

Intel® Unnati Industrial Training Program 2024

~ HEMANTH KUMAR A.R.

PROBLEM STATEMENT

PS - 13

Vehicle Movement Analysis and Insight Generation
in a College Campus using Edge AI

BRIEF SOLUTION

The proposed project aims to tackle the issue by monitoring the movement of vehicles entering or exiting the college campus using a camera, which is then further processed to detect the vehicle along with its license plate using a real-time object detection algorithm called YOLO (You Only Look Once). The extracted number plate location is fed to Tesseract OCR (Optical Character Recognition) algorithm which extracts text from the image.

Further these information is checked with registered vehicles database and all the details such as vehicle registration number, date, time of entry, time of exit, status of registration in the database, type of vehicle is stored.

This information is used to generate insights such as parking space availability, peak time of parking space occupancy, average time of vehicle parked for the day, type of vehicles entered or exited, origin state of the vehicle and many more which can be really useful when parking space efficiency have to be maintained with the limited number of resources.

FEATURES OFFERED

- **Object Detection:** Utilizes a YOLO-based (You Only Look Once) deep learning model to detect vehicles entering and exiting the parking lot. It successfully identifies the vehicles as 2-wheeler or 4-wheeler.
- **License Plate Recognition:** Employs Tesseract OCR, trained on custom fonts, to accurately read and recognize license plates.
- **Real-Time Tracking:** Monitors vehicle entry and exit, updating the status in real time.
- **Parking Space Availability:** Keeps track of occupied and available parking spaces, dynamically updating as vehicles enter or exit.
- **Historical Data Storage:** Stores vehicle data, including date, time, vehicle class, license plate number, registration status, and parking status in an ODS (Open Document Spreadsheet) file.
- **Visualization:** Provides a user-friendly web interface to display real-time data and historical trends.
- **Graphical Representation:** Features line graphs showing occupied and available parking spaces over time, facilitating quick and easy monitoring.
- **Batch Processing:** Capable of processing multiple images with different parameters in a batch, ensuring efficient and continuous data updates.

PROCESS FLOW PART

1. Vehicle Entry/Exit Detection

- ✓ **Sensor Activation:** Sensors at the entrance/exit detect the presence of a vehicle.
- ✓ **Image Capture:** Cameras capture images of the vehicle's license plate.

2. Image Processing

- ✓ **Preprocessing:** Images are preprocessed to enhance quality (e.g., noise reduction, contrast adjustment).
- ✓ **OCR Application:** An OCR model trained to read custom fonts extracts the license plate number from the image.

3. Data Handling

- ✓ **License Plate Validation:** Validate the extracted license plate number (e.g., format, length).
- ✓ **Timestamp Recording:** Record the timestamp of the vehicle's entry or exit.

4. Database Update

✓ **Vehicle Entry:**

- Add the vehicle's license plate number and entry timestamp to the database.
- Increment the live vehicle count.
- Update the total number of vehicles entered during the day.

✓ **Vehicle Exit:**

- Update the vehicle's exit timestamp in the database.
- Decrement the live vehicle count.

