

Find The Word Mobile Application Research Document

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY

CARLOW

At the Heart of South Leinster

Student Name: Daniel Duda
Student Number: C00208007
Supervisor: Paul Barry
Date: 25.10.2018

Introduction	2
Project purpose	2
Similar Applications	3
CamScanner - Phone PDF Creator	3
Google Docs	4
OCR	5
Digital image	5
Image pre processing	6
Machine Learning	9
Artificial Neuron (Perceptron)	9
Convolutional Neural Network	14
Google Firebase ML Kit text recognition API	17
TensorFlow	17
Tesseract	17
Mobile Technologies	18
Native	18
Android	18
IOS	18
Hybrid	18
Apache Cordova	19
Ionic	19
Xamarin	19
React Native	19
Web Applications	19
References	21

Introduction

This document provides information about research that was carried for this project. First section of this document provide information about free to use and most efficient character recognition technologies that are being used currently in the market. Second section will provide information about mobile technologies that can be used for this project. Third section will provide information about technologies that are used for word search in text documents. The last section will have information about already existing application similar to this project.

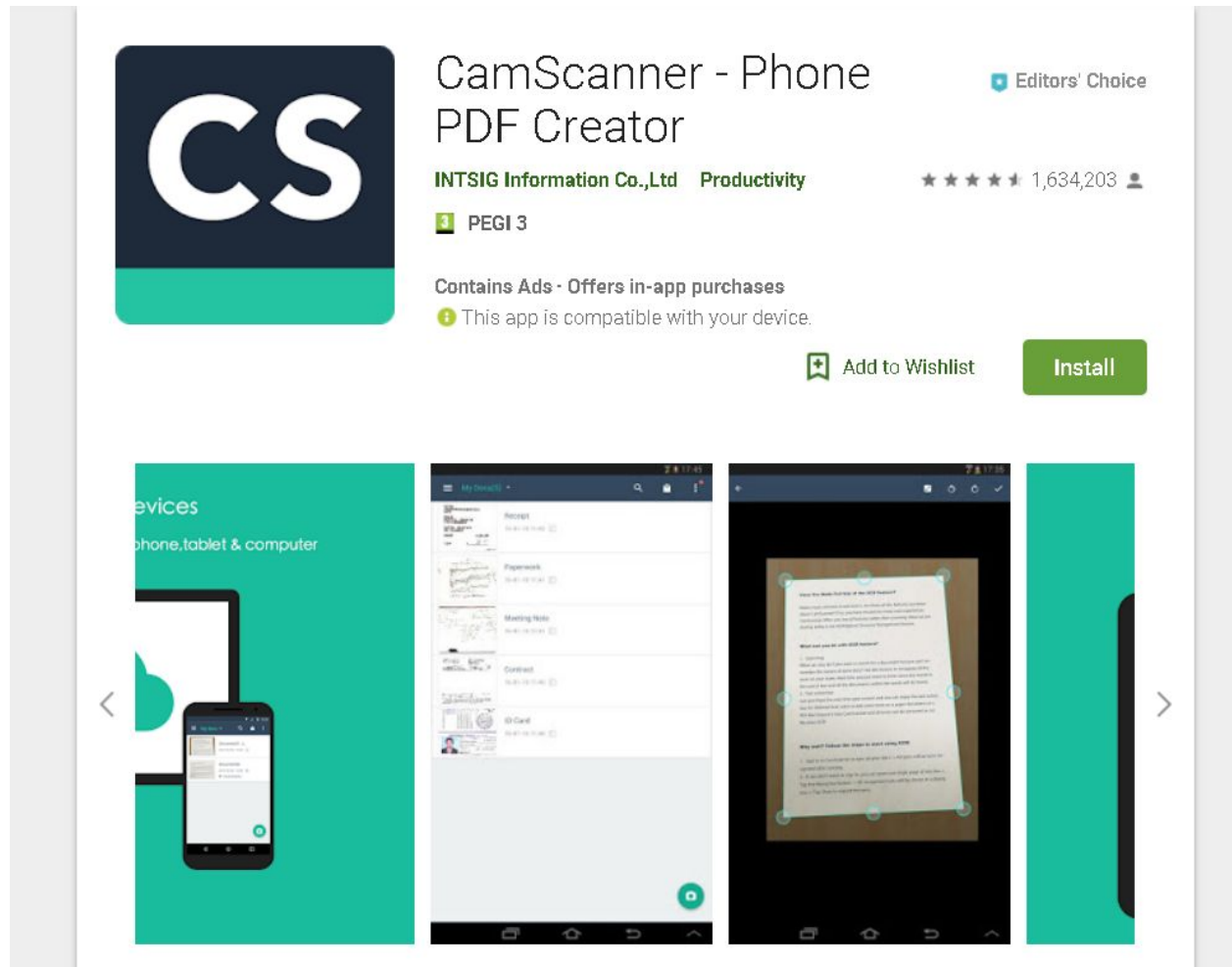
Project purpose

I have found idea for this project while studying, For example the book (a book with big pages and small font), I was reading the book and read in internet at the same time and after i read few pages in a book I had to search in internet page for something related to text in book so I just pressed Ctrl + F and I was able to easy find for what I was looking for, then when I had to find something in the book related to the internet (that was that aha moment when I realized that I understood something important I just need that missing piece from the book!) I had to find where was that piece of text in the book, which if the pages are big and the font is small can be not so easy and can take let say If you need to search four pages it can take two minutes or more and when eyes are tired it will take longer, the app will quickly find the word by using smart phone/tablet camera. The application will convert recorded/pictured book page to the text and it will highlight where the searched word is on the phone.

Similar Applications

CamScanner - Phone PDF Creator

<https://play.google.com/store/apps/details?id=com.intsig.camscanner>



