Simply Parking

Submitted by:

Japman Singh (G12/49/CSE1/2017)
Japneet Singh (G12/50/CSE1/2017)
Jaskaran Singh Kawatra (G12/53/CSE1/2017)
Jaskirat Singh (G12/54/CSE1/2017)

Under the guidance of:

Ms. Basanti Pal Nandi

Department of Computer Science & Engineering
Guru Tegh Bahadur Institute of Technology

Simply Parking

- Introduction
- Objective
- Technologies Used
- Procedure for Model
- Procedure for App
- Flow Diagrams
- Concept
- Conclusions
- References

Introduction

Simply Parking, a one stop solution for all parking problems. Our project takes care of available parking spaces, ticketing and all parking related charges and payments with support for both Android and IOS devices.

Problem Statement

With increasing modernization and automation in daily tasks, one task that still stays manual is parking systems. We aim to solve the problem of manual parking management systems which are labour intensive contactless and automated to minimize any inconveniences caused.

Objective

The aim of the project is to identify and detect the number plate of a car and then use that information to help the user pay their parking fee through an application. It involves the following steps:

1. Getting user registered on the application
2. Setting up a camera on parking boom barrier
3. Deploying the AI model with OCR and uploading results to Firebase
4. Providing user with an application to manage account and logs

All these processes are implemented using Open CV, Google Colab, Anaconda, Jupyter Notebook, Python Tesseract, Firebase, Flutter, Dart, Vehicle Info Restful API and numerous other libraries and technologies.

Our project consists of 3 elements

1. Deeplearning Model trained for number plate recognition.
2. OpenCV program for plate detection and OCR.
3. Flutter application to manage data and make payments.

Technologies used:

1. Labelbox
2. Colab
3. Jupyter Notebook
4. Keras
5. OpenCV
6. PyTesseract
7. Firebase
8. Flutter

Open CV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

Tensor Flow

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming. It is used for both research and production at Google.

Colab

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited. It also offers GPU support for training Deeplearning models.

Py-Tesseract

Py Tesseract or Python-tesseract is an Optical Character Recognition (OCR) tool for Python. It will read and recognize the text in images, license plates etc. Python-tesseract is actually a wrapper class or a package for Google’s Tesseract-OCR Engine. It is also useful and regarded as a stand-alone invocation script to tesseract, as it can easily read all image types supported by the Pillow imaging libraries

Firebase

Firebase is an ecosystem of Google tools that can be used to make full-stack, scalable applications in the Google Cloud (called Cloud Firestore) or Realtime Database. It is categorized as a backend-as-a-service (or BaaS) which gives developers the opportunity to create applications without the hassle of setting up the backend. It is a blend of SQL and NoSQL databases.

Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows and the web from a single code base. The first version of Flutter was known as codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit, with the stated intent of being able to render consistently at 120 frames per second.

Procedure for Deep Learning Model

1. Install a Python distribution with Jupyter Notebook support or use Google Colab’s Jupyter Notebook.
2. Click pictures manually as well as collect from the internet.
3. Label number plates data using Label Box.
4. Export this data to JSON file.
5. Load it into Pandas dataframe.
6. Train the detection model.
7. Test the model on test cases.
8. Make the model converge.
9. Crop detected image and feed it to Python Tesseract OCR
10. Upload OCR detected output on firebase.

Procedure for Flutter Application

1. Get the Flutter SDK: Download the installation bundle to get the latest stable release of Flutter.
2. Setup Android Studio for flutter.
3. Connect application with Firebase Backend.
4. Code the Sign-In module with support for Google sign-in from Firebase.
5. Create Parking, Logs and MyAccount screens and build the flow.
6. Code the Add Vehicle screen and integrate VehicleInfo-RestfulAPI to obtain Car details from number plate.
7. Add vehicle details in Cloud-Firestore.
8. Fetch vehicle number from AI-Model-PyTesseract output and generate parking log in Firebase.
9. Respond to ping when same number plate is detected at exit. Calculate fare using timestamp and debit amount from payment gateway.

