|  |
| --- |
|  |
| AUTOMATIC LICENSE PLATE RECOGNITION FUNCTIONAL SPECIFICATION |
|  |
| **By****Anthony Tierney** |
|  |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |
| AUTOMATIC LICENSE PLATE RECOGNITION FUNCTIONAL SPECIFICATIONAnthony TierneyI.T. Carlow, Kilkenny Road, Carlow |
| 11-Dec-13 |
|  |

|  |
| --- |
|  |

# Abstract

This document describes the functionality of the app. The main functionalities are to capture an image. This image is then processed with noise reduction and edge detection filters to improve the rate of accuracy. The characters are then separated into separated images with a character separation algorithm. These images are then put through an optical character recognition algorithm to get the licence plate number. This licence plate number is then cross referenced in a local database that contains a list of legal vehicles for the car park. The database is updated daily so that the car park attendant has the latest and most up to date database available to them.

Table of Contents

[1. Introduction 1](#_Toc374692505)

[1.1. Purpose 1](#_Toc374692506)

[1.2. Scope 1](#_Toc374692507)

[1.3. Intended Audience 1](#_Toc374692508)

[2. Overview 2](#_Toc374692509)

[3. Inspiration for this project 3](#_Toc374692510)

[4. User Group 4](#_Toc374692511)

[5. Project Timeline 5](#_Toc374692512)

[6. Functionality 6](#_Toc374692513)

[6.1. Crop image taken from the camera 6](#_Toc374692514)

[6.2. Edge Detection and Noise filtering 6](#_Toc374692515)

[6.3. Character Separation 6](#_Toc374692516)

[6.4. Optical Character Recognition 6](#_Toc374692517)

[6.5. Cloud Functionality 6](#_Toc374692518)

# Introduction

## Purpose

The aim of this document is to advertise the functionalities of this project, the Automatic Licence Plate Recognition Android App.

## Scope

This manual provides details of the core functionalities of the project; it does not include any design aspects of the app.

## Intended Audience

This document is intended for those who wish to gain knowledge of the core functionality of the ALPR Android app.

# Overview

This project is an app that allows car park management firms to speed up the verification of vehicles in a car park. The app is installed on a camera enabled android device that is given to the car park attendant.

The attendant aligns the licence plate with a visible boundary box on screen and captures the image of the licence plate which is then cropped to the boundary box. The app takes this image as input and applies noise reduction and edge detection filters to increase the accuracy of the next phase.

The next phase is character separation. The image is scanned for the characters on the licence plate and separates these characters into separate images. These separate images are then run through an optical character recognition algorithm to retrieve the characters contained in the images.

This licence plate number is then checked to verify if the vehicle is present in the local database. This database is downloaded daily from servers that contain the master copy, so that the device always has the most up to date version of the database.

Statistics are also kept of the vehicles that were parked illegally.

# Inspiration for this project

The inspiration for this project came from the knowledge that car park attendants may suffer from expending an unnecessary amount of time checking whether a vehicle in a car park is permitted to park there.

The current procedure for validating vehicles parking rights is slow and monotonous. The parking attendant must enter each individual vehicle registration number separately by hand to a device that contains a list of vehicles that are permitted to park there and that device checks a local database and verifies the vehicle or declines it..

The aim of this app is to speed up this process by using a quick and accurate licence plate recognition algorithm while simultaneously reducing expenditure. The app will be run from a mobile device (such as an Android device) and will have a similar database. This database will be stored locally but updated from a cloud application that contains all the information required to check and validate vehicles.

There are currently similar applications on the Google Play Store that offer similar functionality. The main different between these applications and this project is that the apps currently on the market do not include cloud functionality

# User Group

The user group for this project is a minority of companies that manage car parks with out-dated hardware and architecture. Such companies may wish to upgrade their systems to more modern and up to date variants. Companies may wish to reduce expenditure by incorporating the company mobile phone with the device that checks whether a vehicle is parked legally or not.

# Project Timeline



# Functionality

## Crop image taken from the camera

The image that is processed in this app is taken by the camera. The camera is only temporarily stored on the device, otherwise the local storage will be quickly used. On screen the user is presented with the image being relayed live from the camera. In this screen, there will be a visible boundary box visible to the user so that they can align the licence plate within the box for greater accuracy. When the camera shutter is released, the image is saved temporarily and cropped to the contents of the boundary box.

## Edge Detection and Noise filtering

This functionality scans over the cropped image received from the camera and runs a noise filter to remove any unwanted artefacts in the image that may cause unwanted edged to appear in the post-processed image. A matrix operation is then performed on the image, after noise reduction, to produce an image of edges. This two-step process is called canny edge detection, which was discussed in the research document.

## Character Separation

This component takes the processed image, after edge detection has been completed, and uses it as input. The next stage in this process is to get the grey scale image and hence, the binary (black and white) image. Performing this will enhance the accuracy of the character separation. The next phase is to locate the characters. The image is scanned from left to right and a matrix operation check for a difference in the intensity of the colour in the image (made easier by using the binary image) and noting where there is a change in intensity. With these locations stored, a top-down scan is performed to get a finer position of the character. Each of these characters separated are stored as separate images.

## Optical Character Recognition

The separate images from the Character Separation are used as input. Each image is put through an optical character recognition algorithm to retrieve the character contained in the image. This character is then stored in memory and the next image is processed. The images are processed in order to maintain the order of the licence plate number. This string of numbers and characters are compared to the entries in the local database to corroborate whether a vehicle is legally parked or not.

## Cloud Functionality

The cloud functionality accounts for the possibility that no network connection may be available at the car park location and also; that companies may not wish to pay for costly data plans from a network operator. To tackle this problem, the database is copied from the cloud database to local storage (the device) over a wireless network such as the companies LAN in their office. The database is downloaded daily so that the most up to date version is always installed on the attendant’s device. This has other benefits such as increased performance as network latency is not a factor to consider when contacting the database; hence the performance of the app is increased. Statistics are also kept of illegally parked vehicles and this data is uploaded to the server for statistics generation such as the total amount of fines per day per car park.

# 7. Glossary

**Artefacts:** in the context of this app, an artefact is a distortion or imperfection in the image 6

**Local database:** A database that is stored on the devices' memory 2

**matrix operation:** A calculation that manipulates the image 6

**network latency:** The time it takes for the device to send a request to a server/website and the time it takes to get the information back 6