



# **Space Checker**

**DT228**  
**BSc in Computer Science**

**Jianyu He**

**D17124591**

**Martin McHugh**

**School of Computer Science  
Technological University, Dublin**

**10/04/2020**

# **Abstract**

The goal of this project is to develop the efficiency of drivers, is able to help the driver find suitable car space using the website.

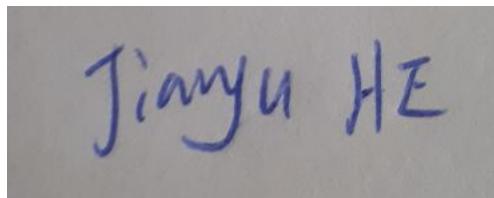
Our project is to design an application used to check the amount of car, car space and space status in the car park by CCTV. The application is going to show car space available and a simple car space map on the website live that driver can check car space on the website, this application is helpful for a driver to check how many cars spaces available in the car park and do not need waste time to check car space in the car park. we are going to create a database for our project. This database will store details of the user, car park and space.

For this project, we do not want to cost money to purchase a CCTV and set it into car space, so we would like to use a video from the car park to test this program.

# Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

A photograph of a handwritten signature in blue ink. The signature reads "Jianyu HE".

---

D17124591

10/04/2020

## **Acknowledgements**

I wish to acknowledge my supervisor Martin McHugh for his guidance and support from the proposal stages and interim report of the project and throughout. I also wish to thank Damian Gordon for him tireless work coordinating our final year project module.

## Table of Contents

1. Introduction	7
1.1. Project Background	7
1.2. Project Description	7
1.3. Project Aims and Objectives	8
1.4. Project Scope	8
1.5. Thesis Roadmap	8
2. Literature Review	9
2.1. Introduction	9
2.2. Alternative Existing Solutions to Your Problem	9
2.3. Technologies you've researched	13
2.4. Other Research you've done	16
2.5. Existing Final Year Projects	26
2.6. Conclusions	27
3. Prototype Design	29
3.1 Introduction	29
3.2. Software Methodology	29
3.3. Overview of System	31
3.4. Feature List	32
3.5. Use Case Diagram	33
3.6. Front-End	34
3.7. Middle-Tier	39
3.8. Back-End	39
3.9. Conclusions	41
4. Prototype Development	42
4.1. Introduction	42
4.2. Prototype Development	42
4.3. Image Processing	42
4.4. Machine Learning	54
4.5. Front-End	58
4.6. Middle-Tier	64
4.5. Back-End	68
4.6. Conclusions	69

5. Testing and Evaluation	69
5.1. Introduction	69
5.2. Plan for Testing	69
5.3. Black Box Testing and Integration Testing	70
5.4. White Box Testing and unit Testing	71
5.5. Plan for Evaluation	74
5.6. Conclusions	74
6. Personal Conclusion and Future Work	75
6.1. Personal Conclusion	75
6.2. Future Work	75
Bibliography	76

## 1. Introduction

This chapter is about the introduction of the project, it lists project background, project description, project aim and objectives, project scope and Thesis Roadmap.

### 1.1. Project Background

As the Population expansion and for the convenience of travel, more and more people buy cars [1]. As results some car parks may very busy, that we design this application to help drivers do not have to waste time to check does any space left [2]. Python is the best programming language uses computer vision to process images.[3] [4]

### 1.2. Project Description

This project is about Machine learning, Classification, Deep learning, Convolutional neural networks, and image processing.

Space checker is a live web application deployed by Django (Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.). this web application has a login page, register page, and home page. the home page can allow the user to select a different car park to show statues of space. The home page contains a status map shows the position of free space and space id.

This application is going to design by OpenCV using Python programming language (Python is a programming language that's soaring in popularity with web and software developers. OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library.). OpenCV is helpful for image processing.

The database of application is set on PostgreSQL (PostgreSQL is a general-purpose and object-relational database management system, the most advanced open-source database system.).

This application is a 3-tier structure, it has presentation tier, application tier, and data tier.

### 1.3. Project Aims and Objectives

The aim of this project provides a website to show information about the car park, this helps users easy to understand how many car spaces are free and where is the car space.

The goal of this project is to provide a simple website shows the name of the car park, total car space, number of free space live, number of cars exist and space map. It allows the driver to check details of car space anytime and anywhere, the benefit of this application is to help drivers do not have to waste time check where free car space. It also will help the car manager to create a report using data from the database.

To achieve these aims some milestones were set. These milestones involved setting flexible dates to complete certain parts of the project. By setting specific time frames to have certain parts of the application developed it was ensured that the final application would be completed to the best attempt by the end.

### 1.4. Project Scope

This project is about Machine learning, Classification, Deep learning, Convolutional neural networks, and image processing.

### 1.5. Thesis Roadmap

This section will provide a summary of each of the chapters covered in this report.

#### **Research**

In this chapter some of the key areas of research that are important in this project will be presented, It all about necessary gained knowledge of space checker and its associated problems

#### **Design**

This chapter delves into the methodology chosen for this project and how these choices came to be. Following this, detailed use-cases and personas related to the desired system will be presented.

#### **Development**

This chapter continues with the issues explored in the previous chapter and will outline the development process undertaken in this project. This chapter will present the key development processes and the challenges encountered during the creation of this system.

#### **Testing and Evaluation**

This chapter describes how all the testing and evaluation of the system was executed. Each phase of testing will be described in detail.

## Personal Conclusion and Future Work

This chapter will reflect on the personal conclusion of project complete, and the future work planned for the project.

## 2. Literature Review

### 2.1. Introduction

In this chapter some of the key areas of research that are important in this project will be presented, It all about necessary gained knowledge of space checker and its associated problems.

### 2.2. Alternative Existing Solutions to Your Problem

Vehicle detection in allocated parking space using camera image [6]



Figure 1(6)

This application finds a white line of the car space to detect the position of car space.

The application implemented is an AI algorithm that takes in an annotated image and draws a green-coloured bounding box around the parking space if it is available, and a red-coloured bounding box if it is occupied. This included training a deep convolutional neural network[23] to detect if a parking space is occupied or not. It trained a VGG model[25] using transfer learning. It means to create an image of an empty car space of each rectangle and then compares those images with each frame of video. That allows the system to detect if it is free of car space.[6]



Figure 2(6)

**Advantages:**

1. The trained model was able to achieve 99.5% accuracy on the validation set.
2. This application shows each rectangle of the car space very clearly.

**Disadvantages:**

1. If the barrier exists on the screen to hide the line of car space, the system cannot detect those positions of car space.
2. Must train the VGG model by manual work, can only use for one car park.
3. It is difficult to detect white lines when white lines getting dim.

Snagging Parking Spaces with Mask R-CNN and Python [7]

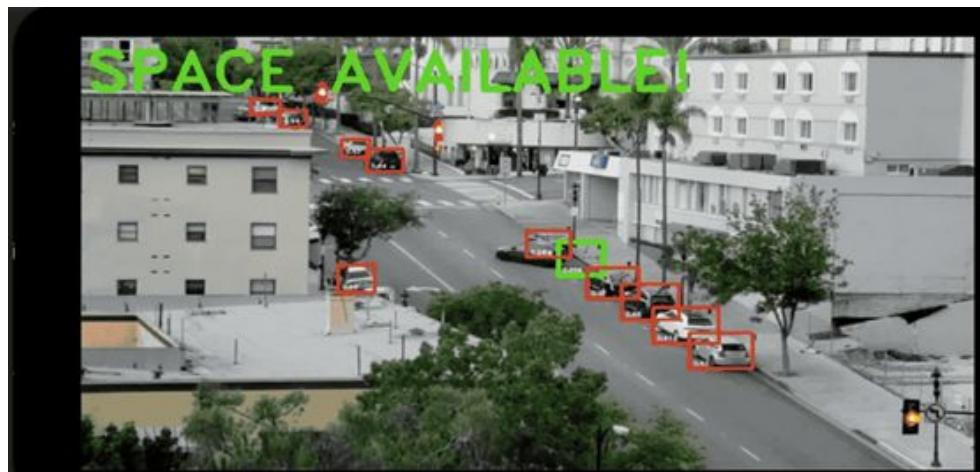


Figure 3(7)

This application is using mask R\_CNN[8] to detect location and outline of the car(ima1). This application through moving of car to check which position is car space, for example, if this car doesn't move over 5 mins, then this position is car space, then draw a box in this position, if car

move then change green box to red box finally send image and text by message to the user, tell user there have a car space available(ima2).[7]



Figure 4(7)

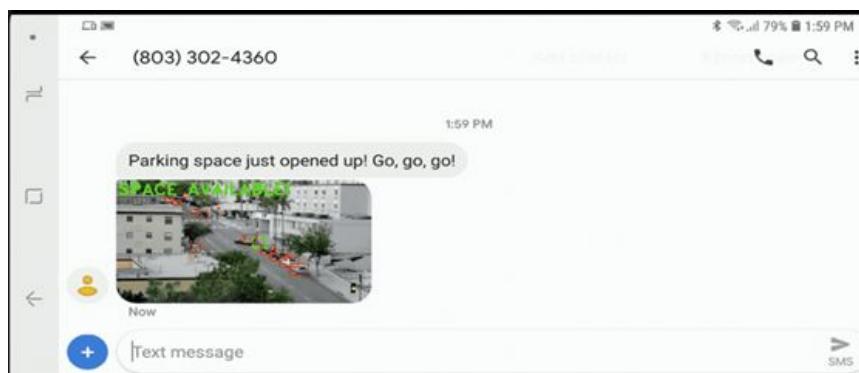


Figure 5(7)

#### Advantages:

1. It is easy to detect where are cars.
2. Don't need manual work, the system going to solve everything.
3. Shows a red box and the green box clearly.

#### Disadvantages:

1. It is difficult to detect cars when some barrier hides the car.
2. If a car stays more than 5 mins on the other space, the system will Determine the position is car space

Use OpenCV TensorFlow to detect car space [5]

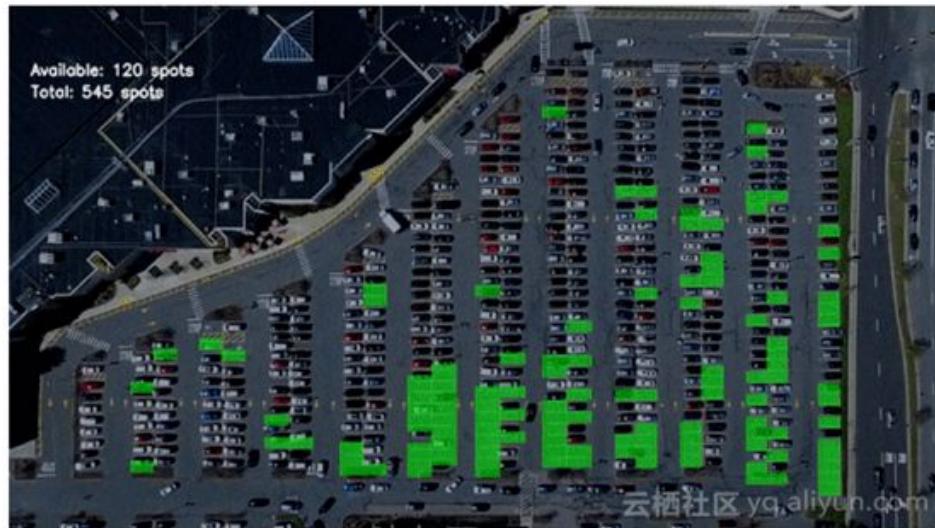


Figure 6(5)

This application is using edge detection, Hough Transform[26] to draw all the car space(img3). Then allocate id for each car space. This application creates a CNN model[8] to create an image of empty car space and car space which have a car exists, that allows the system to detect is free of car space. (img4) [5]



Figure 7(5)

#### **Advantages:**

1. High accurate
2. Able to recognition small cars.
3. Shows number of car space available and total car space

#### **Disadvantages:**

1. The system may recognition space is not belonging car space.
2. Still, have mistaken exists.

## 2.3. Technologies you've researched

### **Library used:**

Our project is about Image processing, first we have to choose a library of programming function mainly aimed at real-time computer vision to help us process images.

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. In simple language, it is a library used for Image Processing. It is mainly used to do all the operations related to Images.[13]

#### **Pros**

1. Faster & Free - With BSD license for OpenCV, you get all the features for free. Also, optimized code runs faster than MATLAB!
2. Richer & bigger community - With regular improvements and over 40,000 developers, it's one huge family. Your back is taken care of.
3. Portability - Due to its source-code in C/C++, any machine capable of running C/C++ (Linux), can have OpenCV running as well.

#### **Cons**

1. Challenging for beginners - More so on C++ than Python definitely. Python's syntax helps but MATLAB is hard to beat when dealing with large matrices.
2. Weak documentation - Again, more challenging for beginners. Lesser tutorials and documentation. This is slowly changing[28]

MATLAB is a computing and visualization software package, as well as a fourth generation programming language, published by MathWorks. It performs matrix manipulations, function plotting, algorithm implementation, and many high-level mathematical operations. Its companion package, Simulink, is used for graphical multi-domain simulation.[27]

#### **Pros**

1. Easy-to-perform linear algebra - Incredibly simple & capable of computing large matrices - inverses, multiplication, rank etc.
2. Simpler language - Much easier to write lines of code. Similar functionality might be just 1–2 lines of code compared to OpenCV.
3. Picture worth a 1000 words - Visualizing in MATLAB is easy. With its integrated IDE, you can edit code, display output, plot figures, manipulate data etc.

## **Cons**

1. Costly - Unless you are provided with a license or your bank balance has more than 5 digits to the left of the decimal, the budget is not for the faint of heart.
2. Slower - Comparatively slower especially for real-time processing of videos on incrementally lower-level machines.[28]

We decided to use opencv, because Matlab is too expensive but opencv is free, and this project have to use image processing to process video, so use opencv is faster.

## **Programming language choose**

After choose library use for project then we have to think about which programming language can use opencv that suitable for the project, we compare the c, java and python language as below.

### **Python**

The Python language has diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, etc.

#### **Extensive Support Libraries**

It provides large standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python.

#### **Integration Feature**

Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also processes XML and other markup languages as it can run on all modern operating systems through the same byte code.[29]

### **C**

C programming language is the structured programming language, So, It helps you to think of the problem in terms of function modules or blocks, Collection of these modules makes a complete program, This modular structure makes the program debugging, testing and maintenance easier.

C language has the ability to extend itself , It is the collection of functions which are supported by the C library this makes us easier to add our own functions to the C library, Because of the availability of a large number of functions , the programming task becomes simple.[30]

### **Java**

**Java is Simple:** Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages. The reason that why Java is much simpler than C++ is

because Java uses automatic memory allocation and garbage collection where else C++ requires the programmer to allocate memory and to collect garbage.

**Java is Object-Oriented:** Java is object-oriented because programming in Java is centered on creating objects, manipulating objects, and making objects work together. This allows you to create modular programs and reusable code.

**Java is Platform-Independent:** One of the most significant advantages of Java is its ability to move easily from one computer system to another.[31]

First we have to exclude C programming language, because we do not have much experience about using C programming language, java and python is good programming language can use for our project, but Python is used widely by enterprises for scientific computing, big data, and artificial intelligence projects. Also, it is currently most preferred programming language for machine learning and data science.[32] That help us easy to get research from internet about image processing, So we decide using python to design our project.

### **Web application deploys**

after programming language chose, then we have to think about how to deploy application on the website.

Django is a high-level Python Web framework that encourages rapid development and pragmatic, clean design. Django is a high-level Python Web framework that encourages rapid development and pragmatic, clean design. A web application framework is a toolkit of components all web applications need. The goal here is to allow developers to instead of implementing the same solutions over and over again, focus on the parts of their application that are new and unique to their project. In fact, Django is much more fully featured than many other frameworks out there. It takes care of a lot of the hassle of Web development, letting you focus on writing your application without any need to reinvent the wheel. It's free and open source. Additionally, the Django framework enables you to model your domain and code classes, and before you know it, you already have an ORM.[16]

### **Software Used**

Programmers usually tackle this situation with a tool called a debugger, which allows running their program step-by-step. Unfortunately, most debuggers are optimized for professional usage and assume the user already knows the semantics of language constructs.

