International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

SecurePass: AI-Driven Intelligent Vehicle Access Control System

Pratik Bhawkar[1], Devika Rainak[2], Riya Demapure[3], Sahil Deshpande[4], Sakshi Bhoirkar[5], Varsha Chavan[6]

[1,2,3,4,5]Student, Artificial Intelligence and Data Science, Anantrao Pawar College of Engineering and Research, Pune, India
[6]Assistant Professor, Artificial Intelligence and Data Science, Anantrao Pawar College of Engineering and Research, Pune, India

Abstract
Vehicle Access Management plays a major role in ensuring the security of private organizations. Traditional methods might work at a small scale, but as data piles up, they tend to slow things down and invite human errors. To overcome these hurdles, SecurePass offers an AI-driven vehicle access management solution, enabling accurate and real-time monitoring, number plate identification, and validation. The system combines OpenCV, YOLOv8, and OCR technology to efficiently identify and extract license plates, validating them in real-time with the organization's secure records. The system features two core modules—an admin dashboard for data management, and a security interface for verifying and recording vehicle entries and exits. SecurePass streamlines the entire process, boosting security and operational efficiency.

Keywords: SecurePass, OpenCV, YOLOv8, OCR

1. Introduction

In today's fast-moving business environment, secure vehicle access isn't just a convenience—it's a necessity. Traditional manual systems often fail under pressure, becoming unreliable and error-prone when the vehicle data starts piling up. To meet the rising need for automation and reliability, SecurePass was developed as an intelligent vehicle access solution that harnesses the power of Machine Learning. This system is designed to streamline vehicle monitoring by utilizing cutting-edge technologies such as OpenCV, YOLOv8, and Optical Character Recognition (OCR). SecurePass enables real-time detection and extraction of vehicle registration numbers from input images, which are then accurately validated against a secure organizational database.

SecurePass features a clean, user-friendly dashboard tailored for two key roles: Administrators and Security Personnel. Admins can fully manage, update, and oversee vehicle data, ensuring everything stays accurate and up to date. Security staff handle real-time vehicle entry verification, making sure only authorized vehicles are granted access. The system also logs every vehicle's entry and exit, giving organizations a clear view of all movements

A key strength of SecurePass lies in its ability to manage high volumes of data effortlessly, making it perfectly suited for modern-day access control and parking management. What makes SecurePass even

more powerful is its customizable framework, allowing organizations to adapt the system to fit their unique security needs. It strengthens security, streamlines operations, cuts down on manual effort, and significantly lowers the risk of unauthorized access. integrating SecurePass allows organizations to streamline vehicle access with precision and efficiency, perfectly aligning with modern security and automation standards.

2.Proposed System

Overview of the System:

SecurePass is an intelligent, machine learning-based vehicle access management system designed to automate and streamline vehicle monitoring within private organizations. The system utilizes advanced technologies such as OpenCV, YOLOv8, and OCR to accurately detect and extract vehicle registration numbers from both real-time and uploaded image sources, validating them against a pre-defined dataset. With separate modules for administrators and security personnel, SecurePass ensures smooth management of vehicle records and real-time verification at access points. The system significantly cuts down on manual errors and enhances overall security operations.

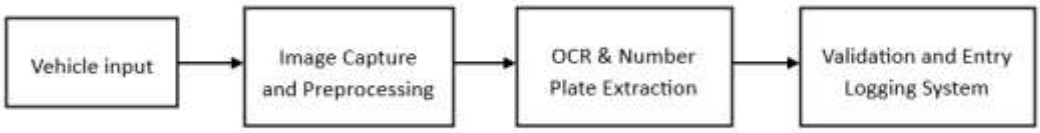

Figure1: Workflow of the system

System Architecture: SecurePass features a streamlined architecture aimed at automating vehicle entry processes in private facilities, leveraging real-time validation and role-specific access. The system supports three core user roles—Admins, Security Personnel, and Managers—all accessible via a unified login interface.