Flow Diagrams

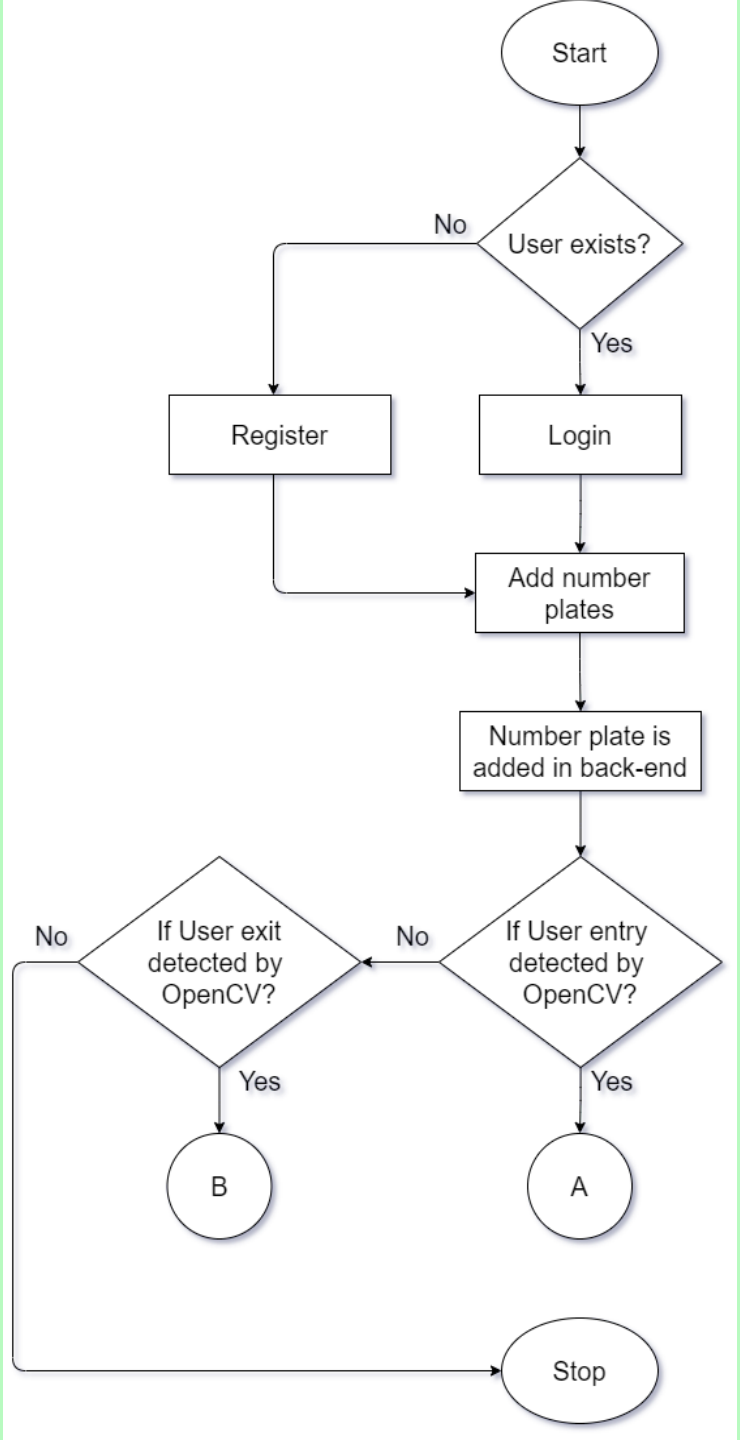


Fig – Application flow Simply Parking (1)