The screenshot shows the Google Play Store listing for the application "CamScanner - Phone PDF Creator". The app icon is a dark blue square with the white letters "CS" and a green horizontal bar at the bottom. The title "CamScanner - Phone PDF Creator" is displayed in a large, dark font. To the right of the title, it says "Editors' Choice" with a blue shield icon. Below the title, the developer is listed as "INTSIG Information Co.,Ltd" and the category is "Productivity". The app has a rating of 4.5 stars and 1,634,203 reviews. A green badge indicates a PEGI 3 rating. A warning icon and text state "Contains Ads - Offers in-app purchases". A green plus icon and text state "This app is compatible with your device." There are two buttons: "Add to Wishlist" and "Install". Below the main listing are four preview images: the first shows the app's interface on a phone, the second shows a list of scanned documents, the third shows a document being scanned with a camera overlay, and the fourth is a solid green background.

Very popular text recognition application with free and paid options. Local and cloud scanning available.

Google Docs

Google drive have OCR option. Picture must be save od google drive then opened with google docs. Google OCR will copy the text from image automatically.

OCR

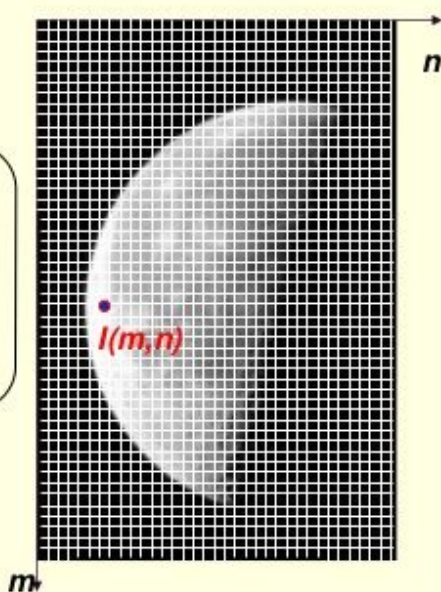
OCR(Optical character recognition) - "Is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo or from subtitle text superimposed on an image."[WIKI18]

Digital image

Is represented as a matrix of pixels. Pixel - "In digital imaging, a pixel, dots, or picture element is a physical point in a raster image, or the smallest addressable element in an all points addressable display device; so it is the smallest controllable element of a picture represented on the screen."[WIKIPIXEL 18]

Digital Image Representation

- A digital image can be represented as a two-dimensional matrix.

$$I(mn) = \begin{pmatrix} i(1,1) & i(1,2) & \dots & i(1, n-1) \\ i(2,1) & i(2,2) & \dots & i(2, n-1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ i(m-1,1) & i(m-1,2) & \dots & i(m-1, n-1) \end{pmatrix}$$


- Each element is called a *pixel* (picture element).
- A color (RGB) image is represented by a 3-dimensional matrix $I(m \times n \times 3)$

CSC447: Digital Image Processing Prof. Dr. Mostafa GadaHagg. 9

[SLIDE16]

“The RGB color model is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.”[WIKI RGB 18] The RGB representation can be seen as matrix of $m \times n \times 3$ for example: 100x100 pixels x (0-255, 0-255, 0-255) where numbers from 0 to 255 represents intensity of each colour. The white colour have value of R-255 G-255 B-255, black colour R-0, G-0, B-0 and red colour R-255 G-0 B-0.