Thonny is a beginner-friendly Python IDE, developed in the University of Tartu, Estonia, which takes a different approach as its debugger is designed specifically for learning and teaching programming.[22]

although we can just use command line to deploy django, but we think use software to deploy django is best to show each layout of django, and make code design easier.

Pycharm is an IDE (Integrated Development Environment) by Jetbrains. It is used for development in Python and frameworks like Django. You can customize it with themes and plugins. It lets you enhance productivity while coding by providing some features like suggestions, Local VCS, etc. If you

are a beginner, We would recommend you to use PyCharm as it will ease your learning. Plus it's advanced features will ease development process. [17]

### **Database used**

PostgreSQL, also known as Postgres, advertises itself as “the most advanced open-source relational database in the world”. It was built to be feature-rich, extendable and standards-compliant. In the past, Postgres performance was more balanced - reads were generally slower than MySQL, but it was capable of writing large amounts of data more efficiently, and it handled concurrency better.

The performance differences between MySQL and Postgres have been largely erased in recent versions. MySQL is still very fast at reading data, but only if using the old MyISAM engine. If using InnoDB (which allows transactions, key constraints, and other important features), differences are negligible (if they even exist). These features are absolutely critical to enterprise or consumer-scale applications, so using the old engine is not an option. On the other hand, MySQL has also been optimized to reduce the gap when it comes to heavy data writes.

When choosing between MySQL and PostgreSQL, performance should not be a factor for most run-of-the-mill applications – it will be good enough in either case, even if you consider expected future growth. Both platforms are perfectly capable of replication, and many cloud providers offer managed scalable versions of either database.

Postgres is an object-relational database, while MySQL is a purely relational database. This means that Postgres includes features like table inheritance and function overloading, which can be important to certain applications. Postgres also adheres more closely to SQL standards.

Postgres handles concurrency better than MySQL.[10]

## [\*\*2.4. Other Research you've done\*\*](#)

### **Tensorflow**

Machine learning is a complex discipline. But implementing machine learning models is far less daunting and difficult than it used to be, thanks to machine learning frameworks—such as Google’s TensorFlow—that ease the process of acquiring data, training models, serving predictions, and refining future results.

Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.[23]

## Mask R-CNN

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding-box object detection, and person key point detection. Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners. We hope our simple and effective approach will serve as a solid baseline and help ease future research in instance-level recognition. Code has been made available at this [https URL \[8\]](https://github.com/facebookresearch/maskrcnn-benchmark)

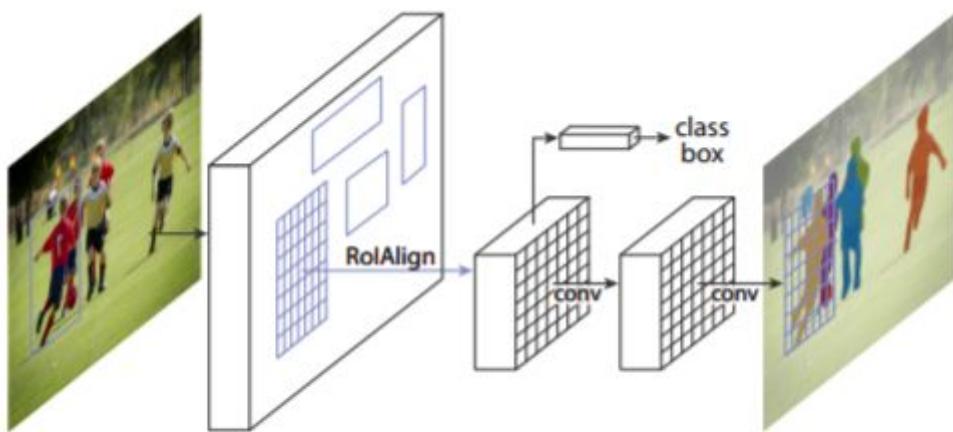
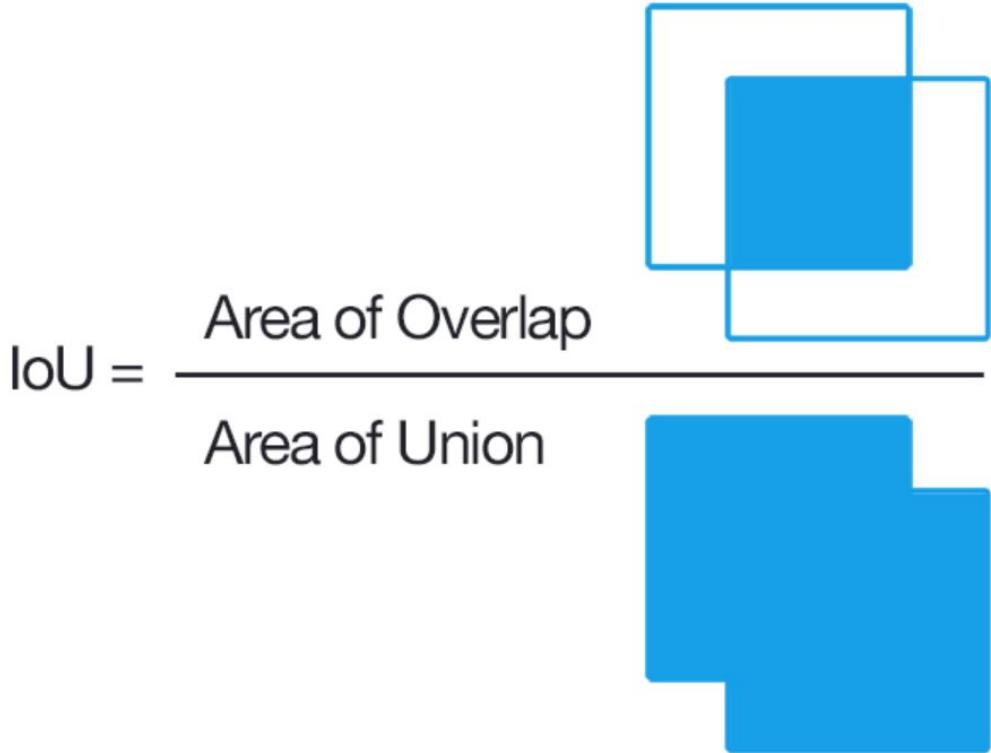


Figure 1. The **Mask R-CNN** framework for instance segmentation.

Figure 8[8]

## Intersection Over Union

metric used to measure the accuracy of an object detector on a dataset. We often see this evaluation metric used in object detection challenges such as the popular PASCAL VOC challenge.[9]



**Figure 2:** Computing the Intersection over Union is as simple as dividing the area of overlap between the bounding boxes by the area of union (thank you to the excellent [Pittsburg HW4 assignment](#) for the inspiration for this figure).

*Figure 9[9]*

## Hough Transformation

It's more common to think of a line in rectangle coordinates, i.e.  $y = mx + b$ . As the Wikipedia article states, a line can also be expressed in polar form. The Hough transform exploits this change of representation (for lines, anyway. The discussion can also be applied to circles, ellipses, etc.).

The first step in the Hough transform is to reduce the image to a set of edges. The Canny edge-detector is a frequent choice. The resulting edge image serves as the input to the Hough process.

To summarize, pixels "lit" in the edge image are converted to polar form, i.e. their position is represented using a direction theta and a distance r - instead of x and y. (The center of the image is commonly used as the reference point for this change of coordinates.)

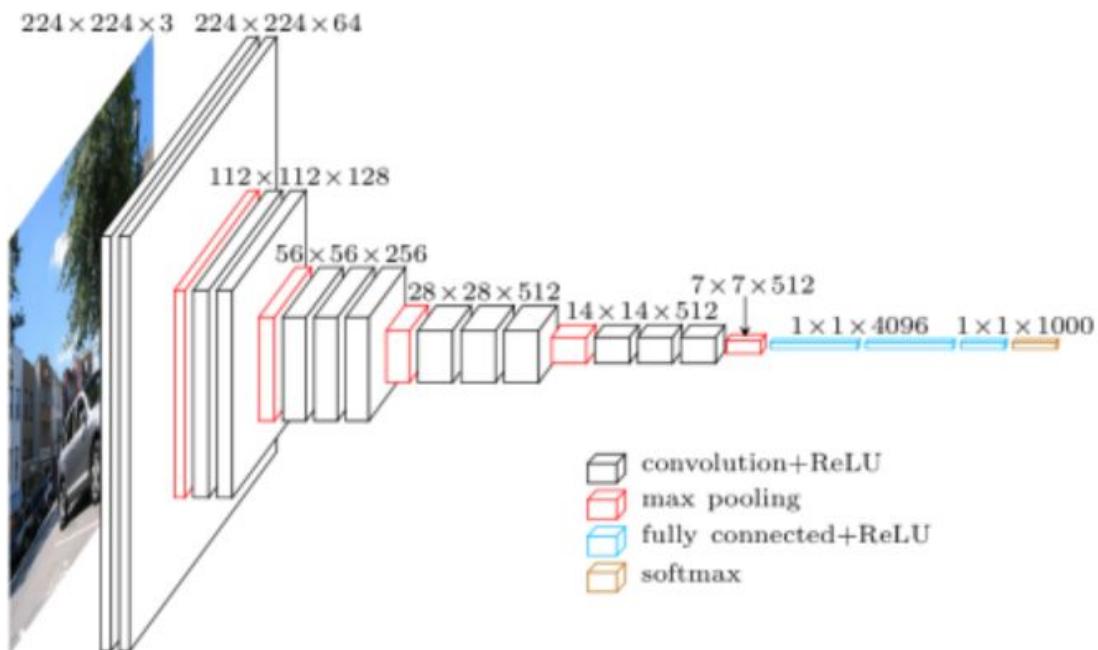
The Hough transform is essentially a histogram. Edge pixels mapping to the same theta and r are assumed to define a line in the image. To compute the frequency of occurrence, theta and r are

discretized (partitioned into a number of bins). Once all edge pixels have been converted to polar form, the bins are analyzed to determine the lines in the original image.

It is common to look for the N most frequent parameters - or threshold parameters such that counts smaller than some n are ignored[26]

## VGG

VGG is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” . The model achieves 92.7% top-5 test accuracy in ImageNet , which is a dataset of over 14 million images belonging to 1000 classes.[25]



**FIG. 2 - MACROARCHITECTURE OF VGG16<sup>[5]</sup>**

*Figure 10[25]*

## Convolutional Neural Networks

Convolutional Neural Networks have a different architecture than regular Neural Networks. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a last fully-connected layer — the output layer — that represent the predictions.

Convolutional Neural Networks are a bit different. First of all, the layers are organised in 3 dimensions: width, height and depth. Further, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension.[34]

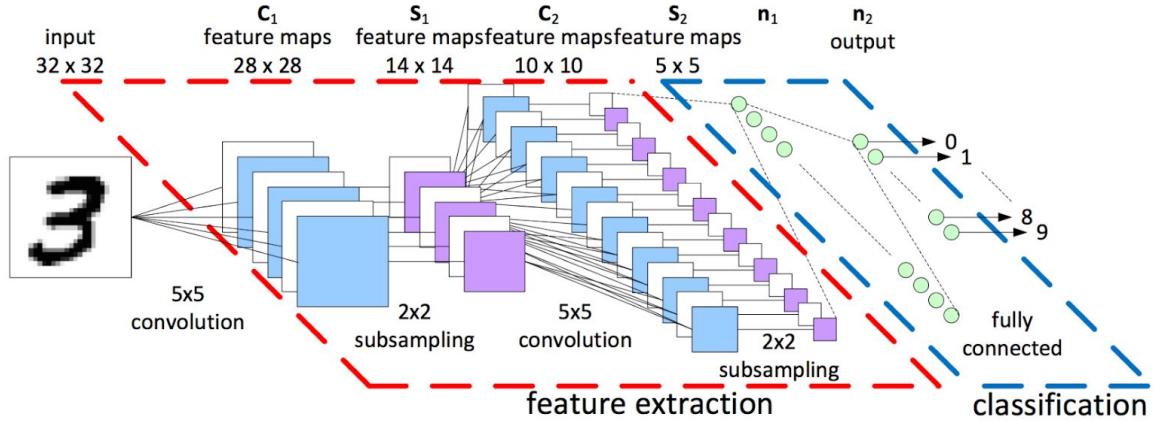


Figure 11[34] Convolutional Neural Networks architecture

### K-Nearest Neighbour

KNN falls in the supervised learning algorithms. This means that we have a dataset with labels training measurements ( $x, y$ ) and would want to find the link between  $x$  and  $y$ . Our goal is to discover a function  $h: X \rightarrow Y$  so that having an unknown observation  $x$ ,  $h(x)$  can positively predict the identical output  $y$ . [35]

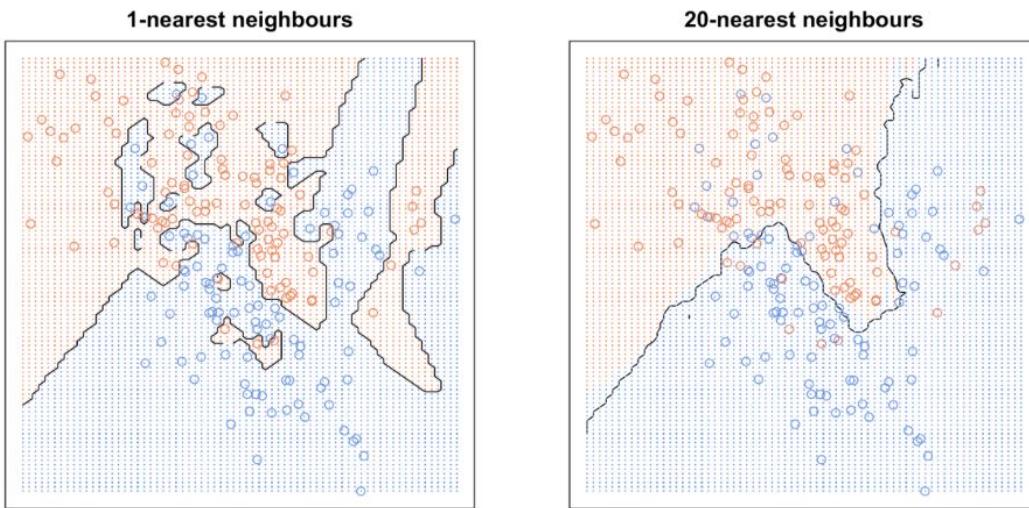
For distance metrics, we will use the Euclidean metric.

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + \dots + (x_n - x'_n)^2}$$

Finally, the input  $x$  gets assigned to the class with the largest probability.

$$P(y = j | X = x) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(y^{(i)} = j)$$

Figure 12[35] knn algorithms



*Figure 13[35]*

### **support vector machine**

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.

So you're working on a text classification problem. You're refining your training data, and maybe you've even tried stuff out using Naive Bayes. But now you're feeling confident in your dataset, and want to take it one step further. Enter Support Vector Machines (SVM): a fast and dependable classification algorithm that performs very well with a limited amount of data.[36]

### **Back Propagation Neural Network**

Back-propagation is the essence of neural net training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and to make the model reliable by increasing its generalization.