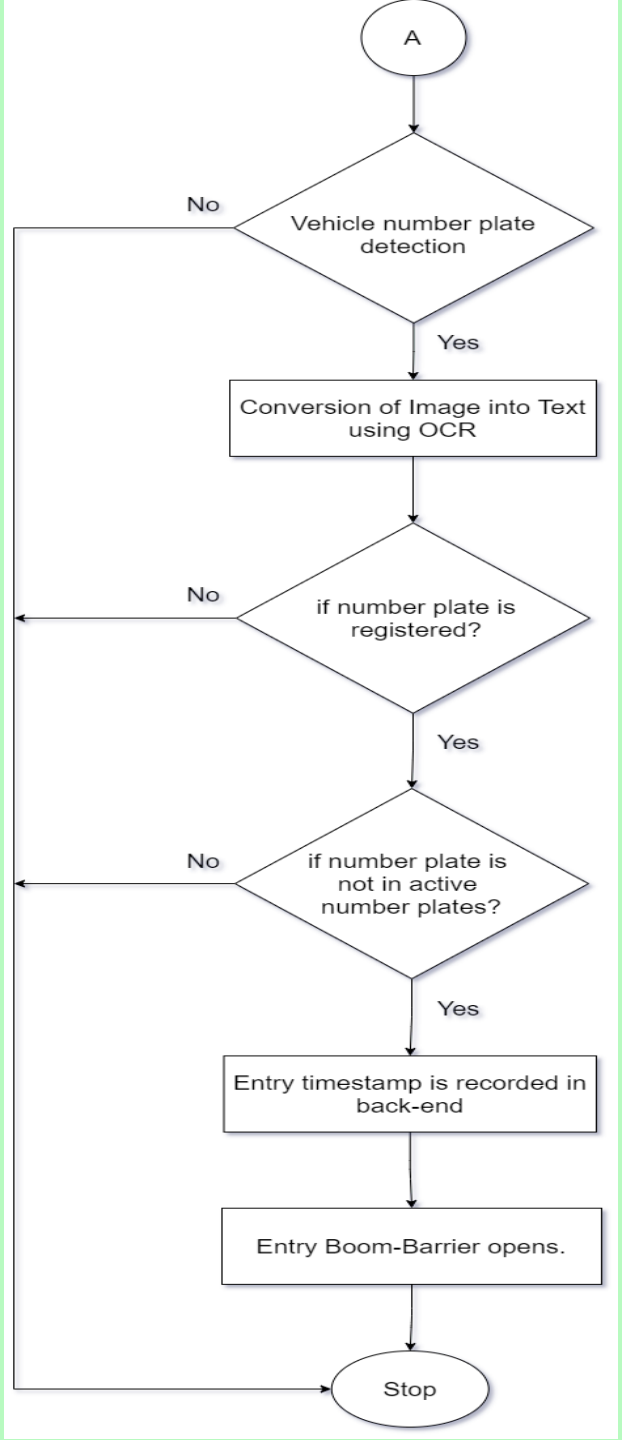


Fig – Application flow Simply Parking (2)

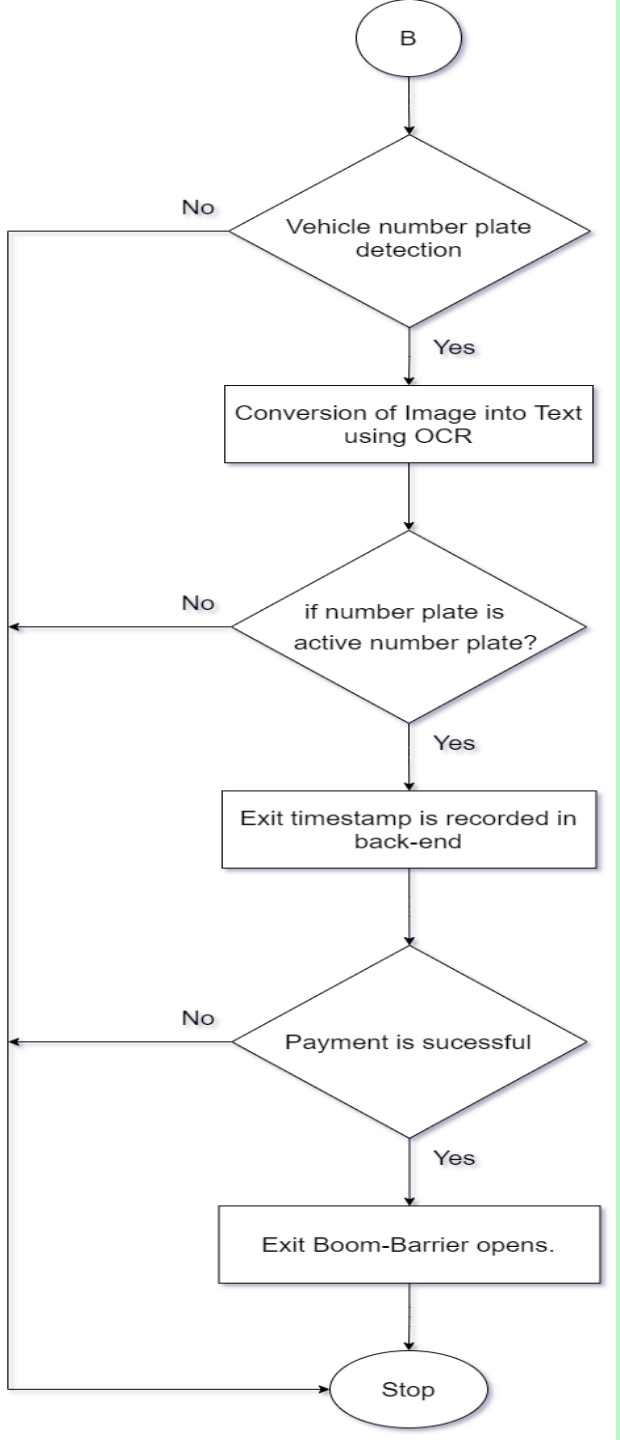
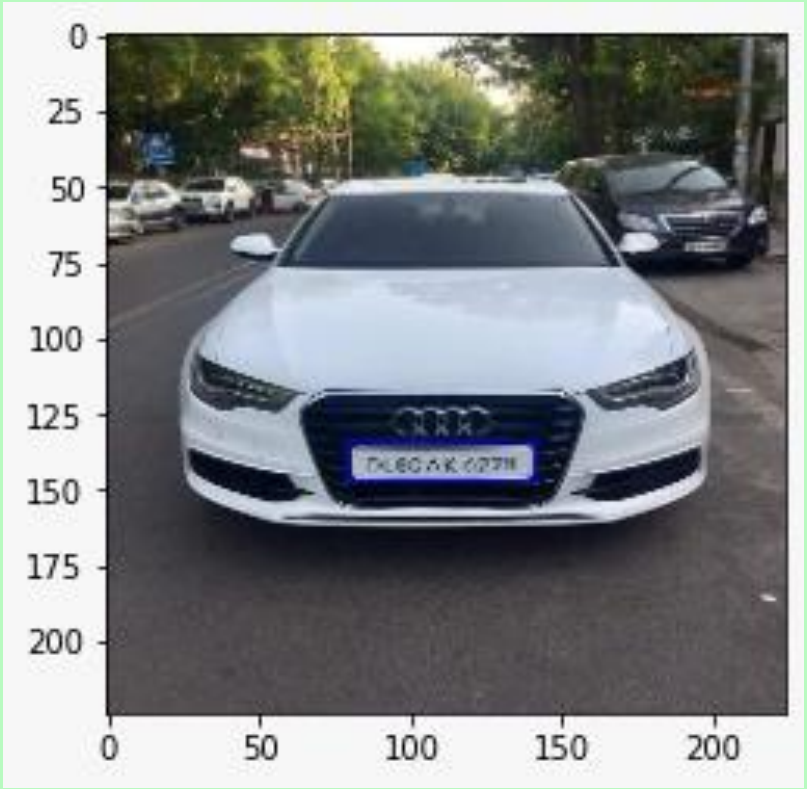


