**Artificially Intelligent**

**Parking Management System**

# 

## Abstract

Managing parking is an important issue that finds a lot of automation nowadays. This work utilizes *Artificial Intelligence to manage the parking system*. The implemented features are related to priority parking categorization, nearest parking slot assignment, and security authentication of vehicles to be parked. Working on finding an accurate, time saving and fast [vehicle to park slot] mappers is also considered. Parking agents are trained on different data sets of cars from kaggle, github and et cetera and eventually minimize the vision to number plate reading. A run time database/storage mechanism is used to make the system work on priority parking, available parking slots and et cetera to map the solution on virtual scenarios. The management done by the mentioned artificially intelligent system can help in saving time for parking a vehicle, by more than 50% as in simple terms a car is mapped directly to its nearest allowed parking.

# Introduction

As the world is now moving towards automation, vehicles are important for humans to manage without any fuss. Factors like priority of person, priority of car, security domain covering the authentication of vehicle entering in the parking region, allocating nearest available parking slot. It is seen that parking management is an issue around the globe, moreover it takes more energy, attention than required due to human mismanagement.

So, modern problems require modern solutions. Therefore, the parking management system enables many important persons to get their time saved as we see government officials, army officials, professional workers and many other people gets their time wasted in parking vehicle, due to the reason they can't locate the nearest place to park, as well as, their reserved parking space is no more reserved or say someone from outside of the office have utilized the space for no reason that makes up to mass issues to be addressed at many places especially issues specific to Pakistan. Proper authentication is as important as allocating the nearest parking place in places like Pakistan where we have to maintain boundaries of people as it is reported that many outsiders do park at the reserved parkings and inside the domain of other people’s property, as the parking management crisis is among the issues in the country.

The detection phase is introduced first as we first need to collect relevant information from the frame captured, then the information collection is fed into the authentication agent that validates from the database that whether the vehicle is allowed to park in the parking area desired or access is denied.

Authentication leading to AI embedded in terms of graph traversal to find the shortest route is the hidden implementation of the system. Moreover, the desired allocated parking space of the user is communicated accordingly. This system efficiently solves much of the problem faced in parking the vehicle.

Further, other contributors towards parking management are considered in the project to cover the background knowledge and implementation ideas and its working. So, that the implementation of parking management in Pakistan can become more dynamic and adaptable.

### Literature Review

In this section, some of the most representative proposals in this area are described.

In [ **8** ],an engineering for savvy stopping is proposed utilizing a portable application associated with the cloud. The framework provides parking facilities in offices and users can hold places and make reservations before showing up at the parking spot. There exist different frameworks that utilize remote correspondence to save spots and suggestions of closest parking spots through a global positioning system (GPS).

In [ **5** ], The framework transmits the accessibility of spaces each 2 min. On the off chance that all parking spots are involved, no activities are thought of; in the other case, any client can save a spot inside 2 km of their area. The GPS facilitates access for any cell phone; at that point, the client gets a message with directions. The application requires a WiFi association and no activity is performed if all parking slots are reserved.

In [ 6 ], which calculates the ideal parking spot for the client dependent on the distance of trajectory and time. It ought to be noticed that the framework doesn't have the booking administration and is dependent upon the accessibility of the space around then.

in [ 7 ], the researchers introduced the administration of parking spots continuously utilizing the cloud as a method for correspondence and a database. The framework has a portable application to reserve space spots and an ultrasonic sensor set on the ground associated by means of Ethernet.

The authors clarified that all the current smart parking technologies and applications are not appropriate for open parking garages because of changing natural conditions and high expenditure.

These current smart parking systems allow parking a vehicle quickly and efficiently. However, they require special infrastructure that is expensive in terms of installation, time, and money. The preceding raises an area of innovation opportunity in the automation of private parking lots looking for practicality in its installation, modularity, and design, which may trigger the introduction of new business models.

