**PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA**
**MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH**
**UNIVERSITY OF MOHAMED BOUDIAF-M'SILA**

FACULTY OF MATHEMATICS
COMPUTER SCIENCE
COMPUTER SCIENCE DEPART
**N°** :……


**Domain**: Mathematics and Computer Science
**Branch**: Computer Science
**Specialty**: IA

A Dissertation
*Submitted in partial fulfillment of the requirements for the degree of*
***Master in Artificial intelligence***

**By**
TALLAI Meriem
MEKDOUR Asma

**Title of the thesis**

# Recognition of Printed

# Arabic Characters

Under the supervision of
**Dr. Rahima Bentrcia**

Composition of the jury

| | | |
|---|---|---|
| **Dr.** Bentrcia Rahima | University of Msila | Supervisor |
| **Dr.** Kadri Said | University of Msila | Reporter |
| **Dr.** Brahimi Belgacem | University of Msila | Examiner |

**Academic Year: 2021/2022**

**PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA**
**MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH**
**UNIVERSITY OF MOHAMED BOUDIAF-M'SILA**

FACULTY OF MATHEMATICS AND
COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTM
N° :……

**Domain**: Mathematics and Computer Science
**Branch**: Computer Science
**Specialty**: IA

A Dissertation
*Submitted in partial fulfillment of the requirements for the degree of*
***Master in Artificial intelligence***

**By**
TALLAI Meriem
MEKDOUR Asma

**Title of the thesis**

# Recognition of Printed

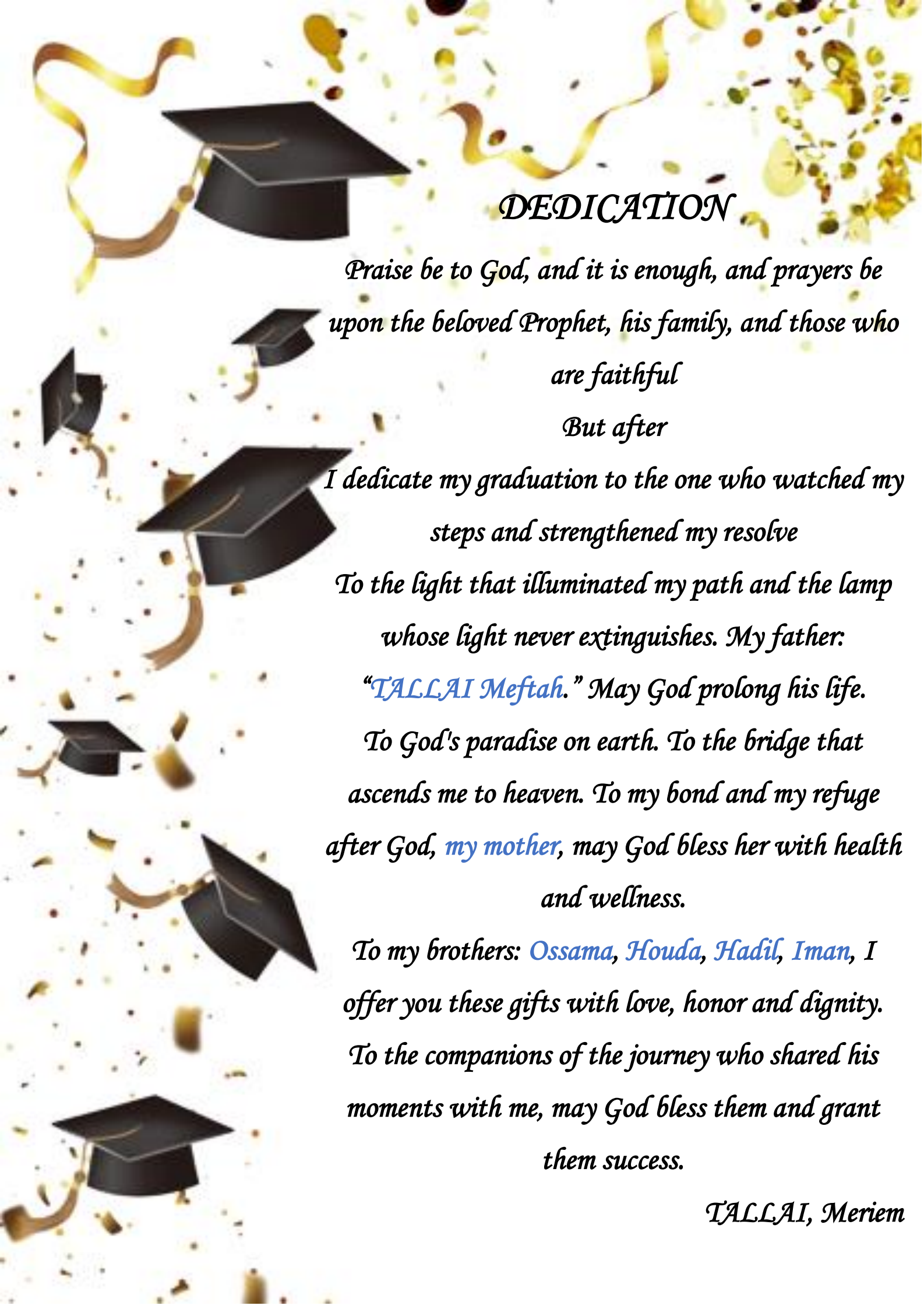
# Arabic Characters

Under the supervision of
**Dr. Rahima Bentrcia**

Composition of the jury

| | | |
|---|---|---|
| **Dr.** Rahima Bentrcia | University of Msila | Supervisor |
| **Dr.** Kadri Said | University of Msila | Reporter |
| **Dr.** Brahmi Belgacem | University of Msila | Examiner |

**Academic Year: 2021/2022**



# *DEDICATION*

*Praise be to God, and it is enough, and prayers be upon the beloved Prophet, his family, and those who are faithful*

*But after*

*I dedicate my graduation to the one who watched my steps and strengthened my resolve*

*To the light that illuminated my path and the lamp whose light never extinguishes. My father: "TALLAI Meftah." May God prolong his life.*

*To God's paradise on earth. To the bridge that ascends me to heaven. To my bond and my refuge after God, my mother, may God bless her with health and wellness.*

*To my brothers: Ossama, Houda, Hadil, Iman, I offer you these gifts with love, honor and dignity.*

*To the companions of the journey who shared his moments with me, may God bless them and grant them success.*

*TALLAI, Meriem*

# DEDICATION

*I dedicate this work*

*To the one whose grace is not above his bounty, accept the dust of his feet...*

*To the one whose affiliation with him and his remembrance makes me proud...*

*Who made the effort of the years for me to climb the ladders of success and made me grow in the purest and purest virtue to my father, the master: mekdour el-bahi, May God prolong his life.*

*To my companion and partner in this work, may God protect and preserve her: Meriem Tallai.*

*For those who gave me advice and guidance, my sister: Rania, for she was like my arm and my back.*

*To my little girl and sister: israa.*

*To all those whom fates willed to bring me together with the gardens of study and make them brothers: Echyma, Samira, Meriem, Amira, Haniya, Sounia, Fatima.*

*MEKDOUR, Asma*

# *ACKNOWLEDGMENTS*

*First and above all, we thank "God" for giving us the patience, the health, the courage to have come this far.*

*We would like to thank Dr. Rahima Bentrcia who honored us with her supervision, for her direction, her Guidance, her advice and all her constructive remarks for the good progress of our work.*

*We also thank the jury: Dr. Kadri Said and Dr. Brahimi Belgacem.*

*TALLAI, Meriem*
*MEKDOUR, Asma*

# Table of Contents

# List of Tables

# List Of Figures

## List Of Figure

# CHAPTER 1: GENERAL INTRODUCTION

## 1. Overview

Artificial intelligence is a successful technology where most of the ideas and methods are impossible. Today, it has become the means used in computer science. It is a study of intelligent behavior (in humans, animals, and machines) and it can be inserted into synthetic machines, which is the machine's capacity to make decisions in a problem, by finding a way to solve the problem or decide by following many different inferential processes.

Artificial intelligence contributes to various fields, including the field of Character Recognition, which seeks to be a direct link between humans and machines through the natural language of humans (Arabic, English, French, Japanese, Chinese, etc.) whether it is through speech or writing. Hence the need for written recognition for exploiting what has been written by modern electronic means, the field of Character Recognition contributed to many studies where it achieved good results in various languages.

In addition, the Arabic language is one of the most widely spoken languages in the world, spoken by more than 467 million people living in the Arabic world besides many other neighboring regions such as Ahwaz, Turkey, Chad, Mali, Senegal, Eritrea, Ethiopia, and South Sudan. Thus, it occupies the fourth or fifth place in terms of the most widely spoken languages in the world and it occupies the third place according to the number of countries that recognize it as an official language [1]. Unfortunately, there is little Arabic computing research, especially in the field of recognition, due to the difficulty of dealing with Arabic texts because of some features in Arabic writing which make the recognition process difficult and ambiguous.

Therefore, the work carried out in the framework of this thesis was related to the development of an offline recognition system for printed Arabic characters. The character classification process was accomplished by using Convolutional Neural Networks (CNN) which have proven their efficiency in obtaining quick and reliable recognition results.

## 2. Problem Statement

We are motivated by the following problems to conduct our research:

• Manual data entry takes a long time using the keyboard, and errors can occur.

• The images containing text (printed or handwritten) are stored in a computer. The computer recognizes this text as an image. This text cannot be processed, searched, or edited.

Therefore, developing a recognition system for printed Arabic characters can present a suitable solution to these problems.

## 3. Challenges Related to Arabic Text Recognition

- **Arabic Characters:** The Arabic language is characterized by 28 consonants (29 if we count the hamza), as shown in Table 1-1.
- **Multiple Character Shapes:** The Arabic language has 28 characters without uppercase or lowercase. Each character has two or four shapes according to its position in the word (Isolate, Beginning, Middle, and End), In Table 1-1, the first column represents the letter's order in the alphabet. The second column represents the letter, the third column represents the letter in its isolated found rm, and the fourth column represents the letter at the beginning of the word. The fifth column represents the letter in the middle of the word, and the last column represents the letter at the end of the word.

| No | Name | Isolate | Beginning | Middle | End |
|---|---|---|---|---|---|
| 1 | Alif | ا | - | - | ـا |
| 2 | Baa | ب | بـ | ـبـ | ـب |
| 3 | Taa | ت | تـ | ـتـ | ـت |
| 4 | Thaa | ث | ثـ | ـثـ | ـث |
| 5 | Jeem | ج | جـ | ـجـ | ـج |
| 6 | Haa | ح | حـ | ـحـ | ـح |
| 7 | Khaa | خ | خـ | ـخـ | ـخ |
| 8 | Daal | د | - | - | ـد |
| 9 | Dhal | ذ | - | - | ـذ |
| 10 | Raa | ر | - | - | ـر |
| 11 | Zaa | ز | - | - | ـز |
| 12 | Seen | س | سـ | ـسـ | ـس |
| 13 | Sheen | ش | شـ | ـشـ | ـش |
| 14 | Saad | ص | صـ | ـصـ | ـص |

| No | Name | Isolate | Beginning | Middle | End |
|---|---|---|---|---|---|
| 15 | Dhad | ض | ضـ | ـضـ | ـض |
| 16 | Tta | ط | طـ | ـطـ | ـط |
| 17 | Dha | ظ | ظـ | ـظـ | ـظ |
| 18 | Ain | ع | عـ | ـعـ | ـع |
| 19 | Ghain | غ | غـ | ـغـ | ـغ |
| 20 | Faa | ف | فـ | ـفـ | ـف |
| 21 | Qaf | ق | قـ | ـقـ | ـق |
| 22 | Kaaf | ك | كـ | ـكـ | ـك |
| 23 | Laam | ل | لـ | ـلـ | ـل |
| 24 | Meem | م | مـ | ـمـ | ـم |
| 25 | Noon | ن | نـ | ـنـ | ـن |
| 26 | Haa | ه | هـ | ـهـ | ـه |
| 27 | Waaw | و | - | - | ـو |
| 28 | Yaa | ي | يـ | ـيـ | ـي |

**Table 1-1:** The Different Shapes of Arabic Letters.

Figure 1-1 shows the four different forms of the Arabic letter Baa (ب) based on its position in the word.


**Figure 1-1:** The letter (ب) in its four positions: the beginning, middle, end of the word, and Isolated.

Some characters such as the Arabic letter Taa (ت) have more than four shapes, as presented in Figure 1-2.

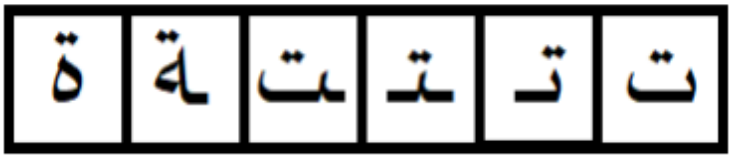



**Figure 1-2:**The six forms of the Arabic character (ت).

The Arabic language is written from right to left, as shown in Figure 1-3.


**Figure 1-3:** Arabic writing direction.

- **Dots:** The dots play an important role in Arabic characters. The shape of some characters is similar **(ف، ق)، (س، ش) (ع، غ)، (ط، ظ)، (ر، ز)، (د، ذ)، (ج، ح، خ)، (ب، ت، ث)** whereas the difference arises with the position of the dot(s); (above or below the characters) and the number of dots (one, two, three, or no dot), as shown in Figure 1-4 Fifteen characters haven't the dots in Arabic characters, ten characters have one dot, three characters have two dots, and two characters have three dots.

| Similar Characters | The Shape of Characters |
|---|---|
| ب،ت،ث | ب |
| ج،ح،خ | ح |
| د،ذ | د |
| ر،ز | ر |
| س،ش | س |
| ط،ظ | ط |
| ص،ض | ص |
| ع،غ | ع |
| ف،ق | ف |

**Figure 1-4:** Arabic characters similarity[2].

- **Connectivity:** Arabic characters are connected (cursive) within the same word, unlike the Latin characters, as appears in Figure 1-5.


**Figure 1-5:** Characters connectivity.

- **Ligatures:** It is the overlapping of two characters in certain syllables of words in most Arabic fonts. Figure 1-6 shows some examples of ligatures.

**Figure 1-6:** Examples of Arabic Ligatures.

- **Diacritics:** They are used to help clarify text content and resolve linguistic ambiguities, as clarified in Figure 1-7.


