors and real-time data, utilizing the Prewitt edge detection technique for occupancy assessment. Machine learning algorithms, particularly Convolutional Neural Networks (CNNs), enhance predictive capabilities but face challenges like data dependency. Libraries such as Numpy, CVZone, and Pickle play crucial roles in system implementation. The smart parking system employs video analysis and algorithms, utilizing computer vision and Prewitt edge detection for parking space identification.

The approach proposed by Denis *et al*. focuses on multi-storied parking areas, utilizing energy conservation and management. Another RFID and LCD-based approach offers a user-friendly system with unique IDs. Praveen & Harini discuss NB-IOT technology, providing efficient communication and mobile applications for slot booking. The GSM-based approach emphasizes network data transfer, while the Cloud-of-Things-based approach aims to reduce user parking time. Image processing and artificial intelligence-based approaches focus on license plate detection for efficient parking. Miscellaneous approaches include real-time reservation using IoT and Wi-Fi technology. Astya *et al.* have also stated various approaches for smart car parking systems.

Devi et al. employ a static camera, initiating with size detection and shape analysis of car images. Unlike motion-based methods, this approach enhances efficiency in identifying images as references. Integration of deep learning minimizes sensor costs and wiring complexities in the development of an intelligent parking system. Image processing techniques, including grayscale conversion, thresholding, and contour detection, are crucial for accurate comparison with a dataset, ensuring precise determination of vacant parking slots. The application of Support Vector Machine (SVM) enhances classification accuracy, as demonstrated in the presented results. This research aligns with the trend of leveraging deep learning for effective and cost-efficient parking management systems.

A deep learning based approach given by Kumeda *et al.* delves into the application of deep learning, particularly Convolutional Neural Networks (CNNs), for image classification in the context of traffic accident detection. Emphasizing the power of neural networks in classification and pattern recognition through computer vision, the study focuses on the architecture of CNNs. CNNs, inspired by the human visual system, employ convolutional layers, ReLU layers, pooling layers, and fully connected layers to extract spatial features and patterns from images. The model, comprising multiple convolutional and fully connected layers, achieves notable accuracy (94.4%) on the Traffic-Net dataset, showcasing the efficacy of deep CNNs in real-world image classification tasks.

An Intelligent Smart Parking System based study by Alsheikhy *et al.* introduces a cost-effective and efficient smart parking system based on real-time image processing using a single camera. The proposed algorithm, illustrated through simulation experiments, effectively manages parking lots by detecting vacant spaces and directing drivers. The system's advantages include cost-effectiveness, environmental friendliness, easy maintenance, and integration capabilities. The algorithm employs a convolutional neural network tool for intelligent adaptation to various conditions. While fault tolerance is addressed with a backup camera, human intervention serves as an alternative. Compared to IoT sensor-based methods, this camera-centric approach stands out for its affordability, operational independence from Internet services, and practicality.

Besides the above work by Puranic *et al.* focuses on addressing key challenges in educational institutions related to vehicle management and theft prevention. The proposed system employs video capture and MATLAB-based application for vehicle number plate recognition using template matching. The system assumes single entry and exit gates, with initial focus on single-lane traffic. Image processing techniques, including contrast extension and median filtering, are applied for number plate localization and character segmentation. Template matching, based on character templates, achieves an average accuracy of 80.8%. The study reviews existing methodologies, highlighting the need for a customized Automatic Number Plate Recognition (ANPR) system tailored to educational settings. Future work includes enhancing accuracy through optimal camera positioning and incorporating neural networks for live video feed analysis. The system's potential extension to recognize multiple vehicles and implement multi-level genetic algorithms is also discussed.