Backpropagation is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.[37]

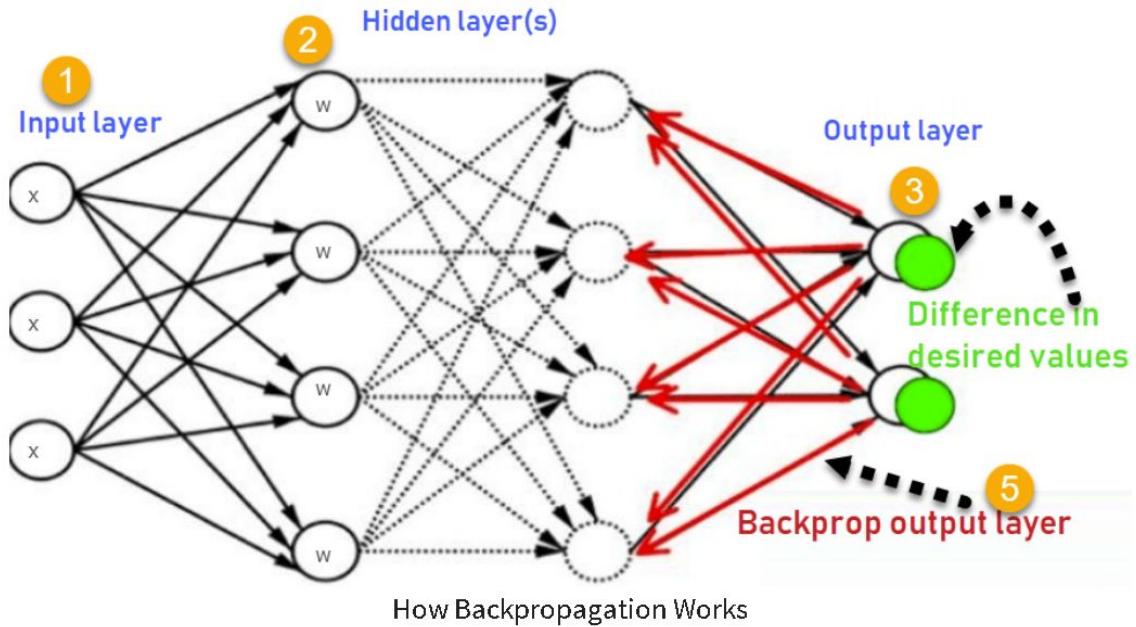


Figure 14[37]

## Apache

Apache is an open-source and free web server software that powers around 40% of websites around the world. The official name is Apache HTTP Server, and it's maintained and developed by the Apache Software Foundation.

It allows website owners to serve content on the web — hence the name “web server.” It’s one of the oldest and most reliable web servers, with the first version released more than 20 years ago, in 1995.

When someone wants to visit a website, they enter a domain name into the address bar of their browser. Then, the web server delivers the requested files by acting as a virtual delivery man.

Here at Hostinger, our web hosting infrastructure uses Apache in parallel with NGINX, which is another popular web server software. This particular setup allows us to get the best of both worlds. It greatly improves server performance by compensating the weaker sides of one software with the strengths of another.[38]

## AWS AND AWS Lightsail

Amazon Web Services (AWS) is a cloud computing platform that offers subscription and pay-as-you-use services to companies, individuals and other entities for virtualized computing, storage, application deployment, analytics, machine learning, IoT and a whole lot more.

What is Amazon Lightsail? It's a cloud service offered by AWS, targeting new or inexperienced cloud users. It includes a static IP address, a management console, secure shell terminal access, key management, domain name server (DNS) management, and virtualized server monitoring.[40]

In this post, we will learn about the significance of Amazon Lightsail and its features, benefits, and costs. However, before we learn about Amazon Lightsail, let's gain an understanding of virtual private servers (VPS), which represent one component of Lightsail.[40]

## The Difference Between Amazon Lightsail and EC2

The two solutions offered by AWS are Amazon Lightsail and Elastic Compute Cloud (EC2). So, let's compare Lightsail and EC2.

Amazon Lightsail	Amazon EC2
1. Used for simple web apps	1. Used for enterprise web apps
2. Used for websites including WordPress, custom sites, and e-commerce	2. Used for Big data, HPC, and analytics workloads
3. Single configurable VPS	3. Works on multiple VMs, migrations from an on-premises environment, including BYOL
4. Used for prototyping and test environments	4. Used for enterprise app development and deployments

[40]

## Erosion and Dilation in OpenCV

Morphological operations are a set of operations that process images based on shapes. They apply a structuring element to an input image and generate an output image.

The most basic morphological operations are two: Erosion and Dilation

**Basics of Erosion:**

- Erodes away the boundaries of foreground object
- Used to diminish the features of an image.

**Working of erosion:**

1. A kernel(a matrix of odd size(3,5,7) is convolved with the image.
2. A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is eroded (made to zero).
3. Thus all the pixels near boundary will be discarded depending upon the size of kernel.
4. So the thickness or size of the foreground object decreases or simply white region decreases in the image.

**Basics of dilation:**

- Increases the object area
- Used to accentuate features

**Working of dilation:**

1. A kernel(a matrix of odd size(3,5,7) is convolved with the image
2. A pixel element in the original image is '1' if atleast one pixel under the kernel is '1'.
3. It increases the white region in the image or size of foreground object increases[41]

## **Keras**

Keras was created to be user friendly, modular, easy to extend, and to work with Python. The API was “designed for human beings, not machines,” and “follows best practices for reducing cognitive load.”

Neural layers, cost functions, optimizers, initialization schemes, activation functions, and regularization schemes are all standalone modules that you can combine to create new models. New modules are simple to add, as new classes and functions. Models are defined in Python code, not separate model configuration files.

The biggest reasons to use Keras stem from its guiding principles, primarily the one about being user friendly. Beyond ease of learning and ease of model building, Keras offers the advantages of broad adoption, support for a wide range of production deployment options, integration with at least five back-end engines (TensorFlow, CNTK, Theano, MXNet, and PlaidML), and strong support for multiple GPUs and distributed training. Plus, Keras is backed by Google, Microsoft, Amazon, Apple, Nvidia, Uber, and others.[42]

## WSGI and uWSGI

WSGI: A Python spec that defines a standard interface for communication between an application or framework and an application/web server. This was created in order to simplify and standardize communication between these components for consistency and interchangeability. This basically defines an API interface that can be used over other protocols.

uWSGI: An application server container that aims to provide a full stack for developing and deploying web applications and services. The main component is an application server that can handle apps of different languages. It communicates with the application using the methods defined by the WSGI spec, and with other web servers over a variety of other protocols. This is the piece that translates requests from a conventional web server into a format that the application can process.

uwsgi: A fast, binary protocol implemented by the uWSGI server to communicate with a more full-featured web server. This is a wire protocol, not a transport protocol. It is the preferred way to speak to web servers that are proxying requests to uWSGI.

### WSGI Application Requirements

The WSGI spec defines the interface between the web server and application portions of the stack. In this context, “web server” refers to the uWSGI server, which is responsible for translating client requests to the application using the WSGI spec. This simplifies communication and creates loosely coupled components so that you can easily swap out either side without much trouble.

The web server (uWSGI) must have the ability to send requests to the application by triggering a defined “callable”. The callable is simply an entry point into the application where the web server can call a function with some parameters. The expected parameters are a dictionary of environmental variables and a callable provided by the web server (uWSGI) component.

In response, the application returns an iterable that will be used to generate the body of the client response. It will also call the web server component callable that it received as a parameter. The first parameter when triggering the web server callable will be the HTTP status code and the second will be a list of tuples, each of which define a response header and value to send back to the client.

With the “web server” component of this interaction provided by uWSGI in this instance, we will only need to make sure our applications have the qualities described above. We will also set up Nginx to handle actual client requests and proxy them to the uWSGI server.[43]

## 2.5. Existing Final Year Projects

## **Project 1**

Title: Euro Coin Classification Using Image Processing & Machine Learning

Student: Yumin Chen

Description (brief):

This project is through a suitable mathematical model to recognize the denomination of euro coin.

The main technologies involved in this project are image processing and machine learning. For Image Processing, computer vision techniques are used to process the image-based sample dataset and extract features. Machine Learning is used for predictive data analytics to build the models of generalized euro coin denominations.

This project allows people to easily calculate a large amount of money.

What is complex in this project

Visual object recognition is one of the most challenging computational problems in machine vision. Humans can easily recognize any euro coin, but a machine cannot so have to create an artificial recognition system.

What technical architecture was used?

JSON language, image processing techniques, machine learning techniques, data mining techniques, computer vision techniques, statistical techniques

Explain the key strengths and weaknesses of this project, as you see it.

The strength of this project is allowed the user easy to calculate the money

The weakness of this project is this application must scan each coin sometime is not necessary.

## **Project 2**

Title: Image Selection Based on Optimal Characteristic Analysis

Student: Jameel Briones

Description (brief):

This project is through comparing Basic image properties such as sharpness, noise level, exposure, and contrast will be analysed to test for the image's quality to find a similar image then category. Users can share the image to social media sites.

It helps the user to tidy up images.

What is complex in this project?

This project must research and implement a good image comparison algorithm that can be used to compare the image's quality.

Have to assessment measure for different image properties that can affect image quality

What technical architecture was used?

Python language, OpenCV, image processing technical

Explain the key strengths and weaknesses of this project, as you see it.

The strength of this project lies in the image quality assessment. The tests performed have produced a higher success rate than the image comparison, often matching human's assessment of image quality. The image assessment is also made more efficient due to multithreading. It also provides a share functionality in the application.

project's weaknesses lie in the image comparison. It may be accurate in a few samples, but it can also have a few outliers resulting from its brute force matching of its descriptors. Because of this, it often leads to an inaccurate match of the images. It can also be quite slow at times, depending on the image size and quantity to be compared.

## 2.6. Conclusions

With the necessary gained knowledge of the space checker and its associated problems, the development stages of the app can begin with this knowledge in mind.

From the research of similar applications, technologies, and similar projects, we have a good understanding of how to design the application.

The technologies best suited for the project were decided after researching many different options.

## Requirements Table

Name	Description	Priority
User Login	Enables a user to log in on a website	HIGH
User Logout	Enables a user to logout of the website	LOW
User create account	Enables user to create an account and enter relevant information about themselves	HIGH
User register	Allow user to register an account on the website	HIGH
Home page	Provide selecting of the car park and shows the number of free space and status of space.	HIGH
database	create a database for the application using PostgreSQL	HIGH
deploy website	using Django to deploy website shows login page and register page.	HIGH
space map	shows space id, status, and position on space map and reference substance for the car park.	HIGH
Select park	user can select the different car park	MEDIUM
show total space	show the total number of car space on the home page	HIGH
show car exist	show the total number of cars exist on home page	HIGH

process video	to process video show status of car spaces	HIGH
---------------	--	------

## 3. Design

### 3.1 Introduction

Following on from the previous chapter, where some of the key background research was presented. The first section will look at the software methodologies employed in this project which describes which methodology was chosen and why. After that, some example use-cases and user personas will be presented. The next section outlines the technical architecture of the system and will discuss in depth how the system architecture is planned to work. This will cover both front-end and back-end aspects of the system. The final section will discuss the plan for testing and evaluation of the system.

### 3.2. Software Methodology

Waterfall relies on teams following a sequence of steps and never moving forward until the previous phase has been completed. The methodology, in its traditional form, leaves almost no room for unexpected changes or revisions.

One of the defining steps of Waterfall is committing to an end product, goal, or deliverable at the beginning, and teams should avoid deviating from that commitment. For small projects where goals are clear, this step makes your team aware of the overall goal from the beginning, with less potential for getting lost in the details as the project moves forward.

Waterfall is based entirely on following a set of steps that keep teams always moving forward. The methodology, in its traditional form, leaves almost no room for unexpected changes or revisions. If your team has loyally followed the steps of Waterfall nearly to the end of the project but then faces an unplanned roadblock that necessitates a change in scope or goals, pivoting won't be easy. You'll have put a considerable amount of work into a project under very specific, rigid assumptions. A sudden change to the parameters of the project could render much of the work you've carried out up to that point useless, which can throw off the entire timeline.[11]

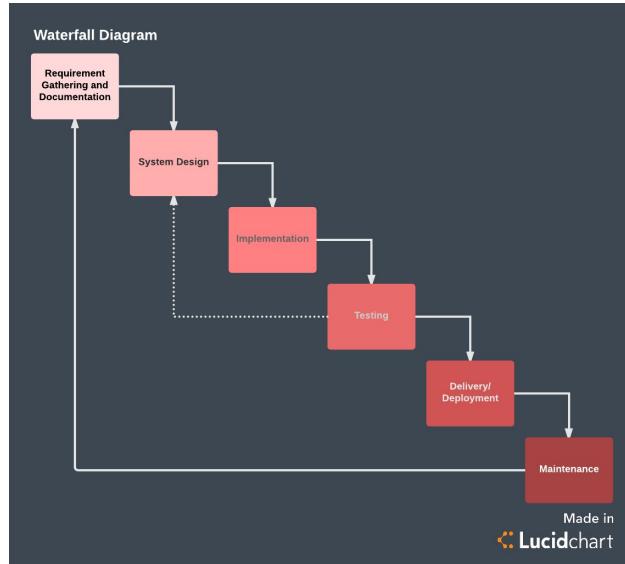


Figure 15(11)

Unlike the more traditional waterfall model, which focuses on a stringent step-by-step process of development stages, the iterative model is best thought of as a cyclical process. After an initial planning phase, a small handful of stages are repeated over and over, with each completion of the cycle incrementally improving and iterating on the software. Enhancements can quickly be recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last.

- Planning & Requirements: As with most any development project, the first step is go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.
- Analysis & Design: Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc) that will be utilized in order to meet the needs of the analysis stage.
- Implementation: With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.
- Testing: Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up.
- Evaluation: Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on.[33]

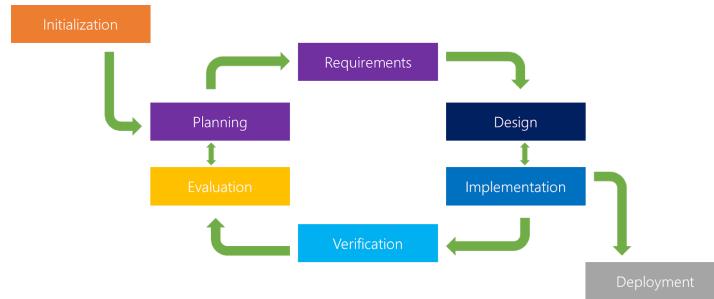


Figure 16(33)

Our project must have many tests and changes, so we can not use waterfall methodology. As results we decide use Iterative methodology to implement our project.

### 3.3. Overview of System

The model-view-controller pattern approach will be used where a feature will be planned.

MVC Stands for Model View Controller and its software architecture design pattern. The main goal of this architecture is to separate functionality, logic, and the interface of an application to promote organized programming.

The Model is responsible for getting and manipulating the data, so it's basically the brain of the application.

the view is the user interface, it's what the user sees and how they interact with the application.

The controller acts as a kind of a middleman between the model and the view. The controller will ask the model to get some data from a database and then the controller will take that data and load a view and pass that data into it.[12]

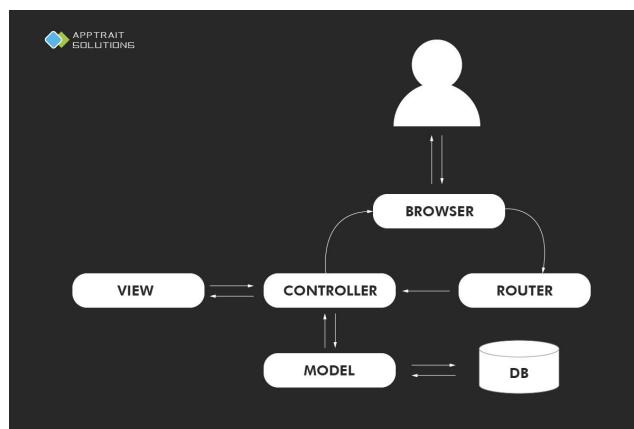
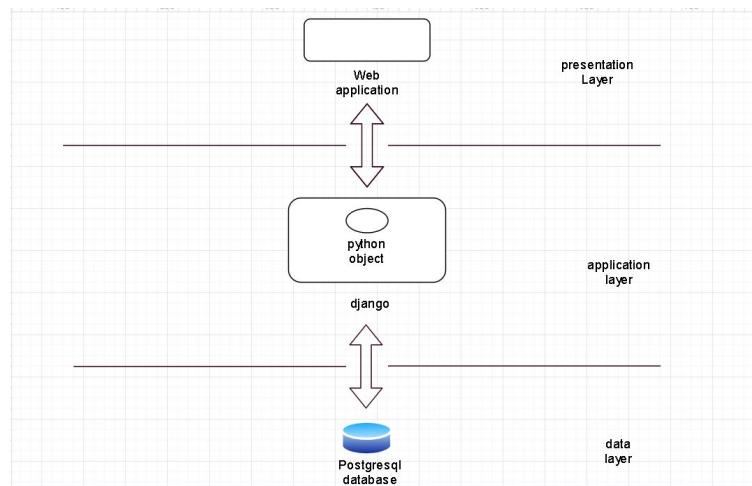


Figure 17(12)

In Django, we use Model-Template-View (MTV) pattern is like MVC pattern, in which Django views are called templates and controllers are called views. Hence our HTML code will be templates and Python code will be in views and models. The template is the view from MVC, View is controller from MVC.