**Figure 1-7:** Arabic Text with Diacritics.

- **Overlaps** Some letters in the word overlap in some types of Arabic fonts, as illustrated in Figure 1-8.

**Figure 1-8:** Some overlapping letters in Arabic fonts.

- **Sub word(s):** some Arabic words are composed of sub-word(s). Figure 1-9 introduces an example of an Arabic word with four sub-words.

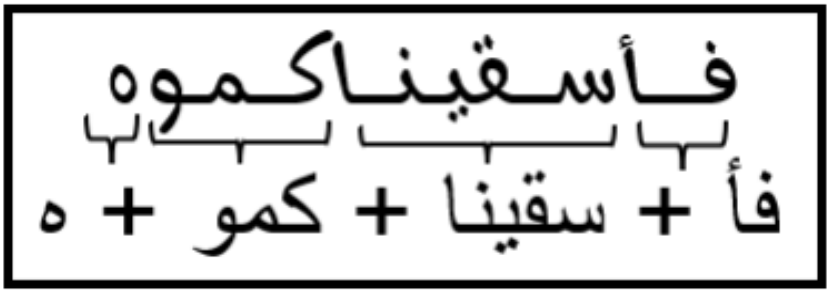



**Figure 1-9:** Example of Arabic word with four sub-words.

- The Arabic Text is a group of words consisting of letters, numbers (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), punctuation marks (!,?), and symbols (/, <, >, ...).

## 4. Objectives of the Present Research

Arabic is the official language of millions of people around the world. Therefore, presenting this famous language with Arabic characters is unified, and specialized researchers needed to improve applications in the field of Arabic writing recognition. The research objectives are:

- Another alternative to the method of entering data into the computer instead of using the traditional keyboard.

- Ease of obtaining a copy of the paper document without the possibility of errors in the entry process.
- Ease of modifying and editing the paper document after converting it to an electronic document using editing programs.
- Recognition of Arabic characters with a high accuracy rate.

## 5. Research Importance

- The system is used in the field of Arabic characters recognition by entering a large number of data (printed characters), where a large number of paper-typed data is entered into the computer in a very short time and without errors compared to entering it in the traditional way (by the keyboard).
- Solve the problem of easy damage and loss of paper documents.
- Improving the performance and outcome of OCRs especially for Arabic texts.
- Fraud detection for manuscript characters.

## 6. Research Scope

There are two types of Optical Character Recognition systems (OCR): offline and online systems as shown in Figure 1-12:

### 6.1. Online Character Recognition

Online Character Recognition is a character recognition process performed at the same time that characters are written. Usually, an optical pen is used for writing characters to be displayed on the computer.

### 6.2. Offline Character Recognition

Offline Character Recognition is recognizing the character that is present on the image and scanned through a digital device like a scanner or camera, then the image is stored in the computer.

#### 6.2.1. Machine Printed

Machine printed characters can be classified into two categories:

- **Single font:** Characters in the scanned image are written only in one type of font such as Andalus shown in Figure 1-10.
- **Omni-font:** Characters in the scanned image are written in more than one type of font.


**Figure 1-10:** Example of printed Arabic script: Andalus Font.

### 6.2.2. Handwritten

This writing style is the most challenging because the same letters may differ from one person to another due to a person's mode, health, and speed. Figure 1-11 presents an example of a handwritten sentence.

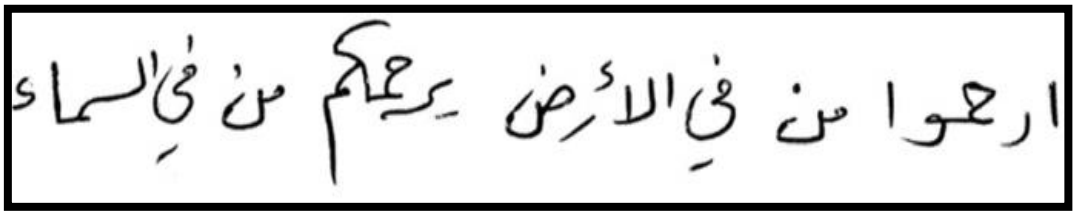


**Figure 1-11:** Example of Arabic handwriting.

Therefore, this research works on the offline recognition of printed Arabic characters - **Omni-font**-.

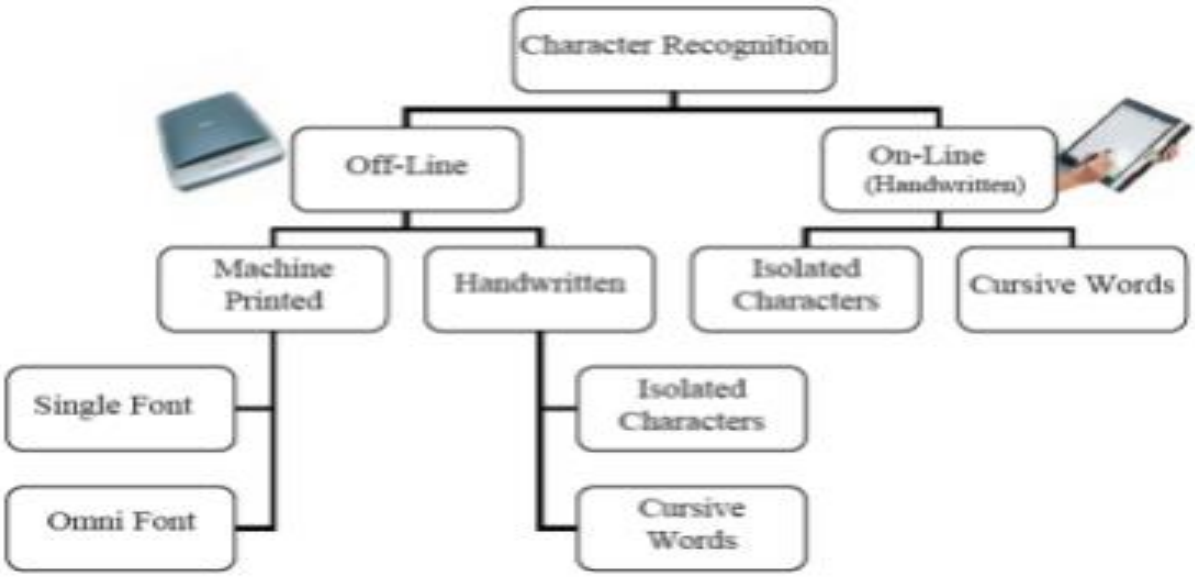



**Figure 1-12:** Character Recognition Mechanisms[3].

## 7. Thesis Organization

The proposed structure of our manuscript can be presented as follows:

- **Chapter 1:** In it, we present a general introduction to artificial intelligence and its contribution to the development of the field of natural language processing, and the status and importance of the Arabic language among the peoples of the world. And Our

problem statement. It presents the most important Challenges that delayed the emergence of Arabic Character recognition systems. We also discussed the objectives and importance of the research. and mentioning the types of Characters Recognition.

- **Chapter 2:** Here we show the definition of a dataset in the Oxford Dictionary. and the importance of data set in machine learning and artificial intelligence projects. We also show the definition of a database for our project and how it is collected and stored.
- **Chapter 3:** This chapter presents previous studies.
- **Chapter 4:** In this chapter, the printed Arabic character recognition system is described Where a detailed presentation of the system is presented.
- **Chapter 5:** It contains the most important findings of the research in addition to discussing the obtained results.

And in the end, the conclusion, which contains the summary of the research as well as the prospects for this system.

# CHAPTER 2: THE USED DATASET

## 1. Introduction

Artificial Intelligence is the transfer of the cognitive ability of the computer and the performance of any intellectual task with the same accuracy that comes with the human being. Machine Learning is an analytical approach associated with Artificial Intelligence and technology to learn regularity and judgment criteria from relatively large data and to predict and judge unknown things. The main idea behind machine learning is that a computer can be trained to complete tasks that are impossible for the human mind. Deep learning contributes to achieving the goals of machine learning on the system more systematically.

In most cases, computers are programmed to simulate the task of the human brain using Artificial Neural Networks ANNs.

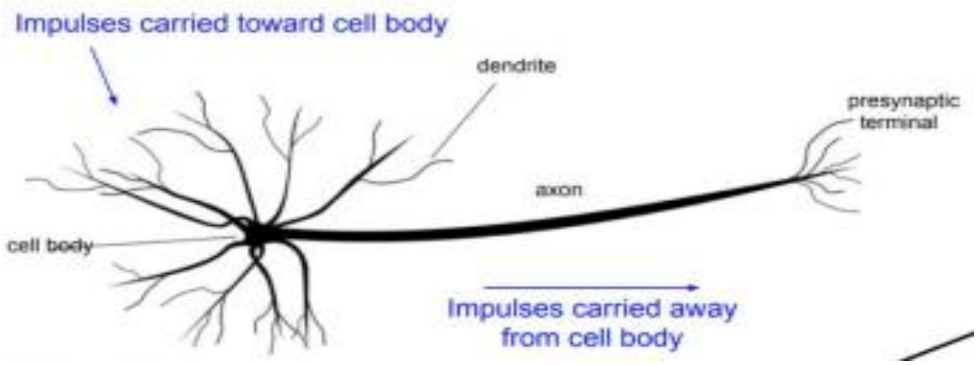



**Figure 2-1:** A Biological Neuron

As shown in figure 2-1, a neuron is a cell whose task is to transmit an electrical signal under certain conditions. It consists of a nucleus and the body of the neuron is connected on one side by a group of dendrites (the entrance to the cell) and on the other hand the axon (the entrance to the exit). While neurons receive electrical signals at the level of dendrites, they transmit them to the axon. In addition, the synapse is the part where the entire signal is transmitted from cell to cell. The human brain consists of countless neurons trying to process, interpret and classify data and compare it with previous data.


**Figure 2-2:** Relation between AI, Machine Learning, and Deep Learning

However, deep learning is a subset and complementary to machine learning algorithms and takes another concept in the name of Deep Neural Networks. Figure 2.2 shows the relationship between Artificial Intelligence, Machine Learning, and Deep Learning.

Deep Learning is trained to handle unlimited amounts of data by adding layers to a network. The deep learning model can extract properties from raw data without the need for specific rules. This is done through multiple processing layers consisting of several linear and nonlinear transformations.

Supervised learning is an approach to create Artificial Intelligence (AI), in which a computer algorithm is trained on input data that has been labeled for a given output. The model is trained so that it can detect basic patterns and relationships between input data and output labels, enabling it to obtain accurate classification results when presented with data it has never seen before.

Supervised learning is good for classification and regression problems, such as character recognition. In supervised learning, the goal is to understand the data in the context of a specific question.

## 2. DataSet

Oxford Dictionary defines a dataset as “a collection of data that is treated as a single unit by a computer”. This means that a dataset contains a lot of separate pieces of data but can be used to train machine learning algorithms, to find predictable patterns inside the whole dataset.

Now it's important to collect this data into datasets. Sufficient volumes of data must be analyzed to extract hidden patterns and make decisions based on the dataset. However, this process is more complicated since it requires a proper treatment of the data to be usable.

Usually, a dataset is used not only for training purposes. A single training dataset that has already been processed is usually split into several parts, which is needed to check how well the training of the model went. For this purpose, a testing dataset is usually separated from the data. Next, a validation dataset, while not strictly crucial, is quite helpful to avoid training your algorithm on the same type of data and making biased predictions [4].

Figure 2-3 depicts the data division used in machine learning projects, where a large amount of data is required because machine learning models and AI projects cannot be trained without data. One of the most important aspects of creating machine learning and artificial intelligence projects is gathering and preparing the data set. Any machine learning project's technology cannot function properly unless the data set is properly prepared and processed beforehand. Developers rely entirely on data sets when working on a machine learning project. When developing machine learning applications, the data set is split into two parts:

- Training dataset.
- Testing dataset.
- Also, a part of the training set is allocated for validation, which is called the validation set.

The training set is the data that the algorithm will learn from. Learning looks different depending on which algorithm is used. Following training with the training set, the image in the validation set is used to calculate the workbook's accuracy or loss, where accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy is the number of correct predictions out of total number of predictions and loss is the penalty for a bad prediction. That is, loss is a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples.

The basic idea here is that we know what the true labels are for each image in the validation set, but we're pretending we don't. Each image in the validation set can be used as input to our classifier. After that, we'll get a rating for that image. We can now check the actual label of the verification image to see if we were correct. We can calculate the validation error if we do this for each image in the validation set. Following training with the training set, the image in the

validation set is used to calculate the workbook's accuracy or loss. The basic idea here is that we know what the true labels are for each image in the validation set, but we're pretending we don't. Each image in the validation set can be used as input to our classifier. After that, we'll get a rating for that image. We can now check the actual label of the verification image to see if we were correct. We can calculate the validation error if we do this for each image in the validation set.

