AUTOMATIC FARE GENERATION SYSTEM FOR PARKING LOTS

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

CSE2006 – Microprocessor and Interfacing

by

Name	Registration Number
Mohd Ayan Khan	20BCE0644
Nunnaguppala Babi	20BCE2851
Jayanth T	20BCE0967
A Aryan Rajesh	20BCE0718

Under the guidance of Prof. Dr Arun Kumar Chandrasekhar

SENSE

VIT, Vellore

DECLARATION

I hereby declare that the thesis entitled “AUTOMATIC FARE GENERATION SYSTEM FOR PARKING LOTS” submitted by me, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering* to VIT is a record of bonafide work carried out by me under the supervision of Prof. Arun Kumar Chandrasekhar.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: 6.04.2022

CERTIFICATE

This is to certify that the thesis entitled "AUTOMATIC FARE GENERATION SYSTEM FOR PARKING LOTS" submitted by Mohd Ayan Khan (20BCE0644), Nunnaguppala Babi (20BCE2851), Jayanth T (20BCE0967), A Aryan Rajesh (20BCE0718), SCOPE, VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by him / her under my supervision during the period, 07.12.2022 to 05.05.2023, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Vellore

Date: 05.04.2023

Signature of the Guide

Internal Examiner **External Examiner**

Head of the Department Programme

ACKNOWLEDGEMENTS

We would like to thank our Professor for Microprocessor and Interfacing, Prof. Arun Kumar Chandrasekhar, to give us her guidance and support to work on this project during the duration of our course.

We extend this acknowledgement to the authors and publishers of the different papers that we used as our citations and without whom this project would be impossible to complete.

We would also like to thank our institute, Vellore Institute of Technology, for giving us the opportunity to work on this project.

Mohd Ayan Khan
(20BCE0644)

Nunnaguppala Babi
(20BCE2851)

Jayanth T
(20BCE0967)

A Aryan Rajesh
(20BCE0718)

EXECUTIVE SUMMARY

'The measure of intelligence is the ability of change' - Albert Einstein

Our project aims to provide an efficient and automatic fare generation system for parking lots that cater to multitudes of cars on a daily basis. The streets and areas that usually bustle with visitors every day require deployment of human sources to manage the parking of the visiting cars, calculate the estimated fares and collect the payment at the time of exit. This project will automate this procedure, thus eliminating the need to employ workers, preventing intentional over charging of fares for personal interests, providing an additional security measure in the form of image and database records, and improving the efficiency by reducing the human errors in the process.

CONTENTS Page No.

List of Figures

KEY WORDS

- Fare Generation
- Machine Learning
- Computer Vision
- Automobile
- Parking System
- Deep Learning
- Neural Networks

1. INTRODUCTION

1.1. OBJECTIVE

In India, owing to a large population, car sales growth hikes every year. In the year 2017 itself, the growth was reported to be 9.2 percent, highest in past four years. This drives the need for strategizing methods for efficient systems that simplifies the management of these automobiles. One such domain is the Parking Lots at various locations across the country.

Our project aims to provide an efficient and automatic fare generation system for parking lots that cater to multitudes of cars on a daily basis. The streets and areas that usually bustle with visitors every day require deployment of human sources to manage the parking of the visiting cars, calculate the estimated fares and collect the payment at the time of exit.

This project will automate this procedure, thus eliminating the need to employ workers, preventing intentional over charging of fares for personal interests, providing an additional security measure in the form of image and database records, and improving the efficiency by reducing the human errors in the process.

To implement this, the type of the vehicle and its identification, the license plate, are necessary to be determined by the system designed. This project engages Computer Vision and Machine Learning tools for the same.

1.2. MOTIVATION & BACKGROUND

Over the last decades the number of cars on Indian roads have expanded rapidly. As the size of cities has reduced even further due to population overload, the need for systems that aim to efficiently manage expulsion of automobiles came to existence. The parking of vehicles has always been a matter of concern especially in metropolitans and with that insight the creation smart-parking systems took the foreground.

Even with the development of techniques that improve the efficiency of the parking space available, the responsibility of fare collection is left to the human workforce, which more often than not, has proven to be a drawback.

The conventional fare collection system usually involves issuing of slips mentioning the entry and their collection along with the decided fare at the time of exit. This process, at the time of entry, slows down invariably due to less workforce and larger number of visiting cars. And the slips then issued are required to be kept safe and handy, proves to be quite a task for regular users.

Furthermore, it’s a common occurrence that the users are overcharged by the employed workers for personal gains by printing misleading prices of the slips that don’t abide by the set rules. This, along with no record of the parked cars apart from the paper slips, have an added security risk to the vehicles. All of these discrepancies not only make the process cumbersome for the users but prove to be a hindrance in the making of a smart parking system.

As the world treads the path of Artificial Intelligence, renovating the systems to make them as human management independent as possible has utmost importance. An automatic system thus not only pave way for a hassle free fare generation but increases the overall efficiency thus saving time and money of the users.

2. PROJECT DESCRIPTION AND GOALS

2.1. LITERATURE REVIEW

Computer Vision is the core of Artificial Intelligence which allows computers to take intelligent decisions based on the visual details around them. It includes various takes such as object detection and recognition, image classification and segmentation. Image classification is the task of categorizing images into one of several predefined classes, is a fundamental problem in computer vision.

Vehicle classification done using a visual-based dimension estimation method [17] ,extracts moving vehicles from traffic image sequences and fits them with a simple deformable vehicle model. A set of coordination mapping functions are derived from a calibrated camera model and relying on a shadow removal method, vehicle's width, length and height are estimated [17].

Jun-Wei Hsieh et al. [15] developed a Make- and -Model Recognition system to detect vehicles and recognize using a symmetrical SURF descriptor. The vehicle's front region is separated into several grids and different weak classifiers trained on these grids are integrated to build an accurate ensemble classifier [15].

In recent years, deep learning models that exploit multiple layers of nonlinear information processing, for feature extraction and transformation as well as for pattern analysis and classification, have been shown to overcome these challenges. Among them, CNNs (LeCun, Boser, Denker,Henderson, Hubbard, & Jackel, 1989a, 1989b) have become the leading architecture for most image recognition, classification, and detection tasks (LeCun, Bengio, & Hinton, 2015) Zhen Dong et al. [16] used semi-supervised convolutional neural network to perform vehicle type classification using vehicle frontal-view images extracts rich and discriminative information of vehicles is captured using sparse Laplacian filter learning to obtain the filters of the network with large amounts of unlabeled data . The softmax classifier,used as the output layer is trained by multitask learning with small amounts of labeled data [16].

ResNet is a newly developed CNN architecture which won the 1st place in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015 classification task [3]. ResNet enhances the number of layers by inserting shortcut connections which turn the network into its counterpart residual version. These shortcut connections perform the identity mappings in the network [3]. ResNet-based vehicle classification method has been used before by Heechul Jung [18] for real-time traffic surveillance recordings by utilizing a MIOvision traffic dataset, which comprises 11 categories such as bicycle, motorcycle, car,bus etc which achieved an accuracy of around 97.95% on average.

License Plate Detection systems have been employed widely over the world for real time tracking of vehicles, curb criminal activities etc. Various techniques have been developed using different softwares like LabView, MATLAB, OpenCV using different approaches like edge based, neural networks, sliding window techniques amongst others were used. While the initial edge based techniques didn't produce accurate results, with no license plate detection or incorrect detection [13], the techniques developed later on which used median-filters and neural networks, though gave a better accuracy, but could only be used to detect english characters.[14]

2.2 GOALS AND OBJECTIVES

Our project aims to provide an efficient and automatic fare generation system for parking lots that cater to multitudes of cars on a daily basis. The streets and areas that usually bustle with visitors every day require deployment of human sources to manage the parking of the visiting cars, calculate the estimated fares and collect the payment at the time of exit. This project will automate this procedure, thus eliminating the need to employ workers, preventing intentional over charging of fares for personal interests, providing an additional security measure in the form of image and database records, and improving the efficiency by reducing the human errors in the process.

2.3 SOLUTION

In this project, we will be creating such a device which will detect the approaching vehicle towards it with the help of a ultrasonic sensor and after a certain threshold it will send the signal for execution of code which will scan the number plate of the vehicle and register it in the database. At the time of exit, again the ultrasonic sensor will repeat the same process and again give the signal for execution of code. Since the vehicle is already registered, the device will fetch the entry time from the database and based on the current time calculate the hours and generate fare based on the type of vehicle. The fare generated will be auto debited from the wallet of the individual if there is sufficient balance available.

3. DESIGN AND ARCHITECTURE

3.1 HARDWARE REQUIREMENTS

3.1.1 Raspberry Pi 3

Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. It is an SoC (System on chip), which integrates all the components of a computer or other electronic systems on a single chip like a GPU, Wifi-Module etc.

Raspberry Pi 3 Specifications:
SoC: Broadcom BCM2837
CPU: 4× ARM Cortex-A53, 1.2GHz
GPU: Broadcom VideoCore IV
RAM: 1GB LPDDR2 (900 MHz)
Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage: microSD
GPIO: 40-pin header, populated
Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI) [20]
OS USED: Raspbian Stretch

Fig 3.1: Raspberry Pi 3

3.1.2 UltraSonic Sensor (HC-SR04)

The human ear can hear sound frequency around 20HZ ~ 20KHZ, and ultrasonic is the sound wave beyond the human ability of 20KHZ.[21] HC-SR04 is an ultrasonic ranging module that provides 2 cm to 400 cm non-contact measurement function. The ranging accuracy can reach to 3mm and effectual angle is < 15°. It can be powered from a 5V power supply.[19]

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Fig 3.2: HC-SR04 Specifications

Fig 3.3: Ultrasonic Sensor

3.1.3 DC MOTOR

The human ear can hear sound frequency around 20HZ ~ 20KHZ, and ultrasonic is the sound wave beyond the human ability of 20KHZ.[21] HC-SR04 is an ultrasonic ranging

module that provides 2 cm to 400 cm non-contact measurement function. The ranging accuracy can reach to 3mm and effectual angle is < 15°. It can be powered from a 5V power supply.[19]

Fig 3.4: DC Motor

3.2 SOFTWARE REQUIREMENTS

3.2.1 TENSORFLOW

TensorFlow is an open source software library for high performance numerical computation, originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning. Owing to its flexible architecture, it allows deployment across different platforms ranging from desktops to mobile and EDGE devices. [3]

3.2.3 KERAS

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano, two of the top numerical platforms in that provide the basis for Deep Learning research and development. [5] It was developed with a focus on enabling fast experimentation. Keras Python library that provides a clean and convenient way to create a range of deep learning models.