Image pre processing

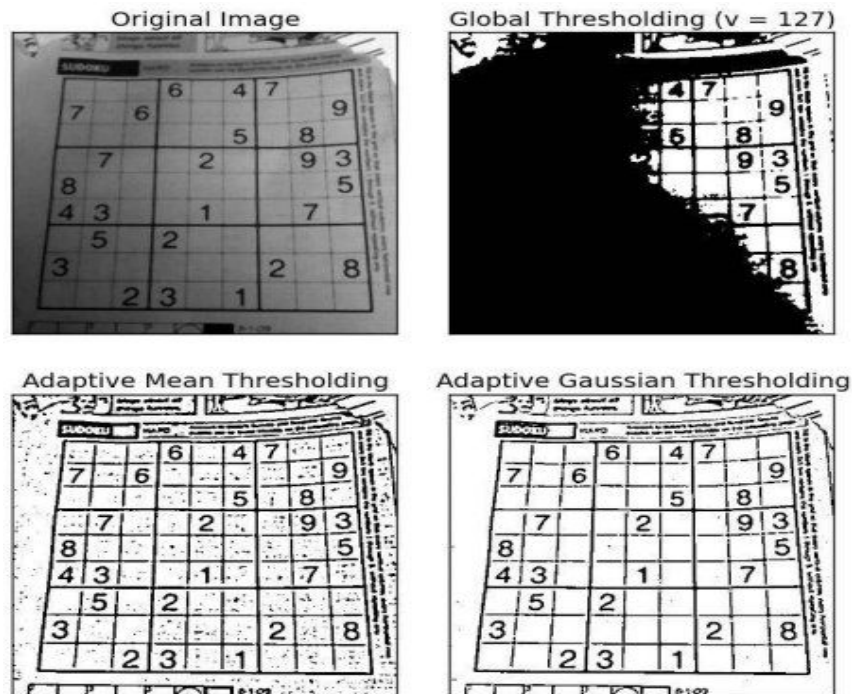
In order to read text from image with highest accuracy possible the image have to be prepared for it especially if the image have bad quality.

- Resize image if needed
 - Some OCR engines works best with particular image size it depend on OCR engine.

- Grayscale the image
 - To change image to grayscale (RGB) Red Green Black values of image are being removed leaving only one value for each pixel, there is several ways to achieve this. Human eyes are more sensitive to green colors and humans have more receptors for red than blue colors the method weighted coefficients is the most popular method for changing image to grayscale.
 - Weighted coefficients: $GRAY = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B$ [IMD15]

- Binarization/Thresholding
 - Convert image to black and white. Method of obtaining a binary image from color image or from shades of gray in image. It determines a

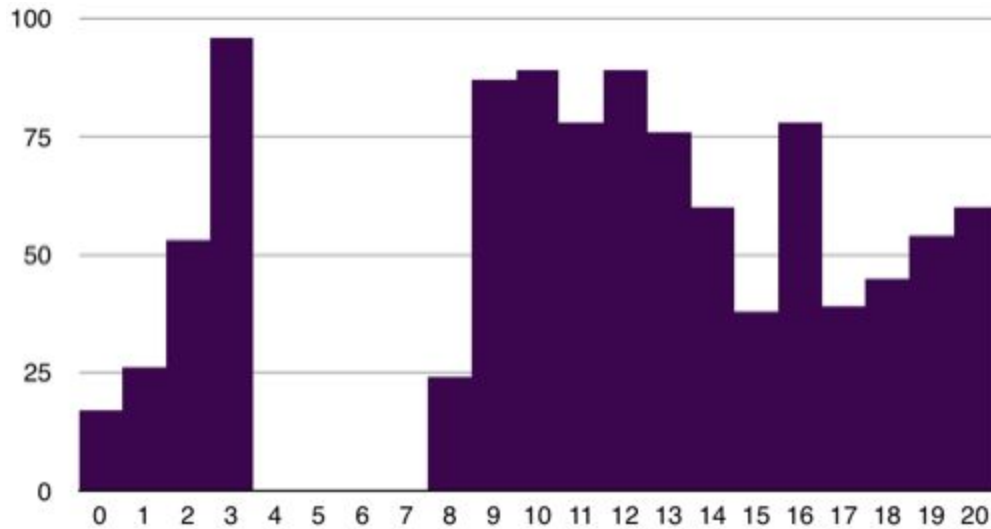
brightness threshold for a given image and pixels lighter than the designated threshold receive one value, and darker ones the second value.



[OPEN18]

Along with methods showed in picture above and the other several methods the popular method called by its author name Otsu is used for image thresholding. This method use image bimodal histogram (bimodal histogram have two peaks) this method use the threshold value between this peaks

Image histogram is a graph which show image pixel intensity. Like in on the graph below the x axis show the intensity value of the colour (grayscale in this case) and the y axis show how many of each pixel with particular colour/grayscale there is in image.



[VERPIT18]

- Noise removal - “Noise is random variation of brightness or colour in an image, that can make the text of the image more difficult to read.” [TESSERACT 18]
Some of the methods used to deal with noise in the images:
 - Averaging(Box Filter) this method simply changes value of each pixel based on the value of the picture itself and value of its neighbour pixels. By using convolution each pixel value is being multiplied by matrix(kernel) showed below then sum up values and divide by the size of matrix which is nine in this case.