The technical architecture shows how many layers there are in the application and how the layers communicate with each other. For this project, a 3-Tier model will be used as seen below. This model was chosen as changes to one layer of the model should not affect the other layers.



18

### 3.4. Feature List

#### Login webpage:

- User login
- Condition presentation.
- Register page transform button

#### Register webpage:

- User register
- Condition presentation.
- Login page transform button

#### Car park video page:

- status of space displayed

- number of total space displayed
- number of free space displayed
- space id displayed
- location page transform button
- feedback page transform button

Feedback webpage:

- Feedback Update

Transform webpage:

- Car park page transform button
- Feedback page transform button

### 3.5. Use Case Diagram

Use case diagrams are used to identify system functionality and communicate system behaviour. The use case diagrams in the below figures show the progression of the system functionality.

#### 1st Iteration Use Case Diagram

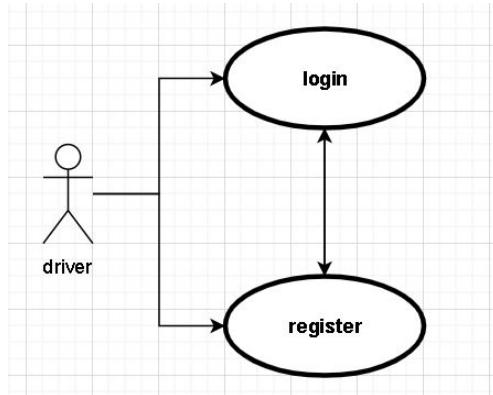


Figure19

## 2nd Iteration Use Case Diagram

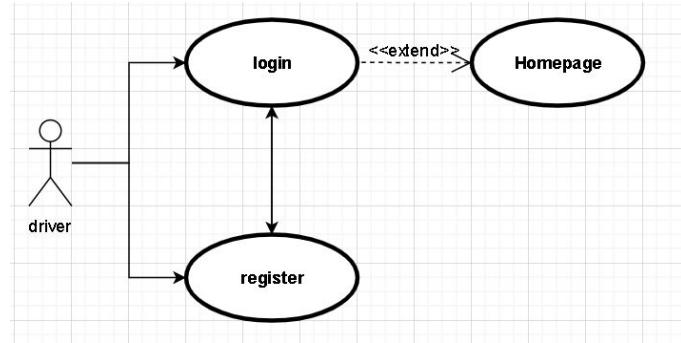


Figure 20

## 3rd Iteration Use Case Diagram

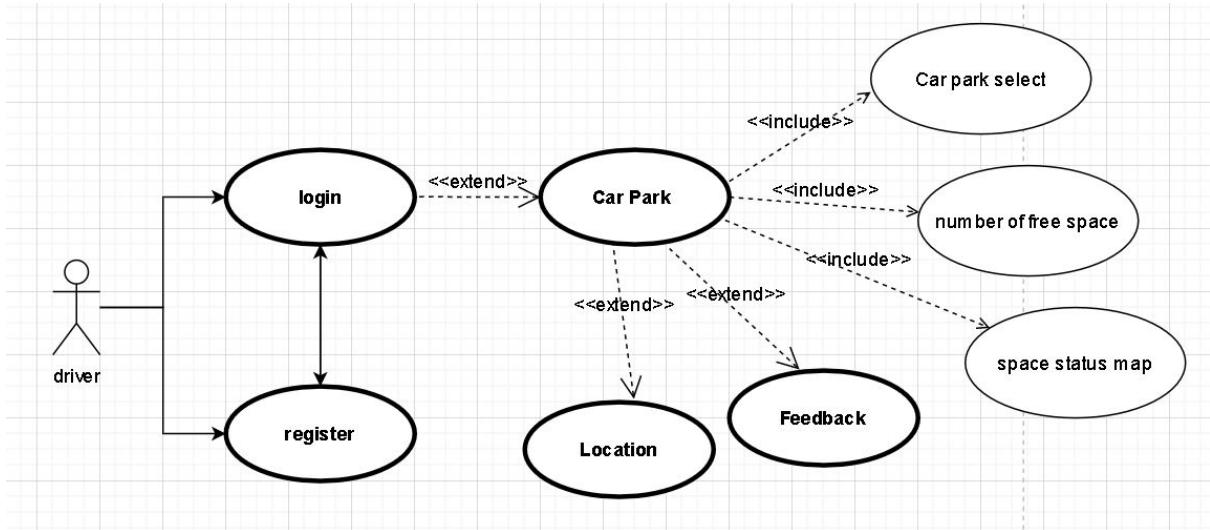


Figure 21

## 3.6. Front-End

Front end development is mostly focused on what some may coin the "client-side" of development. Front end developers will be engaged in analysing code, design, and debugging applications along with ensuring a seamless user experience. You manage what people first see in their browsers. As a front-end developer, you are responsible for the look, feel and ultimately the design of the site.[14]

## Web application Architecture

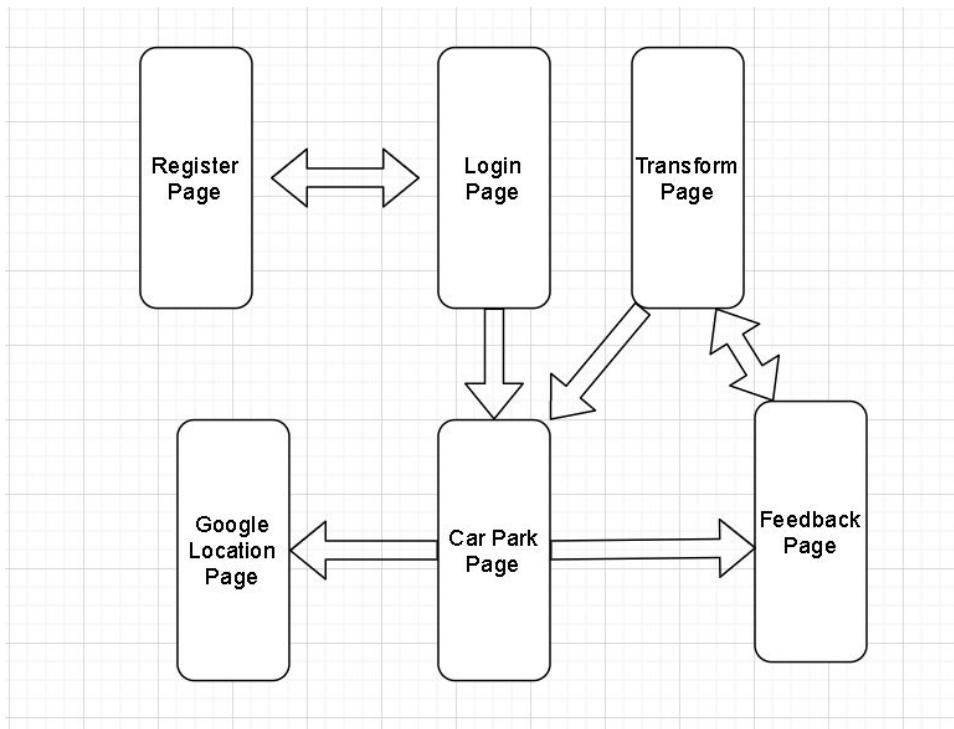


Figure22

## Medium-fidelity Design.

### Register Page

---

Username:

Password:

>Password Confirm:

---

---

Figure 23

register page has 2 buttons: register and logs in. Click the register button able to tell user registration is successful or false. Click the login button will go to the login page.

## Register Page

['Please enter right password']

Register

Login

Figure 24

enter a different password between passwords and passwords confirm.

## Login Page

Username:

Login

Password:

Register

Figure 25

login page

# Login Page

This user not exist, Please try again

Login

Register

Figure 26

if the username or password wrong.

# EPL Parking

Location

Feedback

Figure 27

if login success will go to the homepage, and it should be contained a car park video shows status, space id, number of total space and free space.

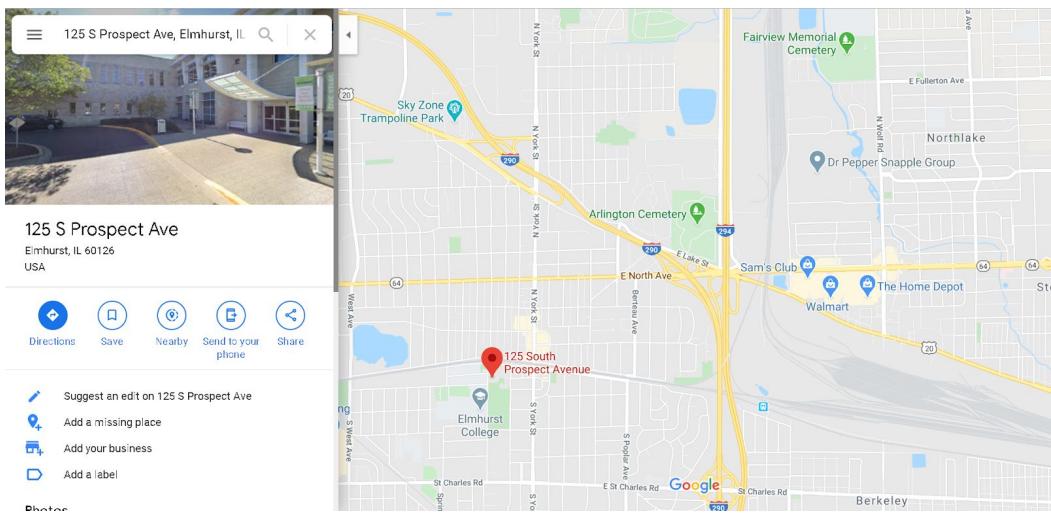


Figure 28

Click the Location button directly to park location in the google map.

## Feedback

**Update**

Figure 29

Feedback page allow user to update feedback.



*Figure 30*

If feedback update successful will tell the user update successful then direct to transform page.

### 3.7. Middle-Tier

In most three-tier web database systems, most of the application logic is in the middle tier. The client tier presents data to and collects data from the user; the database tier stores and retrieves the data. The middle tier serves most of the remaining roles that bring together the other tiers: it drives the structure and content of the data displayed to the user, and it processes input from the user as it is formed into queries on the database to read or write data. It also adds state management to the HTTP protocol. The middle-tier application logic integrates the Web with the database management system [15].

our middle tier is Django, Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. [16]

The goal of Django is to allow developers to instead of implementing the same solutions repeatedly, focus on the parts of their application that are new and unique to their project. In fact, Django is much more fully-featured than many other frameworks out there. It takes care of a lot of the hassle of Web development, letting you focus on writing your application without any need to reinvent the wheel. It's free and open source. Additionally, the Django framework enables you to model your domain and code classes, and before you know it, you already have an ORM. Let's take a closer look to understand its acclaim better.[16]

we use PyCharm to design code.

### 3.8. Back-End

The Data Layer comprises the data storage system and the data access layer. This is the DBMS layer and is generally accessed through the middle layer applications.

we use PostgreSQL as our data layer.

PostgreSQL is an object-relational database management system (ORDBMS). It is an open-source system that is known for its close alignment with the SQL standard.[10]

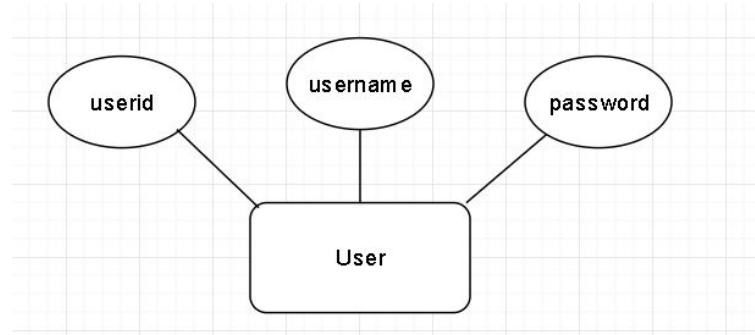


Figure 31

1st Iteration ERD

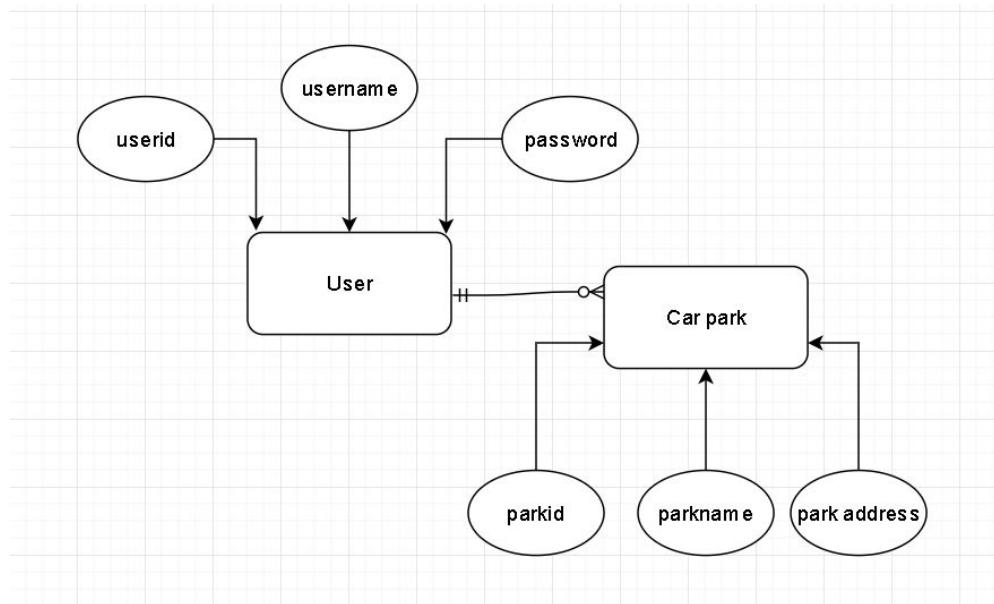


Figure 32

2nd Iteration ERD

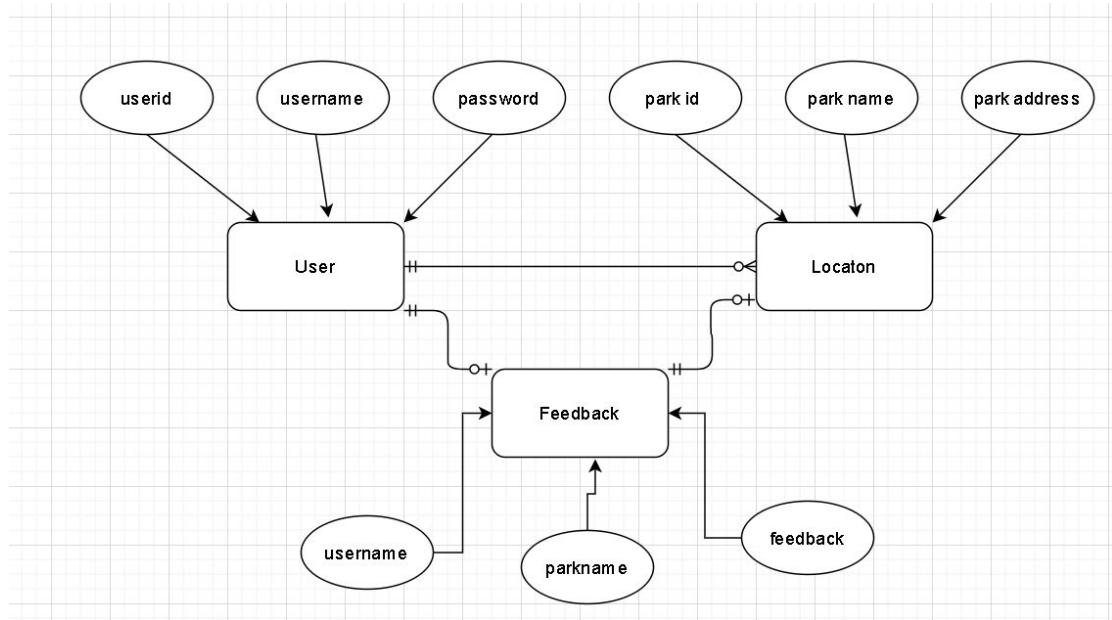


Figure 33

3rd Iteration ERD

### 3.7. Conclusions

In this chapter we looked at the design of the system, first exploring the methodology that will be used in the development process, next a broad overview of the technical architecture was outlined, then into detail the front and back-end design. Finally, testing was discussed regarding the different testing methods that will be followed and the software test plan.