3.2.3 PYTHON 3.6.4

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

3.2.4 LIBRARIES USED

The following Libraries were used in this project:

- Open CV
- Pytesseract
- NumPy
- Pandas
- Matplotlib
- Augmentor
- Theano

3.2.5 GITLAB

GitLab is the leading integrated product for modern software development. It connects issue management, version control, code review, CI, CD, and monitors into a single, easy-to-install application which helps teams go faster from planning to monitoring. [28]

3.3 ARCHITECTURE

3.3.1 Object Detection

When a vehicle approaches the entry barrier, its presence is detected using an ultrasonic sensor. The ultrasonic sensor intimates the processor of the presence of the vehicle which then turns on the camera. The camera captures the image of the vehicle and that image is then used to recognize license plate detection and vehicle classification. Ultrasonic Sensor is the most commonly used sensor to detect a moving target and estimate the approximate distances it is present at.

Ultrasonic sensors work on the principle of emitting short, high-frequency sound pulses at regular intervals. These propagate in the air at the velocity of sound. If they strike

an object, then they are reflected back as echo signals to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo. As the distance to an object is determined by measuring the time of flight and not by the intensity of the sound, ultrasonic sensors are excellent at suppressing background interference. Virtually all materials which reflect sound can be detected, regardless of their color. Even transparent materials or thin foils represent no problem for an ultrasonic sensor. [11]

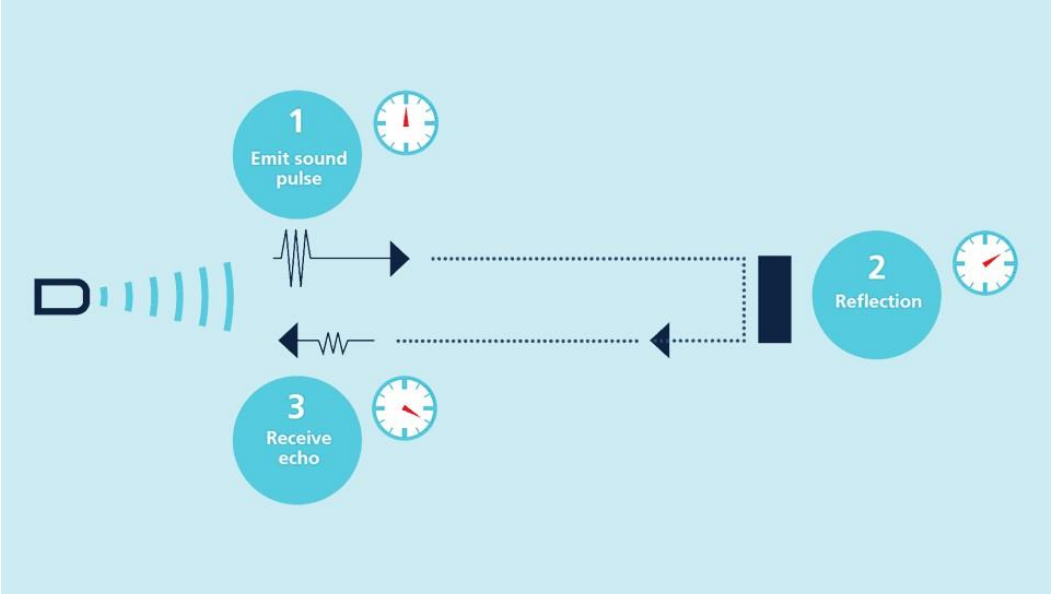


Fig 3.5: Ultrasonic Sensor Working

3.3.2 LICENSE PLATE RECOGNITION

License Plate Recognition uses the concepts of Digital image Processing and Optical Character Recognition to extract the registered vehicle number from the license plate. This procedure is divided into three steps:

3.3.2.1 PLATE DETECTION

The image is first converted into its grayscale equivalent and then into binary image. A certain threshold is decided and all gray level values above the threshold

are mapped to 1 and below it to 0. Thresholding an image converts it into one with black background with all high frequency components in white.

$Y = 0 ; GL < Threshold\ 1 ;$

$GL >= Threshold$

Contours are then drawn for all the objects in the binary image using openCV commands findContours() and drawContours(). A contour is a numpy array of coordinates(x,y) of all the points forming the boundary of images. findContours() returns a list of all the contours in the image;the retrieval mode and contour approximation method are set by the programmer, in this case the same being simple chain approximation(cv2.CHAIN_APPROX_SIMPLE). The list created using findContours() is give as argument to drawContours() which draws contours around all objects of the list. Possible characters are detected from the above image by comparing the dimensions,area and aspect ratio of the bounding rectangle of possible character to predefined pixel dimensions, area and aspect ratio.

```
if (possibleChar.intBoundingRectArea > MIN_PIXEL_AREA and  possibleChar.intBoundingRectWidth > MIN_PIXEL_WIDTH and
possibleChar.intBoundingRectHeight > MIN_PIXEL_HEIGHT and   MIN_ASPECT_RATIO < possibleChar.fltAspectRatio and
possibleChar.fltAspectRatio < MAX_ASPECT_RATIO):
    return True
else:
    return False
```

If the above condition are satisfied it adds the characters to a list called listofPossibleChars. Each possible character is compared with the listofPossibleCharacter to find matching Characters. Distance and angle between characters, change in area, height and width is calculated to do the same. Based on the length on the listofMatchingCharacters found, a new list, ListofListsofMatchingCharacters, of all matching character sequences which could

be possible plates is created. From this list, possible plates are extracted and cropped.

Fig 3.6: Plate Detection

3.3.2.2 CHARACTER DETECTION IN PLATES

All possible plates are resized for the purpose of clarity and preprocessed. The image is inverted, converted to gray scale and then into binary by thresholding. Possible characters are detected as in the case of plate detection following the exact same steps.

After generating the ListofListsofMatchingCharactersinPlate, the overlapping characters are removed. The longest length of matching chars is chosen to be the actual licence plate. The characters are recognized using KNN (K Nearest Neighbor) classification model.

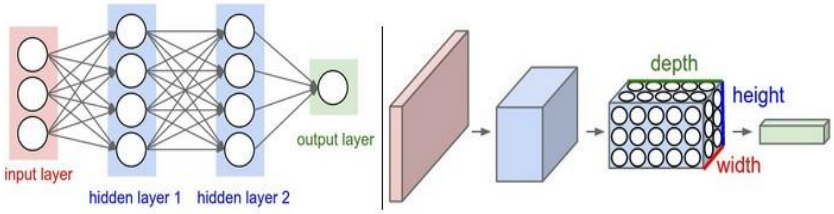
Fig 3.7: Character Segmentation

3.4 VEHICLE CLASSIFICATION

Vehicle classification is performed using Convolutional Neural Networks(CNN or ConvNets) which are a special kind of multi-layer neural networks. Convolutional Neural Networks are designed to recognize visual patterns directly from pixel images with minimal preprocessing [24] . We have used pretrained Residual Network deep learning architecture and then fine-tuned the network to fit our dataset of images of Cars and Bikes.

3.4.1 CONVOLUTION NEURAL NETWORK

Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture accordingly[2]. Unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. Moreover, the neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner[2]. Basically, a ConvNet is made up of Layers. Every Layer has a simple API: It transforms an input 3D volume to an output 3D volume with some differentiable function that may or may not have parameters[2].



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

Fig 3.8: Neural Network and Convolutional Neural Network

3.4.2 RESIDUAL NETWORK

Residual Network, popular as ResNet is the most groundbreaking work done in the Computer Vision/Deep Learning community in the last few years[1]. Developed by Microsoft, ResNet, is a residual network framework which makes training of deep neural networks easier by eliminating the problem of vanishing gradient and performance degradation.

As the more and more layers are added to neural network i.e. as it goes deeper, it faces a vanishing gradient problem. As the gradient back propagates to earlier layers, repeated multiplication may make the gradient infinitesimally small. As a result, the accuracy of the network gets saturated and then degrades rapidly. ResNet with 50 layers has been used for vehicle classification. It requires images with input size of 229x229x3. The linear and non-linear transformations are applied on the image as it passes through the subsequent layers of the ConvNet. Since ResNet50 is originally pre trained on ImageNet dataset comprising of 1.2 million images with 1000 categories, the last layers are stripped off to perform transfer learning by fine-tuning the architecture.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig 3.9: Residual Network Layers

3.5 DATABASE MANAGEMENT

Csv (Comma Separated Values) file is used to store the following information:

1. License Plate No.
2. Vehicle Type
3. Entry Time
4. Balance

As soon as a new number plate is detected, all the above information is stored in the database of the Parking system and whenever an existing number plate is recognized, the fare is calculated on the basis of exit time and all the information regarding the vehicle is removed from the database. This is implemented using Pandas in Python.

After registering the above information in the database, the barricade gets opened so that the vehicle can enter the parking lot. Also, at the time of exit, the fare is calculated and barricade opens again for vehicle to exit.

