**IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK BASED MODELS FOR OPTICAL CHARACTER RECOGNITION OF NIGERIAN LICENSE PLATE**

**BY**

**AJEGBEMIKA ENIOLA BOLAFOLUWA**

**14/ENG04/003**

**A REPORT SUBMITTED TO THE**

**DEPARTMENT OF ELECTRICAL, ELECTRONICS AND COMPUTER ENGINEERING**

**AFE BABALOLA UNIVERSITY, ADO-EKITI**

**EKITI STATE, NIGERIA**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF ENGINEERING**

**May, 2019**

# **CERTIFICATION**

This is to certify that **AJEGBEMIKA ENIOLA BOLAFOLUWA** with matriculation number 14/ENG04/003 carried out this research work under my supervision, in partial fulfillment for the award of Bachelor of Engineering in the Department of Electrical, Electronics and Computer Engineering, Afe Babalola University, Ado-Ekiti.

Dr EjidokunT.O **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Supervisor** Signature and Date

Prof DadaJ.O **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Head of Department** Signature and Date

# **DEDICATION**

I dedicate this report to God Almighty in which through him all things were made possible. I would also like to dedicate this report to my parents, Mr. and Mrs. Ajegbemika, and my siblings, Olajire, Oluwatobiloba and Similoluwa.

# **ACKNOWLEDGEMENT**

I would like to acknowledge my maker and creator, Almighty God for giving me the essence of life. Through Him I live and breathe.

A special thanks to my lovely parents Mr. Babatunde Ajegbemika and Mrs. Folashade Ajegbemika for their unequivocal support in the pursuit of a life career, the countless words of encouragement, guidance and support throughout the journey.

Words cannot express my deepest gratitude to my amazing friends, Susannah Akande, Ale Oluwafeyisayo, Olusola Juliet and Emmanuel Uche-Ihesuilor for the continuous support and for being of benefit to my life.

I specially acknowledge my supervisor Dr. Ejidokun for his insightful comments, invaluable suggestions, helpful information, practical advice and unceasing ideas, which helped me tremendously towards the completion of this project.

Lastly, I would also like to express my great appreciation to my lecturers to mention a few, Engr. Ojo, Engr. Ajibade and Engr. Folorunso who have taught, encouraged and impacted in me during my undergraduate study, I appreciate you all.

# **ABSTRACT**

This study implemented and trained various Convolutional Neural Network (CNN) models for optical character recognition of Nigerian license plate with a view to determine the most suitable model.

Two thousand five hundred (2,500) samples of Nigerian license plates were collected trained and validated on various models. The adopted models were LeNet, LeNet-5, CNN4 and CNN6 which varied based on the number of convolutional layers present in each of them. The performance of these models were evaluated and compared based on the performance metrics, which include the accuracy, loss, validation accuracy and validation loss.

The obtained results indicates that the LeNet and LeNet-5 architecture performed better than others with a validation accuracy of 98.6% and 93% respectively. The study concluded that with the high accuracy achieved by both LeNet andLeNet-5, this will help to enhance the efficiency of future implementation of automatic license plate recognition systems in Nigeria.

# **TABLE OF CONTENTS**

**Pages**

[Certification ii](#_Toc7523208)

[Dedication iii](#_Toc7523209)

[Acknowledgement iv](#_Toc7523210)

[Abstract v](#_Toc7523211)

[Table of Contents vi](#_Toc7523212)

[List of Figures viii](#_Toc7523213)

[**CHAPTER ONE: INTRODUCTION** 1](#_Toc7523214)

[1.1 Background of Study 1](#_Toc7523216)

[1.2 Statement of Problem 3](#_Toc7523217)

[1.3 Aim and Objective of Study 3](#_Toc7523218)

[1.4 Justification 3](#_Toc7523219)

[1.5 Scope of Study 4](#_Toc7523220)

[1.6 Organization of Report 4](#_Toc7523221)

[**CHAPTER TWO: LITERATURE REVIEW** 5](#_Toc7523222)

[2.1 History of OCR 5](#_Toc7523224)

[2.2 Fundamentals of OCR Based Systems 5](#_Toc7523225)

[2.2.1 Image acquisition 6](#_Toc7523226)

[2.2.2 Preprocessing 7](#_Toc7523227)

[2.2.3 Character segmentation 10](#_Toc7523228)

[2.2.4 Character recognition 12](#_Toc7523229)

[2.3 Fundamentals of Deep Learning 12](#_Toc7523230)

[2.3.1 Deep Belief Neural Network (DBNN) 13](#_Toc7523231)

[2.3.2 Convolutional neural Network 14](#_Toc7523232)

[2.4 Related Works 19](#_Toc7523233)

[**CHAPTER THREE: METHODOLOGY** 25](#_Toc7523234)

[3.1 Introduction 25](#_Toc7523236)

[3.2 System Requirement 25](#_Toc7523237)

[3.2.1 Hardware requirements 25](#_Toc7523238)

[3.2.2 Software requirements 25](#_Toc7523239)

[3.3 Data Processing 26](#_Toc7523240)

[3.3.1 Data collection 26](#_Toc7523241)

[3.3.2 Data pre-processing 27](#_Toc7523242)

[3.4.1 Le-Net 29](#_Toc7523244)

[3.4.2 Le-Net5 30](#_Toc7523245)