Box filter

$1/9$

1	1	1
1	1	1
1	1	1

[COLL07]

- Gaussian Blurring - also known as Gaussian smoothing - image modification using the Gauss filter. It is a commonly used function of graphics software, used mostly to reduce noise and distortions in the image or to blur details. Works similarly to Box Filter but uses different kernel.

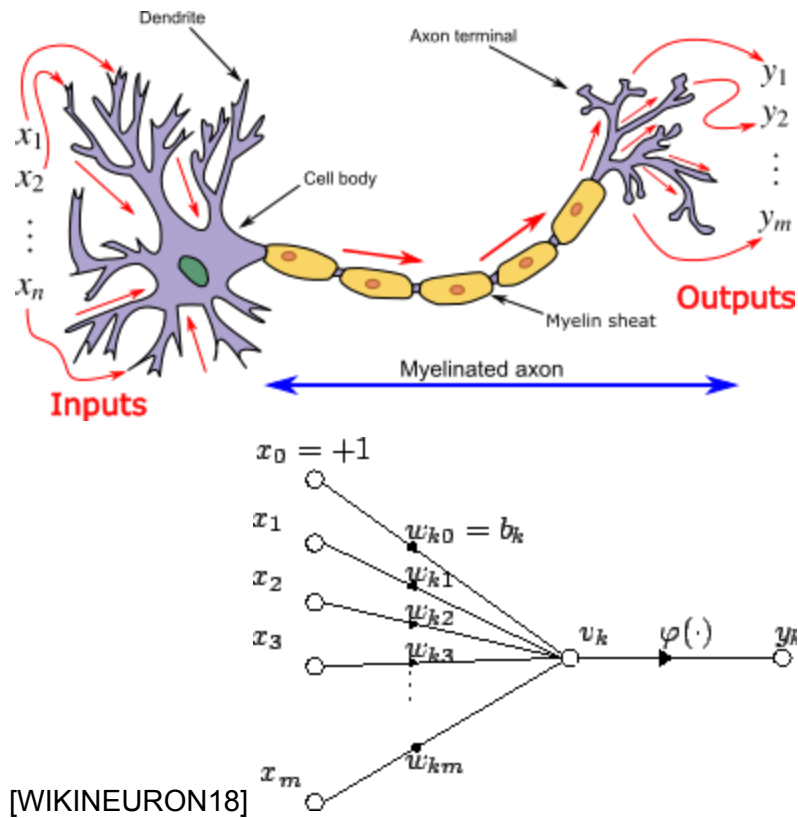
- Edge detection
 - Canny edge detection

- Image rotation - Images which are not straight can cause problems with OCR line detection. One of the most popular techniques applied to detect and correct image skew is:
 - Hough Transform

Machine Learning

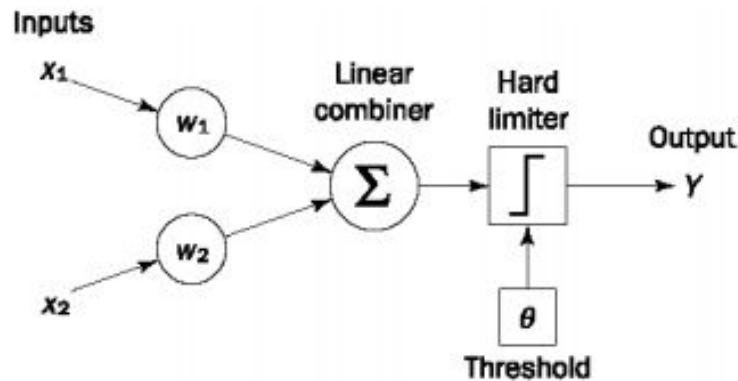
Artificial Neuron (Perceptron)

OCR systems use Neural Networks for character and word recognition. Neural Network is a set of single artificial neurons. Artificial neuron is based on biological neuron.



0, $x_1 \dots$ in the picture above mimic the dendrites in biological neuron which are responsible for input from other neurons and conduct the input to the cell body where neuron decides if it will be activated or no. Then signal is send through axon to other neurons.

Artificial neuron works in similar way. For example with simple single layer perceptron the x inputs are multiplied by their w weights e.g. $x_0 \times w_0 \dots x_n \times w_n$ then result of this operation is summed (some perceptrons use/add bias value too) and passed to activation function. If the sum is bigger than certain threshold then it output 1 otherwise output is 0. Let consider simple perceptron with two input values x_1 and x_2 and two weights w_1 and w_2 .