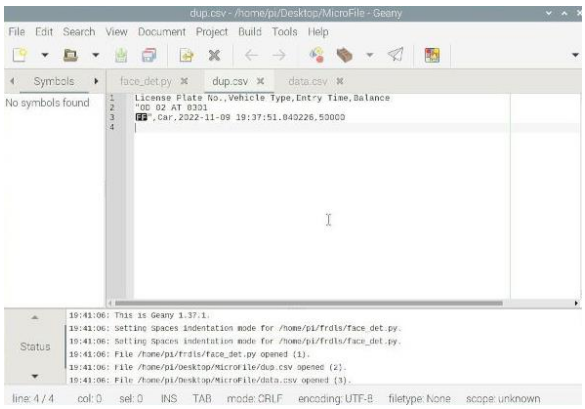


Fig 3.10: Database Schema

4. CODE AND OUTPUT

4.1. CODE for PLATE.py

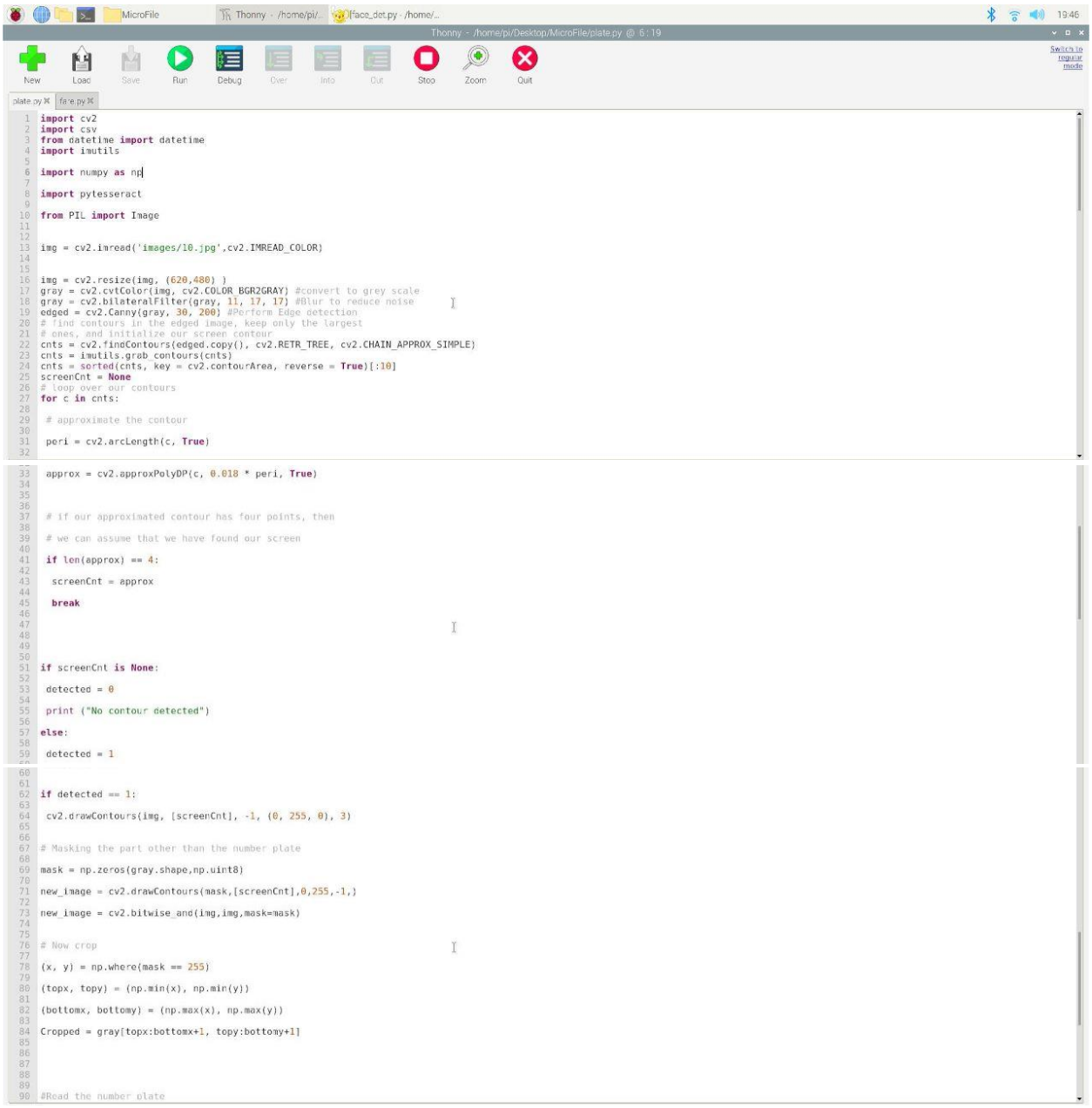
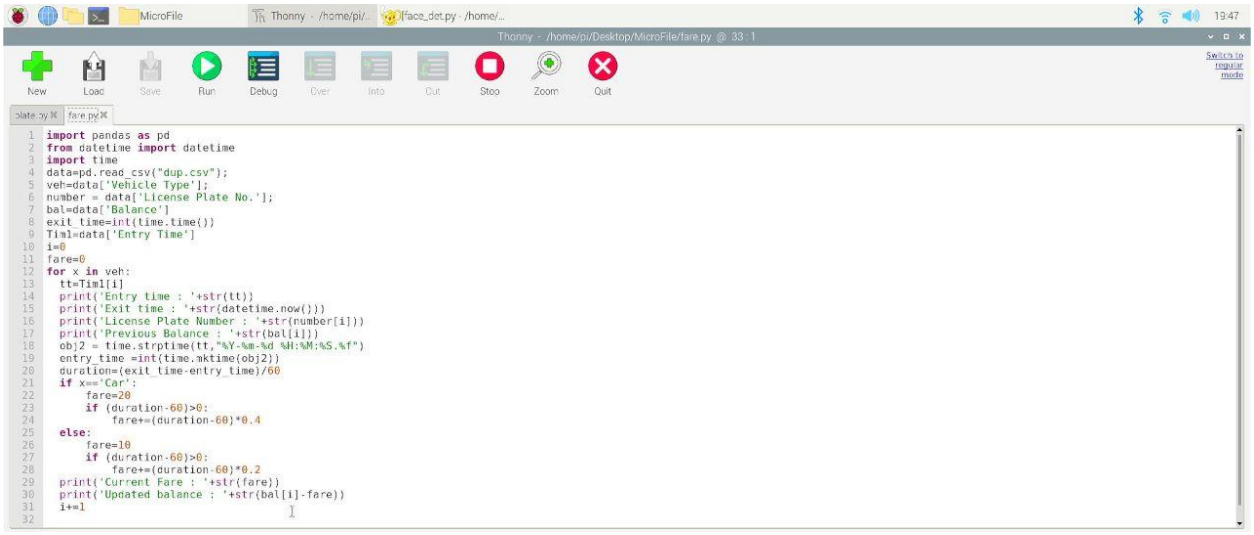


Fig 4.1: Code for PLATE.py

4.2 CODE for FARE.py



```
import pandas as pd
from datetime import datetime
import time
data=pd.read_csv("dup.csv");
veh=data['Vehicle Type'];
number = data['License Plate No.'];
bal=data['Balance']
exit_time=int(time.time())
Tim1=data['Entry Time']
i=0
fare=0
for x in veh:
  tt=Tim1[i]
  print('Entry time : '+str(tt))
  print('Exit time : '+str(datetime.now()))
  print('License Plate Number : '+str(number[i]))
  print('Previous Balance : '+str(bal[i]))
  obj2 = time.strptime(tt,"%Y-%m-%d %H:%M:%S.%f")
  entry_time =int(time.mktime(obj2))
  duration=(exit_time-entry_time)/60
  if x=='Car':
      fare=20
      if (duration-60)>0:
          fare+=(duration-60)*0.4
  else:
      fare=10
      if (duration-60)>0:
          fare+=(duration-60)*0.2
  print('Current Fare : '+str(fare))
  print('Updated balance : '+str(bal[i]-fare))
  i+=1
```

Fig 4.2: Code for FARE.py

4.3 OUTPUT for PLATE.py

Sample Output 1

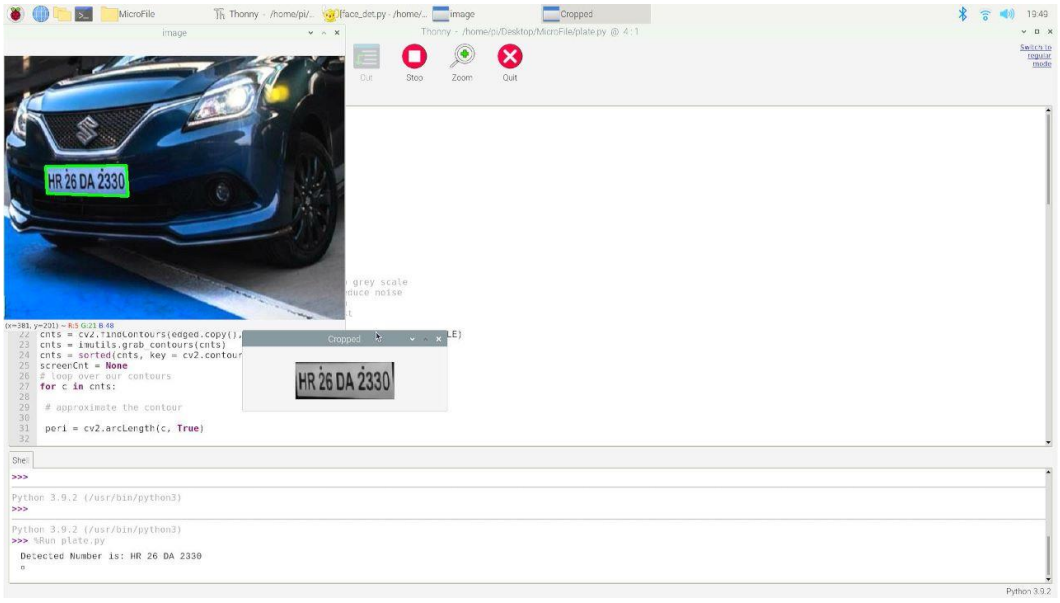


Fig 4.3: Sample Output 1 for PLATE.py

Sample Output 2

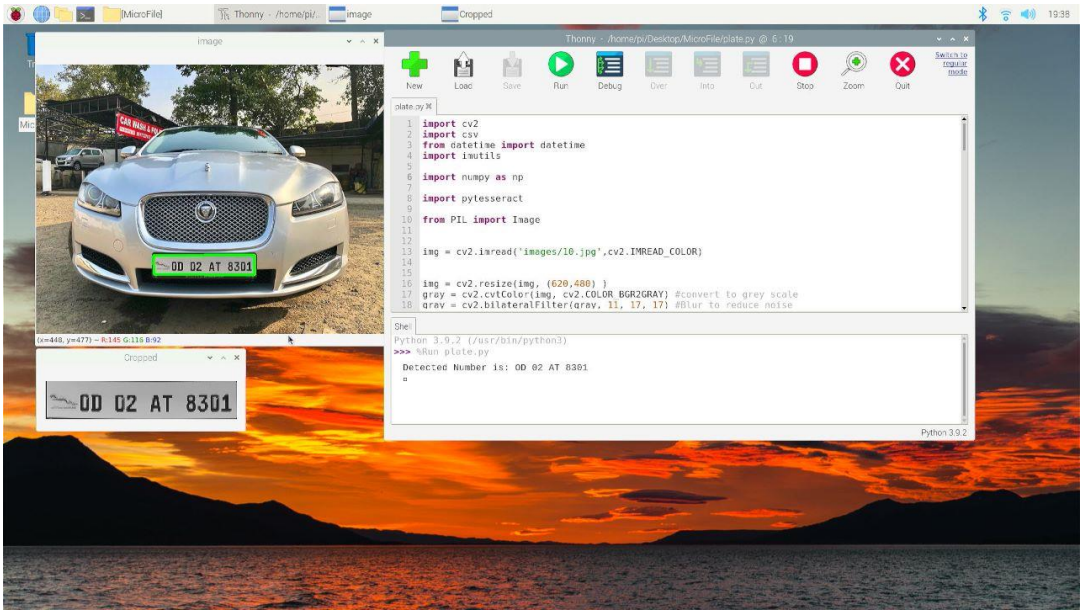


Fig 4.4: Sample Output 2 for PLATE.py

4.4 OUTPUT for FARE.py

Sample Output 1

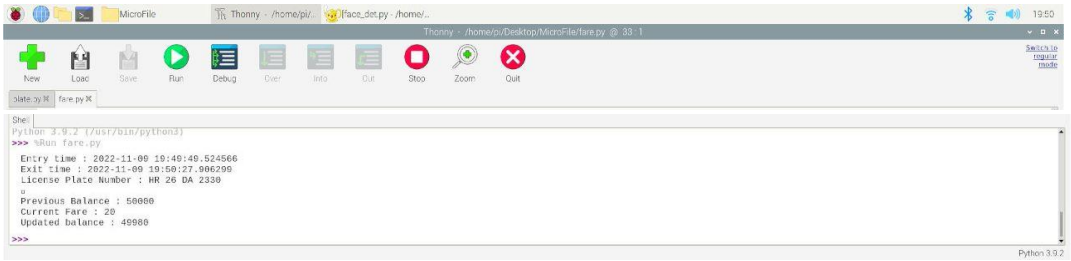


Fig 4.5: Sample Output 1 for FARE.py

Sample Output 2

Fig 4.6: Sample Output 2 for FARE.py

5. CONCLUSION AND FUTURE POTENTIAL

5.1 CONCLUSION

An efficient and time saving parking system can be created with an Automatic Fare Generation System. As the world treads the path of Artificial Intelligence, renovating the systems to make them as human management independent as possible has utmost importance. An automatic system thus not only pave way for a hassle-free fare generation but increases the overall efficiency thus saving time and money of the users.