Based on the key themes discussed in this chapter, the next chapter will cover the development process and will be revisiting many of the same issues covered here. The development chapter will discuss how these designs were implemented including any challenges or changes encountered along the way.

## 4. Project Development

### 4.1. Introduction

This chapter continues with the issues explored in the previous chapter and will outline the development process undertaken in this project. This chapter will present the key development processes and the challenges encountered during the creation of this system.

### 4.2. Project Development

The first step for the development of the application was to set up version control.

We decided to use GitHub to implement version control because GitHub can easily integrate Django or Pycharm, and we can use each change of commit as a version. Git is the most commonly used version control system today and is quickly becoming the standard for version control. Git is a distributed version control system, meaning your local copy of code is a complete version control repository.[18]

For prototype development, we have to process each frame of car park video, we have to draw red and green color rectangle, number of total space and free space on each frame of video, then we use django to create a login page, register page, car park page and feedback page, then create the database for login and register after using PostgreSQL.

### 4.3. Image Processing

We use image processing method to draw red rectangle, green rectangle, number of total space and free space on image as below:

## **1.extract empty space image, train image and test image from video:**

```
#crop a frame from video.
video_name = 'cars.mp4'
cap = cv2.VideoCapture(video_name)
ret = True
count = 0

while ret:
    ret, img = cap.read()
    img = cv2.resize(img, (960, 540))
    cv2.imshow('frame', img)
    cv2.imwrite('frame/test2.jpg', img)
    count = count+1
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    #out.write(image)

cv2.destroyAllWindows()
cap.release()
```

Figure 34

use cv2.videocapture function to load video, then use while loop to show each frame of video, final use cv2.imwrite to output images.

## **2. shows image on the screen**

```
#read a empty space image from car park then change bgr channel to rgb
empty = cv2.imread("frame/extract.jpg")
newl = empty.copy()
empty = cv2.cvtColor(empty, cv2.COLOR_BGR2RGB)
plt.figure(figsize=(10, 10))
plt.imshow(empty)
plt.show()
```

Figure 35

when we use plt to show image have to change color space from BRG to RGB.



*Figure 36*

### 3. find color range of yellow lines

```

cv2.namedWindow('image_win')

value = (0, 0, 0)
def update(x):
    global value
    r_value = cv2.getTrackbarPos('R', 'image_win')
    g_value = cv2.getTrackbarPos('G', 'image_win')
    b_value = cv2.getTrackbarPos('B', 'image_win')
    r_value1 = cv2.getTrackbarPos('R1','image_win')
    g_value1 = cv2.getTrackbarPos('G1','image_win')
    b_value1 = cv2.getTrackbarPos('B1','image_win')
    hsv = cv2.cvtColor(empty, cv2.COLOR_BGR2HSV) #cover color space from bgr to hsv
    lower_white = np.array([r_value,g_value,b_value])
    upper_white = np.array([r_value1,g_value1,b_value1])
    wmask = cv2.inRange(empty, lower_white, upper_white)
    cv2.imshow("ss",wmask)

cv2.createTrackbar('R','image_win',0,255,update)
cv2.createTrackbar('G','image_win',0,255,update)
cv2.createTrackbar('B','image_win',0,255,update)
cv2.createTrackbar('R1','image_win',0,255,update)
cv2.createTrackbar('G1','image_win',0,255,update)
cv2.createTrackbar('B1','image_win',0,255,update)

cv2.setTrackbarPos('R','image_win',165)
cv2.setTrackbarPos('G','image_win',172)

```

*Figure 37*

use trackbars to find a suitable color range of yellow lines.

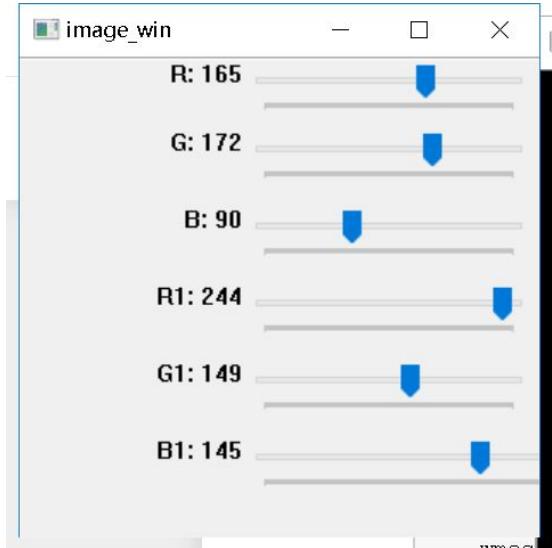


Figure 38

#### 4. show mask of yellow lines from image

```
# image is expected be in RGB color space, then find yellow mask
def yellow_mask(image):
    #hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV) #cover color space from bgr to hsv
    lower_yellow = np.array([220, 190, 44])
    upper_yellow = np.array([246, 255, 102])
    wmask = cv2.inRange(image, lower_yellow, upper_yellow)
    #nmask = cv2.bitwise_not(wmask)
    kernel = np.ones((3,3), np.uint8)
    erosion_img = cv2.erode(wmask, kernel, iterations=1) #erosion
    dilate_img = cv2.dilate(wmask, kernel, iterations=1) #dilation

    return dilate_img
```

Figure 39

use an inrange function to find specified color and do erosion and dilation.

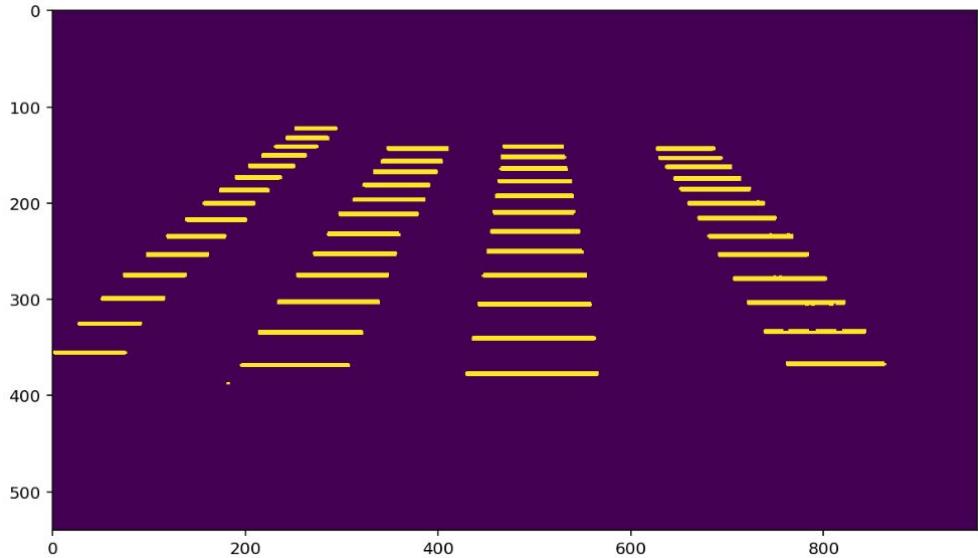


Figure 40

## 5. draw lines on the image

```
#detect each line from yellow mask
hlp= cv2.HoughLinesP(mask, rho=0.1, theta=np.pi/10, threshold=20, minLineLength=35, maxLineGap=0.1)
#seperate park space by column
parkA = []
parkB = []
parkC = []
parkD = []
#becasue hough line function is not sort each line, so I have to sort line by y-axis
#draw line in the copy of original image
def draw_lines(image, lines, color=[255, 0, 0], thickness=1):
    # the lines returned by cv2.HoughLinesP has the shape (-1, 1, 4)
    height, width, channels = image.shape
    part = width /4
    for line in lines:
        for x1,y1,x2,y2 in line:
            if x2> part*2.5 and x2 < width:
                parkA.append((x1,y1,x2,y2))
                cv2.line(image, (x1, y1), (x2, y2), color, thickness)
            elif x2> part*2 and x2< part*2.5:
                parkB.append((x1,y1,x2,y2))
                cv2.line(image, (x1, y1), (x2, y2), color, thickness)
            elif x2 > part * 1.25 and x2 < part *2:
                parkC.append((x1,y1,x2,y2))
                cv2.line(image, (x1, y1), (x2, y2), color, thickness)
            else:
                parkD.append((x1,y1,x2,y2))
                cv2.line(image, (x1, y1), (x2, y2), color, thickness)
```

Figure 41

first use the houghlines function to output the position of each line, then separate all lines into 4 groups to sort lines by x-axis.

No lines detected: 190

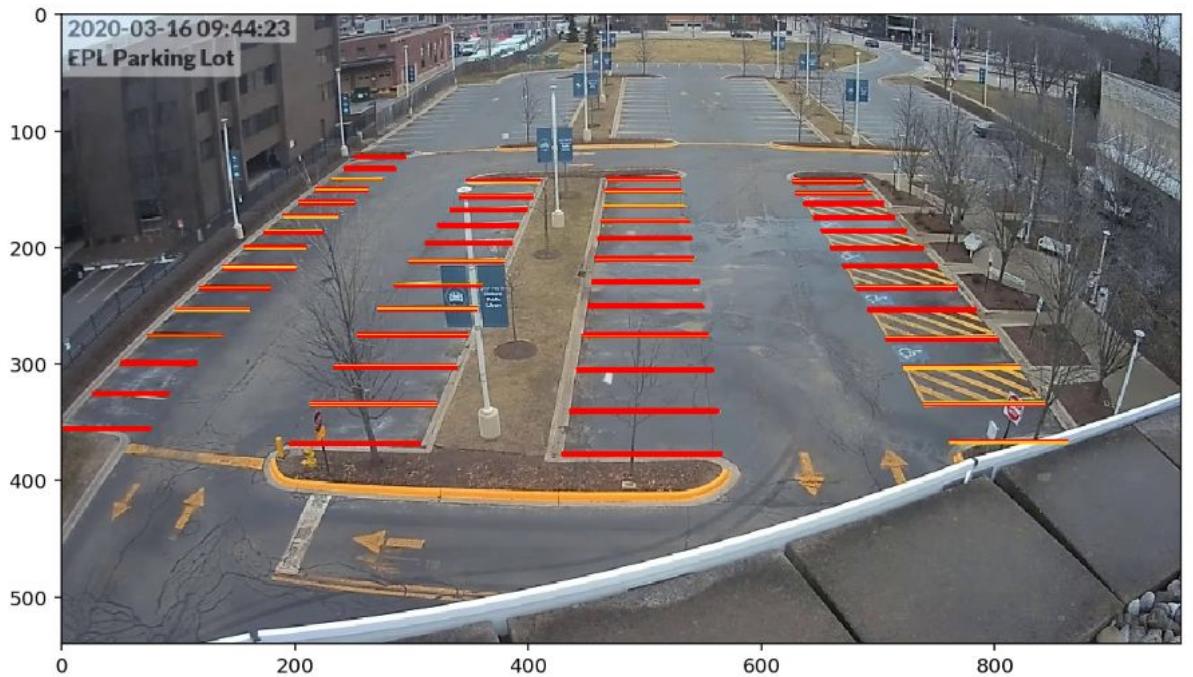


Figure 42

## 6. delete extra lines

```
#use to select specified value
def take(elem):
    return elem[1]
#delete extra line which detect by hough line function
def delclose(park, valuec):
    i = 0
    #sort by specified value
    park.sort(key = take, reverse = True)
    #delete extra line by very close distance
    while i < len(park)-1:
        y1 = park[i][1]
        y2 = park[i+1][3]
        if abs(y2 - y1)<=valuec:
            del park[i]
            print("delete"+ str(i))
        i = i +1
    i = 0
    #use loop make sure it is cleaned
    while i<4:
        delclose(parkD, 5)
        delclose(parkC, 5)
        delclose(parkB, 5)
        delclose(parkA, 5)
        i +=1
```

Figure 43

we detect 190 lines on the image, that we have to delete extra lines, we used sort function to sort each line of each group by y-axis, then find distance of both lines, if distance less than 5, then we delete this line. use a loop to make sure to delete all extra lines.

## 7. draw rectangle on the image

```
#create a empty array list to store all rectangle of car space
all_space = []
#draw all rectangle of car space out
def drawrect(park, miniarea, maxarea, image):
    rect=[]
    i = 0
    #sort each line by y
    park.sort(key = take, reverse = True)

    #use loop to link 2 line to be a rectangle
    while i < len(park)-1:

        x1 = park[i][0]
        y1 = park[i][1]
        x2 = park[i+1][2]
        y2 = park[i+1][3]
        leng = abs(x2 - x1)
        wid = abs(y2-y1)
        area = leng*wid
        #only output specified car space
        if area>miniarea and area<maxarea:
            cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 1)
            rect.append((x1, y1, x2, y2))

        i = i +1
    return rect
```

Figure 44

we can according to the position of lines to draw a rectangle, then exclude the rectangle which is not in the range of size. finally save all rectangle into an array list.



Figure 45

#### 8. crop each rectangle of image as a train image

```

trainimg = cv2.imread("frame/test2.jpg")
trainimg = cv2.resize(trainimg, (960, 540))
# create a array list to save space id
sid = []
# to output each car space use create machine learning model
def save_image_cnn(spot, image, part):
    spot_id = 0
    for x1, y1, x2, y2 in spot:
        spot_img = image[y2:y1, x1:x2]
        spot_img = cv2.resize(spot_img, (144, 48))
        spot_id +=1
        sid.append(part+str(spot_id))
        cv2.imwrite('cnnmodel/' +part +str(spot_id)+'.jpg', spot_img)

save_image_cnn(pa, trainimg, "A")
save_image_cnn(pb, trainimg, "B")

save_image_cnn(pc, trainimg, "C")
save_image_cnn(pd, trainimg, "D")

```

Figure 46

create space id for each rectangle and change all rectangles to the same size, use imwrite function to output each rectangle as image.

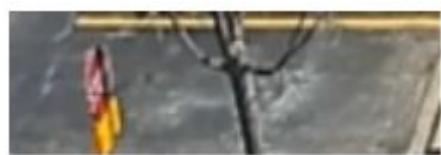


Figure 47

9. tidy up the train images into different label of folders.

