|  |
| --- |
|  |
| AUTOMATIC LICENSE PLATE RECOGNITION RESEARCH MANUAL |
|  |
| **By**  **Anthony Tierney** |
|  |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |
| AUTOMATIC LICENSE PLATE RECOGNITION RESEARCH MANUAL  Anthony Tierney  I.T. Carlow, Kilkenny Road, Carlow |
| 22-Nov-13 |
|  |

|  |
| --- |
|  |

# Abstract

The research in this document covered the different uses, platforms, languages, cloud services, image manipulation libraries, optical character recognition libraries and edge detection algorithms that relate to automatic license plate recognition. It is recommended that the platform used is Android; the cloud service to be used is Google Datastore, the image manipulation library to be used is OpenCV, Tess-two should be used for the optical character recognition and Canny’s edge detection for the edge detection.

Table of Contents

1. [Introduction 1](#_Toc372891761)

[1.1. Purpose 1](#_Toc372891762)

[1.2. Scope 1](#_Toc372891763)

[1.3. Intended Audience 1](#_Toc372891764)

[2. Findings 2](#_Toc372891765)

[2.1. Uses 2](#_Toc372891766)

[2.2. Platforms and Languages 3](#_Toc372891767)

[2.3. Cloud Services 9](#_Toc372891768)

[2.4. Image Manipulation Libraries 10](#_Toc372891769)

[2.4.1. OpenCV 10](#_Toc372891770)

[2.4.2. SimpleCV 10](#_Toc372891771)

[2.5. Optical Character Recognition 11](#_Toc372891772)

[2.5.1. Tesseract 11](#_Toc372891773)

[2.5.2. Tess-two 11](#_Toc372891774)

[2.6. Edge Detection Algorithms 12](#_Toc372891775)

[2.6.1. Edge Detection 12](#_Toc372891776)

[3. Recommendations 16](#_Toc372891777)

[3.1. Platform and Language 16](#_Toc372891778)

[3.2. Cloud Service 16](#_Toc372891779)

[3.3. Image Manipulation Library 16](#_Toc372891780)

[3.4. Optical Character Recognition 16](#_Toc372891781)

[3.5. Edge Detection Algorithm 17](#_Toc372891782)

[4. Glossary 19](#_Toc372891783)

[5. Bibliography 20](#_Toc372891784)

# Introduction

## Purpose

The aim of this document is to research platforms, current solutions, toolkits and libraries, edge detection algorithms and the uses of a licence plate recognition system. This includes investigating competitors existing programs.

## Scope

This manual provides details of the research conducted for this project. The research objectives are:

* Uses of Automatic licence plate recognition
* Platforms and languages
* Cloud Services
* Image manipulation Libraries
* Optical Character Recognition
* Edge Detection Algorithms
* Character separation algorithms

## Intended Audience

This document is intended for those who wish to understand the basic concept of licence plate recognition.

# Findings

## Uses

An automatic licence plate recognition program has many uses. To name a few:

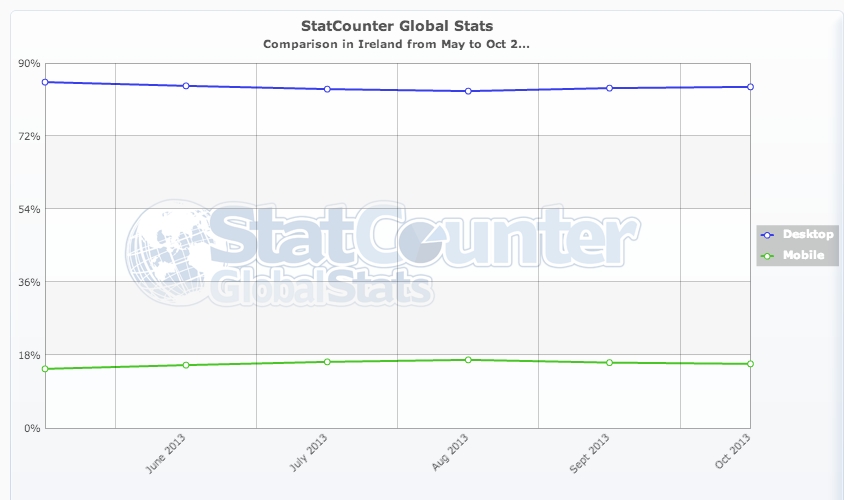
* Border Control, to identify suspect vehicles
* Vehicle Repossessions
* Help police identify stolen vehicles simply by driving past them
* Car park gates may automatically open when a vehicle with a recognised licence plate approaches
* Car park management companies can quickly determine if a vehicle has permission to reside in the car park

These are possibly the main uses for licence plate recognition and are used to great effect. They have resulted in the reduction of crime, better officer efficiency (Association, 2008) and the better management of car parks in airports, shopping centres, universities and so on.