[3.4.3 CNN4 31](#_Toc7523246)

[3.4.4 CNN6 32](#_Toc7523247)

[3.5Testing and Validation33](#_Toc7523248)

[**CHAPTER FOUR: RESULTS AND DISCUSSION** 34](#_Toc7523249)

[4.1 Introduction 34](#_Toc7523251)

[4.1.1 Accuracy 34](#_Toc7523252)

[4.1.2 Loss 36](#_Toc7523253)

[4.1.3 Validation accuracy 39](#_Toc7523254)

[4.1.4 Validation loss 42](#_Toc7523255)

[**CHAPTER FIVE: CONCLUSION AND RECOMMENDATION** 45](#_Toc7523256)

[5.1 Conclusion 45](#_Toc7523258)

[5.2 Recommendation 45](#_Toc7523259)

[**REFERENCES** 46](#_Toc7523260)

[**APPENDIX A: Source Code for Implemented Model** 50](#_Toc7523261)

# **LIST OF FIGURES**

**Pages**

[Figure 2.1: The OCR process 6](#_Toc7536272)

[Figure 2.2: Pictorial representation of convolution process 15](#_Toc7536273)

[Figure 2.3: Pictorial Representation of pooling process 16](#_Toc7536274)

[Figure 2.4: ReLU Layer 17](#_Toc7536275)

[Figure 3.1: Captured license plate 27](file:///G:\Final%20Report.docx#_Toc7536276)

[Figure 3.2: Data acquisition and dataset preparation process 28](#_Toc7536277)

[Figure 3.3: Implemented OCR Model 29](#_Toc7536278)

[Figure 3.4: The LeNet model 30](#_Toc7536279)

[Figure 3.5: The LeNet-5 model 31](#_Toc7536280)

[Figure 3.6: The CNN4 model 32](#_Toc7536281)

[Figure 3.7: The CNN6 model 33](#_Toc7536282)

[Figure 4.1: Accuracy with LeNet architecture 34](#_Toc7536283)

[Figure 4.2: Accuracy with LeNet-5 architecture 35](#_Toc7536284)

[Figure 4.3: Accuracy with CNN4 architecture 35](#_Toc7536285)

[Figure 4.4: Accuracy with CNN6 architecture 36](#_Toc7536286)

[Figure 4.5: Loss with LeNet architecture 37](#_Toc7536287)

[Figure 4.6: Loss with LeNet-5 architecture 37](#_Toc7536288)

[Figure 4.7: Loss with CNN4 architecture 38](#_Toc7536289)

[Figure 4.8: Loss with CNN6 architecture 39](#_Toc7536290)

[Figure 4.9: Validation Accuracy with LeNet architecture 40](#_Toc7536291)

[Figure 4.10: Validation Accuracy with LeNet-5 architecture 40](#_Toc7536292)

[Figure 4.11: Validation Accuracy with CNN4 architecture 41](#_Toc7536293)

[Figure 4.12: Validation Accuracy with CNN6 architecture 41](#_Toc7536294)

[Figure 4.13: Validation Loss with LeNet architecture 42](#_Toc7536295)

[Figure 4.14: Validation Loss with LeNet-5 architecture 43](#_Toc7536296)

[Figure 4.15: Validation Loss with CNN4 architecture 43](#_Toc7536297)

[Figure 4.16: Validation Loss with CNN6 architecture 44](#_Toc7536298)

# **CHAPTER ONE**

# **INTRODUCTION**

## **Background of Study**

With the advent of modern information technology optical character recognition has become a subject of research in recent years. It is the final stage of an automatic license plate recognition system due to its ability to convert human readable form into machine-readable form without alterations, noise variations and other factors, which largely depend on the quality of the input documents (Jørgensen, 2017).

A license plate can be made up of either plastic or metal plate attached to a front and rear of a vehicle for identification purposes (Jain, 2018); it is a standard practice across the globe that no two vehicles should have the same license plate. A basic LPR system is meant to uniquely detect license plate of each vehicles, recognize and extract its characters using various advanced vision based algorithms. However, LPR system is one of the major component of Intelligent Transport System in the transport industry, due to its emergence, it has gradually gained relevance in various areas of applications such as border control, traffic management, recovery of stolen vehicles, automatic electronic toll collection system etc.

In Nigeria, License plate was introduced in 1990; and was later revised in 2011. This was in line with the global effort of proper identification vehicle through integrated vehicle identification system for easy store and retrieval. Nigeria uses the North American license plate standard, it consists of metal plate with a dimension of 152x300mm.

## **Statement of Problem**

Existing License Plates Recognition system of Nigerian Vehicle is plagued with inaccuracy encountered in converting human readable character to machine form, this associated to the size of dataset and framework used for its implementation, complexity of the plates etc., hence this study.

## **Aim and Objective of Study**

The aim of this project is to implement various CNN models for recognition of Nigerian license plate.

The specific objectives of this study are to:

1. Implement various models for optical character recognition of Nigerian License plate.
2. train the models based on (a).
3. evaluate and compare the performance of the model based on (a) and (b).