# Proposed Method/Solution

The methodology of project is divided into following modules:

* Collecting and Allocating the Datasets
* Preprocessing the Images
* Designing the Models
* Gathering and Predicting the images
* Parking Slot Selection
* Guiding to slot through Graphical Representations

**Collecting and Allocating the Datasets:**

The datasets have been collected from multiple sources and preprocessed to make them ready for input to model.

The datasets for detecting the car have been taken from Stanford Cars Dataset - Kaggle[1] and Nienaber Potholes Dataset - Kaggle[2].

For detecting the number plates from the car we have used the datasets from Turkish License Plate Detector[3] repository on Kaggle. The repository had both the negative as well as positive images.

The dataset for detecting the characters on the Number plate has been taken from The Chars74K [4].

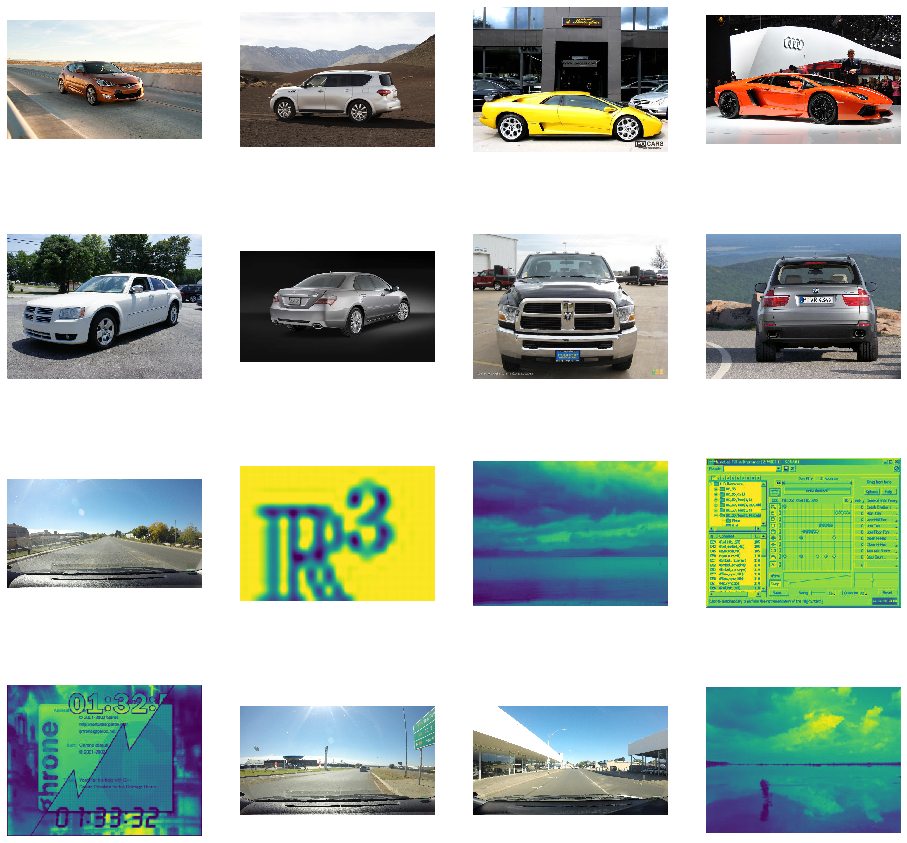
**Preprocessing the Images:**

The images we got from the Pothole Detection Dataset had to be cleaned to serve as Negative Samples for our Car Detection Model.

For this purpose we used the Python Image Library PIL and cv2. We imported the desired number of samples for Training set and Testing set i.e. 2000 samples for training and 500 samples for testing.

For each image we crop the image one-fourth from the bottom which shows the dashboard part of the car. After that, we have images much better than the previous one and can result in better predictions.