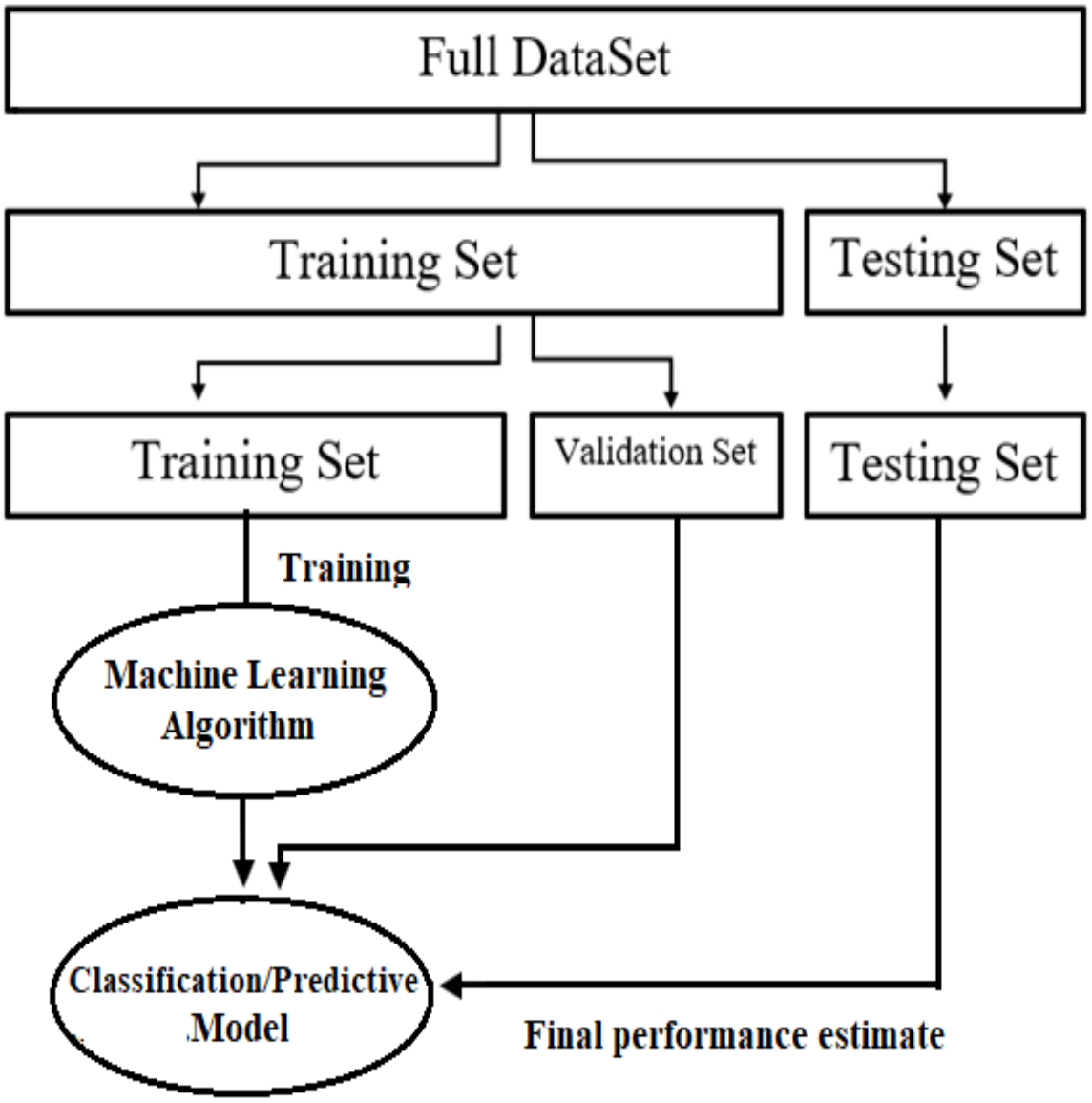



**Figure 2-3:** Splitting of a dataset into training, testing, and validation datasets [5].

## 3. The Used DataSet

Datasets are necessary components of any computer vision system. The normative dataset enables researchers to quickly and fairly compare various methods and techniques for machine

learning. In this section, we describe the dataset which we have used in our research. It contains Arabic printed characters in different positions, the numbers from (0-9), and some symbols (<,>, (,)) which were collected from Microsoft Word software. Data was collected in various forms.

Next, we explain the process of data acquisition and pre-processing, and we provide a detailed description of the dataset.

## 4. Data collection

For data collection, the matrix (11*13) was printed in Microsoft Word software, as shown in Figure 2-4. As the Arabic alphabet has 28 characters, the matrix then represents 143 different characters shapes based on the character positions: at the beginning of the word, at the end of the word, and in the middle of the word, in addition to numbers (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) and some symbols (؟, >, <, (,), /,:,).

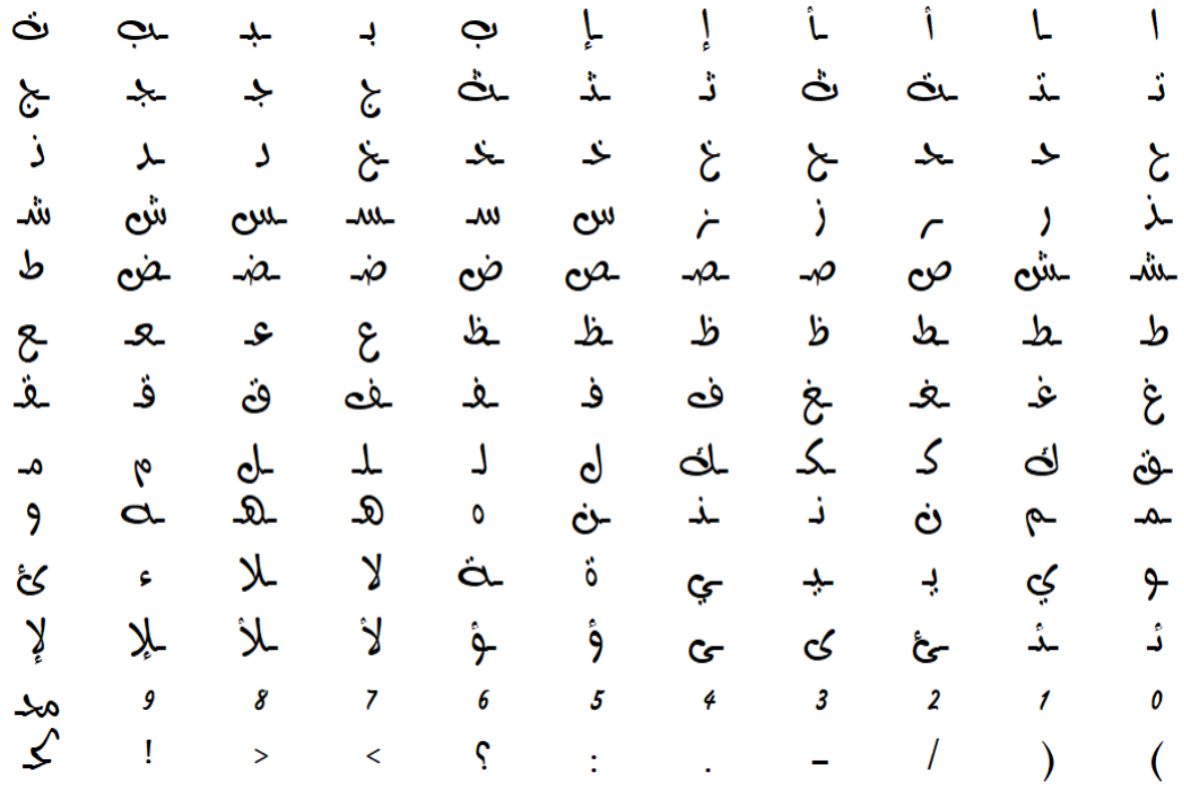



**Figure 2-4:** DataSet Matrix.

## 5. The Compiled DataSet

Several Arabic printed datasets were introduced by the community during the past time, including DARPA, APTI, PATDB, APTID/MF, and RCATSS [6]. Nonetheless, there is no agreement on a standard dataset that the community can use for benchmarking printed text recognition. As a result, the available datasets differ in terms of content, size, formatting styles, and font types. The main objective of this work is to recognize the Arabic text that was printed. For that purpose, no suitable ready dataset. For that, this dataset was prepared using the (21) fonts depicted in Figure 2-5 below where characters were written in different types (21 types of fonts) and sizes (20 to 41) of fonts.

| Font Type : | Font exemple : |
|---|---|
| (A) Arslan Wessam B | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| A Hemmat | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| AGA Aladdin Regular | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Akhbar MT | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Andalus | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Arabic Typesetting | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Aref_Menna | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Arial | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Calibri Light | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| DecoType Naskh | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| DecoType Naskh Special | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| DecoType Naskh Variants | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| FS_Nice | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Hayah | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| K Kamran | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Kamran | التعرف علي النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Monotype Koufi | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Mudir MT | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Sakkal Majalla | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Times New Roman | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |
| Traditional Arabic | التعرف على النصوص العربية المطبوعة باستخدام تقنية التعلم العميق |

**Figure 2-5:** The Set of the Selected Fonts.

## 6. Data Storage

Divide DataSet matrices into characters and save them as images. Next, the set of images of the same letter was saved in different forms (Isolate, Beginning, Middle, and End words). After that, we label the folder in Latin characters to represent each character in the dataset.

As a result, 84162 files were obtained and classified into 58 classes, as shown in Table 2-1, the first column represents the order of the letters in the alphabet. The second column represents the letter, and the third column represents the name of the folder in which the images of the character mentioned in the second column were collected. The name of the folder is named in Latin letters to pronounce the letter to which it belongs, and the fourth column represents the number of images for each letter. The last column represents the ratio of the number of images for each category to the total number of the dataset.

| | Chara-ctrer : | Folder Name(Classe) | N Files : | Percent-age : | | Chara-ctrer | Folder Name(Classe) | N Files : | Percent-age : |
|---|---|---|---|---|---|---|---|---|---|
| 1 | ا | Alif | 1134 | 1.34% | 30 | أ | Hamza Above Alif | 1134 | 1.34% |
| 2 | ب | Baa | 2268 | 2.68% | 31 | إ | Hamza Under Alif | 1134 | 1.34% |
| 3 | ت | Taaa | 2268 | 2.68% | 32 | ئ | Hamza Alif Broken | 1134 | 1.34% |
| 4 | ث | Thaa | 2268 | 2.68% | 33 | ؤ | Hamza_Above_Waaw | 1134 | 1.34% |
| 5 | ج | Jiim | 2268 | 2.68% | 34 | لا | Laaa | 1134 | 1.34% |
| 6 | ح | 7aa | 2268 | 2.68% | 35 | لأ | Laam and Hamza Above Alif | 1134 | 1.34% |
| 7 | خ | Khaa | 2268 | 2.68% | 36 | لإ | Laam and Hamza Under Alif | 1134 | 1.34% |
| 8 | د | Daal | 1134 | 1.34% | 37 | لحـ | Laam+7aa | 260 | 0.3% |
| 9 | ذ | Thaal | 1134 | 1.34% | 38 | محـ | Miim+7aa | 170 | 0.2% |
| 10 | ر | Raa | 1134 | 1.34% | 39 | ى | Alif Broken | 1134 | 1.34% |
| 11 | ز | Zaay | 1134 | 1.34% | 40 | ة | Taaa Closed | 1134 | 1.34% |
| 12 | س | Siin | 2268 | 2.68% | 41 | 0 | 0 | 1085 | 1.34% |
| 13 | ش | Shiin | 2268 | 2.68% | 42 | 1 | 1 | 1085 | 1.34% |
| 14 | ص | Saad | 2268 | 2.68% | 43 | 2 | 2 | 1085 | 1.34% |
| 15 | ض | Daad | 2268 | 2.68% | 44 | 3 | 3 | 1085 | 1.34% |
| 16 | ط | Taa | 2268 | 2.68% | 45 | 4 | 4 | 1085 | 1.34% |
| 17 | ظ | Thaaa | 2268 | 2.68% | 46 | 5 | 5 | 1085 | 1.34% |
| 18 | ع | 3yn | 2268 | 2.68% | 47 | 6 | 6 | 1085 | 1.34% |
| 19 | غ | Ghayn | 2268 | 2.68% | 48 | 7 | 7 | 1085 | 1.34% |
| 20 | ف | Faa | 2268 | 2.68% | 49 | 8 | 8 | 1085 | 1.34% |
| 21 | ق | Gaaf | 2268 | 2.68% | 50 | 9 | 9 | 1085 | 1.34% |
| 22 | ك | Kaaf | 2268 | 2.68% | 51 | ! | Exclamation Marks | 567 | 0.67% |
| 23 | ل | Laam | 2268 | 2.68% | 52 | ؟ | Question Marks | 567 | 0.67% |
| 24 | م | Miim | 2268 | 2.68% | 53 | < | Inferieur a | 567 | 0.67% |
| 25 | ن | Nuun | 2268 | 2.68% | 54 | > | Superieure a | 567 | 0.67% |
| 26 | ه | Haaa | 2268 | 2.68% | 55 | ( | Left | 567 | 0.67% |
| 27 | و | Waaw | 1134 | 1.34% | 56 | ) | Right | 567 | 0.67% |
| 28 | ي | Yaa | 2268 | 2.68% | 57 | / | Slash | 567 | 0.67% |
| 29 | ء | Hamza | 567 | 0.67% | 58 | : | 2Points | 567 | 0.67% |

**Table 2-1:** DataSet Storage.

## Some Images from DataSet

Figure 2-6 depicts a sample of Arabic printed characters written in different word positions and different fonts. Some ligatures are also introduced below.


**Figure 2-6:** Some Images from DataSet.

## 7. Conclusion

that most research literature uses multiple datasets for various uses of some recognition systems, for example, successfully accomplished high accuracy because they used a small dataset. However, the lack of a reference datasets for the Recognition of printed Arabic characters that cover all shapes of Arabic characters and include overlapping characters makes it difficult to compare different approaches and evaluate their accuracy consistently. The proposed dataset is intended to include all shapes of Arabic characters, including those that overlap. It includes 84162 files with character shapes written by 21 different fonts.

# CHAPTER 3: LITERATURE REVIEW

## 1. Introduction

Character recognition refers to technical operations that a computer performs to convert text written on images, whether handwritten or printed, into text files. The computer requires character recognition software to perform this task. This allows you to retrieve the existing text in the image and save it in a file that can be used in a word processor or stored in a database by a computer system. Today, there are many character recognition engines in use, some are paid and some are free, which recognize the Arabic text using a large amount of data and various techniques.