III. Proposed Approach

The proposed approach entails the development of a Smart Parking System through a comprehensive integration of advanced technologies. The system is designed to enhance parking management efficiency by incorporating key components such as Automatic Number Plate Recognition(ANPR), Parking Slot Detection, and Accident Monitoring. Leveraging OpenCV and EasyOCR, the ANPR module ensures accurate identification of vehicle license plates, achieving an impressive accuracy for four-wheelers. Parking Slot Detection is accomplished using OpenCV for image preprocessing, contour detection, and YOLO (You Only Look Once) for real-time detection and classification of parking slots. The YOLOv8 architecture is specifically employed to boost accuracy, enabling precise tracking of parking space occupancy. Additionally, a Convolutional Neural Network (CNN) model is implemented to monitor the parking space for potential accidents or anomalies, contributing to enhanced safety measures. The proposed approach aims to provide a seamlessly integrated system that addresses key challenges in parking management, promising accurate license plate recognition, real-time parking slot occupancy detection, and timely identification of incidents within the parking area.

A. Recognition of License Plate by ANPR System

The proposed Automatic Number Plate Recognition (ANPR) system involves a comprehensive methodology for efficient license plate recognition. The process begins with the Input Module, where diverse images with license plates are ingested, reflecting real-world scenarios in terms of lighting, angles, and plate types. The subsequent Data Collection Module plays a vital role in curating a diverse dataset, ensuring the system's robustness across varied conditions. Following this, the Preprocessing Module applies sophisticated techniques to enhance image quality by addressing challenges like noise and variations in illumination.

The core of the system is the Number Plate Localization Module, utilizing contour detection and approximation to identify potential license plate locations. The algorithm iterates through contours, approximating their shapes and identifying quadrilateral shapes indicative of license plates. The subsequent OCR Implementation Module employs Optical Character Recognition (OCR) directly on localized license plates, converting visual alphanumeric characters into machine-readable text.

Finally, the system concludes with the Output Module, providing a comprehensive result that includes the recognized alphanumeric information. This output serves as a valuable resource for further processing or seamless integration into user interfaces, offering an efficient solution for Automatic Number Plate Recognition.

1. Read image data from standard input. 2. Decode image data into a NumPy array. 3. Preprocess the image: 4. Convert the image to grayscale. 5. Apply bilateral filtering for noise reduction. 6. Use Canny edge detection. 7. Find contours in the edge-detected image. 8. Select potential license plate locations based on contour approximation. 9. Create a mask

for the selected license plate location. 10. Extract the region of interest (license plate) using the mask. 11. Perform Optical Character Recognition (OCR) on the license plate using EasyOCR. 12. Define character mapping dictionaries for OCR misinterpretations. 13. Format the recognized license plate based on different lengths 14. Print the formatted license plate result.

This pseudocode involves a multi-step process for efficient license plate extraction and character recognition. Initially, contours are identified in the edge-detected image, and the top potential license plate locations are determined based on their areas. A mask is then created to isolate the license plate area, and the region of interest (ROI) is extracted from the original image. The extracted ROI is further processed using EasyOCR, a powerful Optical Character Recognition (OCR) tool, to recognize the alphanumeric characters on the license plate. To enhance accuracy, character mapping dictionaries are employed for potential corrections. The system dynamically adjusts to variations in license plate length, accommodating both 9 and 10-character plates. This comprehensive approach ensures robustness in handling diverse real-world scenarios, such as varying lighting conditions, plate angles, and plate types, making it an effective solution for Automatic Number Plate Recognition.

1) Parameteric Configuration: **Bilateral Filtering:**

- Diameter: Diameter of each pixel neighborhood. The value 15 means that the filter considers pixels in a 15x15 neighborhood around each pixel in the image. (Value: 15)
- Sigma color: A parameter that influences the color similarity within the pixel neighborhood. A higher value means that more distant colors will be considered as part of the neighborhood. (Value: 15)

Canny Edge Detection:

- Lower threshold: The lower threshold for the edges. Any gradient value below this threshold is considered not to be an edge. (Value: 100)
- Higher threshold: The higher threshold for the edges. Any gradient value above this threshold is considered to be a strong edge, and values between the lower and higher thresholds are considered as weak edges. (Value: 300)

findContours function:

- Retrieval mode: Retrieval mode specifies how contours are retrieved. RETR_TREE retrieves all of the contours and reconstructs a full hierarchy of nested contours. This hierarchy information is useful when contours are nested or have hierarchical relationships. (Value: cv2.RETR_TREE)
- Contour approximation method: Specifies the contour approximation method. CHAIN_APPROX_SIMPLE compresses horizontal, vertical, and diagonal segments and leaves only their end points. (Value: cv2.CHAIN_APPROX_SIMPLE)