5.2 FUTURE POTENTIAL

The project can be extended to develop an app to convert traditional parking spaces into a smart parking lot. The app functionality will include:

1. Each user can have an account with single or multiple license plates registered
2. The user will get all information regarding payment and receipts available in his account.
3. The user will be able to identify the number of parking slots left in a particular lot, to help the users to find space for parking their car without the assistance of the human workforce.
4. Machine learning can further be deployed to give discount to the users on the basis of their usage of that particular parking lot.
5. By linking Aadhar/syncing account info fare can be automatically deducted.

6. REFERENCES

[1] Andrej Karpathy.”CS231n Convolutional Neural Networks for Visual Recognition”. Internet:http://cs231n.github.io/ , April 13,2018 [April 25,2018].

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition. In CVPR,2016.

[3] Mart Abadi, Ashish Agarwal et al. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.” Internet: https://www.tensorflow.org/ , 2015 [March 1,2018].

[4] Jason Brownie. Internet: www.machinelearningmastery.com/introduction-python-deep-learning-library-keras/, May 10,2016 [March 10,2018].

[5] Itseez. “The OpenCV Reference Manual.” Internet: https://opencv.org/ ,April 23,2018 [March 10,2018].

[6] Travis E, Oliphant. “A guide to NumPy”. USA: Trelgol Publishing, (2006).

[7] Pedregosa et al.”Scikit-learn: Machine Learning in Python.”The Journal of Machine Learning Research,vol. 12,pp. 2825--2830,2011.

[8] Wes McKinney. “Data Structures for Statistical Computing in Python”. Proceedings of the 9th Python in Science Conference,pp. 51 - 56, 2010.

[9] D. K. Basu, M. Nasipuri, S. Basu and S. Saha, “ License Plate Localization from Vehicle Images : An Edge Based Multi – Stage Approach ”, International Journal of Recent Trends in Engineering, Vol. 1 – No. 1, pp. 284 – 288, May 2009.

[10] S. G. Patel, “ Vehicle License Plate Recognition using Morphology and Neural Network ”, International Journal on Cybernetics and Informatics, Vol. 2 – No. 1, pp. 1 - 7, February 2013.

[11] Hsieh, J.W., Chen, L.C., Chen, D.Y. et al.: Vehicle make and model recognition using symmetrical SURF. In: 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 472–477 (2013)

[12] Dong, Z., Wu, Y., Pei, M. et al.: Vehicle type classification using a semisupervised convolutional neural network. In: IEEE Transactions on Intelligent Transportation Systems, pp. 2247–2256 (2015)

[13] Lai, A.H. Fung, G.S., Yung, N.H.: Vehicle type classification from visual-based dimension estimation. In: Proceedings of the IEEE Intelligent Transportation Systems Conference, pp. 201–206 (2001)
[14] Heechul Jung, Min-Kook Choi, Jihun Jung et al.: ResNet-based Vehicle Classification and Localization in Traffic Surveillance Systems. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (2017)

[15] Adam Carlson. “HC-SR04 Datasheet”. Internet: https://www.electroschematics.com/8902/hc-sr04-datasheet/ , Oct. 22,2017 [April 24,2018].

[16] Russell Barnes. “Raspberry Pi 3 is out now!”. Internet: https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/ , March 2 2018 [April 1 2018].

[17] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541-551, Winter 1989.

[18] Y. LeCun. Generalization and network design strategies. Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto, 1989.

[19] LeCun, Yann. Internet:http://yann.lecun.com/exdb/lenet/"LeNet-5, convolutional neural networks", Nov. 16, 2013 [April 24,2018] .

[20] Marcus D. Bloice.”Augmentor: Python library for Data Augmentation”. Internet: https://augmentor.readthedocs.io/en/master/ , March 29,2018 [April 20,2018].

[21] John D. Hunte. “Matplotlib: A 2D Graphics Environment”. Internet: https://matplotlib.org/, April 19,2018 [April 25,2018]

[22] Dmitriy 'DZ' Zaporozhets et al.. Internet: https://gitlab.com/, April 23,2018 [Feb. 1,2018].

[23] LISA Lab,University of Montreal. Internet: http://deeplearning.net/software/theano/, Nov. 21,2017 [March 1,2018].

Automatic Fare Generation System For Parking Lots

*

Babi Nunnaguppala[1] Mohd Ayan Khan[2] Jayanth T[3] A Aryan Rajesh[4]
Microprocessor And Interfacing, BTech Computer Science Engineering Core
Vellore Institute of Technology, Vellore, Tamil Nadu
[1]nunnaguppala.babi2020@vitstudent.ac.in [2]mohdayan.khan2020@vitstudent.ac.in
[3]jayanth.t2020@vitstudent.ac.in [4]aryanrajesh.a2020@vitstudent.ac.in

***Abstract*—For parking lots that accommodate a large number of cars daily, our project intends to create an effective and automatic fare generation method. The deployment of human resources is necessary to manage the parking of the visiting cars, estimate the fares, and collect payment at the point of exit in the streets and locations that typically bustle with visitors every day. This project will automate the process, eliminating the need for labour, avoiding deliberate fare overcharging for personal gain, adding an extra layer of security via image and database records, and increasing efficiency by lowering human errors in the process.To put this into practise, the system created must be able to determine the type of vehicle and its identifier, the licence plate. For the same purpose, this project makes use of computer vision and machine learning techniques.**

***Index Terms*—Fare Generation, Machine Learning, Computer Vision, Automobile, Parking System, Deep Learning, Neural Networks**

I. INTRODUCTION

The traditional method of collecting fares often entails distributing slips that detail the entry, their collection, and the chosen fare at the time of leaving. This process always takes longer at entry because there are fewer workers and more visitors' cars. Regular users find it to be quite a task to keep the slips that are thereafter given safe and accessible. Furthermore, it frequently happens that employees overcharge customers in order to benefit themselves by generating price slips with false information that violates the established regulations. This poses an additional security risk to the automobiles, as does the lack of any other documentation of the parked cars other than the paper slips.All of these discrepancies not only obstruct the development of a smart parking system but also make the procedure difficult for users. Renovation of the systems to make them as human management independent as feasible is of vital relevance as the world moves toward artificial intelligence.Thus, an autonomous system not only paves the way for hassle-free fare creation but also boosts overall efficiency, saving users' time and money.

II. LITERATURE REVIEW

Computer Vision is the core of Artificial Intelligence which allows computers to take intelligent decisions based on the visual details around them. It includes various takes such as object detection and recognition,image classification and segmentation. Image classification is the task of categorizing images into one of several predefined classes, is a fundamental problem in computer vision. Vehicle classification done using a visual-based dimension estimation method,extracts moving vehicles from traffic image sequences and fits them with a simple deformable vehicle model. A set of coordination mapping functions are derived from a calibrated camera model and relying on a shadow removal method, vehicle's width, length and height are estimated [1].License Plate Detection systems have been employed widely over the world for real time tracking of vehicles, curb criminal activities etc. Various techniques have been developed using different softwares like LabView, MATLAB, OpenCV using different approaches like edge based, neural networks, sliding window techniques amongst others were used. While the initial edge based techniques didn't produce accurate results, with no license plate detection or incorrect detection [2], the techniques developed later on which used median-filters and neural networks, though gave a better accuracy, but could only be used to detect english characters. [3]

III. TECHNOLOGY USED

A. HARDWARE REQUIREMENTS

1) Raspberry Pi 3: Raspberry Pi is a series of small singleboard computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. It is an SoC (System on chip), which integrates all the components of a computer or other electronic systems on a single chip like a GPU, Wifi-Module etc.Raspberry Pi 3 Specifications are:

- SoC: Broadcom BCM2837
- CPU: 4× ARM Cortex-A53, 1.2GHz
- Broadcom VideoCore IV
- 1GB LPDDR2 (900 MHz)

Fig1: Raspberry Pi 3

2) UltraSonic Sensor (HC-SR04): The human ear can hear sound frequency around 20HZ 20KHZ, and ultrasonic is the sound wave beyond the human ability of 20KHZ.HC-SR04 is an ultrasonic ranging module that provides 2 cm to 400 cm non-contact measurement function. The ranging accuracy can reach to 3mm and effectual angle is ¡ 15°. It can be powered from a 5V power supply. [4]

- Working Voltage : DC 5V
- Working Frequency : 40Hz

Fig2: Ultrasonic Sensor

3) DC MOTOR: A DC motor in simple words is a device that converts electrical energy (direct current system) into mechanical energy. The direction of current is given by the Fleming's Left Hand Rule.

Fig3: DC Motor

B. SOFTWARE REQUIREMENTS

1) TENSORFLOW: TensorFlow is an open source software library for high performance numerical computation, originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning. Owing to its flexible architecture,it allows deployment across different platforms ranging from desktops to mobile and EDGE devices.
[5]

2) KERAS: Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano, two of the top numerical platforms in that provide the basis for Deep Learning research and development.

3) Python 3.6.4: Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability.Important library of python used here is OpenCV which is a leading open source library for computer vision, image processing and machine learning, and now features GPU acceleration for real-time operation.

4) Anaconda: Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications, that aims to simplify package management and deployment.The packages used from conda distribution as part of this project are :

- Scikit-Learn
- NumPy
- Pandas
- Matplotlib
- Augmentor
- Theano

IV. METHODOLOGY

A. OBJECT DETECTION

When a vehicle approaches the entry barrier, its presence is detected using an ultrasonic sensor. The ultrasonic sensor intimates the processor of the presence of the vehicle which then turns on the camera.The camera captures the image of the vehicle and that image is then used to recognize license plate detection and vehicle classification. Ultrasonic Sensor is the most commonly used sensor to detect a moving target and estimate the the approximate distances it is present at.Ultrasonic sensors works on the principle of emitting short, high-frequency sound pulses at regular intervals. These propagate in the air at the velocity of sound. If they strike an object, then they are reflected back as echo signals to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo.