Login Page: This acts as the entry point for all users. Based on credentials, users are redirected to their respective modules — Admin or Security

Figure 2: Login Interface

Admin Module: The Admin Module enables administrators to oversee the backend vehicle data system. Once authenticated Admins can perform **CRUD operations** (Create, Read, Update, Delete) on vehicle registration entries via the **Manage Data** interface. They can also view **detailed logs** of vehicle access history to support auditing and system diagnostics.

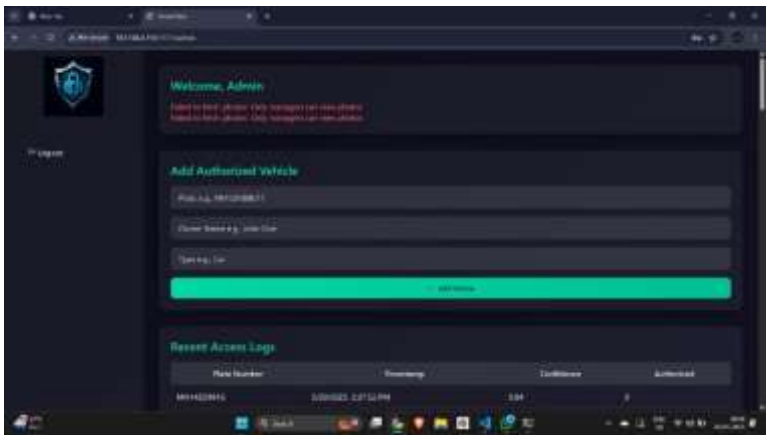

Figure 3: Admin Dashboard

Manager Module: The Manager role is a restricted-access sub-role within the Admin module. Managers cannot edit data, but they are granted access to daily vehicle entry logs for monitoring purposes. Managers can view a summary of how many vehicles have been granted entry each day. This role supports daily operational oversight without compromising database integrity**.**

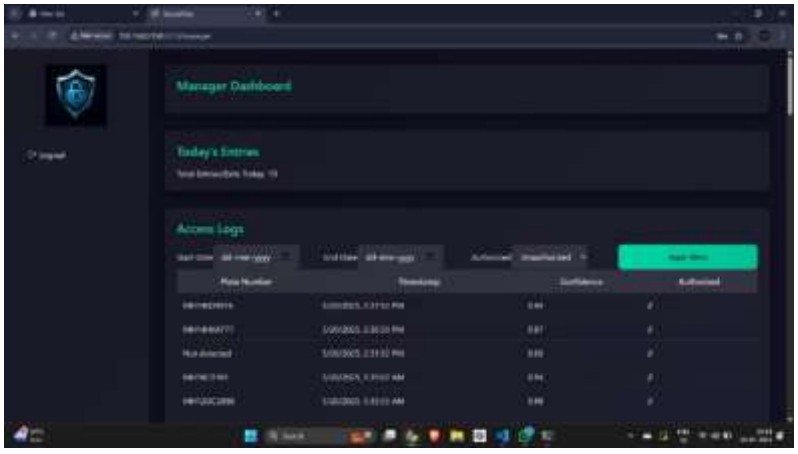

Figure 4: Manager Dashboard

Security Module: Security personnel use this module to perform live vehicle verification. Once logged-in they initiate the **OCR & Validation** process, which captures and processes license plate data. The extracted plate number is matched in real time against the Admin-managed dataset. Depending on the result: **Match Found:** Entry is automatically approved. **No Match / Suspicious Entry:** A manual check is triggered for further review.