OpenCV function cv2.approxPolyDP:

- Epsilon parameter: This is the maximum distance between the original contour and its approximation. It is the precision parameter. A smaller epsilon value will result in a more accurate approximation, but it may also result in more points in the approximation. (Value: 10)
- Closed contour: The boolean value specifies whether the approximated curve should be a closed contour (True) or not (False). If set to True, the function assumes the contour is closed and tries to approximate a closed curve. (Value: True)

B. Parking Spot Classification

The proposed approach for parking spot classification integrates a sophisticated methodology utilizing YOLOv8 for car detection and spatial analysis techniques. The system starts by loading a pre-trained YOLOv8 object detection model, which processes input images and generates predictions, including bounding boxes and class indices. The subsequent steps involve image preprocessing, definition of parking areas through specified vertices, and object filtering based on class labels (specifically 'car'). Each detected car's center coordinates are calculated, and using OpenCV's pointPolygonTest function, the system evaluates whether the car is within the predefined parking areas. The proposed architecture emphasizes spatial analysis, where the positioning of detected cars relative to parking areas is systematically assessed. If a car is confirmed to be within a designated area, the system records its details and updates the count for that particular region. Visualization techniques, such as drawing rectangles around detected cars, circles at their centers, and overlaying information about each car, contribute to a comprehensive representation of the parking scenario. The modular design allows for the adaptability of the system across diverse scenarios, enabling the definition and refinement of parking areas as needed. By leveraging YOLOv8 and spatial analysis, the proposed approach provides an effective and visually accessible solution for parking spot classification, offering a detailed understanding of car distribution within input images.

1.Load YOLO model 2. Define areas 3. Define other areas similarly 4. Initialize lists 5. Initialize other lists similarly 6. Process image 7. Process detected objects and count cars in each area Check each area Check other areas similarly 8. Count the number of cars in each area 9. Display the results on the image 10. Display other areas similarly 11. Show the final image

The presented pseudocode implements a Smart Parking System utilizing computer vision techniques and the YOLOv8 object detection model. The system focuses on automating the classification and monitoring of parking spaces in a given environment. Initially, the YOLOv8 model is loaded to predict objects in input images, specifically targeting cars. The code then defines various parking areas, each represented by a set of vertices. Detected cars are filtered based on their class labels, and their spatial coordinates are analyzed to determine their presence within predefined parking spaces. The system records the count of cars in each area, visually representing the results by drawing rectangles around detected cars, circles at their centers, and polygons around parking areas. The code

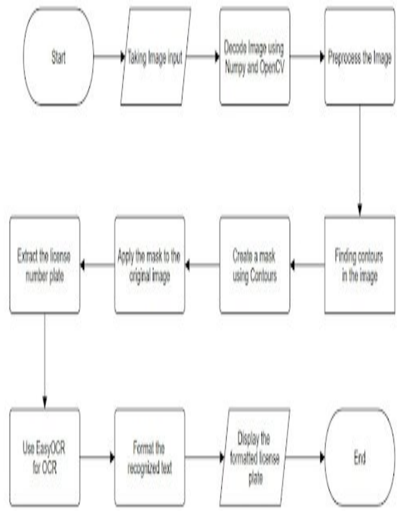


Fig. 1.

further calculates and displays the count of occupied spaces for each area, providing valuable insights into the availability of parking spots. This comprehensive approach to parking spot classification demonstrates the potential for real-world applications, contributing to the efficiency and management of parking spaces through automated image processing and object detection techniques.

1) Pretrained Model used : YOLOv8, or "You Only Look One-level v8," is an object detection algorithm that belongs to the YOLO (You Only Look Once) family of models. The YOLO series is known for its real-time object detection capabilities and has seen several iterations, with YOLOv8 being one of the latest as of January 2023.