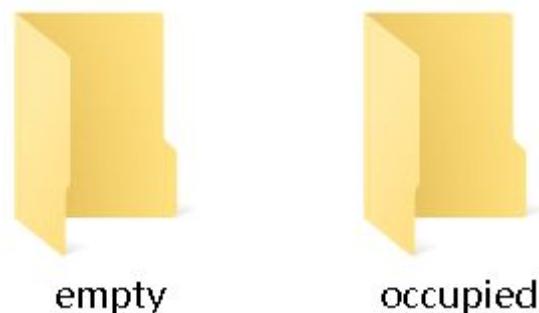


Figure 48

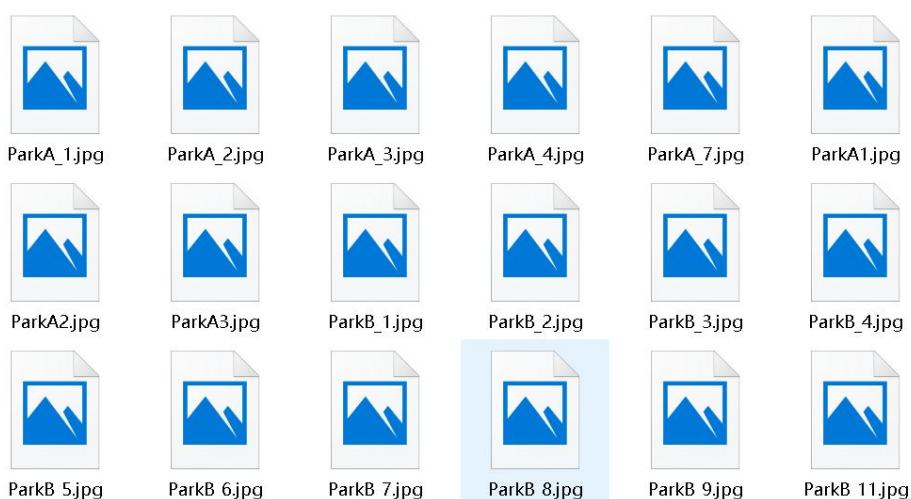


Figure 49

#### 10. according results of vgg model to predict label of each rectangle

```
#this function use to predictions image belong to empty or occupied
def make_prediction(image):
    #Rescale image
    img = image/255.

    #Convert to a 4D tensor
    image = np.expand_dims(img, axis=0)
    #print(image.shape)

    # make predictions on the preloaded model
    class_predicted = model.predict(image)
    inID = np.argmax(class_predicted[0])
    label = class_dictionary[inID]
    return label
```

Figure 50

use model.predict function to predict the label of each car space is empty or occupied.

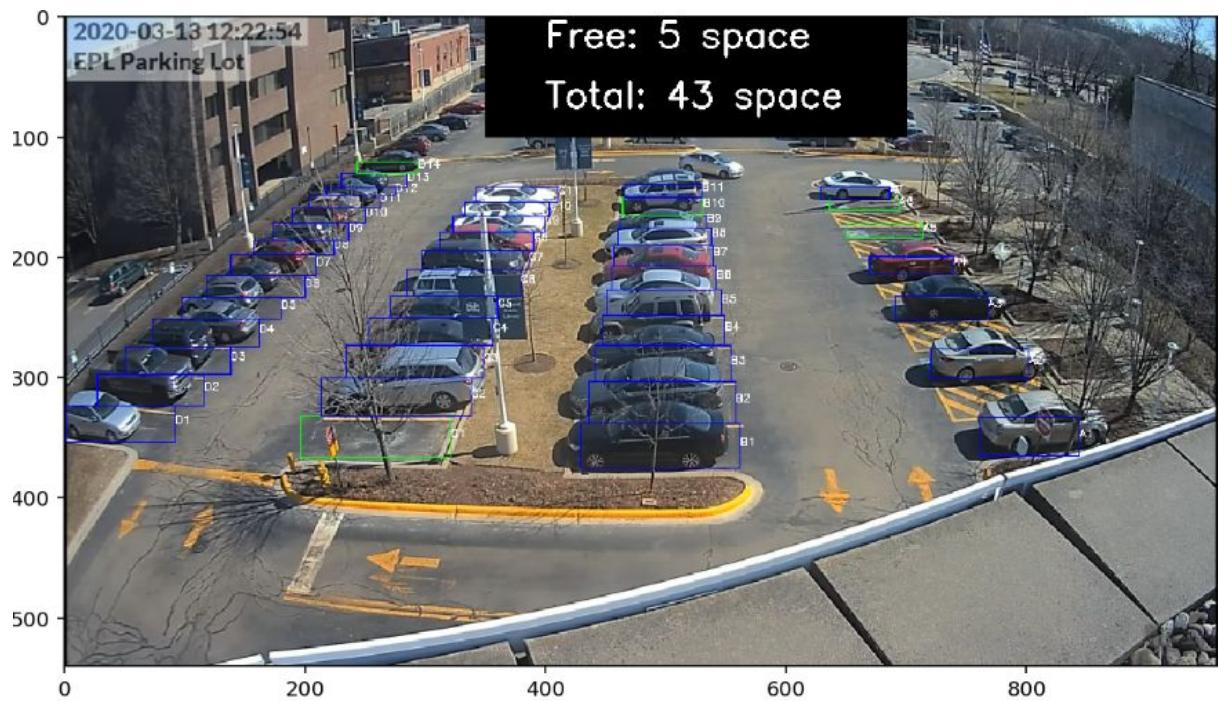
**11. draw blue rectangle with occupied space and green rectangle with empty space, then show id of each space, number of total space and free space.**

```
trainimg = cv2.imread("frame/test.jpg")
trainimg = cv2.resize(trainimg, (960, 540))
trainimg = cv2.cvtColor(trainimg, cv2.COLOR_BGR2RGB)
#use to predictions all space belong to
def predict_on_image(image):
    #init number of all space and free space
    all_count = 0
    free_space = 0
    for x1, y1, x2, y2 in all_space:

        (x1, y1, x2, y2) = (int(x1), int(y1), int(x2), int(y2))
        #set space id beside car space
        cv2.putText(image, "%s" %sid[all_count], (x2, int(y2+(y1-y2)/2)),
                   cv2.FONT_HERSHEY_SIMPLEX,
                   0.3, (255, 255, 255), 1)
        all_count += 1
        #crop this image
        spot_img = image[y2:y1, x1:x2]
        spot_img = cv2.resize(spot_img, (144, 48))
        #draw all space rectangle in red
        cv2.rectangle(image, (x1, y1), (x2, y2), (0, 0, 255), 1)
        label = make_prediction(spot_img)
        #if space is free then draw this space in green
        if label =='empty':
            free_space += 1
            cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 1)
        #set number of free space and all space on the image
        cv2.rectangle(image, (350, 0), (700, 100), (0, 0, 0), -1)
        cv2.putText(image, "Free: %d space" %free_space, (400, 25),
                   cv2.FONT_HERSHEY_SIMPLEX,
                   1, (255, 255, 255), 2)
        cv2.putText(image, "Total: %d space" %all_count, (400, 75),
                   cv2.FONT_HERSHEY_SIMPLEX,
                   1, (255, 255, 255), 2)
    return image
```

Figure 51

use puttext function to show space id, number of total space and free space, then use make\_prediction function to predict status of space.



*Figure 52*

12. use same method which is processed image to process each frame of video

```
video_name = 'cars.mp4'
cap = cv2.VideoCapture(video_name)
ret = True
count = 0
#init output video
fourcc = cv2.VideoWriter_fourcc(*'MP4V')
fourcc = cv2.VideoWriter_fourcc(*'H264')
out = cv2.VideoWriter('output.mp4', fourcc, 12.83, (960, 540))
#to process each frame of video
while ret:
    ret, image = cap.read()
    if ret == False:
        break
    image = cv2.resize(image, (960, 540))
    new_image= predict_on_image(image)
    out.write(image)
    cv2.imshow('frame', new_image)
    if cv2.waitKey(60) & 0xFF == ord('q'):
        break

out.release()
cv2.destroyAllWindows()
cap.release()
```

*Figure 53*

can not shows fluency video, have lag with video, because computer have to spend time to process each frame of video, as results we can only save processed image as a new video.

#### 4.4. Machine Learning

KNN is a classification algorithm which hardly does any learning. So getting poor results is quite obvious. Its efficiency depends on the features you're considering from your dataset. If you have a full-furnished datasets with good features, then your KNN will give better results.

Neural networks are being used for classification purposes since a very long time. They provide benchmark results for the given problem. Its not ours but NN's work to collect features from the input data (image).

You make learning by making changes to the weights and biases associated with each node in the network. Backpropagation with SGD can be used to adjust the weights. Once the weights are adjusted, you just need to perform feed-forward in the network to get the desired result. (In KNN for all the datapoints, you were going through all the previously recorded points for finding k nearest points. This is also one of the reason why KNN performs slow)[39]

in VGG use a very small receptive field (3x3, the smallest possible size that still captures left/right and up/down). There are also 1x1 convolution filters which act as a linear transformation of the input, which is followed by a ReLU unit. The convolution stride is fixed to 1 pixel so that the spatial resolution is preserved after convolution.

so we choose VGG as our project training model.

##### 1. load train and test file

```
cwd = os.getcwd()
folder = 'cnnmodel/train'
for sub_folder in os.listdir(folder):
    path, dirs, files = next(os.walk(os.path.join(folder, sub_folder)))
    files_train += len(files)

folder = 'cnnmodel/test'
for sub_folder in os.listdir(folder):
    path, dirs, files = next(os.walk(os.path.join(folder, sub_folder)))
    files_validation += len(files)
```

*Figure 54*

init test and train folder and count number of image in the folder

## 2. set parameters for vgg model

```
#Build model on top of a trained VGG
model = applications.VGG16(weights = "imagenet", include_top=False, input_shape = (img_width, img_height, 3))
# Freeze the layers which you don't want to train. Here I am freezing the first 5 layers.
for layer in model.layers[:10]:
    layer.trainable = False

x = model.output
x = Flatten()(x)

predictions = Dense(num_classes, activation="softmax")(x)

# creating the final model
model_final = Model(input = model.input, output = predictions)

# compile the model
model_final.compile(loss = "categorical_crossentropy",
                     optimizer = optimizers.SGD(lr=0.0001, momentum=0.9),
                     metrics=["accuracy"]) # See learning rate is very low
```

Figure 55

## 3. Initiate the train and test generators with data Augmentation

```
train_datagen = ImageDataGenerator(
rescale = 1./255,
horizontal_flip = True,
fill_mode = "nearest",
zoom_range = 0.1,
width_shift_range = 0.1,
height_shift_range=0.1,
rotation_range=5)

test_datagen = ImageDataGenerator(
rescale = 1./255,
horizontal_flip = True,
fill_mode = "nearest",
zoom_range = 0.1,
width_shift_range = 0.1,
height_shift_range=0.1,
rotation_range=5)

train_generator = train_datagen.flow_from_directory(
train_data_dir,
target_size = (img_width, img_height),
batch_size = batch_size,
class_mode = "categorical")
```

Figure 56

#### 4. save model according as h5 file and start train

```
checkpoint = ModelCheckpoint("car.h5", monitor='val_accuracy', verbose=1, save_best_only=True, save_weights_only=False,
                            mode='auto', period=1)
early = EarlyStopping(monitor='val_accuracy', min_delta=0, patience=10, verbose=1, mode='auto')

# Start training!
history_object = model_final.fit_generator(
    train_generator,
    samples_per_epoch = nb_train_samples,
    epochs = epochs,
    validation_data = validation_generator,
    nb_val_samples = nb_validation_samples,
    callbacks = [checkpoint, early])
```

Figure 57

We set epochs to 70 to increase accuracy.

```
Epoch 00000
4/4 [=====] - 10s 2s/step - loss: 0.0246 - accuracy: 1.0000 - val_loss: 0.0142 - val_accuracy: 0.9894

Epoch 00066: val_accuracy did not improve from 0.99312
Epoch 67/70
4/4 [=====] - 10s 2s/step - loss: 0.0302 - accuracy: 0.9897 - val_loss: 0.0146 - val_accuracy: 0.9903

Epoch 00067: val_accuracy did not improve from 0.99312
Epoch 68/70
4/4 [=====] - 10s 2s/step - loss: 0.0180 - accuracy: 1.0000 - val_loss: 0.0390 - val_accuracy: 0.9907

Epoch 00068: val_accuracy did not improve from 0.99312
Epoch 69/70
4/4 [=====] - 9s 2s/step - loss: 0.0195 - accuracy: 1.0000 - val_loss: 0.0189 - val_accuracy: 0.9907

Epoch 00069: val_accuracy did not improve from 0.99312
Epoch 70/70
4/4 [=====] - 10s 2s/step - loss: 0.0236 - accuracy: 0.9922 - val_loss: 0.0395 - val_accuracy: 0.9903

Epoch 00070: val_accuracy did not improve from 0.99312
```

Figure 58

## 5. plot diagram shows accuracy and loss

```
import matplotlib.pyplot as plt
print(history_object.history.keys())
plt.plot(history_object.history['accuracy'])
plt.plot(history_object.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

Figure 59

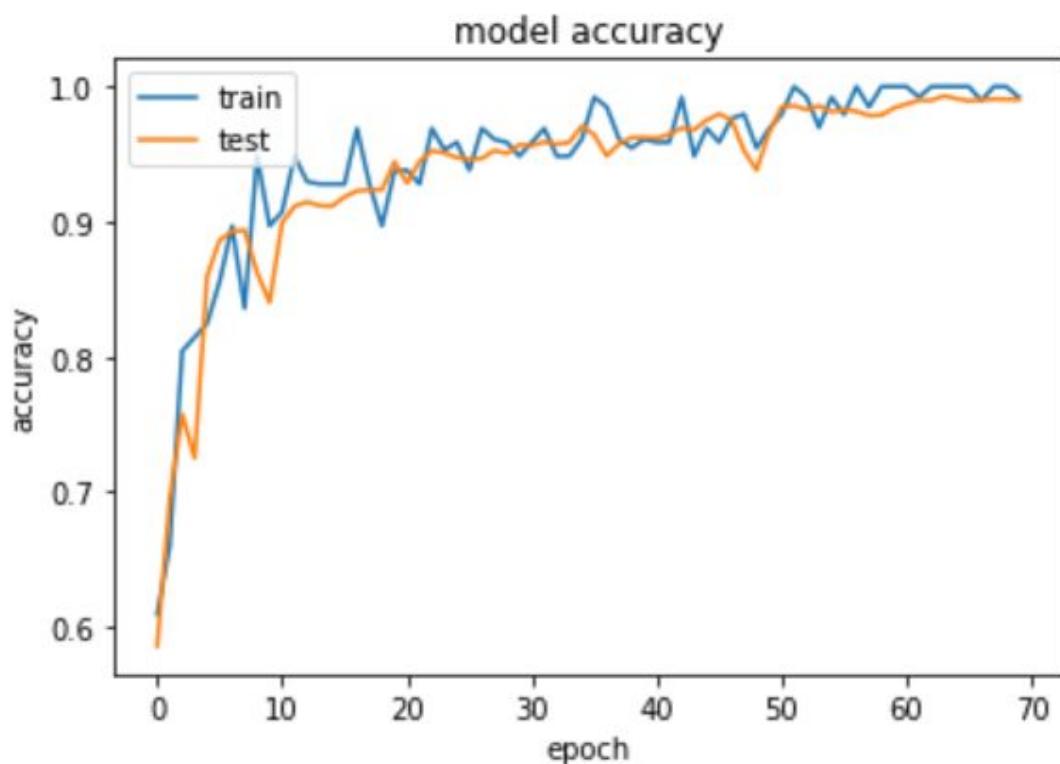


Figure 60

```
plt.plot(history_object.history['loss'])
plt.plot(history_object.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