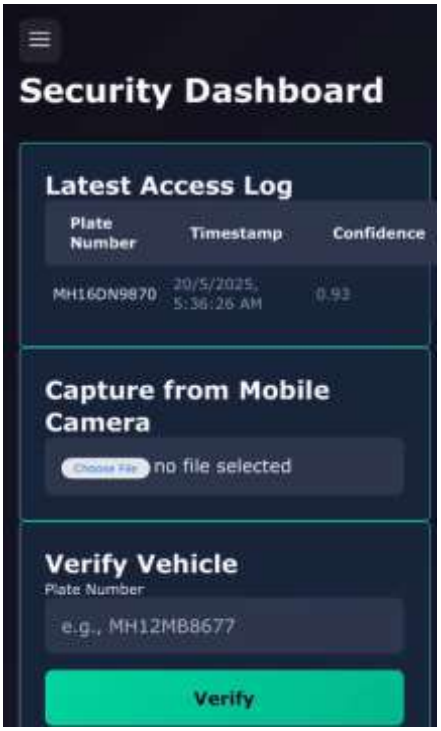

Figure 5: Security Dashboard

3. Implementation details

Technologies Used:

1. **OpenCV**: Captures and preprocesses vehicle images from real-time camera feeds.
2. **YOLOv8 (via Ultralytics)**: Detects and localizes vehicles in the captured images.
3. **PaddleOCR**: Extracts vehicle registration numbers from the detected vehicle regions.
4. **FastAPI**: Acts as the backend engine to process OCR results, verify them against the database, and manage API communication.
5. **SQLite**: Stores authorized vehicle data and logs every vehicle's entry and exit.
6. **SQLAlchemy**: Facilitates smooth interaction between FastAPI and the SQLite database using ORM.
7. **React**: Builds a dynamic and interactive frontend dashboard for security personnel and admins.
8. **HTML,CSS**: Structures and styles the user interface for a clean, user-friendly experience

Phase of Design:

The following methodology presents a comprehensive breakdown of the design and implementation stages for a real-time **Number Plate Detection and Recognition System**. The primary goal of this system is to detect vehicle number plates at entry points, recognize their alphanumeric content, and verify their authorization against a predefined database. The system leverages modern technologies such as computer vision, deep learning object detection (YOLOv8), and advanced optical character recognition (PaddleOCR) to ensure robustness, speed, and scalability. Below, the entire methodology is divided into multiple structured phases to facilitate clarity and traceability of each technical component involved in the system design

Data Collection: The first and most crucial step in building a Number Plate Detection and Recognition System is **data collection**, which involves gathering a diverse set of images containing vehicle number plates under varying real-world conditions. A robust and representative dataset ensures that the model can generalize well across different lighting environments, weather conditions, image resolutions, camera angles, and plate formats.

The dataset may consist of publicly available license plate image datasets, synthetically generated data, or

real-world images manually captured using cameras. These images should ideally include different vehicle types (cars, bikes, trucks), regional number plate styles, and various font types. Special attention is given to ensure the inclusion of images with challenges such as blurred plates, tilted angles, low resolution, and occlusion.

The collected dataset is then annotated by marking bounding boxes around the number plates. These annotations are stored in formats like YOLO (.txt) depending on the detection model requirements. Proper annotation ensures that the detection model (YOLOv8 in this case) can accurately learn the spatial characteristics of number plates.

Figure 6: Data Collection

Image Extraction using OpenCV: In this phase, image extraction is done using **OpenCV**, which processes the provided dataset to extract individual vehicle images. OpenCV helps enhance image quality through operations like resizing, grayscale conversion, and noise reduction. This ensures the images are clean and consistent before being passed on to the YOLO model for license plate detection.

Figure 7: Captured vehicle image processed using OpenCV

Number Plate Detection using YOLOv8: After taking the raw image input, the next step is to detect the number plate area within the frame. For this purpose, we employ YOLOv8 (You Only Look Once version 8), a state-of-the-art object detection algorithm developed by Ultralytics. YOLOv8 is known for its balance between speed and accuracy, making it suitable for real-time applications like vehicle entry systems.
YOLOv8 divides the image into multiple grid regions and predicting bounding boxes and class probabilities for each grid cell. For our system, it is trained to specifically recognize number plates as objects. A bounding box is drawn around the detected number plate area, which is then cropped for further processing. YOLOv8's architecture is composed of convolutional layers, residual blocks, and anchor boxes to localize small objects precisely, even under conditions of motion blur, low lighting, or occlusion.
This phase is the foundation of the system as the accuracy of plate recognition heavily relies on the correct detection and cropping of the number plate region.