## Platforms and Languages

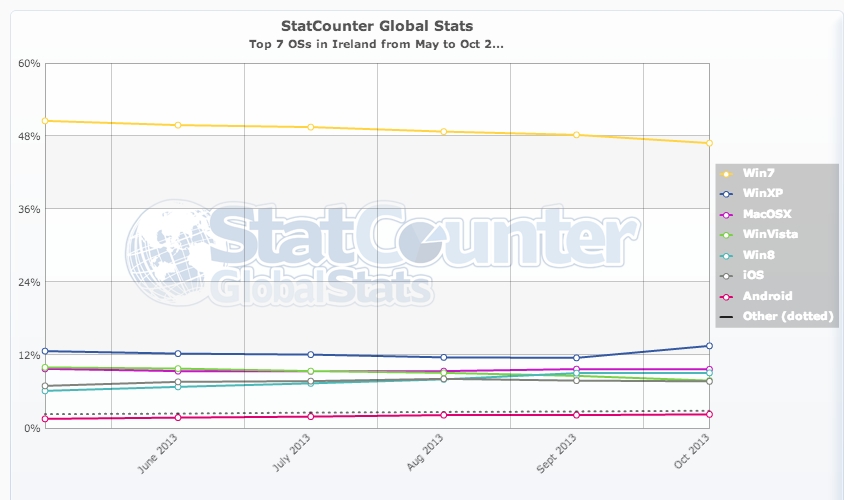
#### PC

The usage of PCs for automatic licence plate recognition is the largest as there are a large number of programs available for the platform that meet the requirements that company car parks require. Below is list a list of some licence plate recognition software available for the PC:

* AutoVu (http://www.genetec.com/solutions/all-products/autovu)
* License Plate Recognition (http://www.licenseplatesrecognition.com/)
* Motorola ALPR (http://bit.ly/1i4B2IH)
* 3M ALPR (http://bit.ly/17eN8wX)

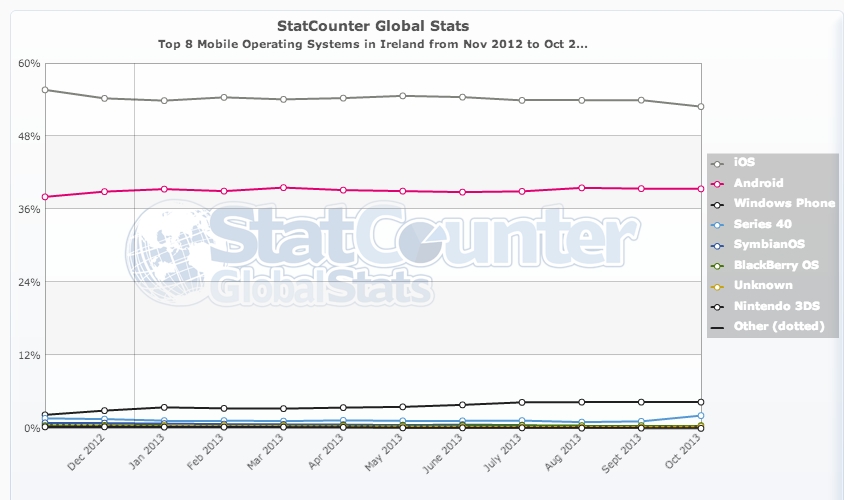
There are many more programs available for this platform as this platform is still the most popular as the graph below shows: (MobileVsDesktop)

From this graph, we see that the PC is still the dominant platform in Ireland for business and residential use. This may be the main advantage and also the main disadvantage of using the PC as the platform of choice. The advantage is that the program has a wider market to its advantage and a better chance of success. Conversely, it may also be the reason for the failure of this project in terms of marketability. This is because of the huge scale of the PC software market. Up until the latest iteration of the Windows operating system, there was no dedicated store for software to be sold. There are many software vendors online but many individuals or companies may not know of the existence of such marketplaces. Another note is that Windows 8, which is the only version of Windows to have a dedicated app store, has the second lowest market share of the Windows OS (DesktopShare).