Architecture of YOLOv8 :

- Backbone Network : YOLOv8 typically employs a backbone network like CSPDarknet53, which stands for Cross Stage Partial (CSP) Darknet 53. This is used for feature extraction from the input image.
- Neck : The network may include a neck architecture, like PANet (Path Aggregation Network), to enhance feature representation and improve detection accuracy.
- Head : The detection head of YOLOv8 is responsible for predicting bounding boxes, object classes, and confidence scores. It uses anchor boxes to predict the bounding box coordinates.
- Output Format : YOLOv8 outputs a tensor containing information about the detected objects. Each bounding box is associated with a class label, confidence score, and the coordinates of the bounding box.

Mathematical Concepts :

- Bounding Box Prediction: YOLOv8 uses anchor boxes to predict bounding box coordinates. The coordinates are usually represented as (x, y, w, h), where (x, y) is the center of the box, and (w, h) are the width and height.
- Object Confidence: YOLOv8 predicts the confidence that an object is present in a bounding box. This confidence score is a measure of how likely it is that the detected object is of the predicted class.
- Class Prediction: The model predicts the probability distribution over all the classes for each bounding box. The class with the highest probability is assigned to the object detected in that box.
- Loss Function: YOLOv8 employs a combination of localization loss, confidence loss, and classification loss. The total loss is calculated as the sum of these individual losses. The loss function guides the model during training to improve its predictions.
- Anchor Boxes: YOLOv8 utilizes anchor boxes to improve bounding box prediction. These anchor boxes are predefined boxes with specific widths and heights. The model adjusts these anchors during training based on the dataset.

C. Accident Detection

The proposed architecture of the Convolutional Neural Network (CNN) is designed for a classification task with four output classes. It consists of:

1. Input Layer: - The model begins with an input layer that expects images with dimensions (224, 224, 3), where 224 represents the height and width, and 3 corresponds to the RGB color channels. This layer serves as the entry point for the input data.

2. MobileNet Base Model: - The core of the architecture is the MobileNet base model, a well-known and efficient convolutional neural network architecture. This pre-trained model is included without the fully connected (dense) layers at the top. The purpose of this base model is to capture hierarchical features from the input images through a series of convolutional and depthwise separable convolutional layers.

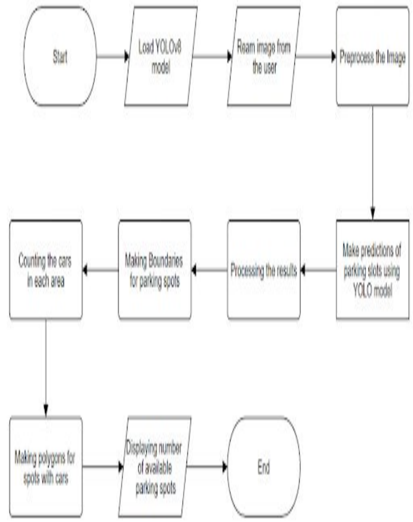


Fig. 2.

3. Global Average Pooling 2D: - Following the MobileNet base, a Global Average Pooling 2D layer is applied . This layer reduces the spatial dimensions of the tensor by computing the average value of each feature map. It serves as a spatial summarization technique, consolidating the learned features across the spatial dimensions.

4. Dropout Layer: - To prevent overfitting and enhance generalization, a Dropout layer with a dropout rate of 0.5 is introduced. This layer randomly sets a fraction of input units to zero during training, thereby introducing a form of regularization and reducing the model's reliance on specific neurons.

5. Dense Layer with Softmax Activation: - The final layer is a Dense layer (‘dense‘) with 4 units, indicating the number of output classes in the classification task. It is equipped with a softmax activation function, converting the raw model outputs into probabilities. The softmax function ensures that the sum of the class probabilities is equal to 1, enabling the model to make predictions across the specified classes.

In summary, this architecture leverages the features learned by the MobileNet base model, applies global average pooling for spatial aggregation, introduces dropout for regularization, and concludes with a dense layer for classification. The model is compiled using the Adam optimizer with a learning rate of 0.0001 and employs categorical crossentropy as the loss function, while accuracy is monitored as the evaluation metric during training.