Figure 61

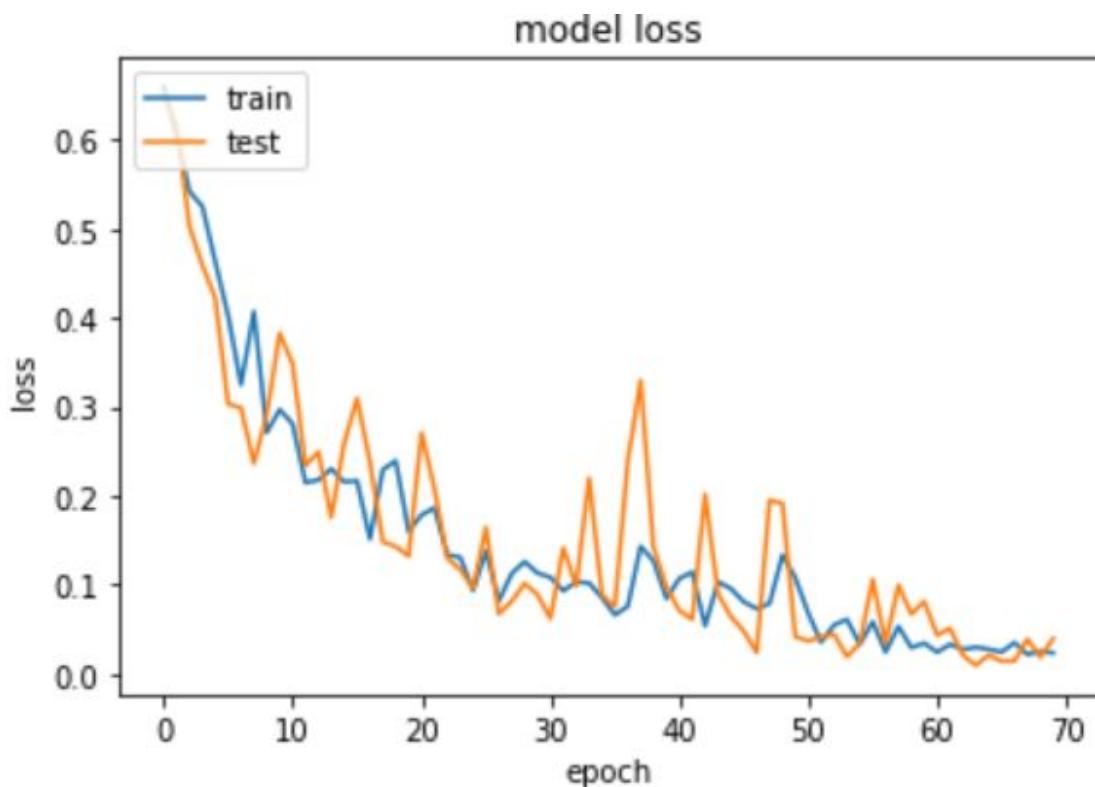


Figure 62

#### 4.5. Front-End

In this park, We are going to explain each template of the django project.

## Login page

```
><body>
<h1 align="center">Register Page</h1>
<form method = 'post' enctype="multipart/form-data" align="center">
{{uf.as_p}}
{{errors}}
{{succ}}
<br>
<input type="submit" value = "Register" />
</form>
<br>
<p align="center">
<button type="submit" onclick="location.href='{{ url 'login'}}'">
    Login
</button>
```

Figure 63

The screenshot shows a web page titled "Register Page". The page has a light blue background with a faint watermark of a hand holding a key. It contains three input fields: "Username:", "Password:", and "Password Confirm:". Below these fields is a green button labeled "Register". At the bottom of the page is another green button labeled "Login".

Figure 64

This is register html, the input form is set in view.py, there is a submit button use to submit form, here have a login button direct to login page.

### Register Page

```
<h1 align="center">Login Page</h1>
<form method = 'post' enctype="multipart/form-data" align="center">

{{uf.as_p}}
    {{ fail }} <br>
<input type="submit" value = "Login" />
</form>
<br>

<p align="center">
<button type="submit" onclick="location.href='{% url 'register'%}'">
    Register
</button>
```

Figure 65

The screenshot shows a web page titled "Login Page". It contains two input fields: "Username:" with the value "jianyu" and "Password:" with the value ".....". Below these fields is a green rectangular button labeled "Login". At the bottom of the page is another green rectangular button labeled "Register". The background of the page has a light blue gradient with faint white cloud-like shapes.

Figure 66

This is login html, the input form is set in view.py, there is a submit button use to submit form, here have a login button direct to register page.

## Car park Page

```
<body>
<h1 align="center">EPL Parking</h1 >
<table style="... ">
| <tr>
|   <th><button type="submit" onclick="location.href='http://maps.google.com/maps?f=q&hl=en&geoc'>
|     Location
|   </button></th>
|   <th><button type="submit" onclick="location.href='{% url 'feedback'%}'">
|     Feedback
|   </button></th>
| </tr>
| </table>
<video id="myVideo" width="960" height="540" muted autoplay="autoplay" loop = "loop" class="center">
  <source src="{% static "cdmlk.mp4" %}" type="video/mp4"/>
</video>
```

Figure 67



Figure 68

This is park html, It play a video shows status of car park, and set 2 buttons, one is direct to location of car park and other one direct to feedback page.

## Location Page

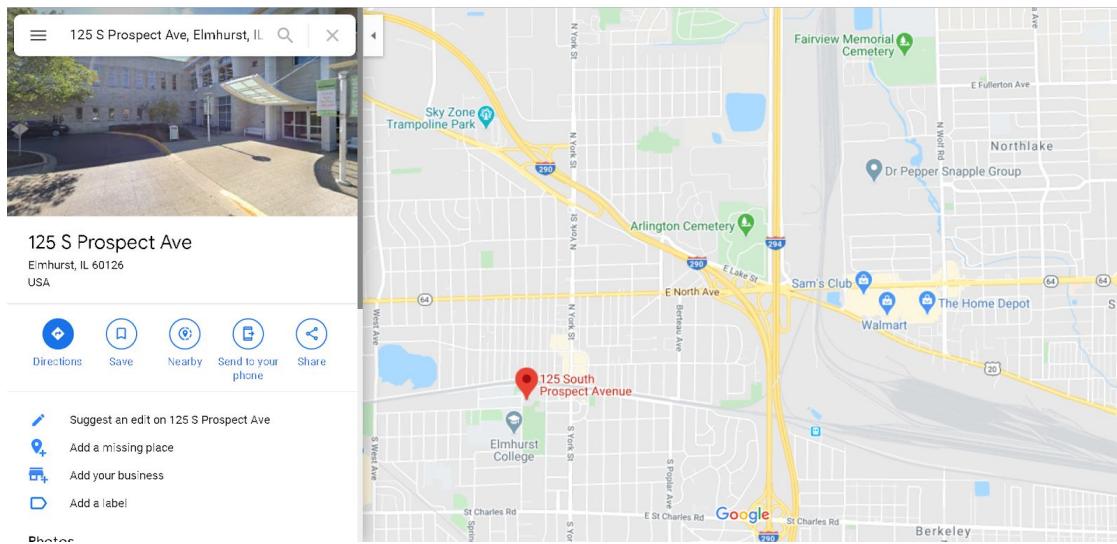


Figure 69

Click location button from park page will direct to google map show address of car park.

## Feedback page

```
<body>
<h1 align="center">Feedback</h1 >
<form method = 'post' enctype="multipart/form-data" align="center">

{{ta.as_p}}


<input type="submit" value="Update">

</form>
<p align="center">{{ submit }}</p>
<br>
```

Figure 70

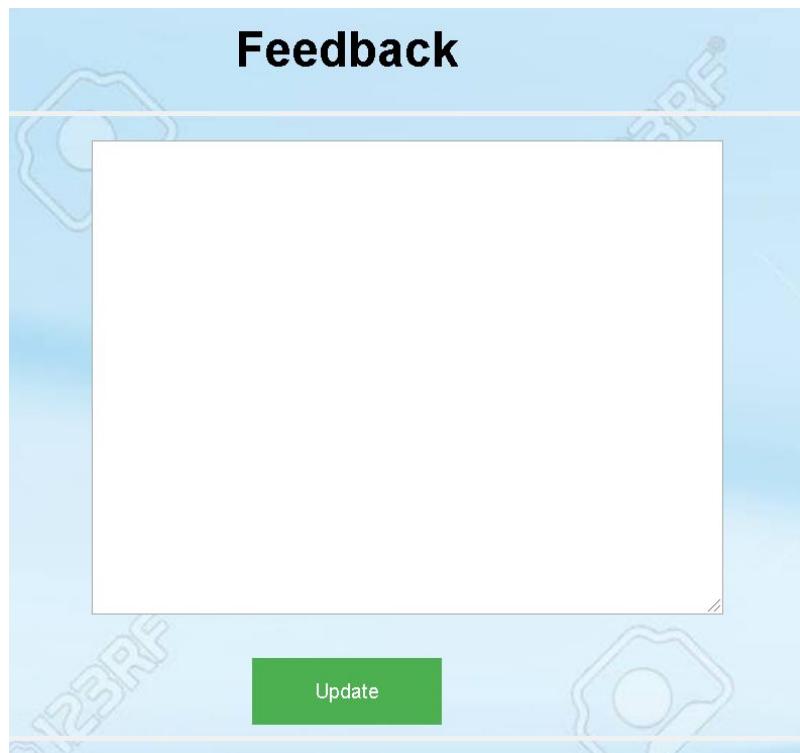


Figure 71

This is feedback html, it have a textarea input form, allow user to input feedback and direct to transfer page.

### Transform page

```
<body>
  <h1 align="center">Feedback</h1 >

  <table style="...">
    <tr>
      <th><button type="submit" onclick="location.href='{{ url 'parkselect'}}'" >
        Park
      </button></th>
      <th><button type="submit" onclick="location.href='{{ url 'feedback'}}'" >
        Feedback
      </button></th>
    </tr>

  </table>
  <h2 align="center">Thanks for your feedback</h2>
  ..
```

Figure 72

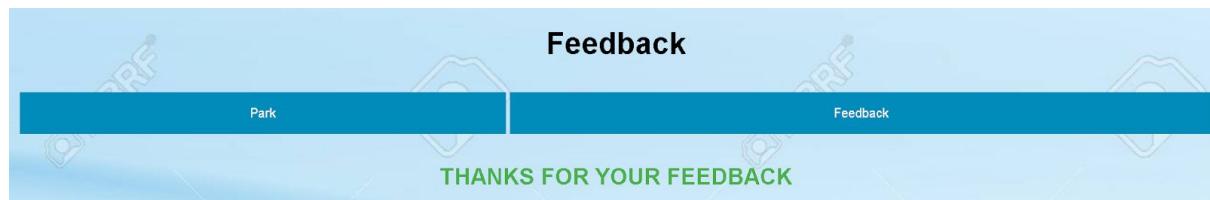


Figure 73

This is transfer page, use to let user knows feedback update successful and direct park page or feedback page.

#### 4.6. Middle-Tier

In this part, we are going to explain each task of the django project.

##### manage.py

This file allows users to use different ways to interact with django.

##### setting.py

This is configuration file of django, use to set such as database, TIME\_ZONE, USE\_I18N与USE\_L10N, template, ect

##### urls.py

This file lists all paths of the webpage.

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', login, name='login'),
    path('register', register, name='register'),
    path('parkselect', parkselect, name='parkselect'),
    path('feedback', feedback, name='feedback'),
]
```

Figure 74

Our project has 4 paths of webpage: login, register, park and feedback.

### model.py

This file init table of database, it allowed developer to set different attribute of table.

### view.py

Creating Views. A view function, or “view” for short, is simply a Python function that takes a web request and returns a web response. This response can be the HTML contents of a Web page, or a redirect, or a 404 error, or an XML document, or an image, etc.

```
def login(request):
    if request.method == "POST":
        uf = UserFormLogin(request.POST)
        if uf.is_valid():

            username = uf.cleaned_data['username']
            password = uf.cleaned_data['password']
            userResult = User.objects.filter(username=username, password=password)
            # pdb.set_trace()
            if (len(userResult) > 0):
                request.session['username'] = username
                Location.objects.create(parkName="EPL parking", address="125 S Prospect Ave Elmhurst, IL 60126").save()
                return redirect('parkselect')
            else:
                return render_to_response('userlogin.html', {'fail': "This user not exist, Please try again"})
        else:
            uf = UserFormLogin()
    return render_to_response("userlogin.html", {'uf': uf})
```

Figure 75

This is login function, it able to receive input of user then check is the input exists in the database, If not exists will tell user your input is wrong, if exist will direct to park page. uf.cleaned\_data is user to get data from input form, user.object.filter is check user input exist in database, request.session is save username in to session.

```

def register(request):
    if request.method == "POST":
        uf = UserForm(request.POST)
        if uf.is_valid():
            username = uf.cleaned_data['username']
            filterResult = User.objects.filter(username=username)
            if len(filterResult) > 0:
                return render_to_response('register.html', {"errors": "This user already exist"})
            else:

                password1 = uf.cleaned_data['password1']
                password2 = uf.cleaned_data['password2']
                errors = []
                if (password2 != password1):
                    errors.append("Please enter right password")
                return render_to_response('register.html', {'errors': errors})
                password = password2
                user = User.objects.create(username=username, password=password1)
                user.save()
                return render_to_response('register.html', {'succ': "register successful"})
            else:
                uf = UserForm()
        return render_to_response('register.html', {'uf': uf})
    
```

Figure 76

This is register function, it use to insert user input into database, also if condition is wrong will tell user the error. user.object.create is to create new data for user table, user.save is save the data into database.

```

def parkselect(request):
    if request.session.get('username'):
        return render(request, 'success.html')
    else:
        return HttpResponse("Please login first!")
    
```

Figure 77

This function is direct to park page, if user did not login then will tell user to login.

```

def feedback(request):
    if request.session.get('username'):
        if request.method == "POST":
            ta = textarea(request.POST)
            if ta.is_valid():
                feedb = ta.cleaned_data['Feedback']
                parkname = "EPL Parking"
                username = request.session.get('username')
                fb = Feedback.objects.create(username=username, parkName=parkname, feedbk=feedb)
                fb.save()
            return render(request, 'note.html')

        else:
            ta = textarea()
        return render_to_response('feedback.html', {'ta': ta})
    else:
        return HttpResponseRedirect("Please login first!")

```

Figure 78

This is a feedback function, used to store user's feedback into database.

```

class UserForm(forms.Form):
    username = forms.CharField(label='Username', max_length=100)
    password1 = forms.CharField(label='Password', widget=forms.PasswordInput())
    password2 = forms.CharField(label='Password Confirm', widget=forms.PasswordInput())

```

This register form.

```

class UserFormLogin(forms.Form):
    username = forms.CharField(label='Username', max_length=100)
    password = forms.CharField(label='Password', widget=forms.PasswordInput())

```

This is login form.

```

class textarea(forms.Form):
    Feedback = forms.CharField(widget=forms.Textarea(attrs={"rows":20, "cols":60}))

```

Figure 79

This is feedback form.

#### 4.7. Back-End

```
from django.db import models

class User(models.Model):
    username = models.CharField(max_length=50)
    password = models.CharField(max_length=50)

class Location(models.Model):
    parkName = models.CharField(max_length=50)
    address = models.CharField(max_length=200)

class Feedback(models.Model):
    username = models.CharField(max_length=50)
    parkName = models.CharField(max_length=50)
    feedbk = models.CharField(max_length=6000)
```

Figure 80

Our project set 3 table for database, user table have 2 attribute use to implement login function, location table have 2 attributes use to store name and address of car park, feedback table have 3 attribute use to store feedback of users.

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'project',
        'USER': 'user',
        'PASSWORD': '13110772322',
        'HOST': 'localhost',
        'PORT': '5432'
    }
}

```

*Figure 81*

This is our project database setting, it link to postgresql.

#### 4.8. Conclusions

In this chapter, we start to design all systems of the project, first explain the methodology that was used in the development process, next a broad overview of the technical architecture was outlined, then into detail the front and back-end design in prototype development.

The next chapter will talk about the testing of the project.

### 5. Testing and Evaluation

#### 5.1. Introduction

This chapter describes how all the testing and evaluation of the system was executed. Each phase of testing will be described in detail.