## 2. Previous Studies

The quality and accuracy of character recognition tools have become more effective and improved over the years. Today, there is a wide range of character recognition systems available for use. Table 3. 1 introduced a sample of previous character recognition studies carried out in the last seven years:

| Year: | Langu-age: | Search Name: | Method: | Accur-acy: | Researchers: | DataSet: |
|---|---|---|---|---|---|---|
| 2020 | Arabic | Algorithm design to recognize continuous Arabic writing using artificial neural networks [7]. | Gradient Descent Algorithm | 82% | -Abdul Hafeez Hamed Muhammad Babiker | 200 Arabic words: -(100) Quranic words -(22) Arab countries -(28) Islamic countries -(50) Sudanese cities |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2020 | Arabic | Developing an efficient CNN model for recognition Handwritten digits [8]. | CNN | 99.86% | -Tahmi hassina | -60.000 training images -10.000 testing images. |
| 2019 | Arabic | A novel features and classifiers fusion technique for recognition of Arabic handwritten character script [9]. | Svm<br>CNN<br>Mlp | 99.2% | - Amani Ali.<br>- Ahmed Ali<br>- M. Suresha, | 16800 characters |
| 2020 | Arabic | Efficient Arabic Handwritten Character Recognition based on Machine Learning and Deep Learning Approaches [10]. | LR, LDA, KNN, DT (CART), NB, SVM.<br>CNN | <50%<br>95,15% | Said Gadri | 13440 training images -3360 testing images |
| 2019 | English Hindian kannaba | Size invariant handwritten character recognition using single layer feedforward backpropagation neural networks [11]. | Svm<br>ANN<br>HMMs | 98.08% Digits<br>96.98% alphabets | A. Yousaf,<br>M. J. Khan,<br>M. J. Khan,<br>N. Javed,<br>H. Ibrahim,<br>K. Khurshid,<br>K. Khurshid, | 19.422 English<br>7.720 digits |
| 2019 | English | Automated grading for handwritten answer sheets using convolutional neural networks [12]. | CNN | 92.86% | -Eman Shaikh,<br>Iman Mohiuddin,<br>-Ayisha Manzoor<br>Ghazanfar Latif, | 250 images |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | -Nazeeruddin Mohammad. | |
| 2019 | English | Performance comparison of ANN and template matching on English character recognition [13]. | ANN | 88% | -Muna Ahmed<br>-Dr. Ali Imam Abidi | 60000 training images<br>10000 testing images |
| 2019 | English | Intelligent character recognition using the fully convolutional neural net-<br>Works [14]. | FCNN (fully convolu-tional neural netWorks) | 92.4% | **-Raymond Ptucha**<br>Felipe Petroski Such ;<br>-Frank Brockler | 12000 word |
| 2018 | Arabic | Optical character recognition system for nastalique Urdu-like script languages using supervised learning [15]. | Multi-channal neural network (MCNN) | 92.10% | -Syed.S.<br>-R.Rizvi's ;<br>-Sagheer Abbas;<br>- Muhammad Adnan Khan;<br>- Muhammad Asadullah. | 98.4% training images and<br>97.3% of testing images |
| 2018 | Arabic | Deep sparse auto-en code features learning for Arabic text recognition [16]. | Sparse autoencoder (SAE) Hidden Markov (HMMs). | 99.40% | - Najoua Rahal<br>-Amir Hussain | 1414 training images<br>6472testing images |
| 2017 | Arabic | Making scanned Arabic documents machine-accessible using an ensemble of SVM classifiers [17]. | Svn | 97.30% | Rand. Elanwar, Wenda. Qin, Margrit. Betke, | 109 Arabic document |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2017 | English Indian | Accurate recognition of words in scenes without character segmentation using recurrent neural network [18]. | Hog Rnn | 90% | - Bolan. Su<br>- Shijian. Lu, | 1156 training images 1110 testing images |
| 2017 | English Chinese Indian | Robust shared feature learning for script and handwritten/machine-printed identification [19]. | CNN | 95.46% | - Z. Feng,<br>- Z. Yang,<br>- L.Jin,<br>- S. Huang,<br>- J. Sun, | 23179 training images 7750 testing images |
| 2016 | Arabic | Arabic Handwritten Names by using Deep learning [20]. | CNN (Holistic approach) | 100% | -Samar Fath AlRahman Babiker Adam<br>-Mayada Hamad Al Obaid | 350 training images 70 testing images |

**Table 3-1:** A sample of previous character recognition studies.

## 3. Novel Contributions

We can introduce our contributions as follows:

- Using an efficient method to recognize mixed fonts printed Arabic characters.
- Collecting a large number of images (thousands) that can be used for educational purposes related to data mining and processing.
- Covering a wide range of characters that should exist in text files besides the letters such as digits and punctuation symbols.

The achieved results are quite encouraging. We propose enhancing these findings by modifying various model parameters such as the number of filters, the number of convolution and max-pooling layers, the size of each filter, and the number of training epochs, and the size of data batches.

# CHAPTER 4: THE PROPOSED RECOGNITION SYSTEM

## 1. Introduction

This chapter presented the working environment and the various steps to implement a recognition system for printed Arabic characters .

This chapter aims to present offline recognition of the printed Arabic characters and evaluate its performance, which was implemented using the Convolutional Neural Network.

There are several major steps in the development of the proposed system. Each step affects the accuracy and performance of the recognition, so we concentrate on improving the character recognition accuracy by employing effective features, which are capable of improving recognition accuracy by extracting information related to the statistical shape of the characters.

## 2. Architecture of the Proposed System:

The system input is a scanned image of an Arabic letter, a photograph taken with the camera, or an image stored on a computer, as shown in Figure 4-1, and the output is the printed Arabic character.

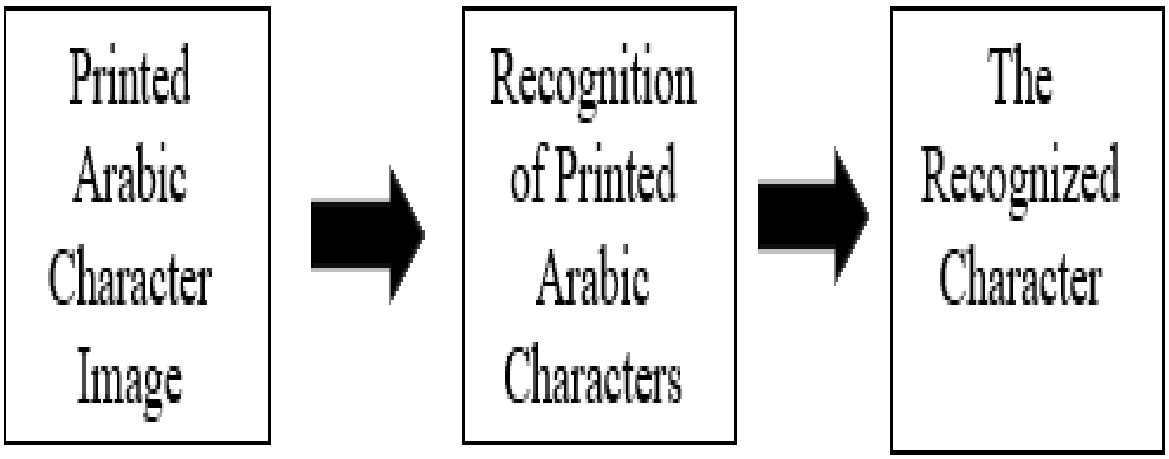



**Figure 4-1:**General view of the system.

Figure 4-2 depicts a detailed view of the system, in which the image goes through four steps before the character is recognized. Each of these steps will be discussed separately in the following sections.

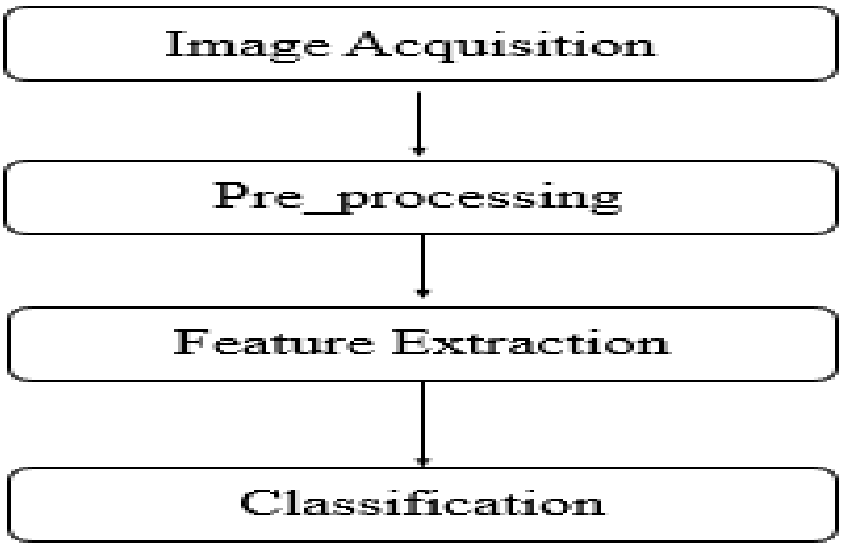



**Figure 4-2:** The Proposed printed Arabic characters recognition system.

## 2.1. Image Acquisition

The first step in the recognition of the printed Arabic characters system is image acquisition.

Image acquisition aims to convert an optical image (real-world data) into a numerical array that can then be manipulated on a computer. Image acquisition is accomplished through the use of scanners as shown in Figure 4- 3:

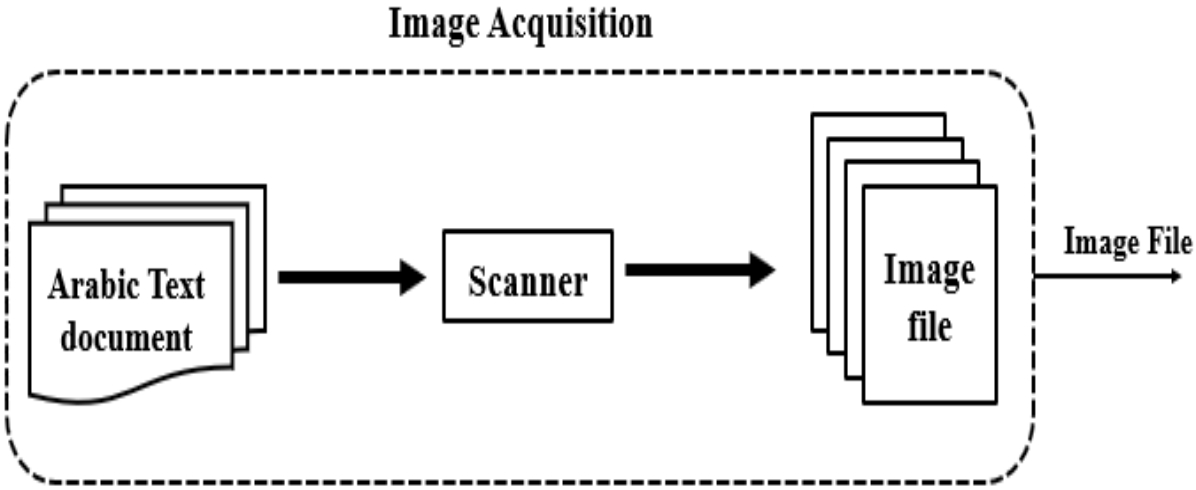



**Figure 4-3:** Image Acquisition.

## 2.2. Preprocessing

It is a series of operations that occur on the image file which are aimed at improving the images and removing redundant in the recognition process. For any image processing

application, the image requires going through the preprocessing step which incorporates the procedures such as shifting, binarization, diminishing, smoothing, and so on. The goal of the preprocessing phase is to reduce the level of noise without changing the information in the document [21], as shown in Figure 4-4:

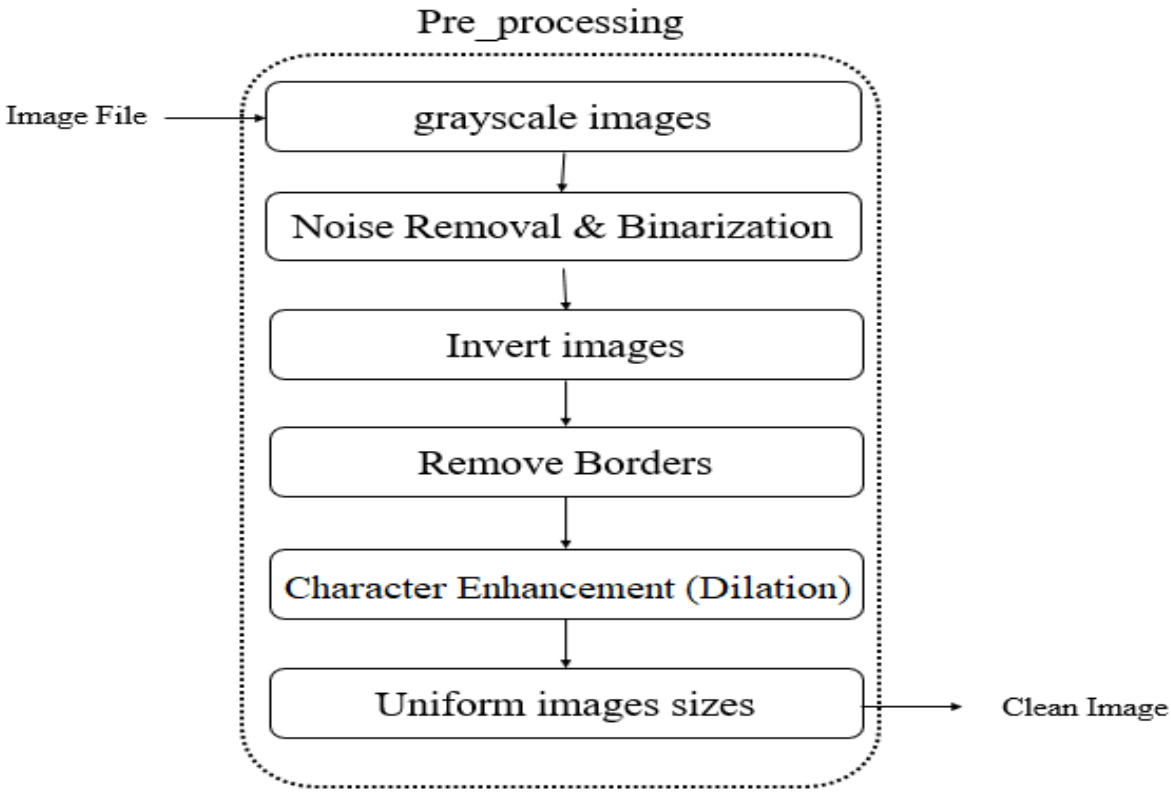



**Figure 4-4:** Image Preprocessing Phase.

### 2.2.1. Adjusting Colors Intensity (grayscale images)

The images have three-channel RGB (height × width × 3), which were converted from a 3-channel color (RGB) into a single channel grayscale image (height × width ×1). The channels were reduced by using an Open CV function as described in the Figure 4- 5.

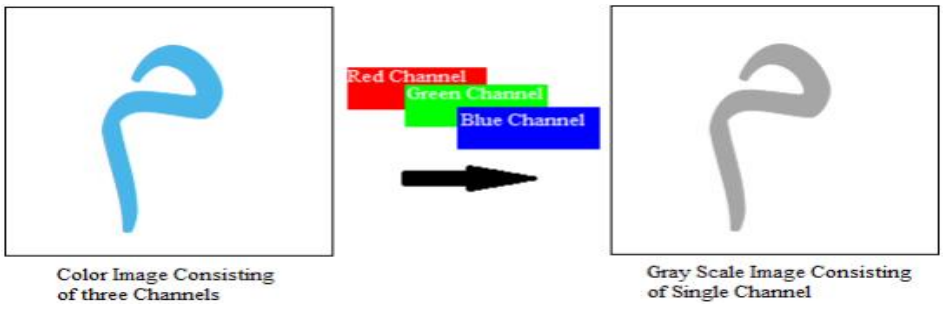



**Figure 4-5:** Grayscale image.

### 2.2.2. Noise Removal and Binarization

Noise is caused by a change in pixel value or the addition of an unwanted bit pattern that has no meaning in the output. It could be introduced during the acquisition procedure as a result of image reproduction and transmission. It's the same as making gaps in document lines, filled loops, or disconnected segments. Noise is commonly found in low-quality documents. This noise appears as either isolated pixels or off-region pixels. Filtering can help us remove this noise.