## **Justification**

Replication of human reading process with the help of machines has been an old area of research in the field of pattern recognition and machine learning. Research interest in OCR based systems is currently receiving lot of attention due to its numerous potentials in business and industry. Generally, automatic license plate recognition of Nigerian is a very complex task due to variations such as different foreground and background colors of various classes license plate and distracting patterns. However, deep learning algorithm have taken top place in object detection due to the great performance they have provided for solving OCR based problems. The key factor is the license plate recognition software. The sophistication of the recognition software, the intelligence and quality of the applied license plate recognition algorithms, determines the capabilities of the recognition software. The better the algorithms are, the highest the quality of the recognition software.

## **Scope of Study**

Optical Character Recognition deals with recognition of optically processed-characters. It is a method of digitizing printed texts and it makes up the final stage of the License Plate Recognition system. The sole purpose of this project is to compare several existing models using the obtained images. It includes recognition of characters from license plate using Convolutional Neural Network (CNN). The various models implemented in this report include LeNet, LeNet-5, CNN4 and CNN6 models. The Optical Character recognition system involves capturing images of license plates and then preprocessing those images and training and testing with the algorithm stated above that converts the images into text and readable form. The observed images are going to be tested and validated based on the accuracy, loss, validation accuracy and validation loss. The performance of the system would then be evaluated using this metrics.

## **Organization of Report**

Chapter one contains background of study, problem statement, aim and objectives, justification of study, scope of study and organization of the report. Chapter two gives detail explanation on the fundamentals of optical character recognition, Convolutional neural network and popular implemented models. The process of data collection and preprocessing, implementation of various CNN models and criteria for evaluation and validation were discussed in Chapter three. Analysis of results and discussion was carried out in Chapter four while the conclusion and direction for future works was presented in Chapter five.

# **CHAPTER TWO**

# **LITERATURE REVIEW**

## **2.1 History of Optical Character Recognition**

Optical Character Recognition has been a topic of interest for many years. It is the process of programmatically processing a document image into its constituent characters. The origin of character recognition can be traced back as early as 1885 when an image scanning device called the Nipkow Disk was created. The first modern OCR tool was created in 1914, it was first implemented on a telegraphy device for the blind, created by Emmanuel Goldberg. The Irish physicist, astrophysicist and chemist Edmund Edward Fournier d'Albe further created an Optophone, an hand held device which when taken across a printed word, produces related tones of the scanned character.

The modern version of OCR appeared in the mid-40s with the development of digital computers before the first scanners capable of reading text was developed but required that the text be in a special font that was easy for the scanning software to recognize. Blur is an important factor that negatively affected OCR accuracy. However in recent times, OCR systems have been developed to digitally recognize symbols and images in different languages.

## **Fundamentals of OCR Based Systems**

OCR is the machine replication of human reading, it can be described as Mechanical or electronic conversion of scanned images where images can be handwritten, typewritten or printed text. It is a method of digitizing printed texts so that they can be electronically searched and used in machine processes. It converts the images into machine-encoded text that can be used in machine translation, text-to-speech and text mining. The advancements in pattern recognition has greatly increased recently and are also computationally more demanding, such evident in Optical Character Recognition (OCR), Document Classification, Computer Vision, Data Mining, Shape Recognition, and Biometric Authentication. The area of OCR is becoming an integral part of document scanners, and is used in many applications such as postal processing, script recognition, banking, security (i.e. passport authentication) and language identification.(Chopra *et al*, 2014).

OCR is a process which separates the different characters from each other taken from an image. OCR is divided into 4 basic components namely; Image acquisition, preprocessing, Character segmentation and character recognition. A detailed explanation is carried out in the subsections below. The figure below shows the block diagram of the OCR process.

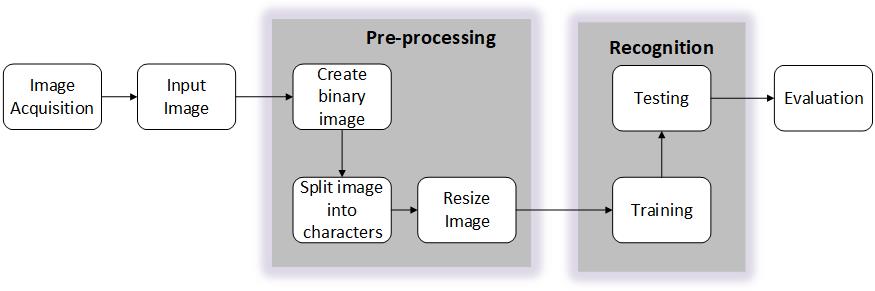


Figure 2.1: The OCR process (Chopra *et al,*2014)

### **Image Acquisition**

Image acquisition is the first step of OCR process, it involves the capturing of a digital image and its conversion into a suitable form that can be easily processed by computer. During digitization of images, some load of compression is carried out by quantization which can either be lossy or lossless in order to conserve storage space and bandwidth (Islam *et al*, 2018).

The image acquisition technology determines the average image quality the license plate recognition algorithm has to work on. Needless to say that the better the quality of the input images are, the better conditions the license plate recognition algorithm has, and thus the higher license plate recognition accuracy can be expected to be achieved.

### **Preprocessing**

Pre-processing aims to enhance the quality of the acquired images. One of the pre-processing techniques is thresholding that aims to binaries the image based on some threshold value. The threshold value can be set at local or global level. Different types of filters such as averaging, min and max filters can be applied.(Islam *et al*, 2018).