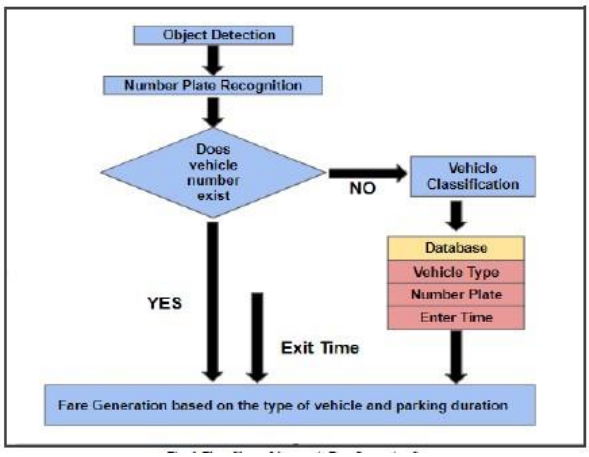


Fig4: Flow Chart of Automatic Fare Generation System

As the distance to an object is determined by measuring the time of flight and not by the intensity of the sound, ultrasonic sensors are excellent at suppressing background interference. Virtually all materials which reflect sound can be detected, regardless of their colour. Even transparent materials or thin foils represent no problem for an ultrasonic sensor. [6]

B. LICENSE PLATE RECOGNITION

License Plate Recognition uses the concepts of Digital image Processing and Optical Character Recognition to extract the registered vehicle number from the license plate. This procedure is divided into three steps:

- Plate Detection
- Character Detection in Plates

1) Plate Detection: The image is first converted into its grayscale equivalent and then into binary image. A certain threshold is decided and all gray level values above the threshold are mapped to 1 and below it to 0.Thresholding an image converts it into one with black background with all high frequency components in white.Y=0 ;GL less than Threshold and 1 ; GL greater than equal to Threshold

Fig5: Cropped number plate from original car image

Contours are then drawn for all the objects in the binary image using openCV commands findContours() and drawContours(). A contour is a numpy array of coordinates(x,y) of all the points forming the boundary of images. findContours() returns a list of all the contours in the image;the retrieval mode and contour approximation method are set by the programmer, in this case the same being simple chain approximation(cv2.CHAINAPPROXSIMPLE).The list created using findContours() is give as argument to drawContours() which draws contours around all objects of the list. Possible characters are detected from the above image by comparing the dimensions,area and aspect ratio of the bounding rectangle of possible character to predefined pixel dimensions, area and aspect ratio.

```
if (possibleChar.intBoundingRectArea > MIN_PIXEL_AREA and possibleChar.intBoundingRectWidth > MIN_PIXEL_WIDTH and
possibleChar.intBoundingRectHeight > MIN_PIXEL_HEIGHT and  MIN_ASPECT_RATIO < possibleChar.fltAspectRatio and
possibleChar.fltAspectRatio < MAX_ASPECT_RATIO):

    return True

else:

    return False
```

Fig6 : Code snippet

If the above condition are satisfied it adds the characters to a list called listofPossibleChars. Each possible character is compared with the listofPossibleCharacter to find matching Characters. Distance and angle between characters, change in area, height and width is calculated to do the same.Based on the length on the listofMatchingCharacters found, a new list, ListofListsofMatchingCharacters, of all matching character sequences which could be possible plates is created.From this list, possible plates are extracted and cropped.

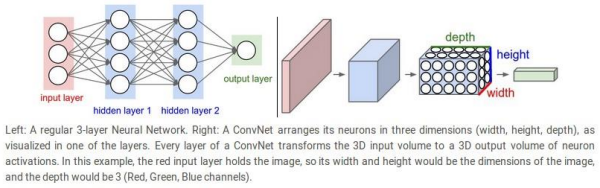
2) Character Detection in Plates: All possible plates are resized for the purpose of clarity and preprocessed. The image is inverted, converted to gray scale and then into binary by thresholding. Possible characters are detected as in the case of plate detection following the exact same steps.After generating the ListofListsofMatchingCharactersinPlate, the overlapping characters are removed. The longest length of matching chars is chosen to be the actual licence plate. The characters are recognized using KNN (K Nearest Neighbor) classification model.

Fig7 : Number plate extraction

3) VEHICLE CLASSIFICATION: Vehicle classification is performed using Convolutional Neural Networks(CNN or ConvNets) which are a special kind of multi-layer neural networks. Convolutional Neural Networks are designed to recognize visual patterns directly from pixel images with minimal preprocessing.We have used pretrained Residual Network deep learning architecture and then fine-tuned the network to fit our dataset of images of Cars and Bikes.

- Convolutional Neural Network : Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture accordingly.

[7] Unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. Moreover, the neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner [7] Basically, a ConvNet is made up of Layers. Every Layer has a simple API: It transforms an input 3D volume to an output 3D volume with some differentiable function that may or may not have parameters [7]



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

Fig8 : Neural Network and Convolutional Neural Network

- Residual Network : Residual Network, popular as ResNet is the most groundbreaking work done in the Computer Vision/Deep Learning community in the last few years [8].Developed by Microsoft, ResNet, is a residual network framework which makes training of deep neural networks easier by eliminating the problem of vanishing gradient and performance degradation.
- Database Management : Csv (Comma Separated Values) file is used to store Number Plate of the vehicle ,Type of Vehicle (Car or Bike) ,Entry Time of the vehicle.This is implemented using Pandas in Python.

V. IMPLEMENTATION

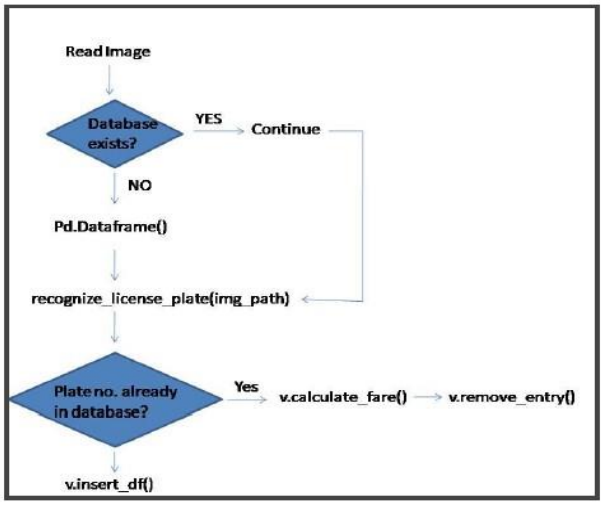


Fig9 : Code Flow

As soon as ultrasonic sensor detects the object,camera clicks the image and that image is passed as argument to the 'main' function . The main function opens the database of the vehicles or creates one if it doesn't exist already.The license plate is recognized and stored in variable licenseplate .An object of class FareCalculator is created. Image, dataframe and license plate are the members of class which get instantiated upon making of the object.The record of license plate just recognized is checked in the database via the method 'checkexistence()'.

- If record exists, Fare is calculated using 'calculatefare()' and printed on the screen.The record is removed via 'removeentry()' method.
- Else if record does not exist, Vehicle type is identified using the method 'testtype()'.The vehicle information like license plate number,vehicle type and entry time are inserted in the database by 'insertdf()' method.

VI. RESULTS

Vehicle classification trained with 672 photos and obtained training accuracy of 94% and validation accuracy of about 92%. By adjusting the hyperparameters more precisely and lowering the regularisation, this accuracy can be raised even more.

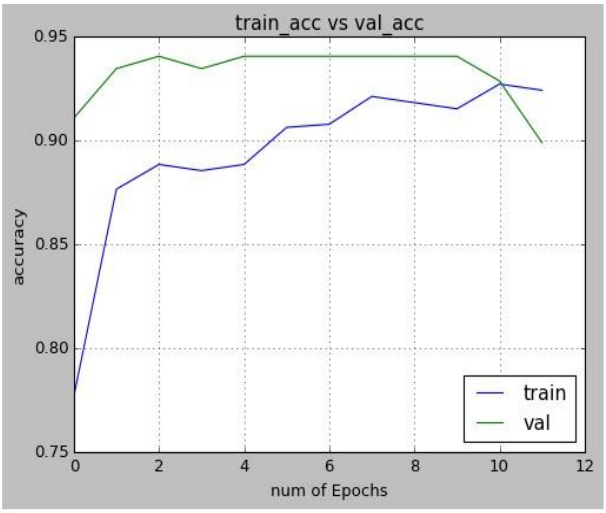


Fig10 : Training v/s Validation accuracy for vehicle classification

The majority of the time, License Plate Recognition can recognise number plates on cars, but it can be difficult to reliably identify plates on bikes. Instead of utilising K Nearest Neighbor, which has a high time and space complexity and produces results that are somewhat erroneous, SVM (Support Vector Machine) classifier can produce results that are more accurate.

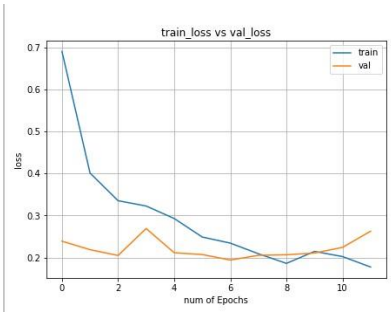


Fig11 : Training v/s Validation loss for vehicle classification The whole process takes around 25 seconds on an Intel Core i5-4210U 1.70GHz CPU which can be further optimised by using Graphical Processing Unit.

VII. CONCLUSION

An automatic fare generation system can be used to build a parking system that is effective and quick. However, the system can be enhanced by using a licence plate detection system with an SVM model rather than a KNN to get better accuracy.The project can be extended to develop an app to convert traditional parking spaces into a smart parking lot.

The app functionality will include the following

Each user can have an account with single or multiple license plates registered. The user will get all information regarding payment and receipts available in his account. Machine learning can further be deployed to give discount to the users on the basis of their usage of that particular parking lot.The functionality of the app will consist of the following : Each user is allowed to register one or more licence plates on their account. The user will receive all payment and receiptrelated information that is available in his account. Machine learning can also be used to provide discounts to customers based on how often they utilise a certain parking lot. Fare can be automatically deducted by syncing account information and linking Aadhar.

REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[2] A. Lai, G. Fung, and N. Yung, "Vehicle type classification from visualbased dimension estimation," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pp. 201–206, 2001.

[3] H. Jung, M.-K. Choi, J. Jung, J.-H. Lee, S. Kwon, and W. Young Jung, "Resnet-based vehicle classification and localization in traffic surveillance systems," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 61–67, 2017.

[4] Y. LeCun *et al.*, "Lenet-5, convolutional neural networks," *URL: http://yann. lecun. com/exdb/lenet*, vol. 20, no. 5, p. 14, 2015.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[6] J.-W. Hsieh, L.-C. Chen, D.-Y. Chen, and S.-C. Cheng, "Vehicle make and model recognition using symmetrical surf," in *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 472–477, IEEE, 2013.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition. cvpr. 2016," *arXiv preprint arXiv:1512.03385*, 2016.

[8] A. Karpathy, "Stanford university cs231n: Convolutional neural networks for visual recognition," *URL: http://cs231n. stanford. edu/syllabus. html*, pp. 13–35, 2018.