[MSLABASIA18]

The steps for training perceptron:

1. Initialise:
 - a. Weights $w_1 = 0.3$, $w_2 = -0.1$
 - b. Learning rate $\alpha = 0.1$
 - c. Threshold $\theta = 0.2$
2. Calculate the sum and actual output Y_p using sum function and activation function:

$$Y(p) = \text{step} \left[\sum_{i=1}^n x_i(p)w_i(p) - \theta \right], // Y(X) = \text{step}[X] = \begin{cases} 1 & \text{if } X \geq 0 \\ 0 & \text{if } X < 0 \end{cases}$$

- a. [MSLABASIA18]
- b. The (p) is the iteration (which training example is going through functions) and n is the number of perceptron inputs
3. Adjust weights if necessary (activation function output):
 - a. $w_i(p + 1) = w_i(p) + \Delta w_i(p)$
 - b. $\Delta w_i(p) = \alpha x_i(p) \times e(p)$
 - i. $e(p)$ is the error calculated by $e(p) = Y_d(p) - Y(p)$
 1. $Y(d)$ is desired output
 2. Y_p is actual output

Presented below table of learning algorithm for perceptron learning logic AND gate:

Epoch	Inputs		Desired output	Initial weights		Actual output	Error	Final weights	
	x_1	x_2	Y_d	w_1	w_2	Y	e	w_1	w_2
1	0	0	0	0.3	-0.1	0	0	0.3	-0.1
	0	1	0	0.3	-0.1	0	0	0.3	-0.1
	1	0	0	0.3	-0.1	1	-1	0.2	-0.1
	1	1	1	0.2	-0.1	0	1	0.3	0.0
2	0	0	0	0.3	0.0	0	0	0.3	0.0
	0	1	0	0.3	0.0	0	0	0.3	0.0
	1	0	0	0.3	0.0	1	-1	0.2	0.0
	1	1	1	0.2	0.0	1	0	0.2	0.0
3	0	0	0	0.2	0.0	0	0	0.2	0.0
	0	1	0	0.2	0.0	0	0	0.2	0.0
	1	0	0	0.2	0.0	1	-1	0.1	0.0
	1	1	1	0.1	0.0	0	1	0.2	0.1
4	0	0	0	0.2	0.1	0	0	0.2	0.1
	0	1	0	0.2	0.1	0	0	0.2	0.1
	1	0	0	0.2	0.1	1	-1	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1
5	0	0	0	0.1	0.1	0	0	0.1	0.1
	0	1	0	0.1	0.1	0	0	0.1	0.1
	1	0	0	0.1	0.1	0	0	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1

[MSLABASIA18]

The epoch is one iteration over all training examples. The first epoch calculation explained:

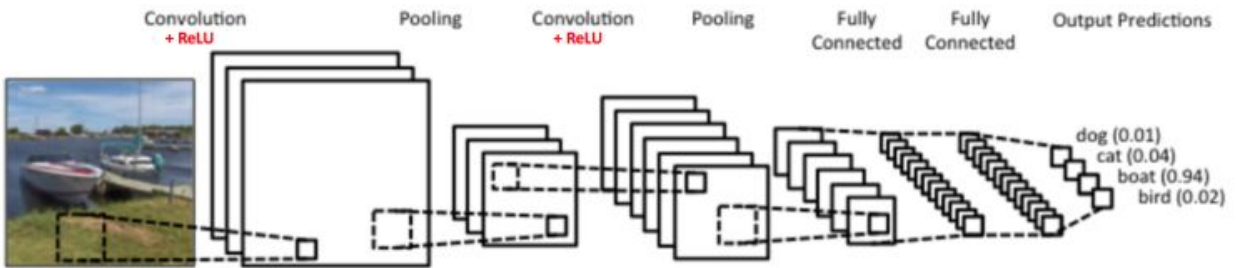
1. $p = 1, x_1 = 0, x_2 = 0, Y(d) = 0, w_1 = 0.3, w_2 = -0.1$
 - a. $Y(p) = \text{sum}(x_1 \times w_1 + x_2 \times w_2) - \theta$
 - i. $= (0 \times 0.3 + 0 \times (-0.1)) - 0.2 = -0.2$
 - b. $\text{step}(-0.2) = \quad \Rightarrow 0 \quad (-0.2 < 0)$
 - i. desired output $Y(d) = 0$ equals $Y(p) = 0$ actual output hence no weights adjusting
2. $p = 2, x_1 = 0, x_2 = 1, Y(d) = 0, w_1 = 0.3, w_2 = -0.1$
 - a. $Y(p) = (0 \times 0.3 + 1 \times (-0.1)) - 0.2 = -0.3$
 - b. $\text{step}(-0.3) = 0 \quad \Rightarrow (-0.3 < 0)$