In the field of image processing, many algorithms are included in the process image pretreatment step. Its purpose is to make the following algorithms have better results, faster processing speed, or to be more in line with the image of the required information, and other algorithms. Among a wide range of applications in the algorithm, the former license plate recognition in the field of image processing is particularly common, study its content because that contains the target image. In addition to the license plate itself, there are others, such as complex background information, environmental factors, where the target, and the target of the state itself and its surroundings, and so on. With the main algorithm is how to apply the results of the former premise, try to use the appropriate pre-treatment methods, thus becoming with license plate recognition systems.

In recognition of the license plate area, a common type of image pre-processing to its target, it may roughly include the following categories: first, to reduce the amount of image

## **Fundamentals of Deep Learning**

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans. It is achieving results that were not possible before. In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. It discovers intricate structures in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech, audio. Deep learning is an innovative field of machine learning analysis which has introduced with the objective of driving closer to one of its primary goals: Artificial Intelligence. Various deep learning architectures are; convolutional neural networks (CNN), convolutional deep belief neural networks (CDNN) and deep belief networks (DBN). The following architectures are discussed below:

### **Deep belief Neural Network (DBNN)**

A deep belief network (DBN) is a [generative](https://en.wikipedia.org/wiki/Generative_model) [graphical model](https://en.wikipedia.org/wiki/Graphical_model), or alternatively a class of [deep](https://en.wikipedia.org/wiki/Deep_learning) [neural network](https://en.wikipedia.org/wiki/Artificial_neural_network), composed of multiple layers of [latent variables](https://en.wikipedia.org/wiki/Latent_variables) ("hidden units"), with connections between the layers but not between units within each layer. When trained on a [set of examples](https://en.wikipedia.org/wiki/Training_set) [without supervision](https://en.wikipedia.org/wiki/Unsupervised_learning), a DBN can learn to probabilistically reconstruct its inputs. The layers then act as [feature detectors](https://en.wikipedia.org/wiki/Feature_learning). After this learning step, a DBN can be further trained with [supervision](https://en.wikipedia.org/wiki/Supervised_learning) to perform [classification](https://en.wikipedia.org/wiki/Statistical_classification). DBNs can be viewed as a composition of simple, unsupervised networks such as [restricted Boltzmann machines](https://en.wikipedia.org/wiki/Restricted_Boltzmann_machine) (RBMs) or [auto encoders](https://en.wikipedia.org/wiki/Autoencoder), where each sub-network's hidden layer serves as the visible layer for the next. An RBM is an [undirected](https://en.wikipedia.org/wiki/Undirected_graph), generative energy-based model with a "visible" input layer and a hidden layer and connections between but not within layers. This composition leads to a fast, layer-by-layer unsupervised training procedure, where [contrastive divergence](https://en.wikipedia.org/wiki/Contrastive_divergence) is applied to each sub-network in turn, starting from the "lowest" pair of layers (the lowest visible layer is a [training set](https://en.wikipedia.org/wiki/Training_set)) (Wikipedia, 2013). Deep belief neural network represents many-layered perceptron and permits to overcome some limitations of conventional multilayer perceptron due to deep architecture. The supervised training algorithm is not effective for deep belief neural network and therefore in many studies was proposed new learning procedure for dep neural networks.(Golovko *et al*, 2014).

### **Convolutional neural network**

A convolutional neural network (CNN, or ConvNet) is a class of [deep neural networks](https://en.wikipedia.org/wiki/Deep_neural_network), commonly used to analyze visual images. It can be of different variation of [multilayer perceptrons](https://en.wikipedia.org/wiki/Multilayer_perceptron) designed to require minimal [preprocessing](https://en.wikipedia.org/wiki/Data_pre-processing), known as shift invariant or space invariant artificial neural networks (SIANN). It is based on their shared-weights architecture and [translation invariance](https://en.wikipedia.org/wiki/Translation_invariance) characteristics. CNNs use relatively little pre-processing compared to other [image classification algorithms](https://en.wikipedia.org/wiki/Image_classification). This means that the network learns the [filters](https://en.wikipedia.org/wiki/Filter_(signal_processing)) that in traditional algorithms were [hand-engineered](https://en.wikipedia.org/wiki/Feature_engineering). This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in [image and video recognition](https://en.wikipedia.org/wiki/Computer_vision), [recommender systems](https://en.wikipedia.org/wiki/Recommender_system), [image classification](https://en.wikipedia.org/wiki/Image_classification), [medical image analysis](https://en.wikipedia.org/wiki/Medical_image_computing), and [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing).

A convolutional neural network consists of an input, multiple hidden and output layers. The hidden layers of a CNN typically contains convolutional layers, RELU layer (i.e. activation function), pooling layers, fully connected layers and normalization layers.

Convolution is the first layer that extracts features from an input image and preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Convolutional layer apply convolution operation on the input layer, passing the results to next layer. The operation computes a dot product between their weights and a small region that are connected (currently overlapping) to the input volume. In general, it is mathematically modelled as:

(2.1)

Where is the output of the current layer, is the previous layer outputs, is kernel for present layer, is the bias for current layer and represents a selection of input maps (Rahman *et al*, 2018). This will change the dimensions depending on the filter size used and number of filters used. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.