After preprocessing we will have following images for Car Detection and Number Plate Extraction modules:



**Designing the Models:**

We will be detecting the presence of cars first of all. Then we will extract the number region from the car and then we will proceed to read the number plate. Which will then be used to guide the driver to his respective parking slot.

We will be designing three models for this:

1. Model for Car Detection
2. Model to Extract License Plate
3. Model to Extract and Recognize the Characters from License Plate

**Model for Car Detection:**

***Architecture***

We have used a sequential model based on Convolutional Neural Networks. The architecture is as follows:

1. Convolutional Layer
2. MaxPooling Layer
3. Convolutional Layer
4. MaxPooling Layer
5. Convolutional Layer
6. MaxPooling Layer
7. Convolutional Layer
8. MaxPooling Layer
9. Dropout Layer
10. Flatten Layer
11. Fully Connected Layer
12. Fully Connected Layer

**Convolutional Layer:**

This layer is the first layer of our model. It inputs an image of (150, 150, 3) shape and performs convolutions with 32 filters with a kernel size of 3x3.

**MaxPooling Layer:**

This layer takes the input from Convolutional Layer (148,148,32) and performs MaxPooling on them reducing size to (74, 74, 32).

**Convolutional Layer:**

This second convolutional layer again performs the convolutions with 64 filters and kernel of 3x3.

**MaxPooling Layer:**

This layer takes the input from Convolutional Layer (72, 72, 64) and performs MaxPooling on them reducing size to (36, 36, 64).

**Convolutional Layer:**

This second convolutional layer again performs the convolutions with 128 filters and kernel of 3x3.

**MaxPooling Layer:**

This layer takes the input from Convolutional Layer (34, 34, 128) and performs MaxPooling with a window of 3x3 this time reducing size to (11, 11, 128).

**Convolutional Layer:**

This last convolutional layer with reduce to (9, 9, 128).

**MaxPooling Layer:**

And this last MaxPooling Layer will return the (4, 4, 128) output.

**Dropout:**

The dropout layer will be used to output 10% of weights to avoid overfitting that we would be facing otherwise.

**Flatten Layer:**

The flatten layer will reshape or flatten the Matrix into 1 Dimensional array containing 2048 weights.

**Fully Connected Layer:**

This layer will reduce the weights from 2048 to 512.

**Fully Connected Layer:**

This is the final output layer and with sigmoid function for binary output, it will return 1 output. This output will give us the classification of the input image.

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

==========================================================

conv2d (Conv2D) (None, 148, 148, 32) 896

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d (MaxPooling2D) (None, 74, 74, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_1 (Conv2D) (None, 72, 72, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 36, 36, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 34, 34, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 11, 11, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_3 (Conv2D) (None, 9, 9, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_3 (MaxPooling2 (None, 4, 4, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 4, 4, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 2048) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 512) 1049088

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 1) 513

**Model to Extract Number Plate:**

***Architecture***

We have used a sequential model based on Convolutional Neural Networks. The architecture is as follows:

1. Convolutional Layer
2. MaxPooling Layer
3. Convolutional Layer
4. MaxPooling Layer
5. Convolutional Layer
6. MaxPooling Layer
7. Convolutional Layer
8. MaxPooling Layer
9. Dropout Layer
10. Flatten Layer
11. Fully Connected Layer
12. Fully Connected Layer

**Convolutional Layer:**

This layer is the first layer of our model. It inputs an image of (150, 150, 3) shape and performs convolutions with 32 filters with a kernel size of 3x3.

**MaxPooling Layer:**

This layer takes the input from Convolutional Layer (148,148,32) and performs MaxPooling on them reducing size to (74, 74, 32).

**Convolutional Layer:**

This second convolutional layer again performs the convolutions with 64 filters and kernel of 3x3.

**MaxPooling Layer:**

This layer takes the input from Convolutional Layer (72, 72, 64) and performs MaxPooling on them reducing size to (36, 36, 64).