- i. desired output $Y(d) = 0$ equals $Y(p) = 0$ actual output hence no weights adjusting
3. $p=3$, $x_1 = 1$, $x_2 = 0$, $Y(d) = 0$, $w_1 = 0.3$, $w_2 = -0.1$
- $Y(p) = (1 \times 0.3 + 0 \times (-0.1)) - 0.2 = 0.1$
 - $\text{step}(0.1) = 1 \Rightarrow (0.1 > 0)$
 - desired output $Y(d) = 0$ not equals $Y(p) = 1$ actual output so weights adjusting :
 - $w_1(p+1) = w_1(p) + \Delta w_1(p)$
 - $= 0.3 + (-0.1) = 0.2$
 - $\Delta w_1(p) = \alpha \times x_1(p) \times e(p)$
 - $= 0.1 \times 1 \times (-1) = -0.1$
 - $e(p) = Y_d(p) - Y(p)$
 - $= 0 - 1 = -1$
 - $w_2(p+1) = w_2(p) + \Delta w_2(p)$
 - $= -0.1 + 0 = -0.1$
 - $\Delta w_2(p) = \alpha \times x_2(p) \times e(p)$
 - $= 0.1 \times 0 \times (-1) = 0$
 - $e(p) = Y_d(p) - Y(p)$
 - $= 0 - 1 = -1$
4. $p=4$, $x_1 = 1$, $x_2 = 1$, $Y(d) = 1$, $w_1 = 0.2$, $w_2 = -0.1$
- $Y(p) = (1 \times 0.2 + 1 \times (-0.1)) - 0.2 = -0.1$
 - $\text{step}(-0.1) = 0 \Rightarrow (-0.1 < 0)$
 - desired output $Y(d) = 1$ not equals $Y(p) = -0.1$ actual output so weights adjusting :
 - $w_1(p+1) = w_1(p) + \Delta w_1(p)$
 - $= 0.2 + 0.1 = 0.3$
 - $\Delta w_1(p) = \alpha \times x_1(p) \times e(p)$
 - $= 0.1 \times 1 \times 1 = 0.1$
 - $e(p) = Y_d(p) - Y(p)$
 - $= 1 - (-0.1) = 1.1$
 - $w_2(p+1) = w_2(p) + \Delta w_2(p)$
 - $= -0.1 + 0.1 = 0.0$
 - $\Delta w_2(p) = \alpha \times x_2(p) \times e(p)$
 - $= 0.1 \times 1 \times 1 = 0.1$
 - $e(p) = Y_d(p) - Y(p)$
 - $= 1 - (-0.1) = 1.1$

Epoch one ends with with weights adjusted. This calculations showed how early perceptrons worked.

Convolutional Neural Network

Convolutional Neural Network is deep artificial neural network that are used to classify images.



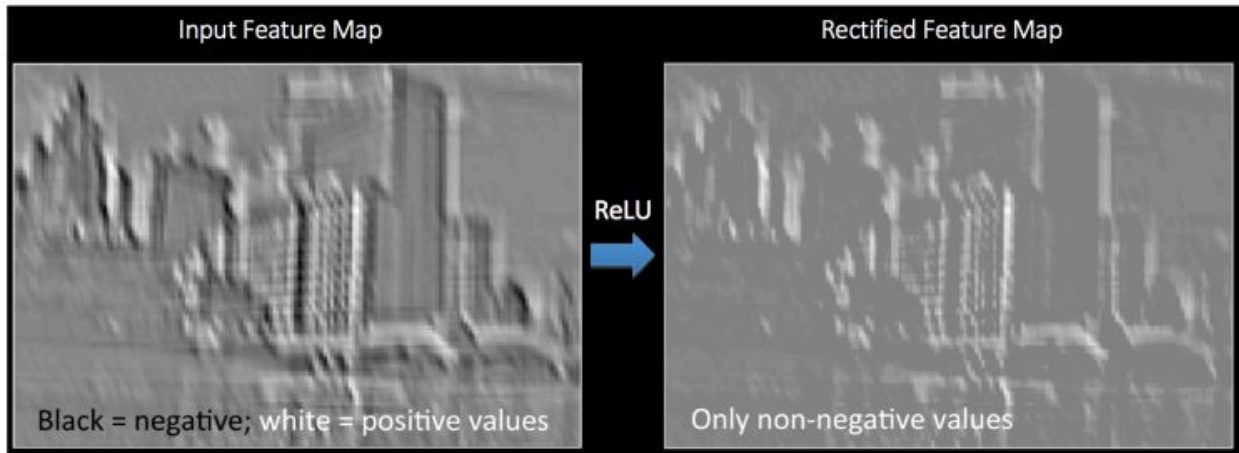
[KARN18]

The convolution layer extract features from image using various filters(kernels) such as Box Filter for noise removal. Each filter produce different features map as shown on picture below.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

[KARN18]