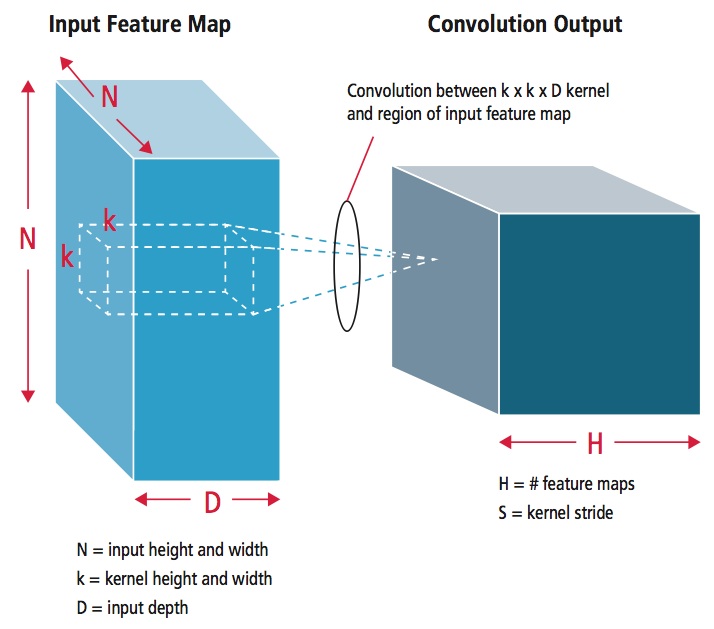


Figure 2.2: Pictorial Representation of convolution process (Hijazi *et al*, 2015)

The pooling layer is the second layer to reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or down sampling which reduces the dimensionality of each map but retains the important information. Pooling is a sample-based discretization process. Its objective is to down-sample an input representation (image, hidden-layer output matrix, etc.) and reducing its dimensionality. Features contained in the sub-regions are assumed to be binned. Two commonly used types are max and min pooling. This layer is also called the subsampling layer which performs down sampling. This operation can be formulated as:

(2.2)

Where represents a subsampling function. This function usually sums up over n × n block of the maps from the previous layers and selects the average value or the highest values among the n × n block maps (Rahman *et al*, 2018). As the name suggests max pooling is based on picking up the maximum value from the selected region and min pooling

## **Related Works**

Kassm and Achkar (2017) developed an algorithm for the detection of Lebanese License Plate to achieve the same or better accuracy than previous known algorithms while attaining a higher speed of processing and a modular yet simple approach. The results were obtained by using a deep convolutional neural network cascade for classification (CNN cascade) which contributed to improving speed and accuracy, a CNN with partially connected deep layers for deskewing and a neural network optimized by neuro-evolution for OCR. In doing this a modular LPR solution that surpassed the conventional solution in terms of speed and accuracy was achieved as well as a deskew module that was able to straighten double lined plates with far better accuracy than its image processing counterpart and an OCR module that was optimized for the best speed and accuracy. Due to this improvement plate detection time was reduced. Moreover, adding an adaptive deskewing module to the system, made character segmentation easier and less resource heavy. Also, with the use of a neuro evolved OCR the system achieved the perfect balance between accuracy and speed making it both reliable and adaptable to any proposed scenario. This made it possible to tolerate errors from the deskew module if they occur.

Saunshi *et al* (2017) developed a license plate recognition system using OCR technology. This system was designed to firstly capture the image of the car, then pass on the image to OCR software which firstly recognizes the location of license plate in the image and then extracts the license plate from it. The Sobel’s edge detection algorithms was used to extract the license plate, a number of image pre-processing steps which include RGB to grayscale conversion, noise removal, and binarization of the image was carried out to enhance the image to get better results. Character segmentation was then carried out using horizontal scanning which was given as input to the CNN in order to recognize the character correctly and individually. Character Recognition was carried out after the segmentation using CNN (Convolutional Neural Networks) trained on large number of data sets. Convolutional Neural Networks was made use of to increase the success rate more than the template matching technique of recognizing the characters. The accuracy of the system was measured and 97% of training accuracy was achieved. The system measured 94%, 96% and 98% of character extraction, segmentation and Character Recognition respectively.

# 

# **CHAPTER THREE**

# **METHODOLOGY**

## **Introduction**

This chapter describes the implementation and validation process of the Optical Character Recognition framework. Furthermore, a precise description of the models employed for the OCR system is discussed. The procedure for data collection, preprocessing and testing for the OCR implementation were also discussed.

## **System Requirement**

The following are the hardware and software components required to develop the application.

### **Hardware requirements**

A PC with an Intel core i5- 7200U processor which operates at 2.7GHz, memory of 12288MB RAM which provided a platform for the system and application software was used for the implementation of the various models. A Nikon digital camera, with a resolution of 20.4megapixels was primarily used for the acquisition License Plate images.

### **Software requirements**

The software components used is as follows; PyCharm is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), specifically for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) language.