Figure 8: Sample image showing annotated number plate extracted using OpenCV and YOLOv8 detection

Optical Character Recognition: Once the number plate is successfully detected and cropped from the input image, the next critical step is to extract the alphanumeric characters present on the plate.For this we use OCR-which converts printed or handwritten text into machine encoded data. In this project, PaddleOCR, a deep learning-based OCR tool, is utilized for its superior accuracy and multilingual support. The role of OCR in this pipeline is to bridge the gap between image processing and actual textual data. PaddleOCR takes in the segmented number plate image and performs two main functions: text detection and text recognition. First, it scans the image and identifies the location of characters using bounding box proposals. This involves detecting the orientation and line structure of the text. Then, it performs recognition — where each detected character or word is interpreted and converted into a string format (like "MH12AB1234").
OCR must work accurately despite challenges such as varying lighting conditions, skewed or blurry text, non-standard fonts, and background noise. PaddleOCR handles these efficiently due to its CRNN (Convolutional Recurrent Neural Network) architecture and CTC (Connectionist Temporal Classification) loss function, which allow it to recognize sequences of characters even if spacing or alignment is imperfect Furthermore, since Indian number plates have a specific format (typically starting with two letters denoting the state, followed by two digits for district code, then an alphabetic series and a unique four-digit number),

the OCR system can be fine-tuned to expect such formats, reducing false positives. For example, if a detected string is “MH12AB1234,” the system recognizes that it matches the expected Indian number plate structure.

Figure 9: OCR

Database Validation: After successfully recognizing the vehicle number through OCR, the system performs a critical verification phase — validating the extracted number against a centralized database maintained by the private society or residential complex.

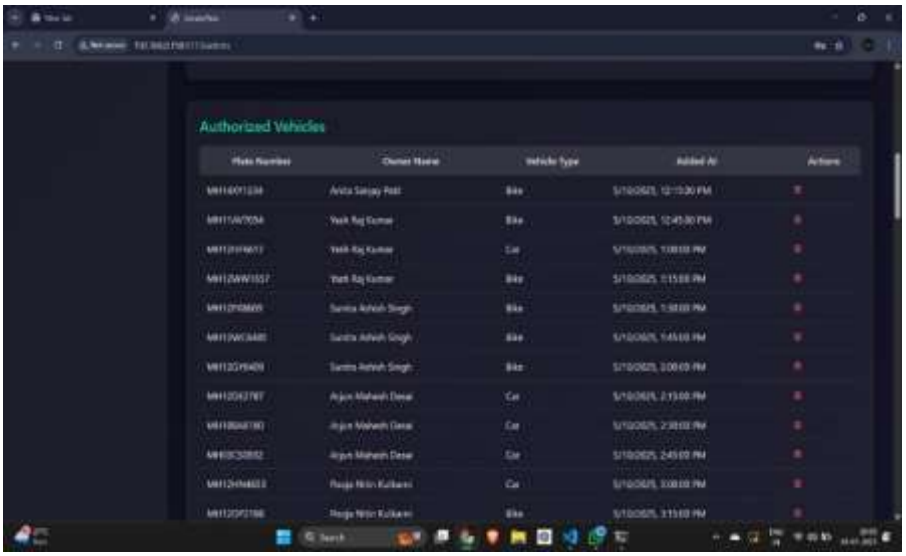

Figure 10: Centralized database storing pre-approved vehicle registration numbers for residents

This step is necessary to identify whether the incoming vehicle is authorized to enter the premises or not. The database typically consists of a pre-approved list of vehicle registration numbers, associated with residents, visitors, or service personnel. Once the number is extracted using PaddleOCR, the system immediately queries the database. If the number is found in the records, the system confirms the vehicle as "Authorized"; otherwise, it is flagged as "Unauthorized", and necessary alerts or logs are generated.

Figure 11: System detects a match - vehicle marked as 'Authorized' and granted entry.