**Convolutional Layer:**

This second convolutional layer again performs the convolutions with 128 filters and kernel of 3x3.

**MaxPooling Layer:**

This layer takes the input from Convolutional Layer (34, 34, 128) and performs MaxPooling with a window of 3x3 this time reducing size to (11, 11, 128).

**Convolutional Layer:**

This last convolutional layer with reduce to (9, 9, 128).

**MaxPooling Layer:**

And this last MaxPooling Layer will return the (4, 4, 128) output.

**Dropout:**

The dropout layer will be used to output 5% of weights to avoid overfitting that we would be facing otherwise.

**Flatten Layer:**

The flatten layer will reshape or flatten the Matrix into 1 Dimensional array containing 2048 weights.

**Fully Connected Layer:**

This layer will reduce the weights from 2048 to 512.

**Fully Connected Layer:**

This is the final output layer and with sigmoid function for binary output, it will return 1 output. This output will give us the classification of the input image.

**Model to Extract and Recognize Characters :**

The model is basically a Multi Layer Perceptron and the structure is as follows:

1. **Input Layer:**

This layer takes 1200 inputs which is basically the flattened version of 20x20x3 image.

1. **Hidden Layer:**

This layer has 2400 nodes that is twice the nodes in the input layer.

1. **Output Layer:**

This Layer has 36 outputs, where each output node represents the presence of a particular alphabet/number.

**Gathering and Predicting the Images:** The images that will be gathered by fetching the frames of the parking entrance will be fed into our model to predict the car. That would return the region of image containing the car. That particular region would then be fed into a model to extract the number plate. And in the end our last model will extract the numbers, recognize each character and return the output.

The output from the models will be then used to find whether the person/car belongs to the respective institution or not. And if it belongs to that institution which category it fits into. We will then check whether a parking slot is free to be assigned to that person.

**Guiding to slot through Graphical Representations:**

After a particular slot is chosen for the visitor, we will use the graphs to guide the visitor to its respective slot through the shortest path.

**Experiments:**

*Due to lockdown and uncertain situation ahead of COVID-19, we could not implement some domains we had planned and there might be some changes in future to make our task more accurate and feasible.*

*As of now, we have designed and implemented the Car Detection Model but we are still planning to make it more and more accurate. We have things sorted out of number plate extraction but we will move on to successfully reading the number plate first. Other than that we have also done a lot of work on Database and implementation of Graphs.*

*So, to manage computational experiments on the data of different cars and text recognition is done in diverse category, whose results are to be discussed.*

**Results:**

Our artificially intelligent is successful enough to detect car, detect its number plate, and finally find the exact route for the car to be followed so that it may get an optimized parking slot, so how much optimized it is for that we see our model have an accuracy as figured below:

**Training Accuracy : 91.87396351575457 %**

**Testing Accuracy : 90.67796610169492 %**

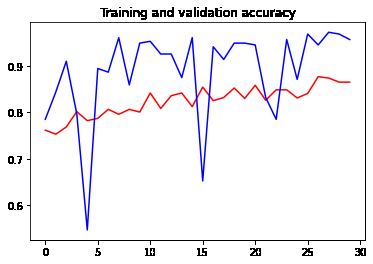
The accuracy is achieved by applying suitable detection and recognition algorithms. As said ***for character recognition***

20/20 - 62s - loss: 0.0652 - acc: 0.9793 - val\_loss: 0.0179 - val\_acc: 0.9966

***And for Number Plate Detection***

loss: 0.3271 - acc: 0.8652 - val\_loss: 0.2162 - val\_acc: 0.9570

***And for Car Detection,*** a plot demonstrates the result of accuracy achieved shown below

******

Finally, after analyzing the car and its number plate and algorithms performing \* category searches makes a path available for the car to follow so that upon following the path the car reaches the accurate and optimized destination for the car to follow in the parking arena.

The result that is plotted after detecting a specific type of car is shown below and the path is made available for the car to follow.