1) Pretrained Model used : MobileNet is a lightweight convolutional neural network (CNN) architecture specifically designed for mobile and embedded vision applications, where computational resources may be limited. Developed by Google researchers Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, MobileNet introduces depth wise separable convolutions to achieve a good balance between model accuracy and computational efficiency.

The basic architecture of MobileNet comprises two key components: depthwise separable convolutions and linear bottlenecks.

Depthwise Separable Convolutions:

MobileNet replaces the standard convolutional layers with depthwise separable convolutions, which consist of two distinct operations: depthwise convolutions and pointwise convolutions.

Depthwise convolutions apply a single filter to each input channel independently, reducing computational complexity by performing convolutions on each channel separately.

Pointwise convolutions then apply 1x1 convolutions to combine the outputs from depthwise convolutions, allowing the network to learn cross-channel correlations efficiently.

Linear Bottlenecks:

To further optimize the model, MobileNet introduces linear bottlenecks, where a 1x1 convolutional layer is followed by a linear activation function. This linear activation function helps prevent information loss during the bottlenecks and allows for better representation learning.

The combination of depthwise separable convolutions and linear bottlenecks significantly reduces the number of parameters and computations, making MobileNet well-suited for resource-constrained environments. MobileNet is flexible and can be easily customized for various applications by adjusting hyperparameters such as depth and width multipliers.

IV. Dataset Desccription

Accident Detection

The dataset employed for training the Accident Detection module is Traffic-Net, curated by DeepQuest AI. Traffic-Net is instrumental in training machine learning systems to recognize and interpret various traffic conditions, with a specific emphasis on accident detection within the traffic environment. The release of Traffic-Net used, consists of 4,400 high-resolution

Type / Stride	Filter Shape	Input Size
Conv / s2	3x3x3x32	224 x 224 X 3
Conv dw/s1	3 x 3 x 32 dw	112 X 112 X 32
Conv /s1	1 x 1 x 32 x 64	112 X 112 x 32
Conv dw/s2	3 x 3 x 64 dw	112 x 112 x 64
Conv /s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw/s1	3 x 3 x 128 dw	56 x 56 x 128
Conv /s1	1x1x128 X	56 x 56 x 128
Conv dw/s2	3 x 3 x 128 dw	56 x 56 x 128
Conv /s1	1x1x128 X 256	28 x 28 x 128
Conv dw/s1	3 x 3 x 256 dw	28 x 28 x 256
Conv /s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw/s2	3 x 3 x 256 dw	28 x 28 x 256
Conv /s1	1 x 1 x 256 x 512	14 x 14 x 256
5x (Conv dw/s1 & Conv/s1)	3 x 3 x 512 dw & 1x1x512x512	14 x 14 x 512 & 14 x 14 x 512
Conv dw/s2	3x3x512dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw/s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 X 1024	7 x 7 x 1024
Avg Pool/s1	Pool 7 x 7	7 x 7 x 1024
FC/s1	1024 x 1000	1 x 1 x 1024
Softmax s1	Classifier	1x1x1000

TABLE I
ARCHITECTURE OF MOBILENET

images, categorized into four distinct classes: Accident, Dense Traffic, Fire, and Sparse Traffic. The Accident class is of particular relevance to our research. Each class encompasses 1,100 images.

Automatic Number Plate Recognition (ANPR) and Parking Slot Detection

For the Automatic Number Plate Recognition (ANPR) and Parking Slot Detection modules, explicit training was not required, as these modules rely on image processing techniques. Instead, random images sourced from the internet were utilized for testing purposes. These images were carefully selected to represent a variety of parking scenarios and license plate configurations, ensuring the robustness and adaptability of the implemented algorithms. The decision to use random internet images for testing aimed to evaluate the generalization capabilities of the ANPR and Parking Slot Detection modules under diverse real-world conditions. This approach aligns with the nature of image processing tasks, where the algorithms are expected to perform effectively on a wide range of input images

V. RESULTS AND DISCUSSION

A. Automatic Number Plate Recognition

Upon getting the coloured picture of the front view of the car containing it's number plate, it is first converted to greyscale image to simplify algorithms and reduce computational complexity as greyscale images contain only intensity information. Then, the image is passed through bilateral filtering and canny edge detection to find the biggest rectangle in the image, that is the license plate itself. Bilateral filtering is applied to reduce noise and the license plate is recognised by finding contours in the edge detected image and the final image is extracted using a mask. Finally, Optical Character Recognition (OCR) is applied on the license plate using EasyOCR to extract the license number.