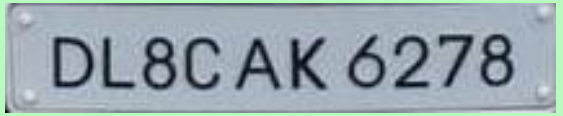
Fig – Application flow Simply Parking (3)

Project Demonstration

The system has been tested on a series of different test cases based on various types of number plates in India and also abroad.

Expected Output

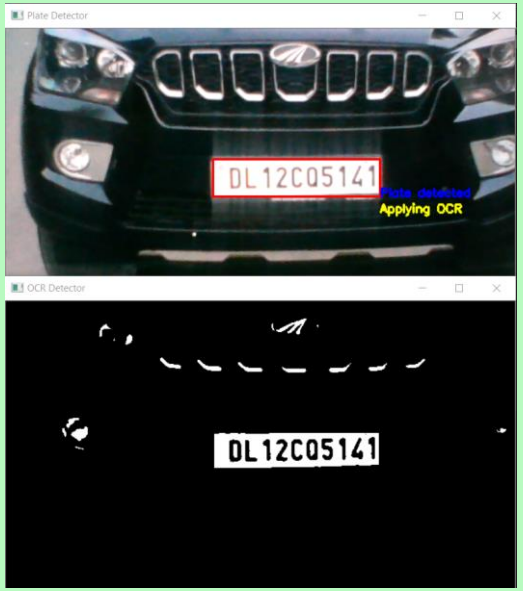




Expected Output Sample (For Reference)