AUTOMATIC FARE GENERATION SYSTEM FOR PARKING LOTS

USING RASPBERRY PI

Team Members

MOHD AYAN KHAN

20BCE0644

BABI NUNNAGUPPALA

20BCE2851

A ARYAN RAJESH

20BCE0718

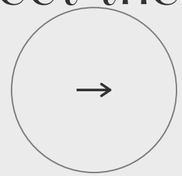
JAYANTH T

20BCE0967

Abstract

In India, owing to a large population, car sales growth hikes every year. This drives the need for strategizing methods for efficient systems that simplifies the management of these automobiles. One such domain are the Parking Lots at various locations across the country. Our project aims to provide an efficient and automatic fare generation system for parking lots that cater to multitudes of cars on a daily basis.

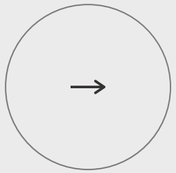
The streets and areas that usually bustle with visitors every day require deployment of human sources to manage the parking of the visiting cars, calculate the estimated fares and collect the payment at the time of exit.



Abstract

This project will automate this procedure, thus eliminating the need to employ workers, preventing intentional over charging of fares for personal interests, providing an additional security measure in the form of image and database records, and improving the efficiency by reducing the human errors in the process.

To implement this, the type of the vehicle and its identification, the license plate, are necessary to be determined by the system designed. This project engages Computer Vision and Machine Learning tools for the same.



Introduction

The traditional method of collecting fares often entails distributing slips that detail the entry, their collection, and the chosen fare at the time of leaving. This process always takes longer at entry because there are fewer workers and more visitors' cars. Regular users find it to be quite a task to keep the slips that are thereafter given safe and accessible. Furthermore, it frequently happens that employees overcharge customers in order to benefit themselves by generating price slips with false information that violates the established regulations.

This poses an additional security risk to the automobiles, as does the lack of any other documentation of the parked cars other than the paper slips.All of these discrepancies not only obstruct the development of a smart parking system but also make the procedure difficult for users. Renovation of the systems to make them as human management independent as feasible is of vital relevance as the world moves toward artificial intelligence. Thus, an autonomous system not only paves the way for hassle–free fare creation but also boosts overall efficiency, saving users' time and money

Literature Survey

Paper Title	Year	Technology Used
Comparison of Vehicle Detection Techniques applied to IP Camera video Feeds for use in Intelligent transport Systems	ELSEVIER, 2019	YOLO, RetinaNet, Regional Based CNN
Monitoring Multimodal Travel Environment Using Automated Fare Collection Data: Data Processing and Reliability Analysis	Springer, 2019	Link Based Algorithm & Path Based Algorithm in Automatic Fare Collection(AFC)
Public Traffic Congestion Estimation Using an Artificial Neural Network	MDPI, 2020	Self Organizing MAPSOM (Artificial Neural Network)

Literature Survey

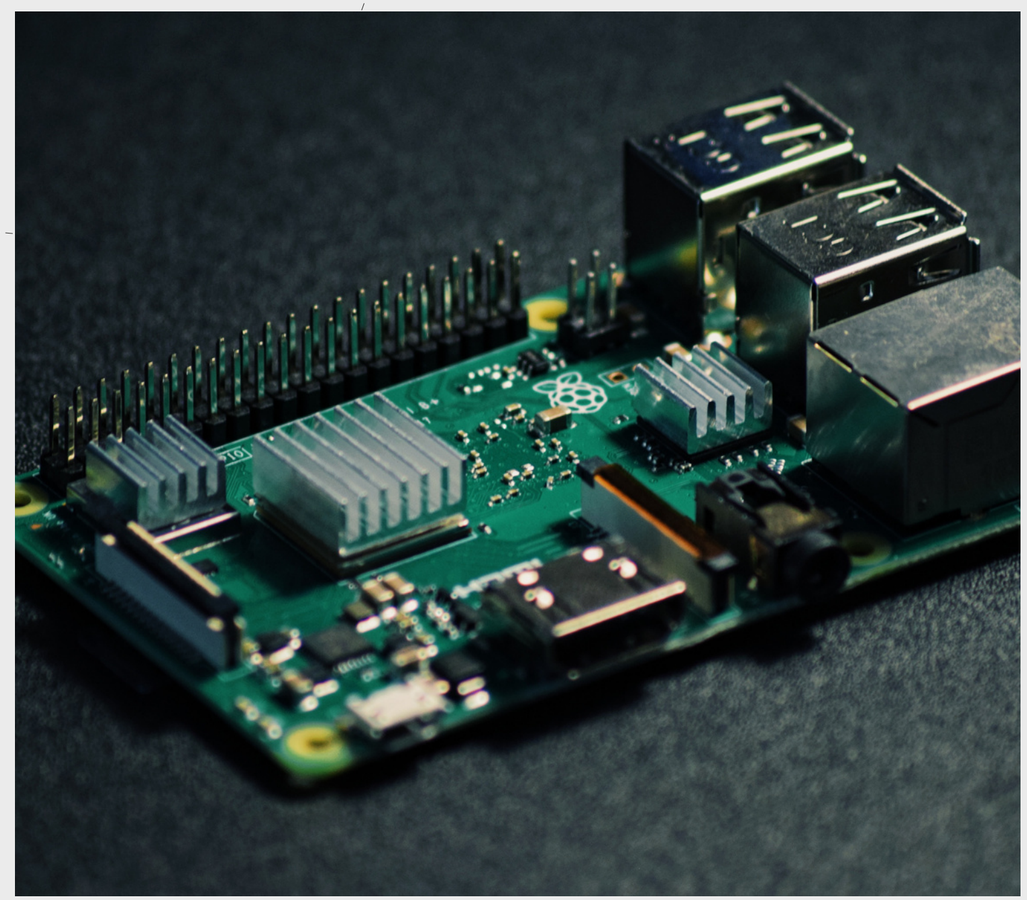
Paper Title	Year	Technology Used
Intelligent System for Vehicles Number Plate Detection and Recognition Using Convolutional Neural Networks	MDPI, 2021	Image Pre-processing, CNN
Vehicle Number Plate Detection and Recognition Techniques: A Review	The ASTE conference-organized by Threads, 2021	Image Pre-processing, CNN
Inferring flow patterns of subway passengers using a dynamic logistics model with automated fare collection and automated vehicle location data	SSRN, 2022	Dynamic Logistic Model,EM Algorithm

Gaps

License Plate Detection systems have been employed widely over the world for real time tracking of vehicles, curb criminal activities etc. Various techniques have been developed using different software's like LabView, MATLAB, OpenCV using different approaches like edge based, neural networks, sliding window techniques amongst others were used. While the initial edge based techniques didn't produce accurate results, with no license plate detection or incorrect detection, the techniques developed later on which used median–filters and neural networks, though gave a better accuracy, but could only be used to detect English characters.

Hardware Components

Raspberry Pi 3



Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. It is an SoC (System on chip), which integrates all the components of a computer or other electronic systems on a single chip like a GPU, Wifi-Module etc.

Hardware Components

Ultrasonic Sensor (HC–SR04)

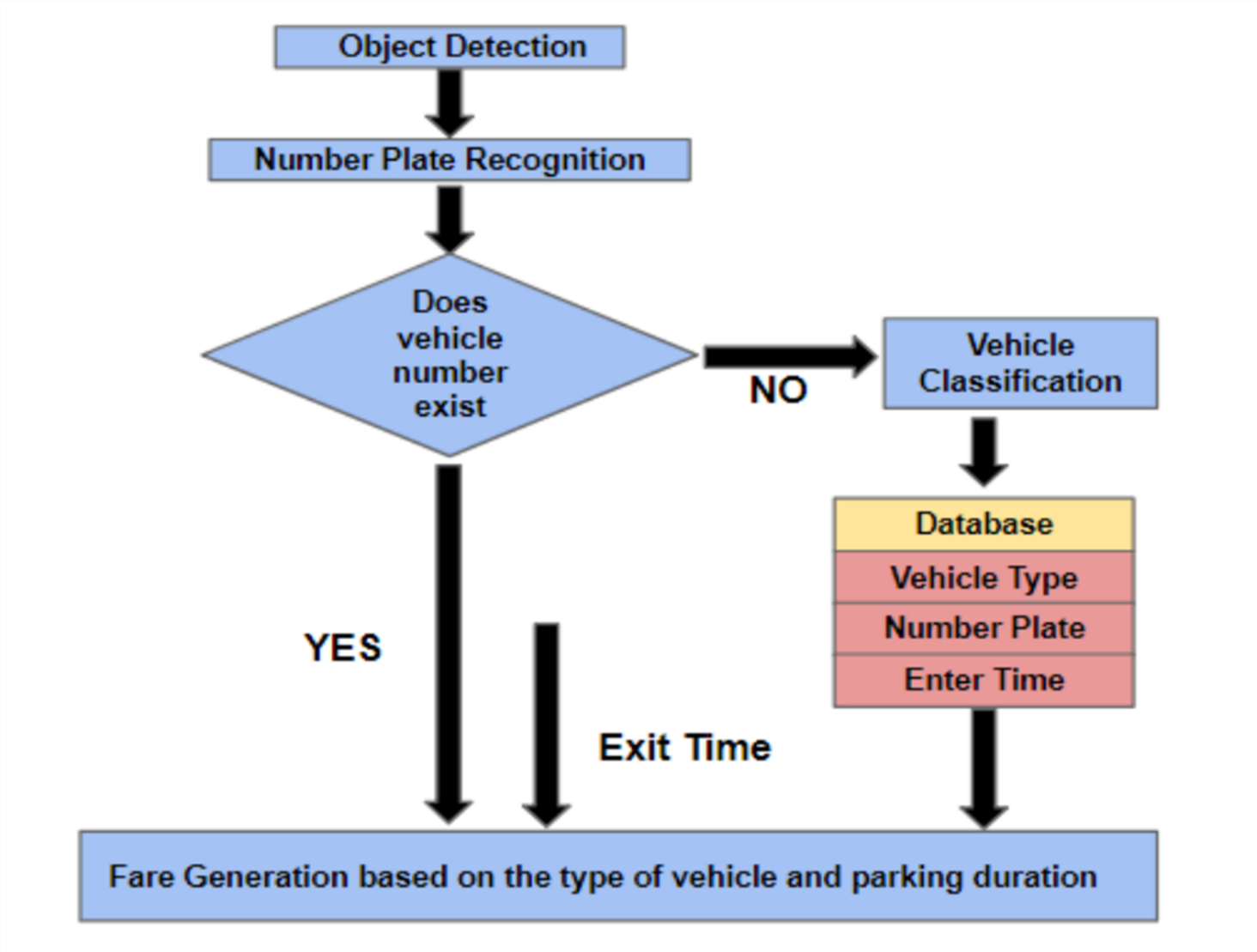
The human ear can hear sound frequency around 20HZ ~ 20KHZ, and ultrasonic is the sound wave beyond the human ability of 20KHZ.[21] HC-SR04 is an ultrasonic ranging module that provides 2 cm to 400 cm non-contact measurement function. The ranging accuracy can reach to 3mm and effectual angle is < 15°. It can be powered from a 5V power supply.

Hardware Components

DC Motor

A DC motor in simple words is a device that converts electrical energy (direct current system) into mechanical energy. The direction of current is given by the Fleming's Left Hand Rule.