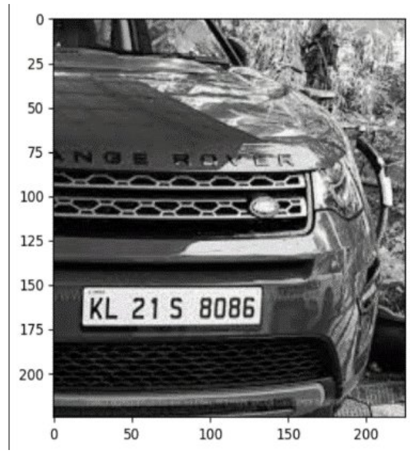
B. Parking Slot Detection

Here, firstly the parking spots are identified and marked with the help of OpenCV. Then, in real time, when the cars arrive, a pre-trained YOLOv8 model is used (yolov8s.pt) to identify the cars by iterating through rows of the DataFrame at that moment and checking if the centroid of the identified object i.e a car falls inside any of the pre-marked parking spots using OpenCV's pointPolygonTest function. If it does, then the respective parking spot will change it's colour showing the presence of a car there as shown in the last picture. Also, a variable called space is maintained to indicate the number of parking spots filled and left out of the total number.

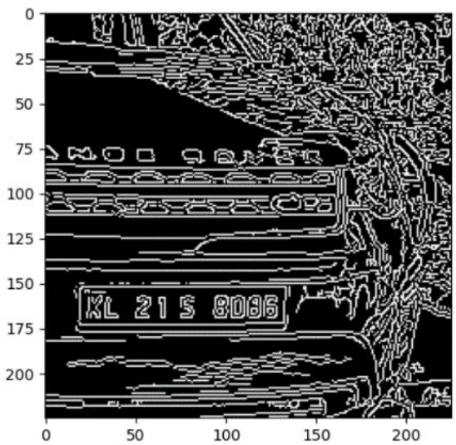
C. Accident Detection

The graph plots the training and validation loss over epochs. Two lines are plotted: one for training loss (labeled as "Train") and another for validation loss (labeled as "Val"). The x-axis is labeled as "epochs" and ranges from 0 to approximately 17.5. The y-axis is labeled as "loss" and ranges from 0 to 0.8. Both the training and validation loss decrease sharply initially and then level off, indicating that the model is learning but then reaches a point where improvements are marginal. The "Train" line is in blue, while the "Val" line is in orange.

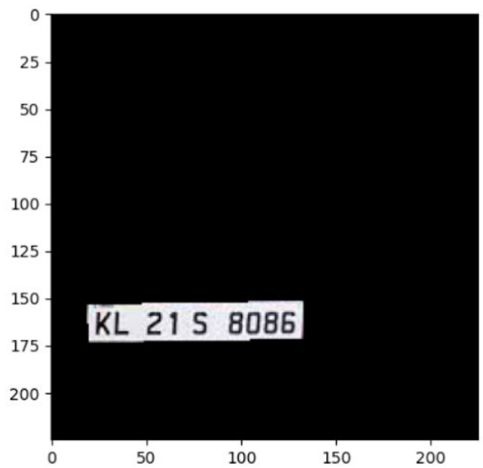
(a) Original image



(b) Gray Scaled Image



(c) After filtering and applying Canny Edge detection



(d) Number Plate extracted from the car

Fig. 3. Stages Automatic Number Plate Recognition

(a) At time t = t0 before

(b) At time t = t0 after

(c) At time t = t1 before

(d) At time t = t1 after

Fig. 4. Parking Slot Detection

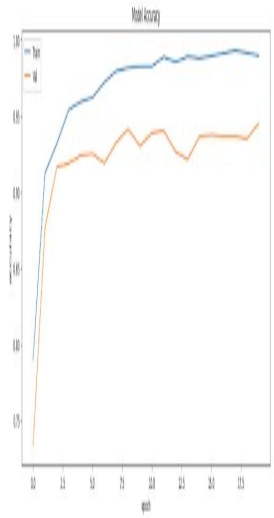


Fig. 5.