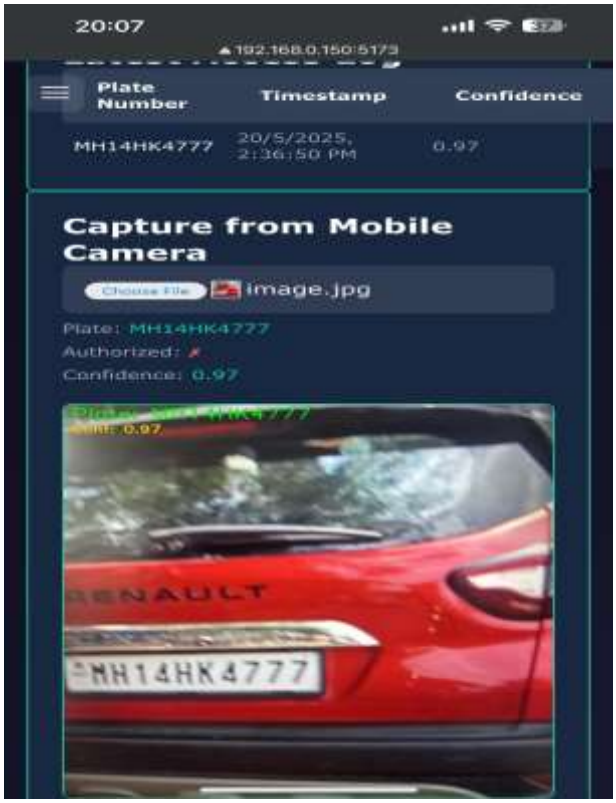

Figure 12: No match found — vehicle flagged as 'Unauthorized' with alert sent to security personnel

Model Training: Model training involves teaching the detection and recognition models to identify

vehicle number plates from a diverse dataset. For the **YOLOv8** model, the dataset consists of annotated images of vehicles, where each number plate is marked with a bounding box. The dataset is augmented with various transformations like rotation, brightness adjustment, and blurring to simulate real-world conditions such as motion blur, occlusion, and different camera angles. The YOLOv8 model, based on convolutional layers, learns to detect number plates by predicting bounding boxes and class probabilities. The **PaddleOCR** model is then trained to recognize the alphanumeric characters from the cropped number plate images, using a CRNN architecture that improves the accuracy of text recognition under varied conditions. Both models are fine-tuned using a loss function to minimize errors and enhance performance, particularly focusing on the expected structure of number plates, such as the Indian format, which helps in reducing false positives.