The artificially intelligent system is made eto evolve artificial intelligence concepts to reside with each other to detect images and plan route by shortest path algorithm and depth to intelligence added by Neural Networks, some of the algorithms and NN used to draw and proceed intelligent system are Convolutional Network, Region Based Detection, A\* Shortest Path, Multi Layer Perceptron.

**Conclusion:**

After all the development a model is chartered to depict as an artificially intelligent parking system in the form of a Front End layer applied as a GUI of the Intelligent system using Tkinter a well re known GUI system. This GUI applied and Artificially developed system allows the user to manage the Parking System efficiently saving space, time and distance to destination.s

**References:**

**[1] jessicali9530/stanford-cars-dataset**

**[2]** <https://www.kaggle.com/felipemuller5/nienaber-potholes-1-simplex>

[3] <https://github.com/muratlutfigoncu/turkish-license-plate-detector>

[4 ] <http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/>

[ 5 ] Orrie, O.; Silva, B.; Hancke, G.P. A Wireless Smart Parking System. In Proceedings of the 41st Annual Conference of the IEEE Industrial Electronics Society (IECON), Yokohama, Japan, 9–12 November 2015; pp. 4110–4114. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=A+Wireless+Smart+Parking+System&conference=Proceedings+of+the+41st+Annual+Conference+of+the+IEEE+Industrial+Electronics+Society+(IECON)&author=Orrie,+O.&author=Silva,+B.&author=Hancke,+G.P.&publication_year=2015&pages=4110%E2%80%934114)]

[ 6 ].Hsu, C.W.; Shih, M.H.; Huang, H.Y.; Shiue, Y.C.; Huang, S.C. Verification of Smart Guiding System to Search for Parking Space via DSRC Communication. In Proceedings of the 12th International Conference on ITS Telecommunications, Taipei, Taiwan, 5–8 November 2012; pp. 77–81. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Verification+of+Smart+Guiding+System+to+Search+for+Parking+Space+via+DSRC+Communication&conference=Proceedings+of+the+12th+International+Conference+on+ITS+Telecommunications&author=Hsu,+C.W.&author=Shih,+M.H.&author=Huang,+H.Y.&author=Shiue,+Y.C.&author=Huang,+S.C.&publication_year=2012&pages=77%E2%80%9381)]

[ 7 ].Karthi, M.; Preethi, H. Smart Parking with Reservation in Cloud based Environment. In Proceedings of the 2016 IEEE International Conference on Cloud Computing in Emerging Markets, Bangalore, India, 19–21 October 2016; pp. 164–167. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Smart+Parking+with+Reservation+in+Cloud+based+Environment&conference=Proceedings+of+the+2016+IEEE+International+Conference+on+Cloud+Computing+in+Emerging+Markets&author=Karthi,+M.&author=Preethi,+H.&publication_year=2016&pages=164%E2%80%93167)]

[ 8 ] .Khanna, A.; Anand, R. IoT based Smart Parking System. In Proceedings of the 2016 International Conference on Internet of Things and Applications (IOTA), Pune, India, 22–24 January 2016; pp. 266–270. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=IoT+based+Smart+Parking+System&conference=Proceedings+of+the+2016+International+Conference+on+Internet+of+Things+and+Applications+(IOTA)&author=Khanna,+A.&author=Anand,+R.&publication_year=2016&pages=266%E2%80%93270)]

[ 9 ].Wang, M.; Dong, H.; Li, X.; Song, L.; Pang, D. A Novel Parking System Designed for Smart Cities. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 3429–3434. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=A+Novel+Parking+System+Designed+for+Smart+Cities&conference=Proceedings+of+the+2017+Chinese+Automation+Congress+(CAC)&author=Wang,+M.&author=Dong,+H.&author=Li,+X.&author=Song,+L.&author=Pang,+D.&publication_year=2017&pages=3429%E2%80%933434)]