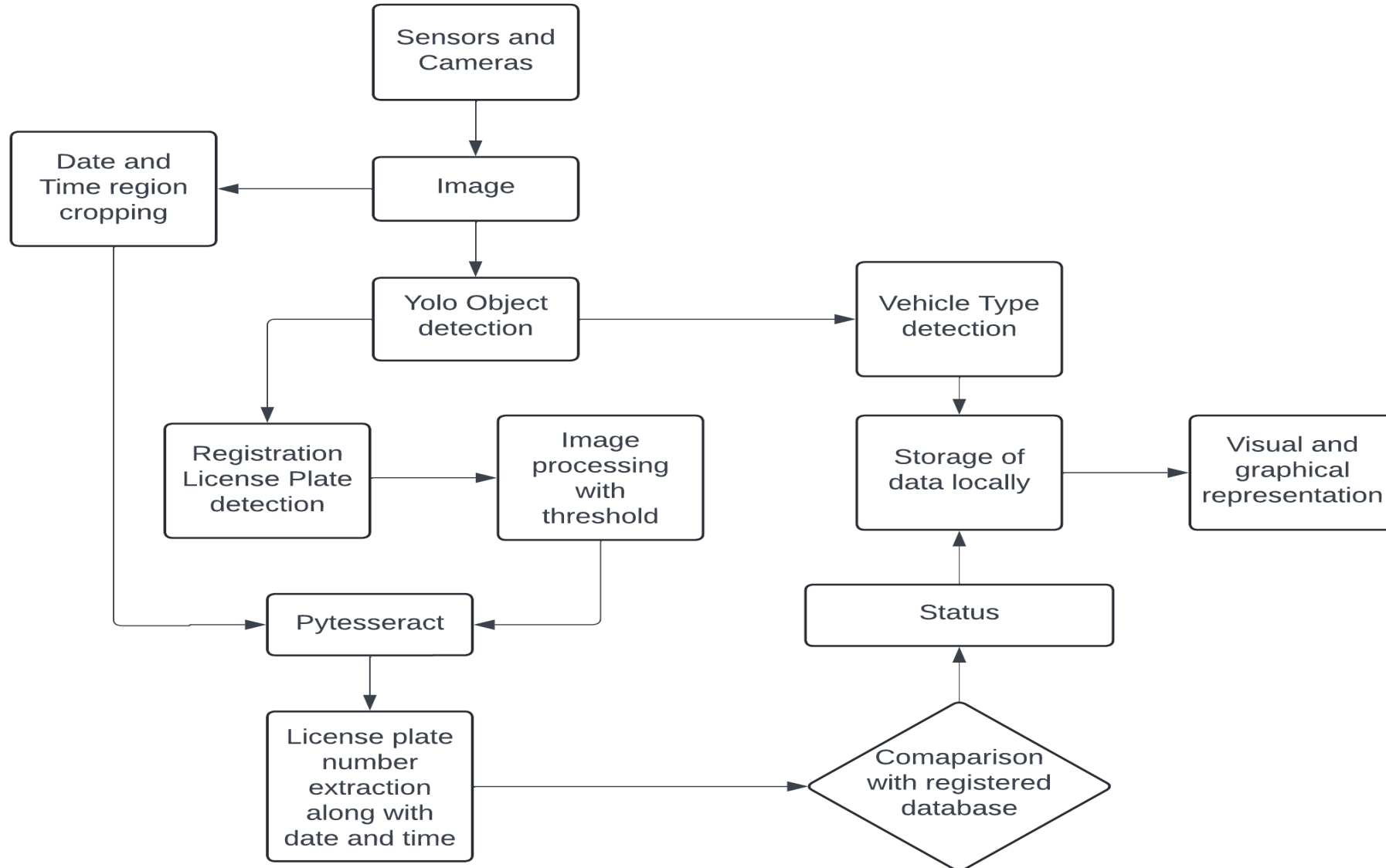
5. Parking Space Management

- ✓ **Space Availability Calculation:** Calculate the number of available spaces based on the live vehicle count and total parking capacity.
- ✓ **Space Allocation:** Assign available parking spaces to incoming vehicles.

6. Data Visualization

- ✓ **Real-Time Dashboard:** Display real-time data on a web interface, including: Live vehicle count, total vehicles entered during the day, available parking spaces.
- ✓ **Historical Data:** Generate graphs and reports showing parking trends over time.

ARCHITECTURE DIAGRAM



TECHNOLOGIES USED

- Python
- OpenCV
- NVIDIA CUDA Graphics Engine
- Pandas
- Numpy
- Yolo
- Pytesseract
- PyQt5
- Pyexcel-ods3
- Flask
- HTML
- Plotly
- Pillow
- Google Colab
- Git
- Pycharm



TEAM INFO

Team Size : 1

Team Leader and Member Name : Hemanth Kumar A. R.

Work Includes:

- Collection of images.
- Processing and annotation of images.
- Training custom Yolov4 Object Detection Model.
- Training Pytesseract for custom font.
- Development of Code.
- Optimisation of performance.
- Report Making.
- Idea Submission PPT making.
- Maintenance of Code Repository.

CONCLUSION

The project successfully addresses the critical need for efficient and accurate management of parking facilities. By integrating advanced OCR technology to read custom fonts, our system ensures precise recognition of vehicle license plates, enhancing the overall accuracy of vehicle entry and exit records.

The implementation of Python scripts to process images and update parking data in real-time has significantly improved the system's responsiveness and reliability. This real-time processing capability, coupled with the ability to calculate both live vehicle count and the total number of vehicles entered during the day, provides comprehensive insights into parking space utilization.

Overall, this project demonstrates the potential of leveraging modern technologies to create a robust and user-friendly parking management system. Future work could focus on improving the model's accuracy, further enhancing the system's scalability and integrating additional features such as predictive analytics for parking space availability and mobile app integration for end-users.