#### Android

Android is easily the most accessible platform for companies as it offers devices at a fraction of the cost of those from Blackberry or Apple. This is a very attractive property to businesses that are looking to cut costs and capital expenditure. The platform also uses a widely used programming language, Java. (Tiobe, 2013) Android may not have the largest market share in Ireland but it, in general, slowly increasing its grasp on the Irish phone market. At the end of October 2013, Android had 40% of the mobile phone market in Ireland. (MobileShare)



It was found that Android also comes with a new version of the development kit that removes the eclipse IDE from the package. It is replaced with a single download from Google called the Android Studio. (Team A. S.) It is a more complete package than the previous Android Developer Toolkit and it is easier to install. A few more benefits of developing on the Android system are:

* Inexpensive to create apps

The Android development kit is available to anyone who wishes to use it. It can used on any PC platform (Windows, Mac, Linux) whereas iOS apps can only be developed on a Mac with a recent version of OSX (Team A. D.), which may result in high capital expenditure, a bad thing for new start-ups. With android, there are also no licencing fees or royalties to pay as the underlying architecture of the Android OS is open source and free. (Darcy, 2009) The only fee is a once off payment of $25 and there is transaction fees of 30% of the purchase price of an app goes to Google for their distribution partner and operating fees.

* Allow deep inter application communication

The Android OS was designed to be as open as possible. Which meant, that the developers of the system allowed app developers have full communication between apps (for example; a photo editing app may have full control over the output from the devices camera). (Darcy, 2009) This also meant that external java toolkits may be used within the application

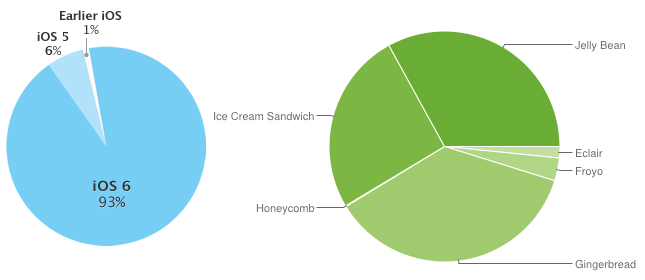
* Excellent support for developers

Within the Android development community, there are multiple levels of assistance available. From tutorials to community support forums, assistance with Android development is thorough and actually helpful. (Support)

There are also a number of negatives for developing on android:

* Fragmentation of the platform

Android is one of the most fragmented systems available today. The graph below show that a version from 2011 (Gingerbread) is of the most widely used versions. (Smith, 2013) This is a major problem for developers as they must choose between legacy support and newer features. Legacy support comes with issues as developers have to spend more time developing the app so that it is compatible with older hardware and OS versions. This will result in a less modern app, a larger budget and longer development times. Conversely, only supporting newer versions will exclude a large section of the possible market and increase the possibility of failure, but it will result in a more modern interface for the user and may encourage users of older hardware to update to a newer version.



* Support for different screen resolutions and sizes

Since Android is open source and free, a lot of companies have made screens of varying sizes and resolutions. This is a headache for app developers as they have to consider these differences when making their app. They must create different icons that suit different resolutions and pixel densities and still maintain the interface layout of the app.

The Play store has a small number of licence plate recognition apps available:

* CarCheck Lite
  + This is the only free automatic licence plate recognition app available on the play store, but it has a daily limit of the number of scans. It’s description states that the app *“…provides lower recognition capabilities and accuracy than the full versions…”* (Immense, 2013)
  + In personal tests, it has been found that this app works relatively well and is reasonably accurate in detecting Irish and British licence plates. Especially those with squared characters such as the German number design such as the example below:



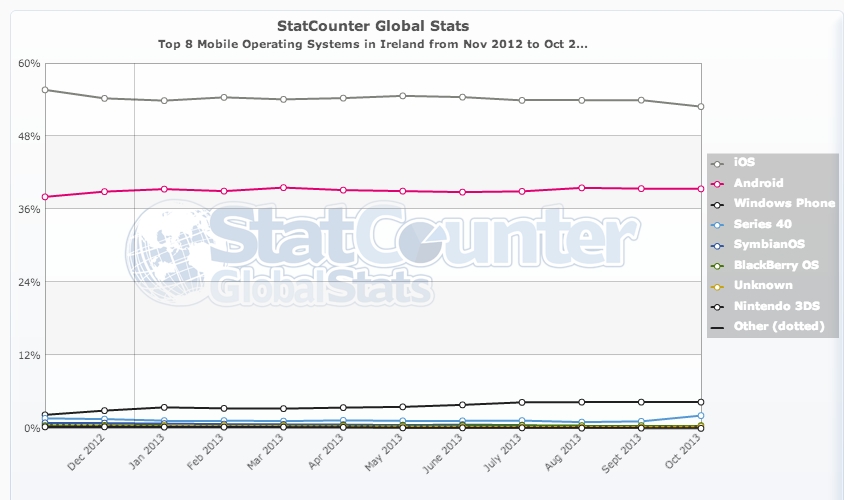
* + However, it does not allow for any other licence plate format other than the standard oblong rectangle format as in the example above.
* ANPReader
  + This is a paid example of an automatic licence plate recognition app.
  + Its description states that the app:

*“…works well in a wide range of lighting conditions, supports oblique angles, can read skewed plates, and can perform recognition across a broad spectrum of distances and sizes...”* (Immense, 2013)

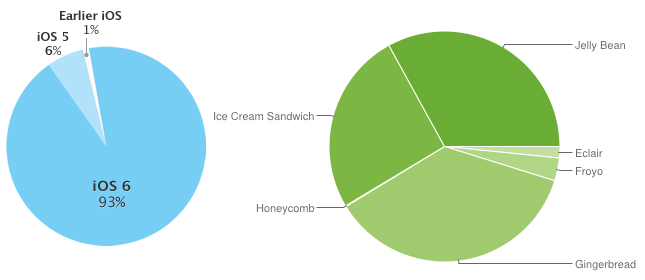
* This app is expensive (€36.82) to purchase considering it offers no cloud functionality.

#### iOS

iOS is a mobile operating system created by Apple for use in their mobile devices, tablet computers and media players. It is currently the most popular mobile OS in Ireland (MobileShare)



This allows for access to a greater market share and possibility of success. The reason for this success boils down to the magnitude of the app store, excellent hardware and an OS that keeps fragmentation to a minimum. Fragmentation of an operating system is when there are multiple versions of the system still in widespread use. Before iOS7 was release recently, iOS6 had the greatest market share of the operating system. (Smith, 2013)



This is great news for developers as it means that they do not have to waste time developing their apps for older systems. Hence, they can either finish the project earlier or allocate more time to the refining of their app. This may also be the reason why the iOS app store is so large, it has over one million apps for iOS and 475,000 dedicated iPad apps. (Ingraham, 2013)

But, that does not mean that it has the greatest app store market share. The Apple app store only has a paltry 18.2% compared to Android’s Play store’s 74.4%. These numbers mean very little as it does not represent the revenue or growth. Even though the Play store has a massive lead in market share, its revenue generation severely lags behind the app store. The app store generates approximately $5.1 million per day compared to $1.1 million for the Play store. (Koetsier, 2013) This clearly indicates that the iOS app store generates more revenue for developers that the Play Store.

This is good news for developers that want a return in revenue from their app. But, for small developers, starting developing for iOS can be a major investment as Apple products are expensive. The cheapest viable Apple laptop for app development is €1,349. The cheapest iOS device is currently €249 for an iPod touch. On top of that, there is a registration fee of $99 per year and Apple take 30% of the apps income. (Apple) By contrast, The Play Store has a once off fee of $25, but they still take 30% of the apps purchase price.

There are currently no apps on the iPhone app store for automatic licence plate recognition for Irish number plates.

## Cloud Services

#### SQL data model versus NoSQL data model

An SQL database is *“…one type of database which can be run on the cloud (either as a Virtual Machine Image or as a service, depending on the vendor). SQL databases are difficult to scale, meaning they are not natively suited to a cloud environment…”* (CloudDB)

NoSQL is *“…another type of database which can run on the cloud. NoSQL databases are built to service heavy read/write loads and are able scale up and down easily, and therefore they are more natively suited to running on the cloud…”* (CloudDB)

#### Cloud Databases

##### Google Cloud Datastore

This is the cloud database option by Google. It is a scalable, schema-less and offers robust data preservation by copying data across multiple locations. It offers developers a choice of languages to program the data in (Ruby, Java, Python and Node.js) and uses an SQL-like database query language. (GoogleCloud)

This service offers a free tier which will allow developers one gigabyte of free storage and 50 thousand read, write and small operations each. It is a database that is fully managed by Google and is scalable.

##### Amazon DynamoDB

*“Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. With the AWS Free Tier you get 100 MB of storage, 5 units of write capacity, and 10 units of read capacity for Amazon DynamoDB.”* (DynamoDB)

All data stored on DynamoDB is stored of solid state drives so speed and reliability are almost guaranteed due to the nature of solid state drives. Like Google Cloud Data Store, there is a free tier of usage. Users get 100 megabytes of storage and 5 write operations per second and 10 read operations per second. DynamoDB uses JSON to send and receive formatted data and the languages that can be used are Java, Node.js, .NET, Perl, PHP, Python, Ruby, and Erlang. (DynamoDBWiki)

##### MongoDB

MongoDB *“…is a cross-platform document-oriented database system. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in [favour] of JSON-like documents with dynamic schemas, making the integration of data in certain types of applications easier and faster.”* (MongoDBWiki)

## Image Manipulation Libraries

### OpenCV

OpenCV is short for Open Source Computer Vision. It is a library that contains the functionality to manipulate images. It is designed for efficiency and real time applications as it is written in C/C++. OpenCV is also available on multiple platforms, such as Android and iOS. (OpenCV) There is a number of programming languages supported in the latest release, these are: (OpenCV, 2013)

* C
* C++
* Java
* Python
* MATLAB

From the about page of the OpenCV website it states that: *“The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality”*

Knowing this, there would be enough functionality in the OpenCV libraries to separate each of the individual characters from the licence plate using the object identification (edge detection) algorithms.