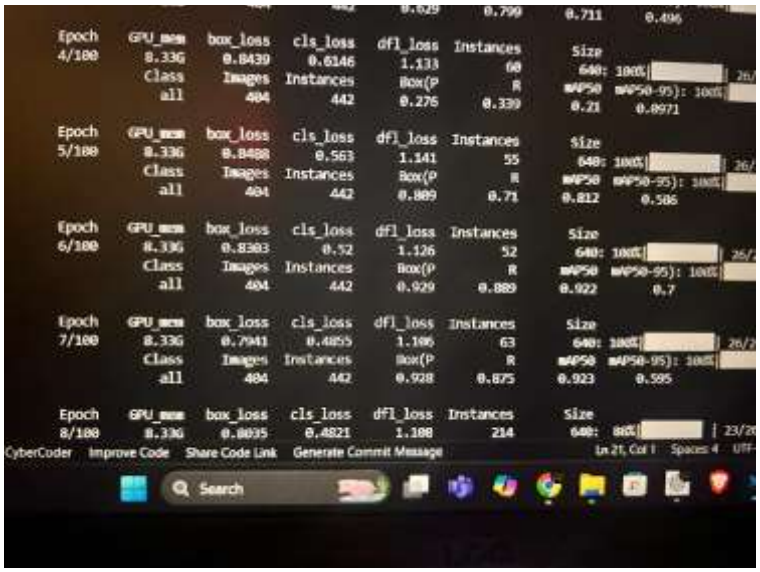

Figure 13: YOLOv8 Training Logs

Model Testing: Model testing evaluates the performance of the trained models on a separate test set that includes new, unseen vehicle images and number plates. For YOLOv8, this involves testing the model's ability to accurately detect number plates in different environmental conditions, measuring metrics such as precision, recall, and IoU (Intersection over Union) to assess the accuracy of bounding box predictions. In PaddleOCR, the model is tested on a range of distorted and noisy number plate images to measure its text recognition accuracy. The accuracy is evaluated by comparing the recognized text with the ground truth. Additionally, real-time performance is tested to ensure the models process images quickly enough for live applications like vehicle entry systems. Error analysis helps identify areas for improvement, such as handling skewed or blurred text, allowing for model fine-tuning to boost accuracy and robustness in real-world scenarios.

Figure 14: Model Testing Log on Real-World Number Plate Dataset

4. Future Scope

Like any evolving technology, SecurePass: Intelligent Vehicle Access Management System has room for further enhancement. While the current system effectively provides vehicle monitoring, validation, and access control, its functionality can be significantly expanded by integrating additional technologies. Future developments may include:

- **Biometric Integration**: Adding fingerprint scanning or facial recognition for secure driver authentication to ensure not only vehicle but also driver validation.
- **Predictive Analytics**: Leveraging machine learning algorithms to detect behavioral patterns and flag suspicious activities proactively.
- **Expanded Vehicle Classification**: Extending support for new categories of vehicles, such as electric and autonomous vehicles.
- **Mobile Application**: Creating a dedicated mobile app for on-the-go access, alerts, and monitoring.
- **Multilingual Support**: Introducing regional language options such as Marathi to make the system more accessible to a wider range of users.

These advancements would make SecurePass even more intelligent, inclusive, and capable of meeting the evolving demands of modern organizations.

5. Conclusion

In this project, we developed SecurePass: Intelligent Vehicle Access Management System, a smart and scalable solution designed to overcome the inefficiencies and security challenges of traditional vehicle monitoring methods in private organizations. SecurePass leverages cutting-edge technologies—such as machine learning, image processing, and OCR—to perform real-time number plate detection, classification, and validation. By automating vehicle access control, the system ensures that only authorized vehicles are permitted entry, while also maintaining detailed logs for accountability and compliance with regulatory standards. This integration of intelligent automation not only enhances security and operational accuracy but also minimizes human error and manual workload. As organizations continue to seek smarter, more efficient ways to manage access, SecurePass provides a flexible, user-friendly, and future-ready solution that can adapt to evolving security needs. With further development—such as biometric integration, mobile app support, and multilingual interfaces—SecurePass holds the potential to become a comprehensive platform for intelligent access management across diverse industries.

Reference:

1. License Plate Detection using OCR (M.L.S.N.S Lakshmi, Prasad Jones Christydass, Sreevardhan Cheerala, 2020).
2. Recognition of Vehicle Number Plate by Using Python and OpenCV (Prof. Aditya Bakshi et al., April 7, 2023).
3. Number Plate Recognition Using OpenCV-Python (Tella Pavani, DVR Mohan, March 2019).
4. Automatic Vehicle Number Plate Recognition Using TensorFlow Object Detection (D.U. Onyishi, E.E. Emina, April 2024),
5. An Automatic Framework for Number Plate Detection Using OCR and Deep Learning (Yash Shambharkar et al., 2023).
6. Automatic Number Plate Recognition System (Shubham Nandanwar, Vijay Rawal, Aditya Parab May 5, 2021).
7. Application of Machine Learning for Automatic Number Plate Recognition Using OCR Engine (Osakwe Anthony Abuchi, Dr. Arul Lenna Rose P. J, Dr. Aarthi E., May 5, 2024).
8. Automatic Number Plate Recognition System for Indian Number Plates Using Machine Learning Techniques (Gayatri Hajare, Utkarsh Kharche, Pritam Mahajan, Apurva Shinde, 2022).