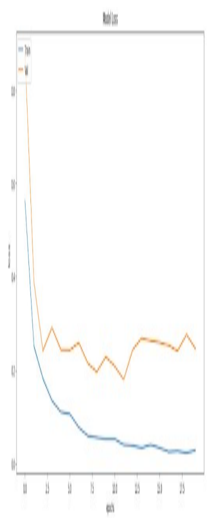


Fig. 6.

The graph shows the model accuracy over epochs for training and validation datasets. Two lines represent “Train” and “Val” (validation) datasets. The x-axis, labeled as “epoch,” ranges from 0 to approximately 17.5. The y-axis, labeled as “accuracy,” ranges from 0.75 to 1.00. The “Train” line is blue and shows a rapid increase in accuracy before plateauing just below 1.00 accuracy. The orange “Val” line also increases but begins to plateau at around 0.90 accuracy, showing some variance compared to the training data.

This is a table of machine learning model evaluation metrics. The table has five columns labeled: “precision”, “recall”, “f1-score”, and “support”. Four rows correspond to different classes (0, 1, 2, 3) with their respective scores in each metric. Below the class rows, there are additional rows for overall model performance metrics including “accuracy”, “macro avg”, and “weighted avg”. The accuracy of the model is 0.94. Macro average and weighted average precision, recall, and f1-score are also provided with values around 0.94 - 0.95.

VI. Conclusion

In this paper, we have used features such as Automatic Number Plate Recognition(ANPR), Parking Slot Detection, and Accident Monitoring in our Smart Parking System. These features provide a huge upgrade on the traditional parking system. ANPR is useful to the organisation for maintaining their parking record. Parking Slot Detection is useful to the user to find a parking spot without any manual input. Accident Monitoring is useful to both the user and the organisation to find any occurrence of mild accidents within the parking space. This dual focus on convenience and safety positions the smart parking system as a comprehensive solution that not only optimizes parking space utilization but also prioritizes the well-being and security of users and their vehicles. Overall, the significance of this research lies in its potential to improve current parking systems, highlighting the importance of continued exploration in this domain.

VII. Future Scope

Looking into the future, there’s a drive towards refining ANPR capabilities to include the recognition of license plates captured at oblique angles, ensuring adaptability in unconventional parking scenarios. Additionally, the expansion of Automatic Number Plate Recognition (ANPR) to encompass two-wheelers, recognizing the diverse array of vehicles in urban landscapes. Also incorporating various types of number plates and the different types of format of vehicle number. There is a promising avenue for the enhancement of smart parking systems to include the identification of minor incidents such as scratches and dents within parking areas among parked vehicles which isn’t currently possible due to lack of data. It would offer a more comprehensive level of security and awareness for vehicle owners. There is room to refine the existing YOLO model for parking slot detection by optimizing its parameters and fine-tuning on a diverse dataset. Improve annotation precision and implement post-processing techniques to enhance detection accuracy. Optimize real-time performance for dynamic parking scenarios and establish a user feedback loop for continuous model improvement. [1]

VIII. References

References

[1] A. Puranic, K. Deepak, and V. Umadevi, “Vehicle number plate recognition system: a literature review and implementation using template matching,” *International Journal of Computer Applications*, vol. 134, no. 1, pp. 12–16, 2016.

	precision	recall	f1-score	support
0	0.97	0.90	0.93	200
1	0.95	0.97	0.96	200
2	0.95	0.98	0.97	200
3	0.92	0.93	0.92	200
accuracy			0.94	800
macro avg	0.95	0.95	0.94	800
weighted average	0.95	0.94	0.94	800

TABLE II
EVALUATION METRICS OF THE PROPOSED APPROACH