Moreover, binarization is the process of converting a colored image into one that only has black and white pixels (black pixel value=0 and white pixel value=255). As a general rule, this can be accomplished by setting a threshold (normally threshold=127, as it is half of the pixel range 0–255). If the pixel value exceeds the threshold, it is considered a white pixel; otherwise, it is considered a black pixel.

- Otsus Binarization Algorithm was used for Image Thresholding and Noise Removal.

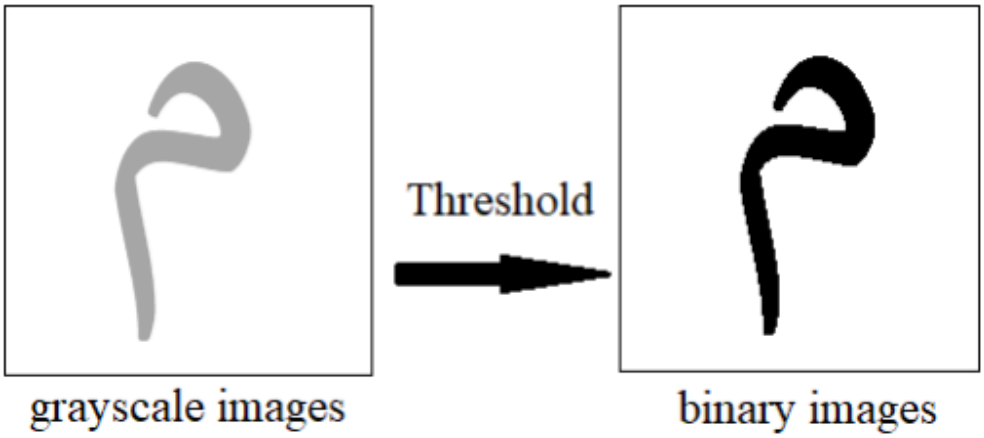



**Figure 4-6:** Binary Image.

• **Otsus Binarization Algorithm:** The objective of this method is to calculate the boundary value of the binary images (the images have two values in the graph). For non-binary images, non-binary images, Otsu's algorithm will not be accurate.

In Figure 4-7, the input image is noisy in the first case, we will apply general Thresholding V=127(Simple Thresholding) to the image. in the second case, we apply Otsus Binarization Algorithm. in the third case, we will first remove the noise from the image by applying a Gaussian filter and then apply the Otsus Binarization Algorithm.

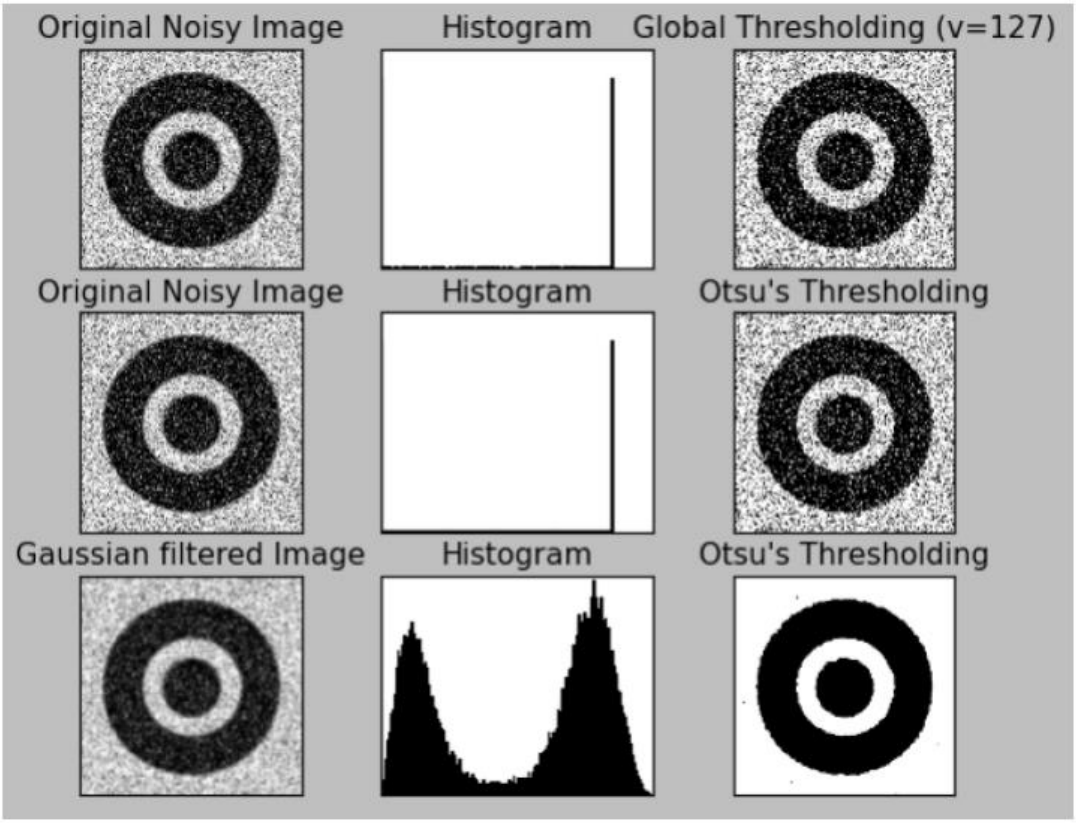



**Figure 4-7:** Otsus Binarization Algorithm.

### 2.2.3. Invert images

Inverting an image is inverting the pixel values, this converts the white pixels to black (background) and the black pixels to white (Characters). As shown in the figure 4-8:

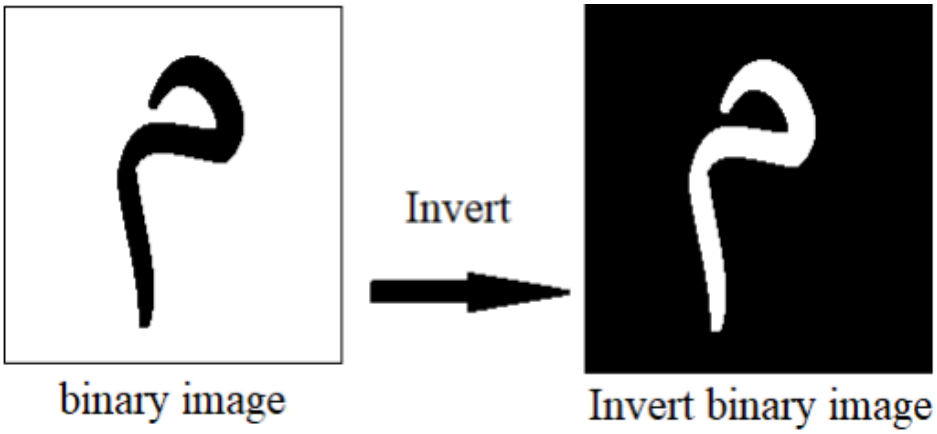



**Figure 4-8:** Invert the image.

### 2.2.4. Remove Borders

The main purpose of this step is to cut the unimportant part from the four sides (Top, Bottom, Left, and Right) of the image (Columns and rows whose sum is 0) and save the image region of interest, as shown in figure 4-9:

**Figure 4-9:** Remove Border.

### 2.2.5. Character Enhancement (Dilation)

Dilation is a primary image operation. Dilation is used to dilate binary images. The primary effect of dilation on a binary image is to continuously increase the boundaries of foreground pixel regions (for example, white pixels, typically). As a result, foreground pixel areas grow in size, while holes within those regions shrink. Figure 4-10(b) shows the Dilation process of Figure 4-10 (a).

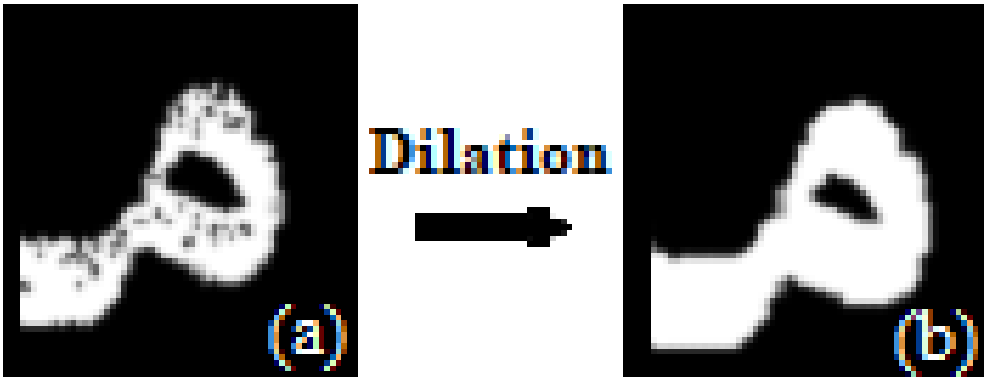



**Figure 4-10:** Character Enhancement (Dilation).

### 2.2.6. Uniform images sizes

The image has been reduced to a fixed size which can be entered into CNN, the larger the image size, the greater the processing time. The delay can be reduced by dimensions reduction. The standard set by the MNIST dataset was followed to reduce the image size to (28x28), as clarified in figure 4-11:

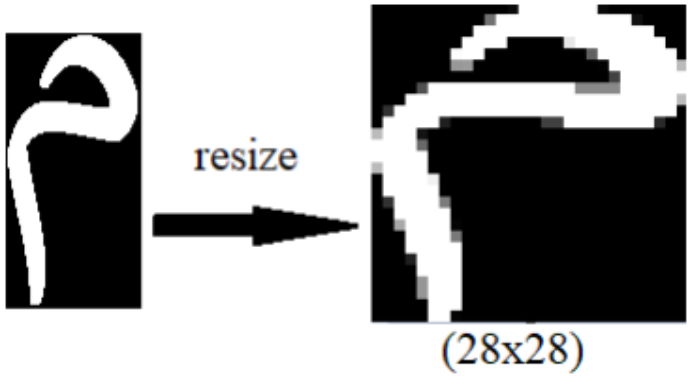



**Figure 4-11:** Resize.

### 2.3. Feature Extraction

The feature extraction phase's goal is to take the most important and discriminating characteristics of the character image to recognize. The selection of various advantages can have a significant impact on the classification performance.

In our system, convolutional neural networks were used for feature extraction.

#### 2.3.1. Convolutional Neural Networks

The name" Convolutional Neural Network" indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural network that uses convolution in place of general matrix multiplication in at least one of their layers [22]. It is a class of Deep Neural Networks (DL) inspired by the visual system of living things. It is more popular for analyzing visual images. It provides effective solutions to many problems and structures due to a strictly mathematical perspective that improves machine-learning methods to recognize letters. Networks are local connections, shared weights, aggregation, and the use of multiple layers. CNN is based on a mathematical operation called convolution. Convolution is a specialized type of linear operation. At least one matrix multiplication in its layers to play each pixel in the image with each value of each kernel.

The goal of the latter is to generate a lot of original images that enhance the different features extracted from the original images, making the classification process more powerful in CNN. The network contains a sequence of layers, as will be explained in detail soon.

First, **Convolution:** extract features and apply filters.

**pooling Max** Reduce size and keep mission features.

**Faltering** Convert to a one-dimensional array.

There are two main parts to a CNN architecture, as shown in the figure 4-12:

- A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction.
- A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

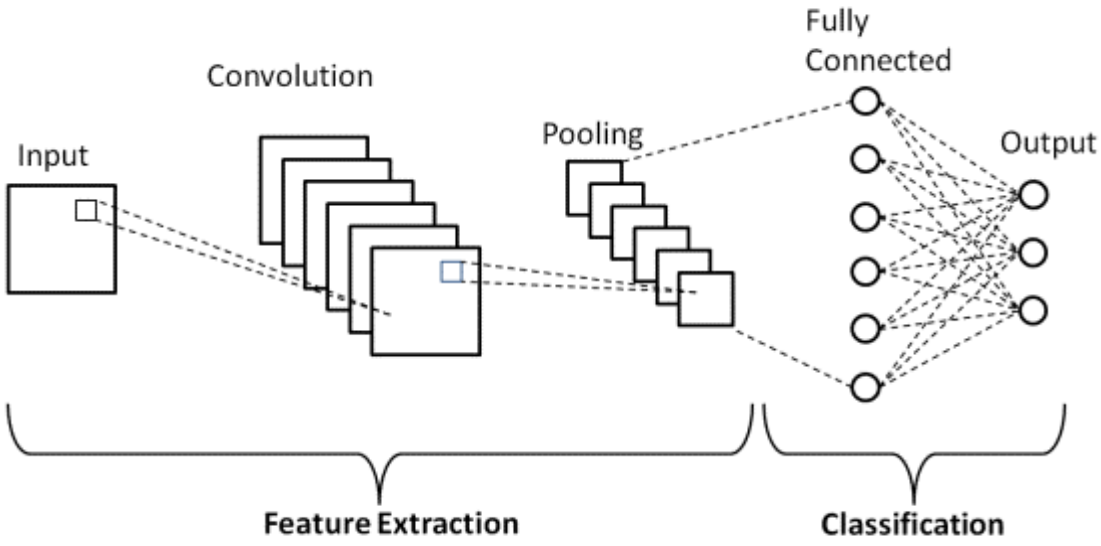



**Figure 4-12**:Convolutional Neural Networks

#### 2.3.1.1. CNN architecture and layers:

Convolutional Neural Networks (CNN) is a type of feedforward network that is also known as a multilayer perceptron and is trained using backpropagation. The primary goal of CNNs is to recognize visual patterns in images. CNN inputs are processed through a series of steps known as layers. The convolutional layer is the first layer of a CNN, and it is followed by the ReLU layer, pooling layer, normalization (flattening) layer, and fully connected layer.

##### 2.3.1.1.1. Convolution Layer

The CNN's core building block is the convolution layer. It carries the majority of the computational load on the network.