### SimpleCV

SimpleCV is an alternative to OpenCV written and coded solely in python. SimpleCV is *“…an open source framework — meaning that it is a collection of libraries and software that you can use to develop vision applications. It lets you work with the images or video streams that come from webcams, Kinects (sic), FireWire and IP cameras, or mobile phones. [It] helps you build software to make your various technologies not only see the world, but understand it too. SimpleCV is free to use, and because it’s open source, you can also modify the code if you choose to. It’s written in Python, and runs on Mac, Windows, and Ubuntu Linux…”* (SimpleCV, 2013)

This platform would work adequately on a non-mobile device as there are currently no mobile devices that natively support the Python language. (Herman, 20131)

## Optical Character Recognition

### Tesseract

Tesseract is an open source optical character recognition program written and coded in C++. It may be used directly or through an API for developers. It supports a multitude of existing languages and can be trained to recognise other languages. (Tesseract, 2012)

Tesseract would only be suitable for a desktop application as there is no support for mobile devices

### Tess-two

Tess-two is *“…A fork of Tesseract Tools for Android (tesseract-android-tools) that adds some additional functions. Tesseract Tools for Android is a set of Android APIs and build files for the Tesseract OCR and Leptonica image processing libraries.”*

This toolkit is suitable for Android because it is a superset of the Tesseract project that was made using the NDK for Android. This can be coded in Java and C++.

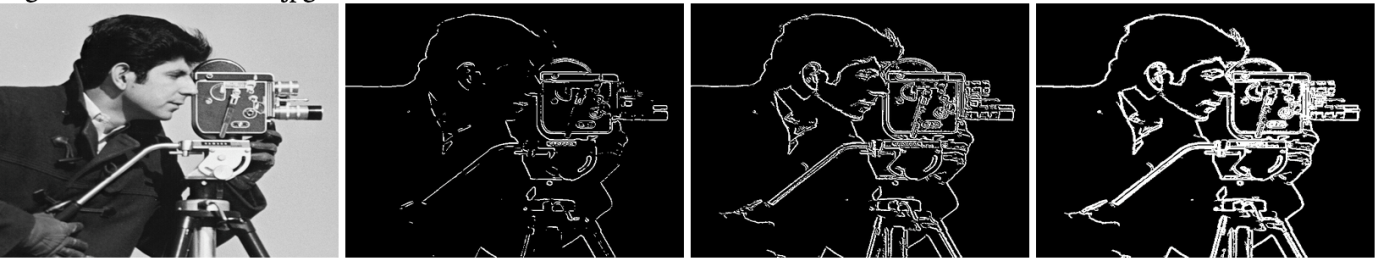
## Edge Detection Algorithms

### Edge Detection

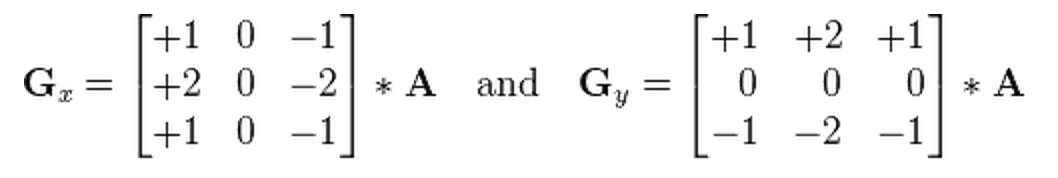
#### Sobel Edge Detection

The Sobel Method of edge detection is *“…used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high frequency variations in the image.”* (Contributors, 2013)

Below is an example image that demonstrated the Sobel method with increasing power:

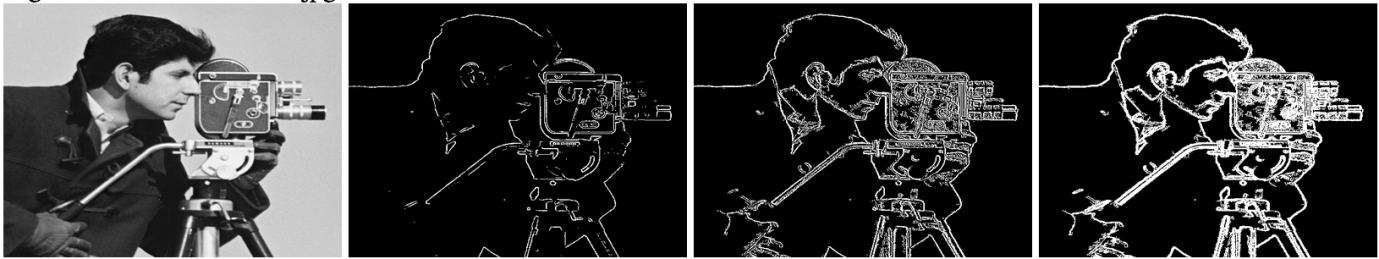
 (School of Information: Sciences, Technology and Arts, University of Arizona)