Block Diagram



Software Components

TENSOR FLOW

TensorFlow is an open source software library for high performance numerical computation, originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning.

KERAS

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano, two of the top numerical platforms in that provide the basis for Deep Learning research and development

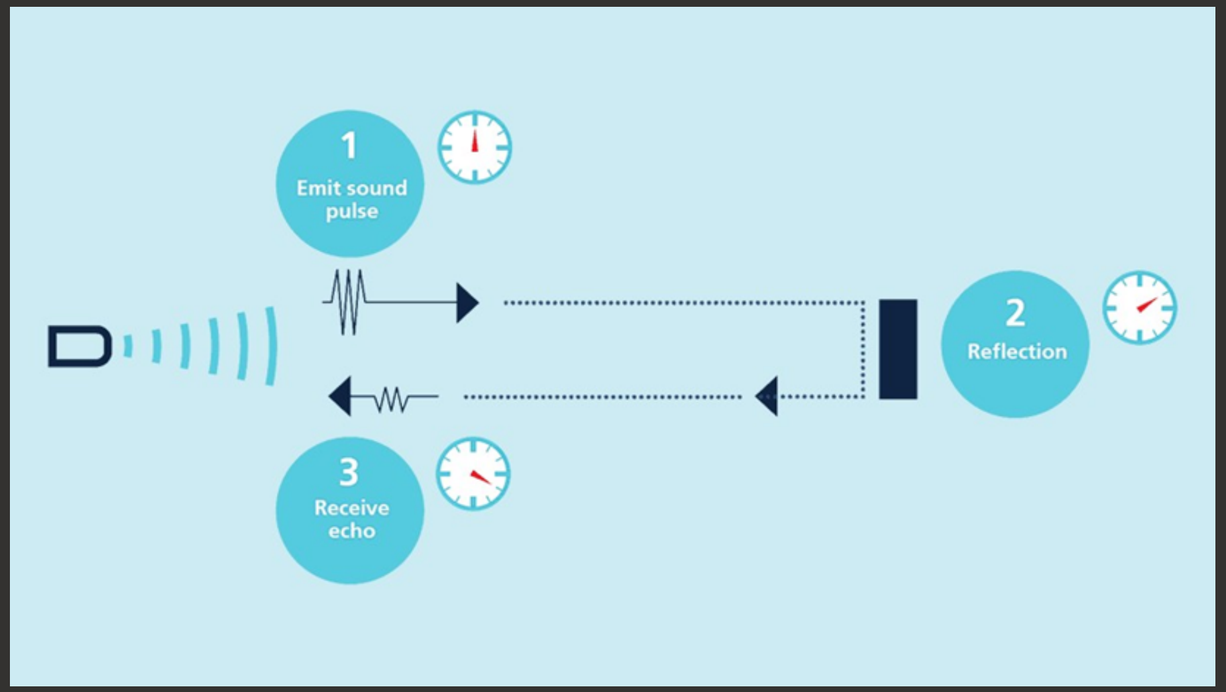
PYTHON 3.6.4

Python is an interpreted high-level programming language for general-purpose programming. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

ANACONDA

Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications, that aims to simplify package management and deployment.

Object Detection



Ultrasonic sensors works on the principle of emitting short, high-frequency sound pulses at regular intervals. These propagate in the air at the velocity of sound. If they strike an object, then they are reflected back as echo signals to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo. As the distance to an object is determined by measuring the time of flight and not by the intensity of the sound, ultrasonic sensors are excellent at suppressing background interference. Virtually all materials which reflect sound can be detected, regardless of their color. Even transparent materials or thin foils represent no problem for an ultrasonic sensor.

When a vehicle approaches the entry barrier, its presence is detected using an ultrasonic sensor. The ultrasonic sensor intimates the processor of the presence of the vehicle which then turns on the camera. The camera captures the image of the vehicle and that image is then used to recognize license plate detection and vehicle classification.

License Plate Detection

License Plate Recognition uses the concepts of Digital image Processing and Optical Character Recognition to extract the registered vehicle number from the license plate. This procedure is divided into two steps:

- Plate Detection
- Character Detection in Plates

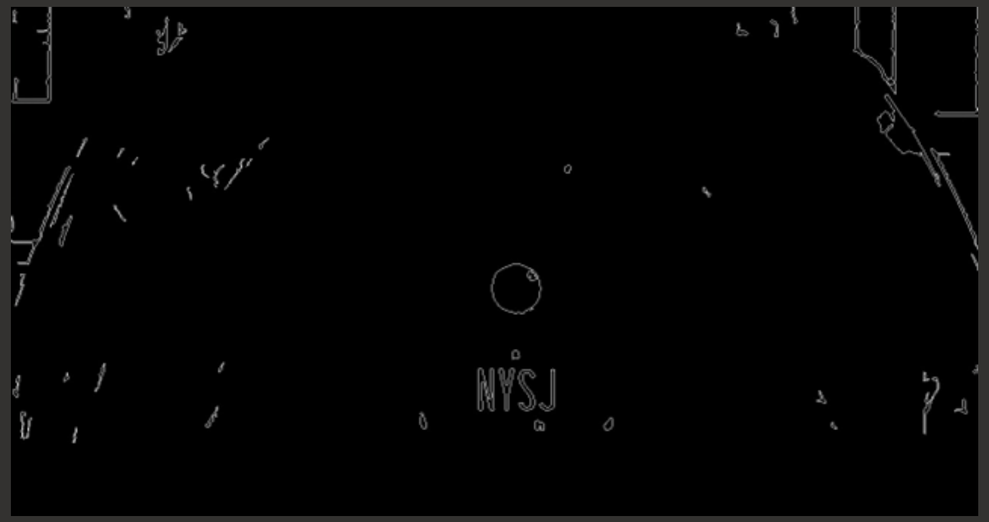
```python
import cv2
import imutils
import numpy as np
import pytesseract
from PIL import Image

img = cv2.imread('4.jpg',cv2.IMREAD_COLOR)
img = cv2.resize(img, (620,480) )
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #convert to grey scale
gray = cv2.bilateralFilter(gray, 11, 17, 17) #Blur to reduce noise
edged = cv2.Canny(gray, 30, 200) #Perform Edge detection

# find contours in the edged image, keep only the largest
# ones, and initialize our screen contour
cnts = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
cnts = sorted(cnts, key = cv2.contourArea, reverse = True)[:10]
screenCnt = None
# loop over our contours
for c in cnts:
    # approximate the contour
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.018 * peri, True)
    # if our approximated contour has four points, then
    # we can assume that we have found our screen
    if len(approx) == 4:
        screenCnt = approx
        break
if screenCnt is None:
    detected = 0
    print("No contour detected")
else:
    detected = 1
if detected == 1:
    cv2.drawContours(img, [screenCnt], -1, (0, 255, 0), 3)

# Masking the part other than the number plate
mask = np.zeros(gray.shape,np.uint8)
new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)
new_image = cv2.bitwise_and(img,img,mask=mask)

# Now crop
(x, y) = np.where(mask == 255)
(topx, topy) = (np.min(x), np.min(y))
(bottomx, bottomy) = (np.max(x), np.max(y))
Cropped = gray[topx:bottomx+1, topy:bottomy+1]

#Read the number plate
text = pytesseract.image_to_string(Cropped, config='--psm 11')
print("Detected Number is:",text)
cv2.imshow('image',img)
cv2.imshow('Cropped',Cropped)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Image Of Car Captured From Camera

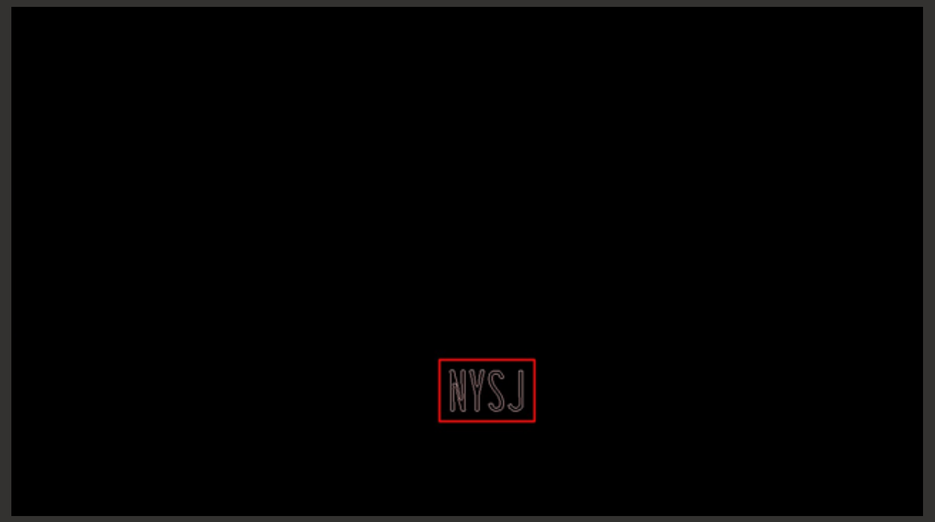
Image Converted to Binary

RGB Image Converted to Grayscale

Contours Drawn Around Connected Regions



Possible Characters Found



Extracted Number Plate

Number Plate Identified

Cropped Number Plate from Original RGB image

Character Detection in Plates

All possible plates are resized for the purpose of clarity and preprocessed. The image is inverted, converted to gray scale and then into binary by thresholding. Possible characters are detected as in the case of plate detection following the exact same steps.

After generating the ListofListsofMatchingCharactersinPlate, the overlapping characters are removed. The longest length of matching chars is chosen to be the actual licence plate. The characters are recognized using KNN (K Nearest Neighbor) classification model.

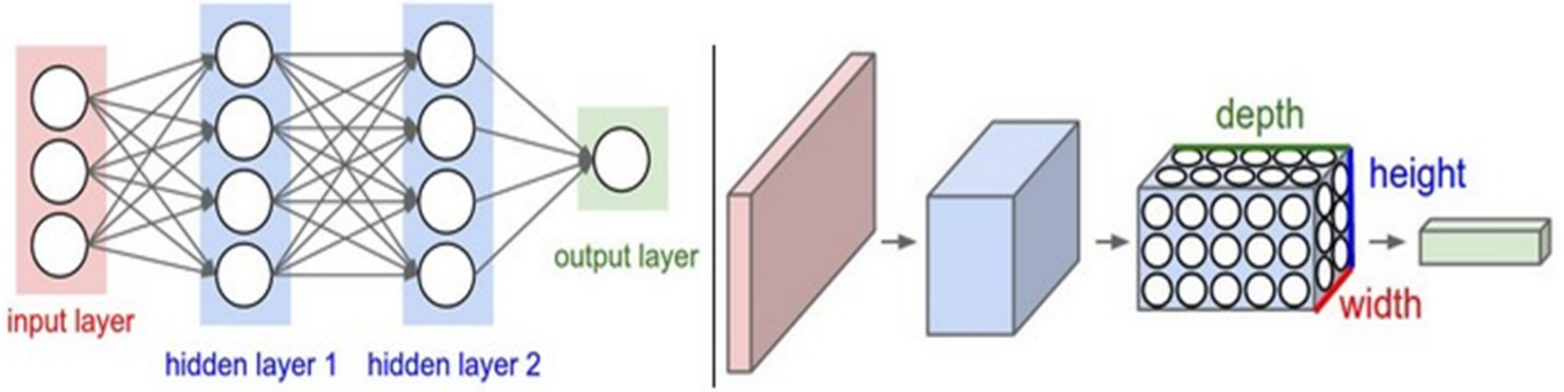
Vehicle Classification

Vehicle classification is performed using Convolutional Neural Networks(CNN or ConvNets) which are a special kind of multi-layer neural networks. Convolutional Neural Networks are designed to recognize visual patterns directly from pixel images with minimal preprocessing.

We have used pretrained Residual Network deep learning architecture and then fine-tuned the network to fit our dataset of images of Cars and Bikes.

Convolutional Neural Networks

Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture accordingly[2]. Unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. Moreover, the neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully–connected manner[2]. Basically, a ConvNet is made up of Layers. Every Layer has a simple API: It transforms an input 3D volume to an output 3D volume with some differentiable function that may or may not have parameters[2].



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