During the course of the implementation of this project several libraries were installed which include; OpenCV (Open source computer vision) it is a [library of programming functions](https://en.wikipedia.org/wiki/Library_(computing)) mainly aimed at real-time [computer vision](https://en.wikipedia.org/wiki/Computer_vision). OpenCV supports the [deep learning frameworks Tensorflow,Torch/Pytorch and Caffe.](https://en.wikipedia.org/wiki/Deep_learning)  Matplotlib is a python programming language library used to produce a 2D interactive graphing, scientific publishing and plots using python scripts across platforms. It makes plotting graphs and making graphical representations on python easy. Keras is an open source high-level neural network API written in python programming language which is used in training deep learning model. Keras was built to be an interface rather than a standalone machine-learning framework. NumPy, which stands for Numerical Python is the fundamental package required for high level scientific computing and data analysis with Python. It is the library that that support large, multi-dimensional arrays and matrices with a collection of high-level mathematical functions to operate these arrays. Using NumPy, mathematical and logical operations on arrays can be performed. Tensorflow is an open source software library for high performance numerical computation, dataflow, across a wide range of tasks. Tensorflow has a flexible architecture which allow easy computation across various platforms such as Central Processing Units (CPUs), Graphics Processing Units (GPUs), and Tensor Processing Units (TPUs). Tensorflow is used for machine learning applications such as in neural networks for research and production.

## **Data Processing**

This aspect is divided into the data collection and the data pre-processing in order to convert the images into an understandable readable form.

### **Data collection**

For the OCR system nearly one thousand five hundred (1500) images of license plates were captured using the Nikon camera. The images were captured using a high resolution camera to ensure better image quality. These captured images were used as input into the OCR design. The use of a camera with more definition and resolution was used in order to increase the success ratio of the system. Among the captured images only the clearly visible images were used as the primary data to ensure we obtain an accurate result.



Figure 3.1: Captured License Plate

### **Data pre-processing**

The License plate was cropped from the main image first and then resized. In order to remove the noise or other unwanted distortion the images were grey scaled in order to reduce the channel of the image from a 3-channel (RGB) image to a 1-channel image to avoid very large images as this is a problem for the CNN model. The new image gotten was used as the input into the model.

The images were resized to a scale of 200x100, the length of the license plate is smaller than the breadth thus giving us a variable-sized layers for the neural networks. Furthermore, the processed images were converted to dataset/ arrays understandable by the machine.

The whole process of data acquisition and dataset preparation is shown in Figure 3.2 below.

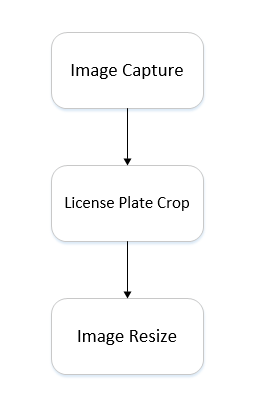


Figure 3.2: Data acquisition and dataset preparation process

## **Implementation of models**

There are three types of layers used in our implementation to the convolutional neural network model. The ﬁrst is the convolution layer, (CONV). The convolutional layer convolve the input images with a ﬁlter and results in a feature map. Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Convolutional layers are the main element in the convolutional neural network architecture. The second type of layer is the pooling layer (POOL). Pooling layer performs downsampling of the data along a spatial dimension, it reduces the number of parameters when the images are too large and results in a reduction of dimension size of the data. Recent state of the art convolutional neural networks often incorporate pooling layers as a vital layer to achieve improved results in image related tasks. The third layer type is the fully connected layer (FC). This units layer is a densely connected to all activation in the previous layers and with the features combined it creates a model. All convolutional layers has a 3 by 3 ﬁlter size and all pooling layers has 2 by 2 size.



Figure 3.3: Implemented OCR model

The various methods employed in this research include the following: LeNet, LeNet5, CNN4 and CNN6 neural networks and are discussed below.

### **Le-Net**

LeNet network is one of the most famous convolution networks. The LeNet model implemented is made up of two convolutional layers with feature maps 4 and 8 respectively both having kernel size 5x5 with a relu activation function. The sub-sampling operation occurs immediately after each convolutional layer with the use of AveragePooling with kernel size 2x2 and strides 1x1. The fully connected layers come up after with two dense layers and a softmax classifier.

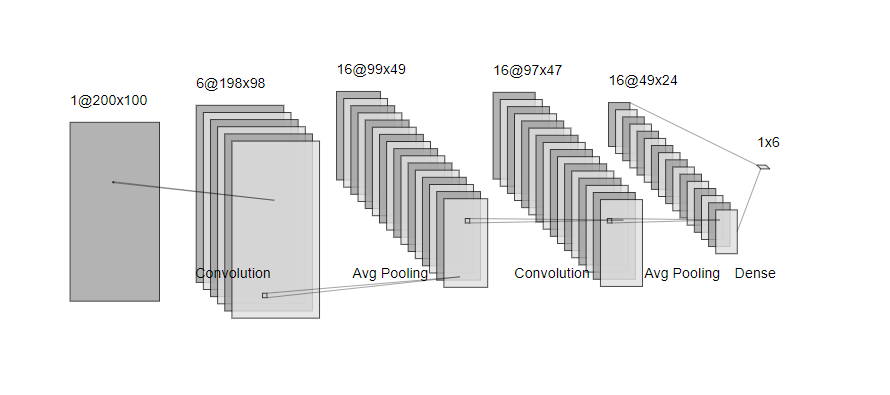


Figure 3.4: The LeNet model

## **Testing and Validation**

For the intent of training, the preprocessed images were used. From the training process carried out we were able to then calculate the accuracy, loss, validation accuracy and validation loss. Training of Nigerian License Plate Optical Character Recognition System using a Convolutional Network was performed with the number of datasets obtained. The dataset was divided into train set and test set in different folders like 70% data is used to train 30% of data to test for comparing the performance of the system. The images were trained in batches based on the epoch, which is the number of iterations for the training. After the implementation of the several models, the matplotlib function was imported in order to plot the model parameters after training, which include, iteration, accuracy and loss. The accuracy and loss of each epoch was derived after each iteration.