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image as shown in the

figure 4-13, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

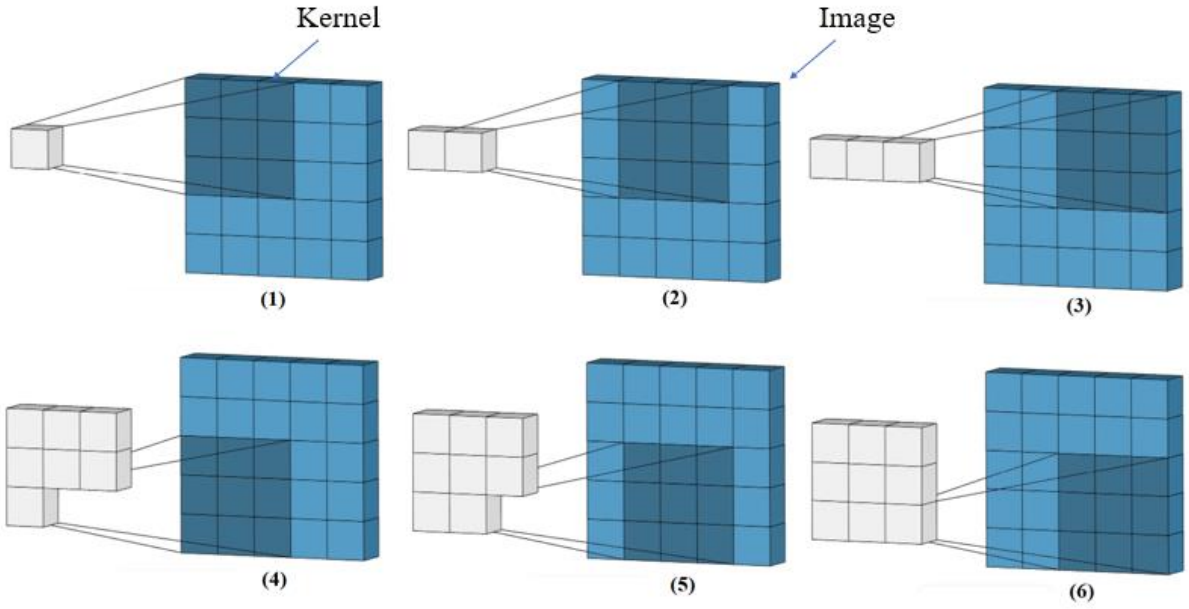



**Figure 4-13:**Illustration of Convolution Operation

This layer computes the dot product of two matrices as shown in the figure 4-14, one of which is the set of learnable parameters known as a kernel and the other is the restricted portion of the receptive field. The kernel is smaller in space than an image but more detailed. This means that if the image has three (RGB) channels, the kernel height and width will be small, but the depth will span all three channels.

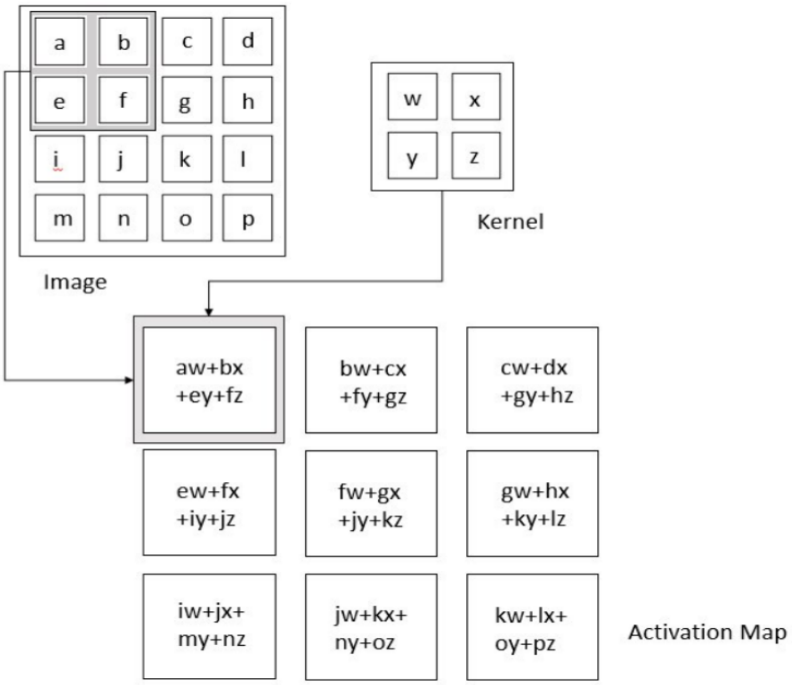



**Figure 4-14**:Convolution Operation

The kernel slides across the image's height and width during the forward pass, producing an image representation of that receptive region. This generates a two-dimensional representation of the image known as an activation map, which contains the kernel's response at each spatial position in the image. A stride is the sliding size of the kernel.

#### 2.3.1.1.2. Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized image section, as shown in the figure 4-15.

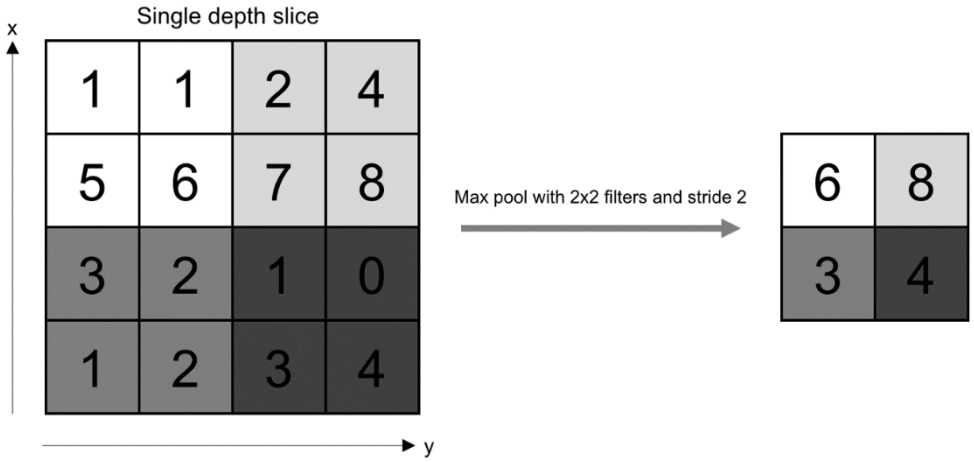



**Figure 4-15:**Pooling Operation

#### 2.3.1.1.3. Flattening Layer

Pooled feature maps are flattened into a vector of input data to be passed through the fully connected layer in this step. Figure 4-16 depicts a flattening operation in action.

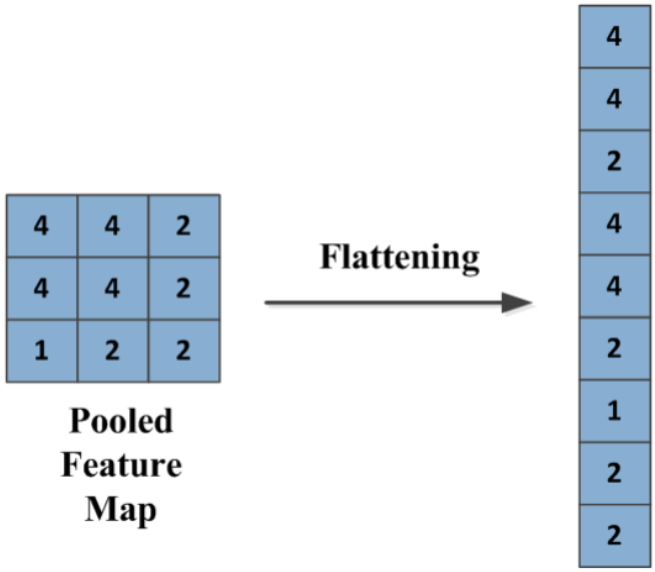



**Figure 4-16:**Example of Flattening Operation.

#### 2.3.1.1.4. Fully Connected Layer

A fully connected layer is one in which a convolutional neural network evolves into an artificial neural network. This layer is shared by all neurons. By the end of this layer, the network has made a prediction, and the cost function, also known as the loss function in Convolutional Neural Networks, has been calculated. The loss function measures how accurate the network is, and to improve its effectiveness, weights, and feature detectors are adjusted.

In general, at the final fully connected layer, a special activation function is used to generate a probabilistic result for each class. For multi-class problems, mostly the Softmax function is preferred.

#### 2.3.1.1.5. Softmax Function

The softmax function calculates the probability of each class. In other words, it shows which class the input is more likely to match and since it outputs the probabilities, the sum of all values equals 1. Softmax function can be expressed as:

$$softmax(x_k) = \frac{e^{x_k}}{\sum_{i=1}^{C} e^{x_i}}$$

Here, $C$ is the number of classes.

### 2.3.2. Building Deep Convolutional Neural Network Model

A framework library in computer programming comprises preset routines for certain purposes. Deep learning frameworks such as Pytorch, Theano, and TensorFlow are popular. Keras, a high-level neural network API, was used for implementation in this study. Keras is a deep learning API open-source written in Python, running on top of TensorFlow's machine learning platform. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research [23].

The time required to train a Deep Convolutional Network may vary depending on the size of the dataset and available processing power.

Furthermore, Anaconda JetBrains PyCharm Community Edition 2019.3, the scientific python development environment was used for programming purposes.

The dataset was divided into training and testing sub-datasets to assess network performance. The training set is used to train the model using known outputs, while the testing set is used to evaluate the final model performance after training. A frequent ratio for partitioning datasets is 80% to 20%, which is also known as the Pareto principle [24]. This means that 80% of the data is used to train the network and 20% is utilized to verify it.

To recognize printed Arabic characters in this study, the dataset was first divided by 80 percent to 20 percent, with 67542 images used for training and 16620 images used for testing. In testing, the images were chosen at random.

Because there is no rule of thumb for suitable layer setup, the best way to start a deep neural network configuration is by trial and error. Because the primary goal of this study is to classify

the given input into 58 classes, multiple layer configurations were tested with the created dataset to attain better testing results.

We created two models to classify and compare the results between them.

### 2.3.3. Training

During Model training, we strive to minimize the loss while also ensuring that the model generalizes effectively on new data for the model to capture the underlying structure of the data. Iterative updates are made to Convolutional Neural Network training. A single training period is insufficient and leads to underfitting where underfitting is a scenario in data science when a data model is unable to correctly represent the relation between input and output variables, resulting in a high error rate on both the training set and unseen data. It happens when a model is overly simplistic, which might happen when a model requires more training time, more input characteristics, or less regularization. When a model is underfitted, it cannot identify the dominating trend in the data, resulting in training mistakes and poor model performance. A model that does not generalize effectively to new data cannot be used for classification or prediction tasks. The ability to generalize a model to new data is ultimately what allows us to utilize machine learning algorithms on a daily basis to make predictions and classify data.

Given the complexities of real-world challenges, training a Convolutional neural network may require hundreds of epochs where in machine learning, an epoch is defined as one complete run of the training dataset through the algorithm. This number of epochs is a critical hyperparameter for the algorithm. It defines the number of epochs or full passes of the whole training dataset through the algorithm's training or learning phase. The dataset's internal model parameters are modified with each epoch. As a result, the batch gradient descent learning algorithm is named after a single batch epoch. An epoch's batch size is often one or more and is always an integer value in what is epoch number.

When developing a Convolutional Neural Network model, we set the number of epochs parameter before the training starts. However, we don't know how many epochs are appropriate for the model at first. We must determine when the network will achieve high recognition accuracy based on the Convolutional neural network design and data set.

To determine model convergence for Convolutional neural network models, learning curve graphs are commonly examined. In general, we produce graphs of loss (or error) vs. epoch or accuracy vs. epoch. As the number of epochs rises during training, we predict the loss to reduce and the accuracy to increase. However, we anticipate that both loss and accuracy will normalize at some time.

As a result, we anticipate that the learning curve graphs will improve over time until they reach convergence. If we continue to train the model, it will overfit, and validation errors will rise.

### 2.3.4. CNN Model 1

In machine learning, data normalization is used to make model training less sensitive to feature size. This enables our model to converge to better weights, resulting in a more accurate model. The model also contains two convolution layers with a filter size of 3x3 in every layer they are followed by a max-pooling layer that has a pool size of 2x2, respectively. The first convolution layer creates 8 filter detectors that are convolved with the input layers to produce a tensor of outputs whereas the second convolution layer creates 16 filters. In this case, the input of the layer is the binarized character images (28x28x1). Then, each output in the convolution layer was activated by using the rectified linear unit as an activation function. Finally, pooled feature maps were flattened and the fully connected layer at the end classified the given input to one of 58 neurons of the output layer. The layers structure of the proposed model is shown in Figure 4-17.

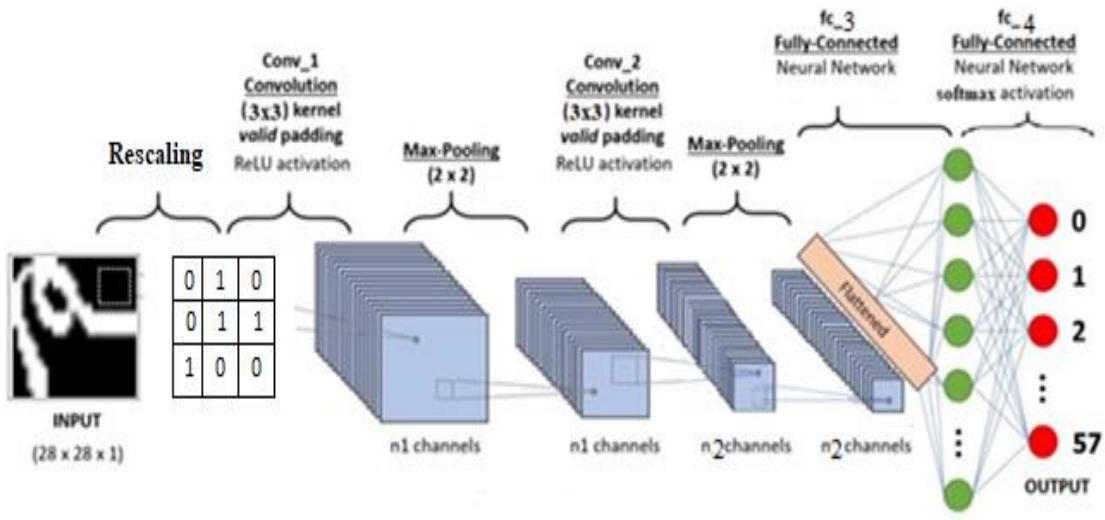



**Figure 4-17:** Structure of the CNNs Model 1.

The Softmax activation function was used to output probabilities of each class to represent which class is more likely to match the given input.