The formula that is convolved is:

 (Contributors, 2013)

#### Roberts Edge Detection

This method is to approximate the gradient of an image through discrete differentiation. This is achieved by getting the sum of the diagonal neighbourhood pixels squared value. (Wikipedia contributors, 2013) Below is an image showcasing the effect of this method at increasing strengths:

 (School of Information: Sciences, Technology and Arts, University of Arizona)

These two matrices are convolved over the image matrix:

 (Wikipedia contributors, 2013)

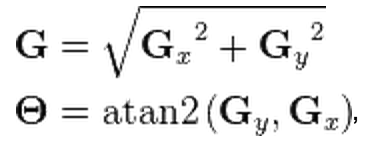
#### Canny Edge Detection

The Canny filter is a multi-stage edge detection algorithm. It was developed by John F. Canny in 1986 and his main objectives while creating this algorithm were: (Wikipedia Contributors, 2013)

* Good detection
  + The algorithm should mark as many real edges in the image as possible
* Good localisation
  + Edges marked should be as close to the original as possible
* Minimal response
  + Image noise should not create extra edges

The stages involved in the Canny Edge Detection algorithm are:

* Noise Reduction
  + This algorithm is sensitive to image noise (it will create edges that are in fact noise in the image)
  + It is necessary so remove this noise to create an accurate detection
  + A Gaussian filter is convolved with the image to reduce the amount of noise in the image
* Find the intensity gradient of the image
  + *“An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (Roberts, Prewitt, Sobel for example) returns a value for the first derivative in the horizontal direction (Gx) and the vertical direction (Gy).”* (Wikipedia Contributors, 2013)
  + From this the edge gradient and direction can be determined:

 (Wikipedia Contributors, 2013)

* Non maximum suppression
  + *“Given estimates of the image gradients, a search is carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction. In many implementations, the algorithm categorizes the continuous gradient directions into a small set of discrete directions, and then moves a 3x3 filter over the output of the previous step (that is, the edge strength and gradient directions). At every pixel, it suppresses the edge strength of the [centre] pixel (by setting its value to 0) if its magnitude is not greater than the magnitude of the two [neighbours] in the gradient direction.”* (Wikipedia Contributors, 2013)
* Tracing edges through the image and hysteresis thresholding
  + *“Thresholding with hysteresis requires two thresholds – high and low. Making the assumption that important edges should be along continuous curves in the image allows us to follow a faint section of a given line and to discard a few noisy pixels that do not constitute a line but have produced large gradients. Therefore we begin by applying a high threshold. This marks out the edges we can be fairly sure are genuine. Starting from these, using the directional information derived earlier, edges can be traced through the image. While tracing an edge, we apply the lower threshold, allowing us to trace faint sections of edges as long as we find a starting point.”*

The first image below is the original image, the second is the image that has the Gaussian filter applied to it, the result is an image that is slightly blurred and the noise is reduced. The third has the edge detection completed:



## Character Segmentation

### Character Extraction using line scanning

Character extraction using line scanning involves the procedures:

* Retrieve the binary image
  + Otsu’s Method

Otsu’s method is used to reduce an image from grey scale to its binary equivalent. It is one of the most popular methods for Thresholding. (Yoon, 2013) Some draw backs of this method are that it may not perform at its highest level in variable lighting.

* Extract separate characters

This method scans the image from left to right to find the start and end points of a character and saves the points. Then is scans top-down to find the remainder of the coordinates. (Singh)

# Recommendations

## Platform and Language

The Platform that is recommended is Android. There are numerous reasons for this choice:

* Android is inexpensive to develop an app on
  + There is a one off cost of $25 to submit as many apps as the user wishes to the Play Store
  + Android devices are inexpensive to purchase. There are many capable devices for under €200 off contract that would be suitable for the needs of this project
  + The development kit is free and can run on any desktop platform
    - With Android, an inexpensive computer should be more than capable to develop apps on
* The communication between apps will prove beneficial for this project
  + Easy access to the camera application to retrieve images
* The market share of Android is increasing gradually in Ireland