# **CHAPTER FOUR**

# **RESULTS AND DISCUSSION**

## **Introduction**

This chapter discusses the experimental result obtained from the different Optical Character Recognition models implemented which include: accuracy, loss, validation accuracy and validation loss

### **Accuracy**

Figure 4.1, Figure 4.2, Figure 4.3 and **Error! Reference source not found.** shows the accuracy of the LeNet, LeNet-5, CNN 4 and CNN6 models respectively.

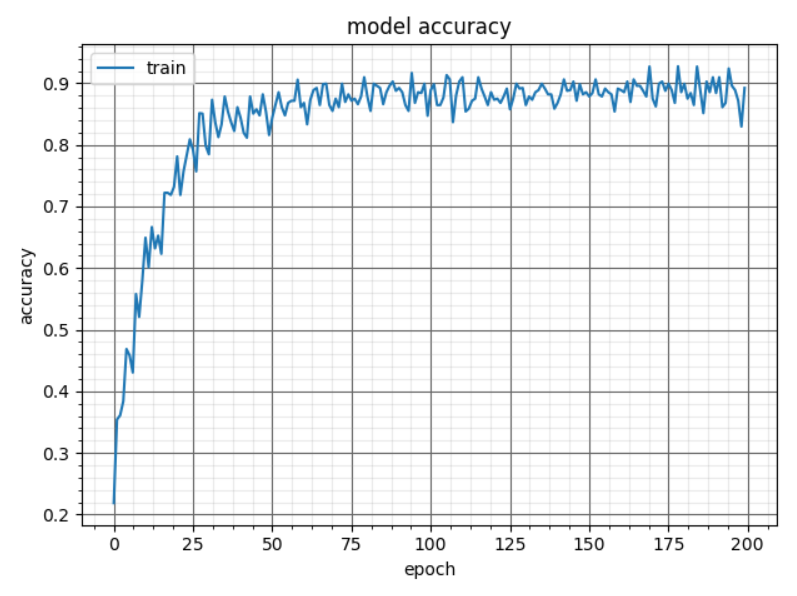


Figure 4.1: Accuracy with LeNet architecture

In Figure 4.1, there was a sharp rise at 0 epoch to an accuracy of 0.36. There was a steady rise till 52 epoch and was readily steady till it converged finally at 0.88. A final accuracy of about 88% was achieved after 200 epoch (iterations). The architecture converged after 52 epoch (iterations) to give an average training rate of 85%.

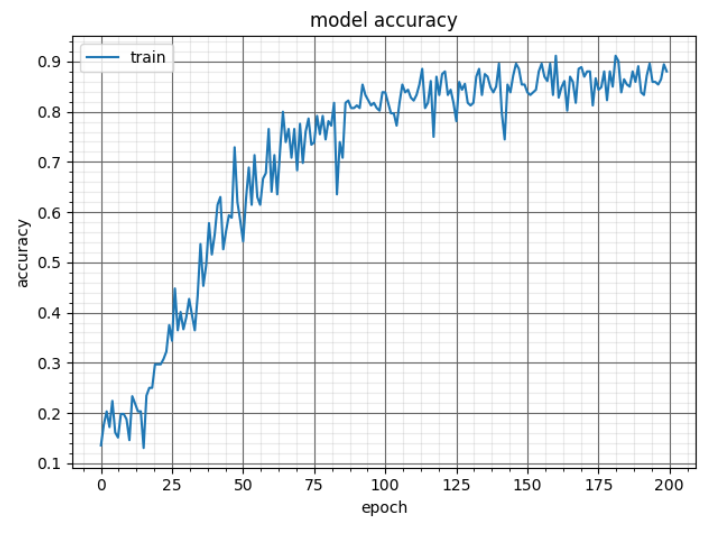


Figure 4.2: Accuracy with LeNet-5 architecture

In Figure 4.2, there was steady fluctuations within 0 to 13 epoch. There was a steady rise with fluctuations till 113 epoch and was readily steady till it converged finally at 0.88. A final accuracy of 88% was achieved after 200 epoch (iterations). The architecture converged after 113 epoch (iterations) to give an average training rate of 88%.