Test Case(1) – Plate Detection
Status - Success

Input Camera Feed



Detected Plate and Mask

Test Case(2) – Plate Detection
Status - Success

Input Camera Feed

Detected Plate and Mask

Test Case(3) – Plate Detection
Status - Success

Input Camera Feed

Detected Plate and Mask

Test Case(4) – Plate Detection
Status - Success

Input Camera Feed

Detected Plate and Mask

Test Case(5) – Plate Detection
Status - Success

Input Camera Feed

Detected Plate and Mask

Test Case(6) – Plate Detection
Status - Success

Input Camera Feed

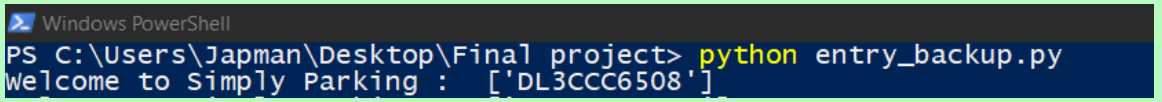
Detected Plate and Mask

Test Case(7) – Plate Detection
Status - Failed

Input Camera Feed

Detected Plate and Mask

Test Case(1) – OCR
Status - Success



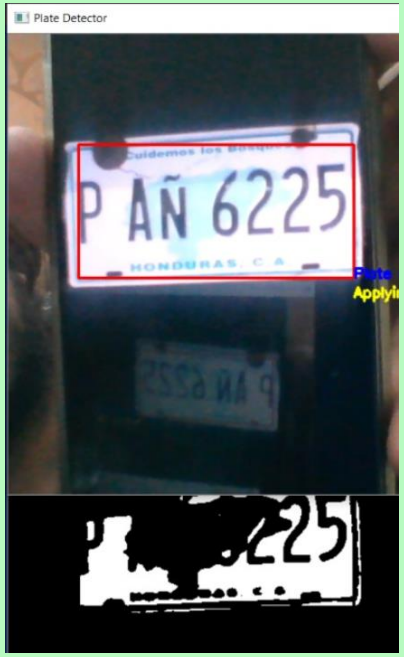
Test Case(2) – OCR
Status - Success

```
Windows PowerShell
PS C:\Users\Japman\Desktop\Final project> python entry_backup.py
Welcome to Simply Parking :  ['DL3C BA 5115']
```

Test Case(3) – OCR
Status - Failed
Reason – Due to Q similar to 0, the system detects last 4 digits as 0514 instead of 5141

```
Windows PowerShell
PS C:\Users\Japman\Desktop\Final project> python entry_backup.py
Welcome to Simply Parking :  ['DL 12C0514']
```

Test Case(4) – OCR
Status - Failed
Reason –
Due to Regex based on Indian format, detection fails



Detected Plate and Mask

Test Case(5) – Plate Detection
Status - Failed
Reason –
Due to Regex based on Indian format, detection fails

Detected Plate and Mask

Conclusion

The system was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of application.

- It provides a friendly graphical user interface which proves to be better for user.
- It will help users to avoid long queues at parking and avoid the wastage of time.
- It makes the parking system less tedious and hassle-free.
- It is a simple and efficient solution for the users to experience an automated parking system.
- The User interacts with an interface which simulates an automated parking system wherein the user’s car number plate can be identified and based on the plate number and the number of hours elapsed at the parking destination, the user is charged on the app itself

Future Prospects

- The software will be further enhanced with the addition of more training sets.
- The OCR in number plate is not 100% accurate and slow therefore we will try different approaches for OCR recognition.
- Optimise OCR to detect difference between similar characters like O, 0 and Q.
- Add option for available number of parking spaces
- Find nearby empty parking spaces using user geo-location.
- Payment gateway needs to be implemented in our mobile application using secure payment channels like PayTM.
- UI changes will be made to make the software more interactive.
- Login from Facebook, LinkedIn, and E-Mail can also be implemented.
- Support option can be provided to the customer in our mobile application in the future.

Test Cases

References

Books, Journals and Articles

[1] O'Reilly - Hands On Machine Learning with ScikitLearn, Keras and TensorFlow, Pg 80-165

[2] Allibai, E. (2018, November 21). Building a deep learning model using Keras. Medium. https://towardsdatascience.com/building-a-deep-learning-model-using-keras-1548ca149d37

[3] Zhang, Wei (1988). "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of Annual Conference of the Japan Society of Applied Physics.

[4] Vishal Gupta – Restful API Guide - https://shrouded-falls-48764.herokuapp.com/docs/readme.html

[5] Venkatachalam, M. (2020, May 14). Introduction to object detection. Medium. https://towardsdatascience.com/introduction-to-object-detection-943f21e26063

[6] Labenz, C. (2020, October 15). Testable flutter and cloud Firestore. Medium. https://medium.com/flutter/testable-flutter-and-cloud-firestore-1cf2fbbce97b

[7] Uther, . (2017). Temporal difference learning. Encyclopedia of Machine Learning and Data Mining, 1233-1240. https://doi.org/10.1007/978-1-4899-7687-1_817

[8] Wright, S. (2013). App logic components. Pro SharePoint 2013 App Development, 245-282. https://doi.org/10.1007/978-1-4302-5885-8_9

Online References

[1] https://flutter.dev/docs/reference/tutorials

[2]. https://www.w3adda.com/flutter-tutorial

[3] https://stackoverflow.com/questions/49072957/flutter-dart-open-a-view-after-a-delay

We would also like to thank the Teaching Faculty at GTBIT, our HOD Dr. Aashish Bhardwaj and our Mentor for the Project, Ms. Basanti Pal Nandi for their immense efforts in making this project successful.

THANK YOU