## Cloud Service

The recommendation for the cloud service is Google Cloud Datastore

* The Google Cloud Datastore is maintained by one of the most influential companies in the industry
* It offers a larger free storage amount that its competitors
  + This is extremely beneficial for start-up companies who wish to reduce initial capital expenditure
* Multiple languages are supported giving the developer a good choice of technologies at their disposal

## Image Manipulation Library

OpenCV is the choice for the image manipulation library:

* OpenCV offers a library designed for Android
* It offers developers a choice of technologies to work with
  + Java, C/C++, Python, MATLAB
* It has a library with the algorithms necessary for edge detection built in

## Optical Character Recognition

The recommended choice for Optical Character Recognition tool is Tess-two:

* Tess-two contains all the functionality of the original Tesseract library but with an additional layer that allows for development on the Android platform using the NDK and C++

## Edge Detection Algorithm

The edge detection algorithm of choice is the Canny method of edge detection:

* This method, by visual inspection, gives the clearest lines
  + Due to the noise reduction with the Gaussian filter, less artefacts are detected as edges

## Character Segmentation

It is recommended that the Character extraction using line scanning to be used because of its ease of use and good detection rates.

# Conclusion

As a conclusion the following components will be used in this implementation:

* Android for the platform
* Google Cloud Datastore for the cloud database storage
* OpenCV for the image manipulation libraries
  + This will only be used for prototyping to get the main functionality of the App functional
  + The necessary libraries and classes will be hand coded to replace OpenCV
* For optical character recognition, tess-two will be implemented
* The Canny edge detection method will be implemented for the needs of this app

The method used for extracting a number plate is as follows:

* In the camera, the user is presented with rectangle with a border that is the same size ratio as a licence plate.
* The user aligns the licence plate within the highlighted rectangle and releases the shutter on the camera
* Hence, the app knows the location of the number plate on the image for further processing
* The Character extraction using line scanning will be used to separate the characters

The plan for development is to create an initial skeleton program and add on the following components individually:

* Add the functionality of the camera
* Add a boundary box for the licence plate and crop the image from this box
* Implement OpenCV for prototyping canny edge detection
* Extract the different characters from the licence plate
* Run the optical character recognition using tess-two for prototyping
* Implement the cloud functionality

# Glossary

**Architecture**: the specification of the relation between parts of a computer system 5

**Capital expenditure**: The cost of long term improvements 4

**Convolved**: combine (one function or series) with another by forming their convolution. 13

**Convolving**: combine (one function or series) with another by forming their convolution. 12

**Development kit**: A software development kit is typically a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform 5

**Fork**: Clone of a software product 11

**Fragmentation**: is a phenomenon in which storage space is used inefficiently, reducing capacity and often performance 5

**Framework**: a software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software. 10

**Gradient**: the vector formed by the operator ∇ acting on a scalar function at a given point in a scalar field. 12

**Image noise**: mage noise is random (not present in the object imaged) variation of brightness or color information in images 13

**JSON**: JSON, or JavaScript Object Notation, is a text-based open standard designed for human-readable data interchange. 9

**NDK for Android**: The Native development kit that allows Android apps to be programmed in C/C++ 11

**Operating system**: the low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals. 3

**Optical Character Recognition**: Method to read text from a digital image 1

**Platforms**: A Base on which to develop an app 1

**Real time applications**: eal-time computing (RTC), or reactive computing, is the study of hardware and software systems that are subject to a "real-time constraint"— e.g. operational deadlines from event to system response. 10

**Scalable**: the ability to resize when needed 9

**Schema-less**: database that contains no fixed table structure 9

**Screen resolution**: The display resolution of a digital television, computer monitor or display device is the number of distinct pixels in each dimension that can be displayed 6

**Software**: the programs and other operating information used by a computer. 3

**Solid state drives**: A solid-state drive is a data storage device using integrated circuit assemblies as memory to store data persistently. 9

**SQL**: SQL is a special-purpose programming language designed for managing data held in a relational database management system 9

**Toolkits**: a set of software tools 5

# Bibliography

Apple. (n.d.). *Purchase and Activation.* Retrieved November 19, 2013, from Apple: https://developer.apple.com/support/ios/purchase-and-activation.html

Association, F. S. (2008, August). *Automated License Plate Recognition Investment Justification and Purchasing Guide .* Retrieved 11 5, 2013, from Federal Signal: http://www.federalsignal.com/pdf/whitePapers/White%20Paper%20-%20ALPR%20Investment%20Justification%20and%20Purchasing%20Guide.pdf

CloudDB. (n.d.). *Cloud Database.* Retrieved November 20, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Cloud\_database