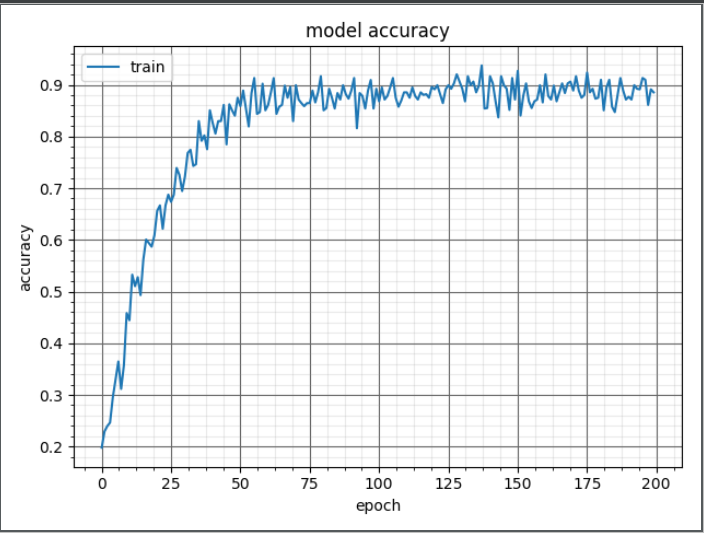


Figure 4.3: Accuracy with CNN4 architecture

# **CHAPTER FIVE**

# **CONCLUSION AND RECOMMENDATION**

## **Conclusion**

A convolutional neural network is the basis for most recognition task ranging from classification, object-detection to segmentation. Convolutional Neural Networks are good for character recognition. In this project, a performance comparison of four network architectures on the task of classifying the Nigerian License plate characters was carried out. After implementing several models for optical character recognition and evaluating these models to obtain the model with a better performance. The obtained results indicates that the LeNet and LeNet-5 perform optimally well due to the optimized nature of the model. In view of the fantastic performance of these models, it is evident that an overall reliable and efficient performance will be obtained if these models are implemented for Nigerian License Plate Recognition System.

## **Recommendation**

Due to the relatively small dataset used for training, it affected the result mildly. It is recommended that a larger and augmented dataset should be employed. Consequently, the use of expanded dataset will improve the overall performance of implemented CNN models. The existence of a Nigerian map at the background of the Nigerian license plate often makes segmentation of characters a difficult task to achieve, it is suggested that the design of Nigerian License plate be reviewed. It is highly recommended that PCs with high computational capabilities and dedicated Graphic processing unit (GPU) should be used in order to reduce training time.

# **REFERENCES**

Ahmad, I. S., Boufama, B., Habashi, P., Anderson, W., and Elamsy, T. (2015). Automatic License Plate Recognition : A Comparative Study, *IEEE International Symposium on Signal Processing and Information Technology* *(ISSPIT)*, 5(6): 635–640.

Amusan, D. G., Arulogun, O. T., and Falohun, A. S. (2015). Nigerian Vehicle License Plate Recognition System using Artificial Neural Network, *International Journal of Advanced Research in Computer and Communication Engineering,* 4(11): 1–5,

Angara, N. S. S.  (2015). Automatic License Plate Recognition Using Deep Learning Techniques, Available on http://www.scholarworks.uttyler.edu, Accessed on 14th January, 2019.

Azad, R., Azad, B., and Shayegh, H. R. (2014). Real-Time and Efficient Method for Accuracy Enhancement of Edge Based License Plate Recognition System. *arXiv preprint arXiv:1407.6498*. 146–155.

Chopra, S. A., Ghadge, A. A., Padwal, O. A., Punjabi, K. S., and Gurjar, G. S. (2014). Optical Character Recognition. *International Journal of Advanced Research in Computer and Communication Engineering,* *3*(1): 4956–4958.

Dhar, P., Guha, S., Biswas, T., and Abedin, Z. (2018). A System Design for License Plate Recognition by Using Edge Detection and Convolution Neural Network. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2),* 8(1): 1-4.

Golovko, V., Kroshchanka, A., Rubanau, U., and Jankowski, S. (2014, June). A Learning Technique for Deep Belief Neural Networks. In *International Conference on Neural Networks and Artificial Intelligence*, 5(4): 136–146.

Ha, P. S., and Shakeri, M. (2016). License Plate Automatic Recognition based on Edge Detection.  In *2016 Artificial Intelligence and Robotics (IRANOPEN),* 5(13): 170-174.

Hijazi, S., Kumar, R., and Rowen, C. (2015). Using Convolutional Neural Networks for Image Recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*.

Himadeepthi, V., Balvindersingh, B., and Srinivasarao, V. (2014). Automatic Vehicle Number Plate Localization Using Symmetric Wavelets. In *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I*: 69-76.

Islam, N., Islam, Z., and Noor, N. (2017). A Survey on Optical Character Recognition System. *arXiv preprint arXiv:1710.05703*.

Jain, M. (2018). Unconstrained Arabic and Urdu Text Recognition using Deep CNN-RNN Hybrid Networks *Doctoral dissertation, International Institute of Information Technology Hyderabad*, 7(1): 6-10.

Jørgensen, H. (2017). Automatic License Plate Recognition using Deep Learning Techniques (Master's thesis, NTNU).

Kassm, G. A., and Achkar, R. (2017). LPR CNN Cascade and Adaptive Deskewing. *Procedia Computer Science*, 11(4): 296–303.

# **APPENDIX**

# Step 1

import cv2 # working with, mainly resizing, images

import numpy as np # dealing with arrays

import os # dealing with directories

from random import shuffle # mixing up or currently ordered data that might lead our network astray in training.

from tqdm import tqdm

import tensorflow as tf # Import Tensorflow

import glob # This will extract all files from the folder

import keras

from keras.preprocessing.image import ImageDataGenerator

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D

from keras.layers import Activation, Dropout, Flatten, Dense

from keras import backend as K

import h5py

from keras.models import model\_from\_json

from keras.models import load\_model

import numpy as np

from keras.preprocessing import image

from keras import backend as K

from keras.preprocessing.image import img\_to\_array, load\_img

from keras.utils import to\_categorical

from keras.utils import np\_utils

import matplotlib.pyplot as plt

# Step 2

# Load images from folder train folder