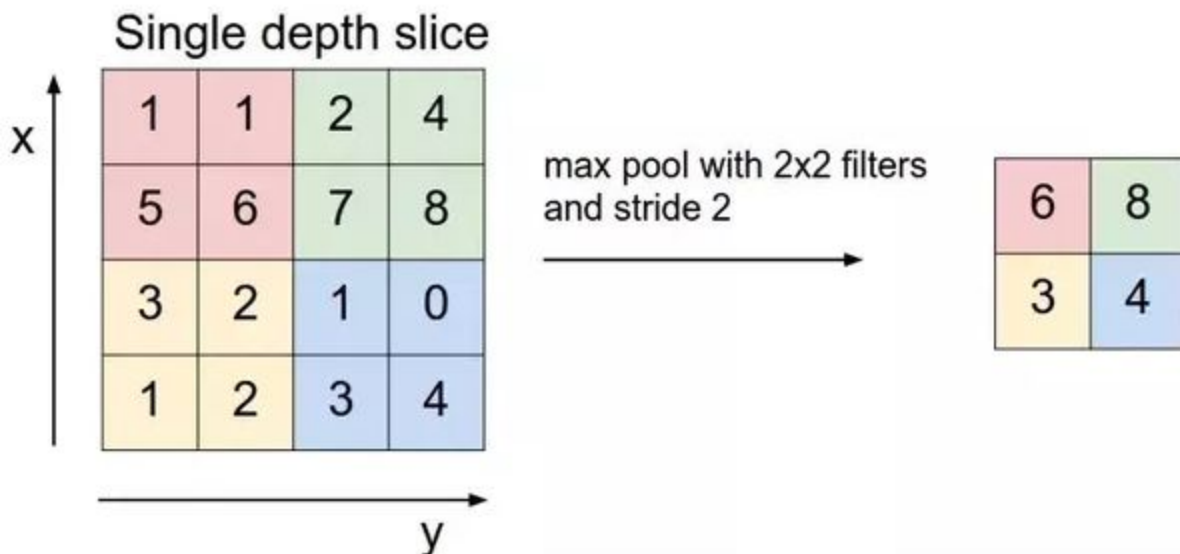
The ReLU operation is applied for every pixel in the feature maps. ReLU change every negative pixel value to zero. ReLU is used to bring non linearity in CNN. Using ReLU function produce so called rectified feature map.



[KARN18]

Pooling

Pooling reduce the size of image. Spatial Pooling lessen the diversity in every feature map and preserve the most important information. There is few different types of spatial pooling such as max, average or sum pooling.



[KARN18]

The next step is to classify the image using deep(many layers) set of artificial neurons (perceptrons) which are connected to each other e.g outputs of one layer of perceptrons are connected to inputs of other/next layer of perceptrons. The images are being classified by training algorithm which provide image and class assigned to each image. In high level if image is classified wrongly then the backpropagation algorithm is applied to adjust the weights on artificial neuron and the filters in first layers.

Google Firebase ML Kit text recognition API

Google Machine Learning SDK which is available for IOS and Android and can be used for tasks such as:

- Face Recognition
- Text Recognition
- Labeling Images
- Landmark Recognition
- Barcode detection

Google ML Kit API can be used on the device without network access or in the cloud with higher accuracy. ML Kit is good for developers who don't have previous experience with Machine Learning. Developers who have experience with Machine Learning can host their own TensorFlow Lite model.

TensorFlow

Open Source library for heavy numerical computation created by google and released in 2015. The word tensor means an matrix and matrices are used in TensorFlow. User provide input matrices then they “flow” through mathematical operations and create output. TensorFlow can be run on many different platforms e.g. desktop, cloud or mobile. Can be run on GPU and CPU. The base layer of TensorFlow is written in C++. It provides many apis for Different programming languages. TensorFlow can be used to create and train models

Tesseract

Is a free OCR software engine. Sponsored by google since 2006. Images needs to be in good quality therefore before extracting text from image, image must be cleaned from noise first.

Mobile Technologies

Native

The Native mobile applications are developed for particular operating system and will work only on that system. These applications have much better performance than hybrid applications. Disadvantage is that application must be developed from scratch for every operating system.

Android

Operating system based on Linux kernel used in mobile devices such as smartphones and tablets. Developed by Android Inc. bought by Google in 2005, currently owned by Open Handset Alliance. At the moment it is the most popular mobile system in the world. Its primary parts are licensed under the Apache License. Most popular IDE for applications development is Android Studio. Official programming language is Java and new programming language Kotlin. Android supports C/C++ programming languages. Applications development can be done on Windows, Mac OS or Linux computers.

iOS

Operating system for mobile devices created by Apple Inc. such as: iPhone, iPod touch and iPad. The current name has been functioning since June 2010, previously the system was known as the iPhone OS. This system is based on the Mac OS operating system. IDE for applications development is called XCode. Programming languages are Objective C and Swift.

In order to develop applications for iOS developer must have Mac OS original computer.

Hybrid

Hybrid applications are developed for Android and iOS operating systems at the same time so it is write once work everywhere. These applications have lower performance than native applications and depend on what framework is used some code cannot be reused e.g. UI have to be done twice for Android and iOS separately.