Contributors, W. (2013, November 18). *Sobel Operator.* Retrieved November 22, 2013, from Wikipedia: http://en.wikipedia.org/w/index.php?title=Sobel\_operator&oldid=582189935

Darcy, C. a. (2009, October 31). *The Top Five Business Benefits of Android.* Retrieved November 12, 2013, from Network World: http://www.networkworld.com/community/node/46936

DesktopShare. (n.d.). *Statcounter Global statistics.* Retrieved November 12, 2013, from Statcounter: http://gs.statcounter.com/#os-IE-monthly-201305-201310

DynamoDB. (n.d.). *AWS Free Usage Tier.* Retrieved November 21, 2013, from Amazon AWS: http://aws.amazon.com/free/

DynamoDBWiki. (n.d.). *DynamoDB.* Retrieved November 21, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Amazon\_DynamoDB

GoogleCloud. (n.d.). *Google Cloud Datastore.* Retrieved November 21, 2013, from Google Cloud Platform: https://cloud.google.com/products/datastore

Herman. (20131, May 5). *Can I write native iPhone, android , windows ,blackberry apps using Python?* Retrieved November 22, 2013, from Stackoverflow: http://stackoverflow.com/questions/16382246/can-i-write-native-iphone-android-windows-blackberry-apps-using-python

Immense. (2013, August 26). *Imense.* Retrieved November 19, 2013, from Android Play Store: https://play.google.com/store/apps/developer?id=Imense

Ingraham, N. (2013, October 22). *Apple announces 1 million apps in the App Store, more than 1 billion songs played on iTunes radio.* Retrieved November 19, 2013, from The Verge: http://www.theverge.com/2013/10/22/4866302/apple-announces-1-million-apps-in-the-app-store

Koetsier, J. (2013, July 17). *Comparing Apples and Googles: The App Store vs. Google Play (infographic).* Retrieved November 19, 2013, from Venturebeat: http://venturebeat.com/2013/07/17/comparing-apples-and-googles-the-app-store-vs-google-play-infographic/

MobileShare. (n.d.). *Statcounter Global Stats.* Retrieved November 12, 2012, from StatCounter: http://gs.statcounter.com/#mobile\_os-IE-monthly-201211-201310

MobileVsDesktop. (n.d.). *Statcounter Gobal Stats.* Retrieved November 12, 2013, from Statcounter: http://gs.statcounter.com/#mobile\_vs\_desktop-IE-monthly-201305-201310

MongoDBWiki. (n.d.). *MongoDb.* Retrieved November 21, 2013, from Wikipedia: http://en.wikipedia.org/wiki/MongoDB

OpenCV. (2013). *About.* Retrieved November 22, 2013, from OpenCV: http://opencv.org/about.html

OpenCV. (n.d.). *OpenCV Main Page.* Retrieved November 22, 2013, from OPenCV: opencv.org

School of Information: Sciences, Technology and Arts, University of Arizona. (n.d.). *Edge Detection Algorithms.* Retrieved November 22, 2013, from arizona.edu: http://vision.sista.arizona.edu/nvs/research/image\_analysis/edge.html

SimpleCV. (2013). *SimpleCV Tutorial.* Retrieved November 22, 2013, from SimpleCV: http://tutorial.simplecv.org/en/latest/

Smith, C. (2013, June 21). *Apple publishes iOS fragmentation chart on its website, target’s Google’s Android.* Retrieved Novenber 19, 2013, from Android Authority: http://www.androidauthority.com/apple-ios-fragmentation-chart-targets-google-android-233600/

Support, A. (n.d.). *Get the Android SDK.* Retrieved November 12, 2013, from developer.android.com: http://developer.android.com/sdk/index.html

Team, A. D. (n.d.). *Xcode.* Retrieved November 12, 2013, from itunes.apple.com: https://itunes.apple.com/us/app/xcode/id497799835?ls=1&mt=12

Team, A. S. (n.d.). *Android Studio.* Retrieved November 12, 2013, from developer.android.com: http://developer.android.com/sdk/installing/studio.html

Tesseract. (2012, October 28). *Tesseract OCR ReadMe.* Retrieved November 22, 2013, from Google Projects: https://code.google.com/p/tesseract-ocr/wiki/ReadMe

Tiobe. (2013, November). *TIOBE Programming Community Index for November 2013.* Retrieved November 12, 2013, from Tiobe: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Wikipedia Contributors. (2013, November 9). *Canny Edge Detection.* Retrieved November 22, 2013, from Wikipedis: http://en.wikipedia.org/wiki/Canny\_edge\_detector

Wikipedia contributors. (2013, June 1). *Roberts Cross.* Retrieved November 22, 2013, from Wikipedia: http://en.wikipedia.org/w/index.php?title=Roberts\_cross&oldid=557862827