Adam optimizer with adaptive learning rate was used to find the minimum point of the cost function.

The CNN Model 1 has seven layers. Rescaling on the first layer. The second layer is made up of an input image with the size (28x28x1). It is convolved using 8 filters of size 3x3, resulting in a 28x28x8 dimension. The third layer consists of a Pooling operation with a filter size of 2x2. As a result, the final image size will be 14x14x8. Similarly, the fourth layer performs a convolution operation with 16 filters of size 3x3, resulting in a dimension of 14x14x16, followed by a fifth pooling layer with a similar filter size of 2x2, resulting in a dimension of 7x7x16. After reducing the image dimension, the eighth layer is a fully connected convolutional layer with 784 units. The final layer will be a Softmax output layer with '58' possible classes, as shown in Table 4-1.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| Rescaling | 28 x28x1 | 0 |
| Conv2D | 28x28x8 | 224 |
| MaxPooling2D | 14x14x8 | 0 |
| Conv2D | 14x14x16 | 1168 |
| MaxPooling2D | 7x7x16 | 0 |
| Flatten | 784 | 0 |
| Dense | 58 | 45530 |
| Total params: 46,922<br>Trainable params: 46,922<br>Non-trainable params: 0 | | |

**Table 4-1:** Architecture of the CNN Model 1.

### 2.3.5. CNN Model 1 Training

The network was trained by using 67542 images while the dataset was split by 80% to 20%. Epoch number was selected 10 while the batch size was 128. This means weights were updated in every 128 samples and this process was repeated 10 times. Training and validation results were shown in following Figure 4-18.

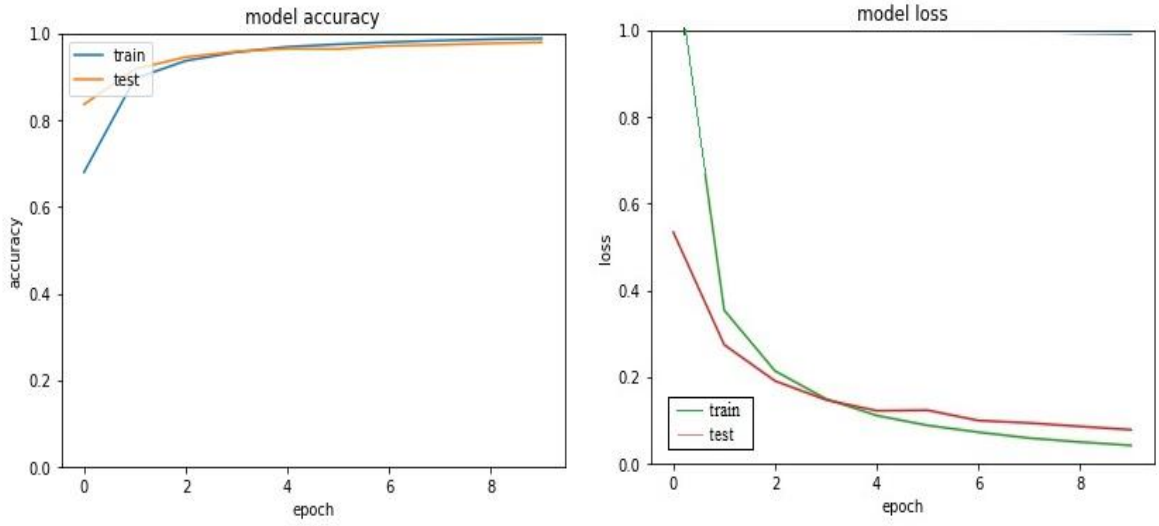



**Figure 4-18:** Accuracy and Loss Results of the CNN Model 1 (Epoch=10).

Then, the number of epochs was increased to 30, and the model was trained and evaluated again by using the same dataset. As seen in Figure 4-19,

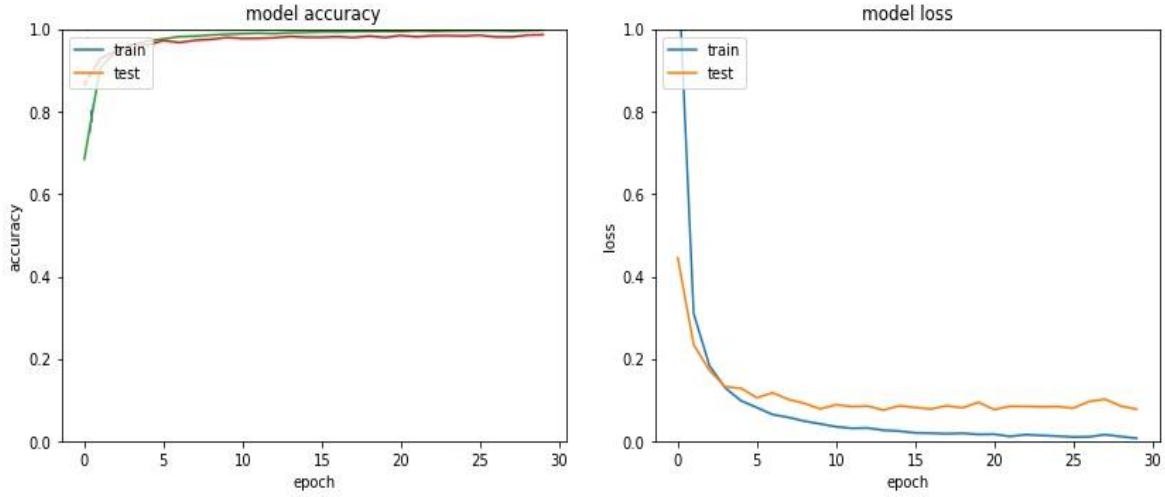



**Figure 4-19:** Accuracy and Loss Results of the CNN Model 1 (Epoch=30).

Furthermore, the number of epochs was raised to 70, and the model was trained and assessed using the same dataset. As shown in figure 4-20,

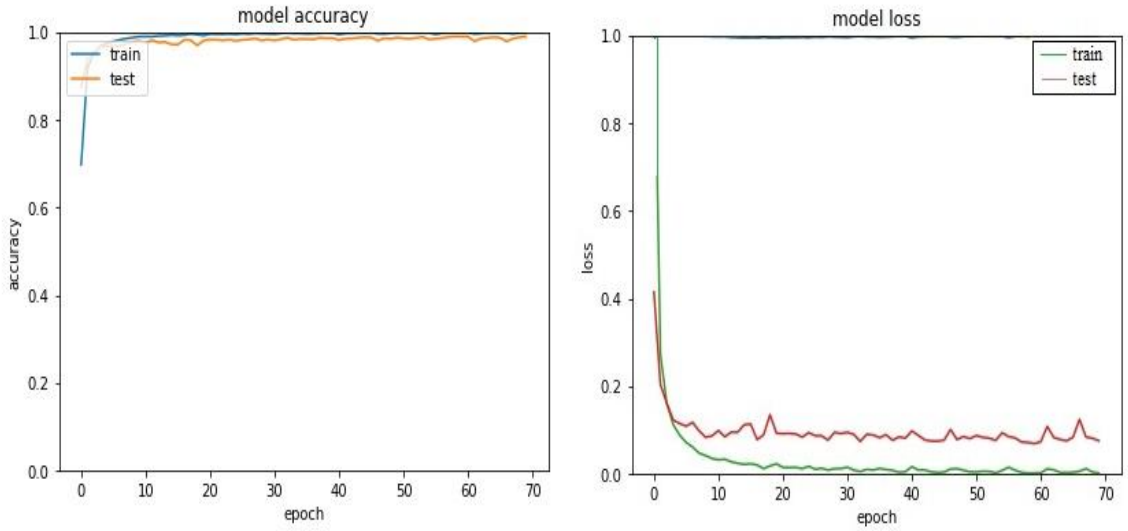



**Figure 4-20:** Accuracy and Loss Results of the CNN Model 1 (Epoch=70).

After that, the number of epochs was raised to 150, As shown in the figure 4-21.

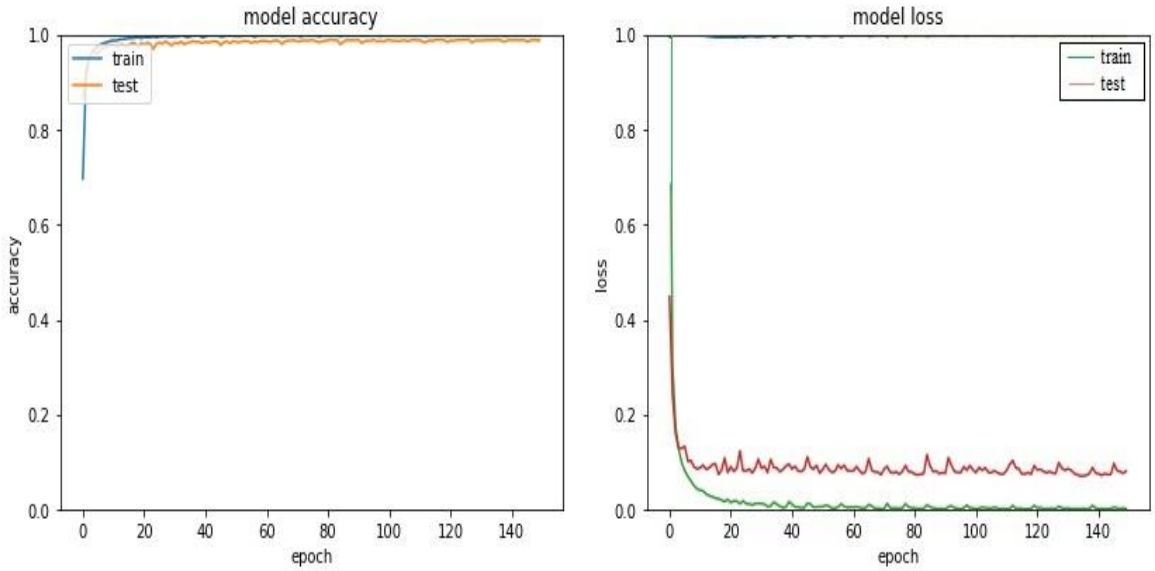



**Figure 4-21:** Accuracy and Loss Results of the CNN Model 1 (Epoch=150).

Finally, the number of epochs was increased to 300, As shown in Figure 4-22.

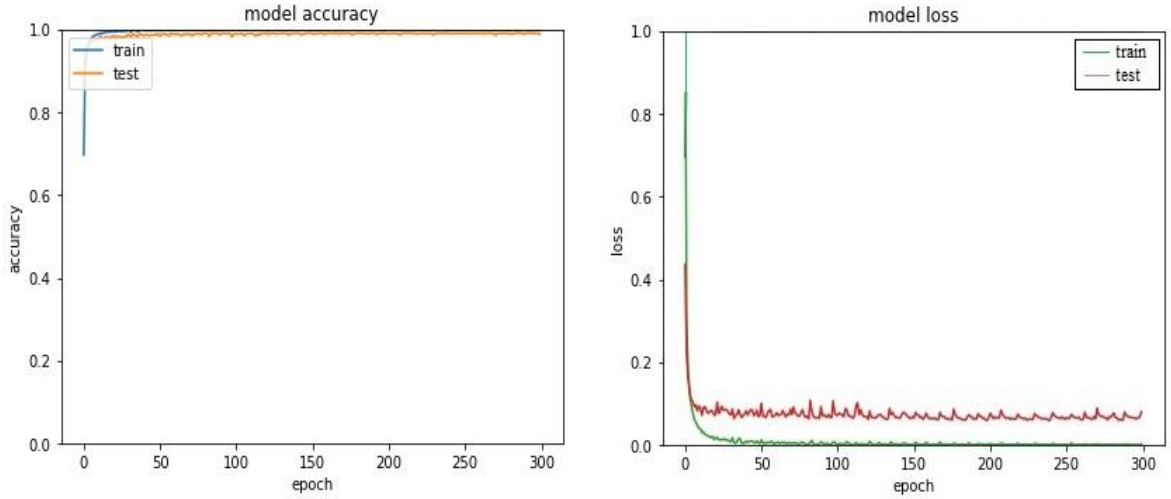



**Figure 4-22:** Accuracy and Loss Results of the CNN Model 1 (Epoch=300).

### 2.3.6. CNN Model 2

A Convolutional Neural Network with a rescaling layer for normalization was created (it is the process of converting real-valued numeric properties into a 0 to 1 range). Data normalization is used in machine learning to reduce the sensitivity of model training to feature size. This allows our model to converge to better weights, resulting in a more accurate model. It also features three convolution layers with a filter size of 3x3 in each layer, followed by a max-pooling layer with a pool size of 2x2 in each layer. The first convolution layer generates eight filter detectors, which are convolved with the input layers to form a tensor of outputs. The second convolution layer generates 16 filters. The third convolution layer generates 64 filters. In this case, the layer's input is binarized character pictures (28x28x1). The rectified linear unit was then used as an activation function to activate each output in the convolution layer. Finally, the pooled feature maps were flattened, and the fully connected layer assigned the supplied input to one of the output layer's 58 neurons. The suggested model's layers structure is depicted in Figure 4-23.