#### 5.2. Plan for Testing

Testing is how you spot the errors. A structured approach to testing, as managed through a test phase, is the way to do it. Methodically working through the various processes and deliverables gives you a solid base for being able to say you are delivering what you said you would. Testing can be a good sense check to make sure our deliverables are fit for purpose [19].

Backing up and committing the project consistently using GIT version control will ensure that any changes can be rolled back if there are any major errors. Having multiple project backups will help in the inevitable “accidentally deleting an important part of the project”.

We may ask other people to evaluate our web application, then we know how to develop our project.

Firstly, we would mix white-box tests make and the unit tests would be an adequate way to test the project. It makes sure no errors in each unit of application. after we use the black box and Integration Testing to test the project, it makes sure no error between each unit and model.

Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing.[20]

White Box Testing is based on applications' internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing usually is done at the unit level.[20]

A unit test is a test written by the programmer to verify that a relatively small piece of code is doing what it is intended to do. They are narrow in scope, they should be easy to write and execute, and their effectiveness depends on what the programmer considers to be useful. The tests are intended for the use of the programmer, they are not directly useful to anybody else, though, if they do their job, testers and users downstream should benefit from seeing fewer bugs.[21]

An integration test is done to demonstrate that different pieces of the system work together. Integration tests can cover whole applications, and they require much more effort to put together. They usually require resources like database instances and hardware to be allocated for them. The integration tests do a more convincing job of demonstrating the system works (especially to non-programmers) than a set of unit tests can, at least to the extent the integration test environment resembles production.[21]

## Black Box Testing and Integration Testing

Test No	Test Description	Expected Outcome	Pass?
1	Click the register button on the register page	return text tells users to register successfully or false.	YES
2	enter different values in password and password confirm on register page	return text tell user password is wrong	YES
3	when login and register leave some fields empty	show text tells the user to enter data.	YES

4	Click the login button in on register page	return to login page	YES
5	enter the same username somebody enters before on the register page.	return text tell user username already exists.	YES
6	Click the register button on the login page	return to register page	YES
7	Click the login button on the login page	if false will return text tell user username or password wrong if success will return to the homepage, and display a video	YES
8	click location button on the index page	display the location of the car park on google map.	YES
9	click feedback button on the index page	go to the feedback page and allow the user to send feedback.	YES
10	send feedback to database	able to store username, park name and feedback in the database	YES

## White Box Testing and unit Testing

### Test Plan

Number	Module	Description
1	extract a space image	extract a empty car park image from car park video, use to extract all yellow line from car parking. extract train image and test image from video.
2	extract car park line	extract yellow line from empty car park image, use to check position of car space.
3	draw rectangle of car space from training and test image	according the line to draw the rectangle of car space on the train image and test image.
4	extract each image of car space	according to the rectangle to extract each car space. use to create a vgg training model.
5	create vgg training model	according to the training image to create a vgg training model as a .h5 file.
6	use h5 file to process test image	use vgg training model to process test image, it use green rectangle shows empty space and red rectangle shows occupied space, and shows total space and empty space on the top of the image.
7	use h5 file to process video	use vgg training model to process each frame of video.

8	save as a new video.	save each frame which is processed as a new video.
9	register page use django	save register information into postgresql
10	login page use django	to confirm the register information from postgresql, then transfer to the index page.
11	park video page	auto play processed video.
12	location page	display location of car park on the google map
13	feedback page	allow user to make feedback and save into postgresql
14	deploy our project in cloud of aws	deploy our project in cloud of aws using lightsail

### Test implement

Id	Test Number	Date	Success/failure	Description (if failure)
a1	1	23/2	S	
b1	2	23/2	F	wrong color mask, show wrong mask.
b2	2	25/2	F	do not extract all lines
b3	2	26/2	F	too many interference factor, show many lines
b4	2	26/2	F	have overlap lines disturb, have to delete extra lines by distance of y-axis
b5	2	28/2	S	
c1	3	28/2	F	lines do not sort, can not use the following line to create a rectangle for each space.
c2	3	1/3	F	have extra rectangle, have to only shows limit area of rectangle
c3	3	1/3	F	the size of the train image is wrong, use the resize function to change the size of the image.
c4	3	1/3	S	
d1	4	2/3	F	the crop image is wrong, because of argument of crop function is wrong
d2	4	2/3	F	output image is not sort, because of wrong position of variable
d3	4	2/3	S	
e1	5	4/3	F	wrong tensorflow version
e2	5	6/3	F	the training image include rectangle, so it have low accurate, have to use original train image to crop image of car space.

e3	5	6/3	F	still low accurate, I increase Epoch.
e4	5	7/3	S	
f1	6	8/3	F	error with wrong argument
f2	6	9/3	F	do not add space count on the image
f3	6	10/3	S	
g1	7	10/3	F	can not shows fluency video, have lag with video, because computer have to spend time to process each frame of video
g2	7	11/3	F	try to change version of tensorflow, but still can not shows fluency video, have to save processed frames as a new video.
g3	7	12/3	S	
h1	8	12/3	S	
i1	9	14/3	F	don't insert information into database
i2	9	15/3	F	input form in wrong
i3	9	15/3	F	have to add conditions to the registration form.
i4	9	16/3	F	add a background image
i5	9	17/3	F	url of button is wrong
i6	9	17/3	S	
j1	10	18/3	F	wrong form
j2	10	18/3	F	add conditions
j3	10	19/3	S	
k1	11	19/3	F	display nothing on this page, can not play avi type of video
k2	11	20/3	F	video not in the middle of webpage
k3	11	21/3	F	video can not auto play
k4	11	22/3	F	video have to spend to load
k5	11	24/3	F	after refresh webpage, the video don't autoplay.
k6	11	24/3	S	
l1	12	25/3	S	
m1	13	27/3	F	error with not enough argument insert into database.
m2	13	27/3	S	
n1	14	29/3	F	can not use database when deployed
n2	14	29/3	F	have to update database
n3	14	30/3	F	error with "attempt to write a readonly database"

n4	14	30/3	F	cant find static file when deployed, video and background missing.
n5	14	30/3	S	

### 5.3. Plan for Evaluation

Evaluation of this system is equally as important as testing. The reason for this is that user experience is one of the main complexities of the application.

we use 10 Usability Heuristics for User Interface Design to evaluate space checkers.

Jakob Nielsen's heuristics are probably the most-used usability heuristics for user interface design. They are called "heuristics" because they are broad rules of thumb and not specific usability guidelines.[24]

01. Visibility of system status - good
02. Match between system and the real world - good
03. User control and freedom - good
04. Consistency and standards - good
05. Error prevention - good
06. Recognition rather than recall - good
07. Flexibility and efficiency of use - very good
08. Aesthetic and minimalist design - very good
09. Help users recognize, diagnose, and recover from errors - bad
10. Help and documentation - bad

### 5.4. Conclusions

This chapter reviewed the testing and evaluation of the system. The testing consisted of performing Unit Testing and Integration Testing both while developing the system and on completion. The evaluation included Nielsen's heuristics.

## 6. Personal Conclusion and Future Work

### 6.1. Personal Conclusion

The main idea of this project is to allow drivers to check the status of car park, that helps the driver to save time to find where free space exists.

I was learning image processing, machine learning and django last semester, some knowledge I do not really understand, this project gives a very good opportunity to digest those knowledge.

The project has provided many new knowledge and challenges. I had to do a lot of research to digest new knowledge and solve the problems.

I think this project is a very good experience to feed my future work.

### 6.2. Future work

The first area I plan to use another machine learning model to process video, because although use VGG model have very high accuracy, but it has low efficiency with process images, I can not directly use the function to show video.

The second area I plan to improve user interface of web applications.

The third area I plan to include some other car parks, this will allow drivers to have multiple choices.

The fourth area I plan to add more features, such as diagrams of the number of free spaces in different times etc.

## Bibliography

References:

1. Aindrila Biswas<sup>1</sup>, Angshu Mukherjee<sup>2</sup> and Mousumi Roy, "Leveraging Factors for Consumers' Car Purchase Decisions- A Study in an Emerging Economy" , June 2014,  
[http://webcache.googleusercontent.com/search?q=cache:http://jmppnet.com/journals/jmpp/Vol\\_2\\_No\\_2\\_June\\_2014/8.pdf](http://webcache.googleusercontent.com/search?q=cache:http://jmppnet.com/journals/jmpp/Vol_2_No_2_June_2014/8.pdf)
2. MariaBörjesson. JonasEliasson, "Experiences from the Swedish Value of Time study", January 2014,  
<https://doi.org/10.1016/j.tra.2013.10.022>
3. Joseph Howse , "OpenCV Computer Vision with Python" , april 2013
4. GiuseppeAmato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, CarloMeghini, ClaudioVairo, "Deep learning for decentralized parking lot occupancy detection", 15 April 2017  
<https://www.sciencedirect.com/science/article/pii/S095741741630598X?via%3Dhub>
5. Feng Xiang, "base OpenCV TensorFlow implement looking for car space", 2018-10-26,  
<https://segmentfault.com/a/1190000016849300>
6. Sabina Pokharel, "Smart Parking — An Application Of AI", 2019-10-10,  
<https://towardsdatascience.com/smart-parking-an-application-of-ai-9a4af90b1de6>
7. Adam Geitgey, "Snagging Parking Spaces with Mask R-CNN and Python, Adam Geitgey", 2019/01/21,  
<https://medium.com/@ageitgey/snagging-parking-spaces-with-mask-r-cnn-and-python-955f2231c400>
8. Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, "Mask R-CNN", 20 Mar 2017  
<https://arxiv.org/abs/1703.06870>
9. Adrian Rosebrock, "Intersection over Union (IoU) for object detection", November 7, 2016,  
<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
10. Krasimir Hristozov, "MySQL vs PostgreSQL -- Choose the Right Database for Your Project", 19/7/2019, <https://developer.okta.com/blog/2019/07/19/mysql-vs-postgres>
11. Lucidchart Content Team, "The Pros and Cons of Waterfall Methodology", November 02, 2017 <https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology>
12. Dipen Patel, " An Introduction to MVC Architecture: A Web Developer's Point of View", 17 Jun 2019,  
<https://dzone.com/articles/introduction-to-mvc-architecture-web-developer-poi>
13. Rahul Galgali, "What is OpenCV?", 27/2/2016, <https://www.quora.com/What-is-opencv>

14. Lauren Stewart, "Front End vs Back End Development", January 14, 2019,  
<https://www.coursereport.com/blog/front-end-development-vs-back-end-development-where-to-start>
15. David Lane, " Web Database Applications with PHP, and MySQL", Hugh E. Williams, March 2002,  
<https://www.oreilly.com/library/view/web-database-applications/0596000413/ch01s03.html>
16. Amit Ashwini, "What Is Django And Why Is It So Popular?" 23/11/2017,  
<https://medium.com/swlh/what-is-django-and-why-is-it-so-popular-2b225620cca0>
17. Anuj Gupta, "What is PyCharm used for?", 12/4/2017,  
<https://www.quora.com/What-is-PyCharm-used-for>
18. Kayla Ngan, "What is Git", 05/04/2018,  
<https://docs.microsoft.com/en-us/azure/devops/learn/git/what-is-git>
19. Erika Rykun, "3 Reasons Why You Should Plan Your Testing Phase", 31 July 2018,  
<https://pmtips.net/article/3-reasons-why-you-should-plan-your-testing-phase>
20. Rajkumar, "Black Box And White Box Testing | Definition And Types", 30/7/2019,  
<https://www.softwaretestingmaterial.com/black-box-and-white-box-testing/>
21. Nathan Hughes, "What's the difference between unit tests and integration tests?", 12/12/2018,  
<https://stackoverflow.com/questions/5357601/whats-the-difference-between-unit-tests-and-integration-tests>
22. Aivar Annamaa, "Learn to code with Thonny – a Python IDE for beginners", 19/02/2018,  
<https://fedoramagazine.org/learn-code-thonny-python-ide-beginners/>
23. Serdar Yegulalp, "What is TensorFlow? The machine learning library explained", 18/6/2019,  
<https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html>
24. "10 Usability heuristics explained", Eranga Liyanage, Oct 2, 2016,  
<https://medium.com/@erangatl/10-usability-heuristics-explained-caa5903faba2>
25. Davi Frossard, "VGG in TensorFlow", 17/06/2016,  
<https://www.cs.toronto.edu/~frossard/post/vgg16/>
26. Hippie trail, "Explain Hough Transformation", 23/04/2017,  
<https://stackoverflow.com/questions/4709725/explain-hough-transformation>
27. Computer Hope, "MATLAB", 30/6/2019,  
<https://www.computerhope.com/jargon/m/matlab.htm>

28. Mouhyemen Khan, "What are the pros and cons of using Matlab vs Open CV for image processing?", 20/9/2017,  
<https://www.quora.com/What-are-the-pros-and-cons-of-using-Matlab-vs-Open-CV-for-image-processing>
29. Mindfire Solutions, "Advantages and Disadvantages of Python Programming Language", 24/04/2017,  
<https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121>
30. HEBA SOFFAR, "C programming language features, advantages and disadvantages", 19/1/2017,  
<https://www.online-sciences.com/programming/c-programming-language-features-advantages-and-disadvantages/>
31. Archan Desai, "What are the advantages and disadvantages of programming in Java?", 22/25/2015,  
<https://www.quora.com/What-are-the-advantages-and-disadvantages-of-programming-in-Java>
32. Mindfire Solutions, "Key Differences between Python and Java", 05/06/2018,  
<https://medium.com/@mindfiresolutions.usa/key-differences-between-python-and-java-3e5dbbc0bd67>
33. Andrew Powell-Morse, "Iterative Model: What Is It And When Should You Use It?", 25/12/2016, <https://airbrake.io/blog/sdlc/iterative-model>
34. Python Lessons, "Convolutional Neural Networks (CNN) explained step by step", 08/05/2019,  
<https://medium.com/analytics-vidhya/convolutional-neural-networks-cnn-explained-step-by-step-69137a54e5e7>
35. i-king-of-ml, "KNN(K-Nearest Neighbour) algorithm, maths behind it and how to find the best value for K" 25/10/2019,  
<https://medium.com/@rdhawan201455/knn-k-nearest-neighbour-algorithm-maths-behind-it-and-how-to-find-the-best-value-for-k-6ff5b0955e3d>
36. Bruno Stecanella, "An Introduction to Support Vector Machines (SVM)", 22/6/2017, <https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/>
37. gvrv99, "Back Propagation Neural Network: Explained With Simple Example",  
<https://www.guru99.com/backpropogation-neural-network.html>
38. Domantas G., "What is Apache? An In-Depth Overview of Apache Web Server", 25/03/2020, <https://www.hostinger.com/tutorials/what-is-apache>

39. Satyendra Singh, ML enthusiast, "Which machine learning algorithms do you suggest for image classification since PCA and KNN are too slow for large datasets (>1,500 images)?", 29/01/2018,

<https://www.quora.com/Which-machine-learning-algorithms-do-you-suggest-for-image-classification-since-PCA-and-KNN-are-too-slow-for-large-datasets-1-500-images>

40. Guest Writer, "What Is AWS Lightsail, and Who Should Use it?", 28/02/2019,  
<https://www.iotforall.com/what-is-aws-lightsail/>

41. geeksforgeeks, "Erosion and Dilation of images using OpenCV in python",  
<https://www.geeksforgeeks.org/erosion-dilation-images-using-opencv-python/>

42. Martin Heller, "What is Keras? The deep neural network API explained", 28/01/2019,  
<https://www.infoworld.com/article/3336192/what-is-keras-the-deep-neural-network-api-explained.html>

43. Justin Ellingwood, "How To Set Up uWSGI and Nginx to Serve Python Apps on Ubuntu 14.04", 05/2015,  
<https://www.digitalocean.com/community/tutorials/how-to-set-up-uwsgi-and-nginx-to-serve-python-apps-on-ubuntu-14-04>