```
Train on 672 samples, validate on 168 samples
Epoch 1/12
672/672 [==============================] - 331s 492ms/step - loss: 0.6903 - acc: 0.7768 - val_loss: 0.2391 - val_acc: 0.9107
Epoch 2/12
672/672 [==============================] - 326s 485ms/step - loss: 0.4012 - acc: 0.8765 - val_loss: 0.2189 - val_acc: 0.9345
Epoch 3/12
672/672 [==============================] - 326s 484ms/step - loss: 0.3353 - acc: 0.8884 - val_loss: 0.2051 - val_acc: 0.9405
Epoch 4/12
672/672 [==============================] - 326s 485ms/step - loss: 0.3227 - acc: 0.8854 - val_loss: 0.2693 - val_acc: 0.9345
Epoch 5/12
672/672 [==============================] - 326s 486ms/step - loss: 0.2930 - acc: 0.8884 - val_loss: 0.2116 - val_acc: 0.9405
Epoch 6/12
672/672 [==============================] - 327s 486ms/step - loss: 0.2490 - acc: 0.9062 - val_loss: 0.2073 - val_acc: 0.9405
Epoch 7/12
672/672 [==============================] - 328s 488ms/step - loss: 0.2345 - acc: 0.9077 - val_loss: 0.1943 - val_acc: 0.9405
Epoch 8/12
672/672 [==============================] - 333s 495ms/step - loss: 0.2092 - acc: 0.9211 - val_loss: 0.2055 - val_acc: 0.9405
Epoch 9/12
672/672 [==============================] - 334s 497ms/step - loss: 0.1863 - acc: 0.9182 - val_loss: 0.2068 - val_acc: 0.9405
Epoch 10/12
672/672 [==============================] - 334s 496ms/step - loss: 0.2148 - acc: 0.9152 - val_loss: 0.2109 - val_acc: 0.9405
Epoch 11/12
672/672 [==============================] - 333s 495ms/step - loss: 0.2025 - acc: 0.9271 - val_loss: 0.2244 - val_acc: 0.9286
Epoch 12/12
672/672 [==============================] - 334s 497ms/step - loss: 0.1777 - acc: 0.9241 - val_loss: 0.2629 - val_acc: 0.8988
```

Screenshot of training done for Vehicle Classification

Database management

Csv (Comma Separated Values) file is used to store the following information:

1. Number Plate of the vehicle
2. Type of Vehicle (Car or Bike)
3. Entry Time of the vehicle

As soon as a new number plate is detected, all the above information is stored in the database of the Parking system and whenever an existing number plate is recognized, the fare is calculated on the basis of exit time and all the information regarding the vehicle is removed from the database. This is implemented using Pandas in Python.

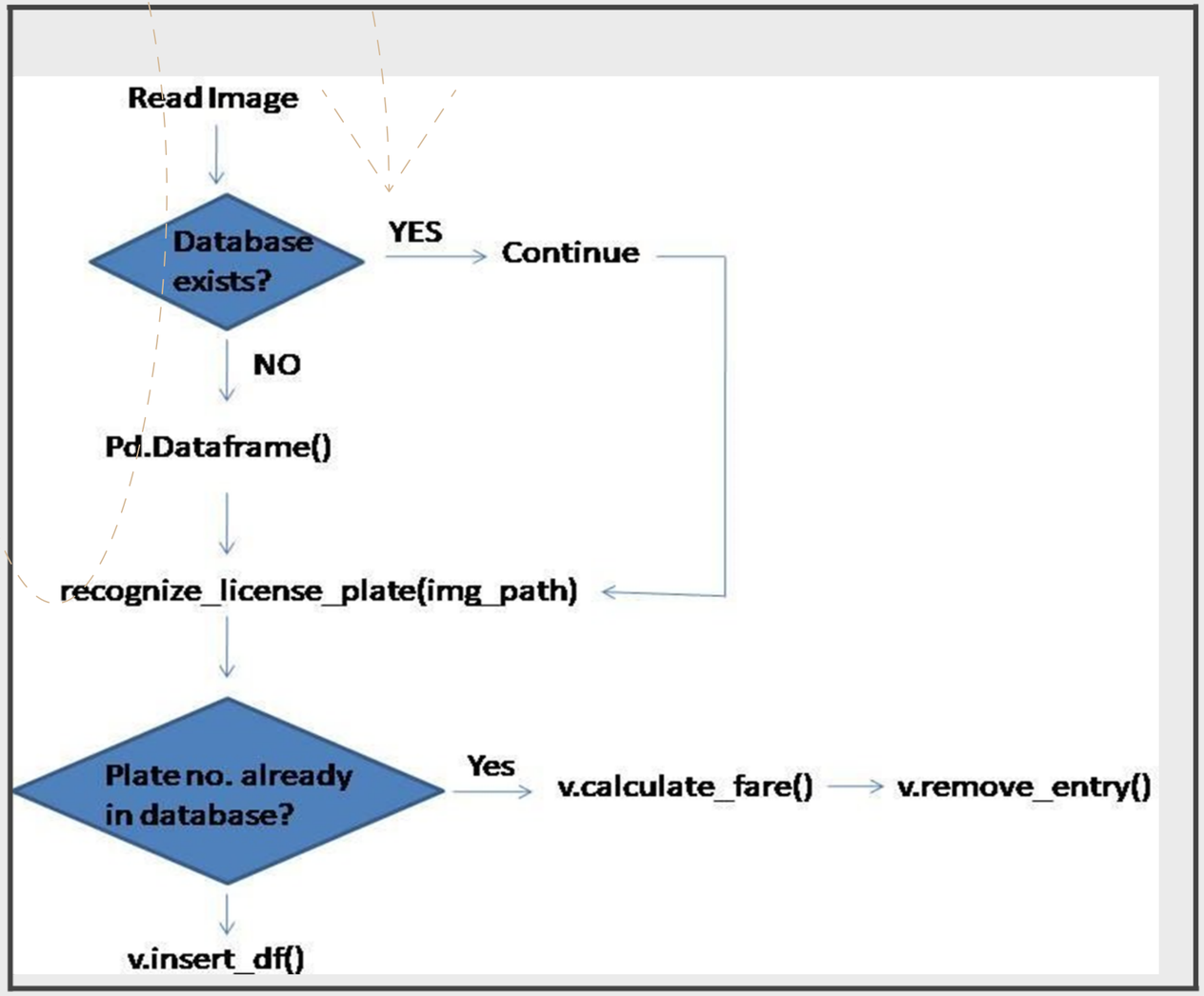
	A	B	C
1	**License_Plate_Number**	**Vehicle Type**	**Entry Time**
2	IC8S	Bike	Wed Apr 25 22:28:53 2018
3	MCLRNF1	Car	Fri Apr 27 23:47:44 2018
4	YINY	Car	Fri Apr 27 23:49:36 2018

Implementation

```
import pandas as pd
from datetime import datetime
import time
data = pd.read_csv("dup.csv")
veh = data['Vehicle Type']
number = data['License Plate No.']
bal = data['Balance']
exit_time = int(time.time())
Tim1 = data['Entry Time']
i = 0
fare = 0
for x in veh:
    tt = Tim1[i]
    print('Entry time : '+str(tt))
    print('Exit time : '+str(datetime.now()))
    print('License Plate Number : '+str(number[i]))
    print('Previous Balance : '+str(bal[i]))
    obj2 = time.strptime(tt, "%Y-%m-%d %H:%M:%S.%f")
    entry_time = int(time.mktime(obj2))
    duration = (exit_time-entry_time)/60
    if x == 'Car':
        fare = 20
        if (duration-60) > 0:
            fare += (duration-60)*0.4
    else:
        fare = 10
        if (duration-60) > 0:
            fare += (duration-60)*0.2
    print('Current Fare : '+str(fare))
    print('Updated balance : '+str(bal[i]-fare))
    i += 1
```

As soon as ultrasonic sensor detects the object, camera clicks the image and that image is passed as argument to the ‘main’ function shown above.

- The main function opens the database of thevehicles or creates one if one doesn’t exist already.
- The license plate is recognized and stored in variable license_plate.
- An object of class FareCalculator is created.
- Image, dataframe and license plate are the members of class which get instantiated upon making of the object.


Read Image
Database exists?
YES
Continue
NO
Pd.Dataframe()
recognize_license_plate(img_path)
Plate no. already in database?
Yes
v.calculate_fare()
v.remove_entry()
v.insert_df()

Code flow

Result

A complete study of the survey papers revealed various new technologies used in Automatic Fare Generation which included convolutional neural network , Image pre-processing , You look only once algorithm AI , Self Organizing MAP etc. Further detailed analysis of those papers revealed some gaps which could be covered in our project. Automatic fare generation system proposed by us makes use of hardware components such as Raspberry Pi 3, Ultrasonic Sensor (HC-SR04), DC Motor which we have configured so far. The Automatic fare generation system will also be using some software as analyzed by us till now which include Tensor Flow, Keras, Anaconda, GIT Lab. Besides this we will also be using Python libraries extensively to make our project efficient and solve problems algorithmically. An analytical approach towards the project reveals us that it would engage Computer Vision and Machine Learning tools too.

Conclusion

An efficient and time saving parking system can be created with an Automatic Fare Generation System. As the world treads the path of Artificial Intelligence, renovating the systems to make them as human management independent as possible has utmost importance. An automatic system thus not only pave way for a hassle free fare generation but increases the overall efficiency thus saving time and money of the users.

Future work

The project can be extended to develop an app to convert traditional parking spaces into a smart parking lot. The app functionality will include:

- Each user can have an account with single or multiple license plates registered.
- Machine learning can further be deployed to give discount to the users on the basis of their usage of that particular parking lot.
- By linking Aadhar/syncing account info fare can be automatically deducted.

Thank You