Apache Cordova

Open source framework that codebase was handed by Adobe(Phonegap) to Apache. Cordova lets developer use HTML, CSS and JavaScript to create native like mobile applications. The application is run in the platform WebView which is different on many different platforms. To access device functionalities developers are using plugins which are written with JavaScript and native code.

Ionic

Free and open source framework that use cordova to port to native platform. It uses AngularJs. It provides many plugins/libraries which help developer build more native look applications than using only cordova.

Xamarin

Microsoft framework for creating mobile applications using Visual Studio and C# programming language. With Xamarin developers create one application that can be run on each target system. Xamarin provides SDK access for Android, iOS and Windows, which is not the case when creating using HTML and JavaScript. In Xamarin there is access to the native API.

React Native

React Native is an Open Source tool created by Facebook. Allowing developer to build applications on iOS and Android systems in JavaScript and JSX syntax. The React Native application is not a typical hybrid application that uses webview to present a mobile site, unlike Apache Cordova instead based on components written in the React syntax, native components are built in Java for Android, or in Objective-C for iOS.

Web Applications

A web application is an application that runs in a web browser(basically it is a website). Such web pages are responsive it means they are customized for different screen sizes such as phone screen or tablet screen. This applications mostly are not static web pages.They communicate with the main server to be able to serve the user and respond to his actions. Internet access in such a case is indispensable, although there are also

applications that are able to temporarily work offline or do not require internet connection at all these applications are called Progressive Web Applications.

Some of the features of Progressive Web Applications:

- The user can add an application icon on the front screen of his device
- Works offline
- Serves push notifications

Front end technologies for web applications:

- HTML
- CSS
- JavaScript

Back end technologies for web applications:

- PHP
- Java
- Python
- c#
- NodeJs
- .NET

References

[WIKI18]Wikipedia contributors.2018.Optical character recognition[online] Available at:
https://en.wikipedia.org/wiki/Optical_character_recognition [Accessed 12 Oct. 2018]

[OPEN18]OpenCV.2018.OpenCV: Image Thresholding[online] Available at:
https://docs.opencv.org/4.0.0-alpha/d7/d4d/tutorial_py_thresholding.html[Accessed 12 Oct. 2018]

[IMD15]IMD.2015.Image Processing in C[online] Available at:
<http://imdepot.blogspot.com>[Accessed 14 Oct. 2018].

[TESSERACT18]Tesseract Ocr.Improve Quality.[online]Available at:
<https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality#noise-removal> [Accessed 15 Oct. 2018].

[WIKIPIXEL 18]Wikipedia contributors.2018.Pixel[online] Available at:
<https://en.wikipedia.org/wiki/Pixel> [Accessed 12 Oct. 2018]

[WIKIRGB18]]Wikipedia contributors.2018.RGB color model[online] Available at:
https://en.wikipedia.org/wiki/RGB_color_model [Accessed 17 Oct. 2018]

[SLIDE16]Mostafa, Mostafa G.M.2016.Digital Image Processing: Digital Image Fundamentals[online]Available at:
<https://www.slideshare.net/MostafaGMMostafa/digital-image-processing-digital-image-fundamentals> [Accessed 20 Oct. 2018]

[VERPIT18]VerpitIT.What is image histogram[online]Available at:
<https://veprit.com/photography-guide/using-histogram/what-is-image-histogram>[Accessed 20 Oct. 2018]

[COLL07] Collins , Robert C.R.2007.Introduction to Computer Vision.Lecture 04: Smoothing[online]Available at:<http://www.cse.psu.edu/~rtc12/CSE486/lecture04.pdf>
[Accessed 19 Oct. 2018]

[SKEW14]Verma, Rajeev & Malik, Latesh. 2014. REVIEW ON SKEW DETECTION AND CORRECTION TECHNIQUES[online]Available at:
https://www.researchgate.net/publication/275138816_REVIEW_ON_SKEW_DETECTIO_N_AND_CORRECTION_TECHNIQUES [Accessed 21 Oct. 2018]

[WIKINEURON18]Wikipedia contributors.2018.Artificial neuron[online] Available at:
https://en.wikipedia.org/wiki/Artificial_neuron [Accessed 16 Oct. 2018]

[MSLABASIA18]Asia University.Algorithm 1: The Perceptron Training Algorithm[online]Available at:
[http://mslab.csie.asia.edu.tw/~ktduc/IS/Lecs/Algorithm%201%20Perceptron%20\(AND%20OR\).pdf](http://mslab.csie.asia.edu.tw/~ktduc/IS/Lecs/Algorithm%201%20Perceptron%20(AND%20OR).pdf) [Accessed 17 Oct. 2018]

[KARN16] Karn, Ujjwal U.K.2016.An Intuitive Explanation of Convolutional Neural Networks[online]Available at:
<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/> [Accessed 22 Oct. 2018]