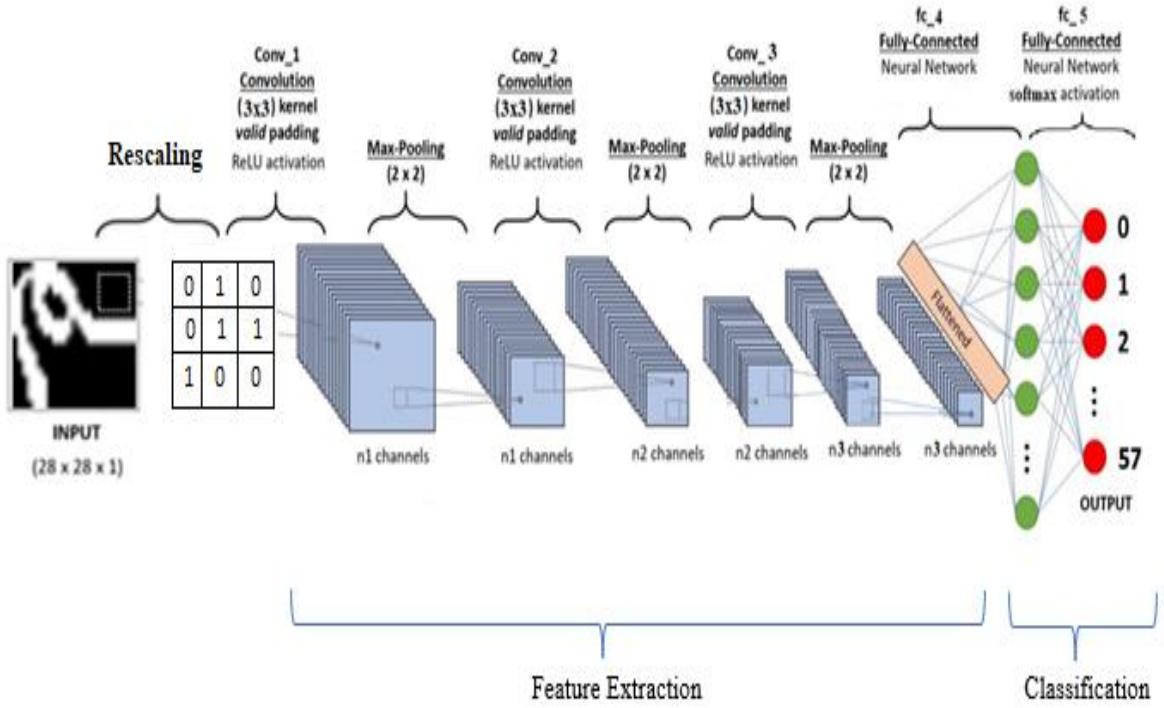



**Figure 4-23:** Structure of the CNNs Model 2.

The Softmax activation function was used to generate probabilities for each class, indicating which class is most likely to match the supplied input. The Adam optimizer with adaptive learning rate was used to identify the cost function's minimal point.

CNN Model 2 consists of 9 layers. The first layer is rescaling. The second layer consists of an input image with dimensions of (28×28x1). It is convolved with 8 filters of size 3×3 resulting in the dimension of 28x28x8. The third layer is a Pooling operation with a filter size of 2×2 and a stride of 2. Hence the resulting image dimension will be 14x14x8. Similarly, the fourth layer is also involved in a convolution operation with 16 filters of size 3×3 resulting in the dimension of 14x14x16, followed by a fifth pooling layer with a similar filter size of 2×2. Thus, the resulting image dimension will be reduced to 7x7x16. Similarly, the sixth layer is also involved in a convolution operation with 64 filters of size 3×3 resulting in the dimension of 7x7x64, followed by a seventh pooling layer with a similar filter size of 2×2. Thus, the resulting image dimension will be reduced to 3x3x64. Once the image dimension is reduced, the eighth layer is a fully connected convolutional layer with 576 units. The final layer will be a Softmax output layer with ‘58’ possible classes. As shown in the Table 4-2:

| Layer (type) | Output Shape | Param # |
|---|---|---|
| Rescaling | 28 x28x1 | 0 |
| Conv2D | 28x28x8 | 224 |
| MaxPooling2D | 14x14x8 | 0 |
| Conv2D | 14x14x16 | 1168 |
| MaxPooling2D | 7x7x16 | 0 |
| Conv2D | 7x7x64 | 9280 |
| MaxPooling2D | 3x3x64 | 0 |
| Flatten | 576 | 0 |
| Dense | 58 | 33466 |
| Total params: 44,138<br>Trainable params: 44,138<br>Non-trainable params: 0 | | |

**Table 4-2:** Architecture of the CNN Model 2.

### 2.3.7. CNN Model 2 Training

The network was trained on the same data set for epoch = 10. The training and validation results are shown in the figure 4-24.

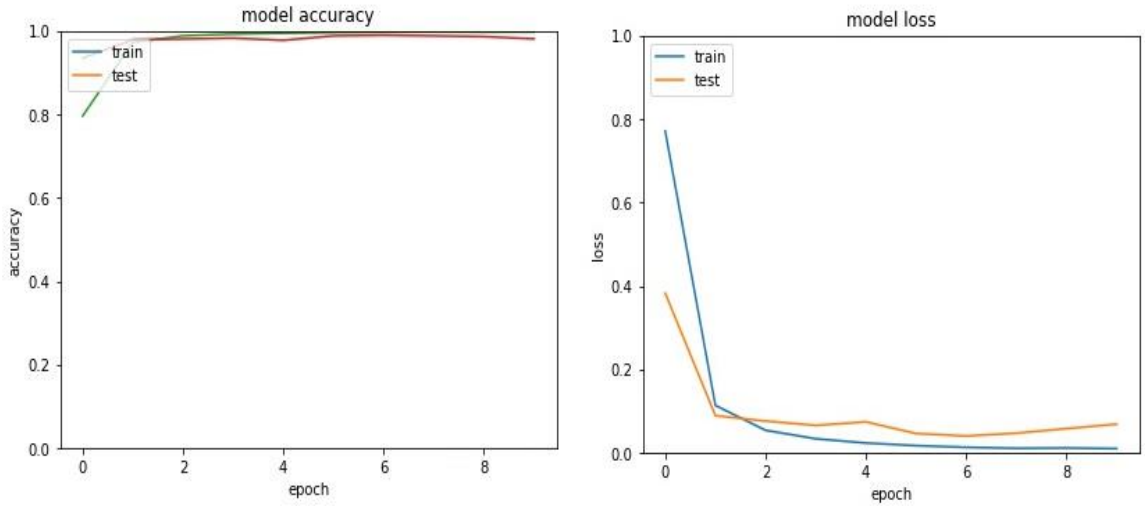



**Figure 4-24:** Accuracy and Loss Results of the CNN Model 2 (Epoch=10).

Then, the model was trained and evaluated again using the same data set. As shown in the figure 4-25., after increasing the number of epochs to 30.

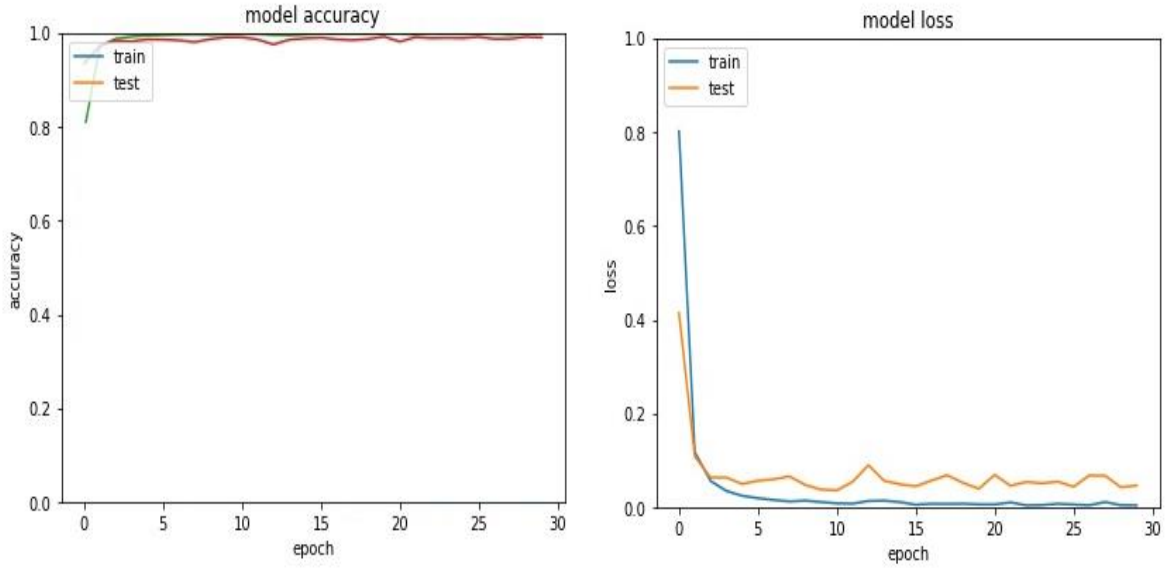



**Figure 4-25:** Accuracy and Loss Results of the CNN Model 2 (Epoch=30).

Increasing the number of epochs to 70. As shown in the figure 4-26.

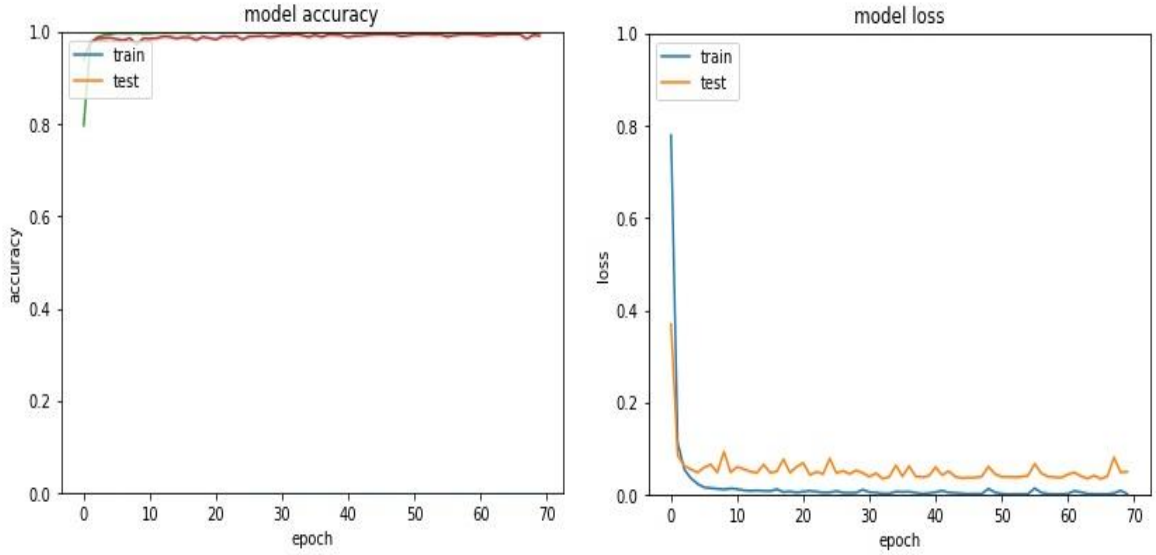



**Figure 4-26:** Accuracy and Loss Results of the CNN Model 2 (Epoch=70).

As shown in figure 4-27, when the number of epochs is increased to 150.

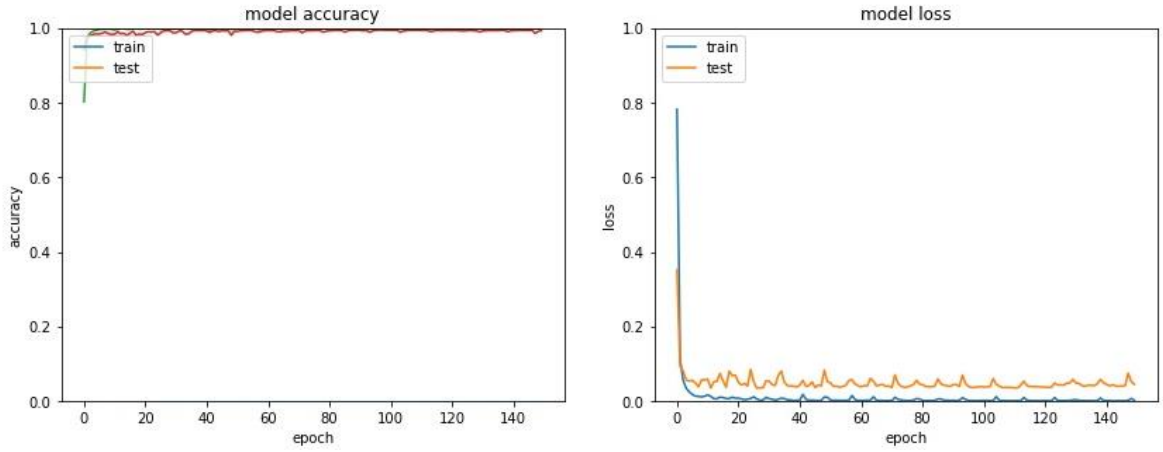



**Figure 4-27:** Accuracy and Loss Results of the CNN Model 2 (Epoch=150).

The number of epochs has been increased to 250, As shown in the figure 4-28.

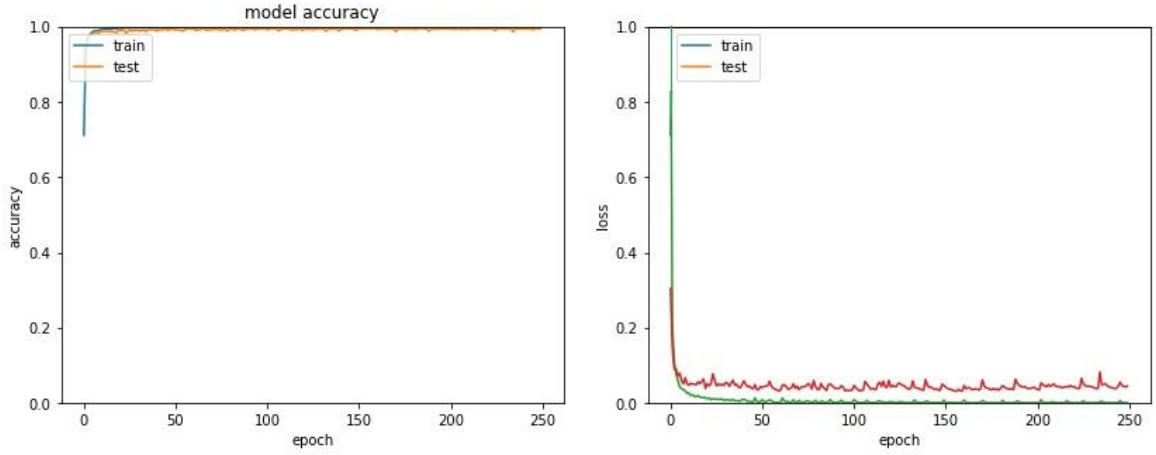



**Figure 4-28:** Accuracy and Loss Results of the CNN Model 2 (Epoch=250).

## 2.4. Classification

Classification is the generic process for grouping entities by similarity. The goal is to make each cell of the classification as similar as possible (to minimize within-group variance). This may involve maximizing between-group variance, by maximizing the distance between each cell. The individual cell or category within the larger classification is called a class. The term "classification" is used to refer both to the process and the result of the process, as in "The classification process produced an excellent classification." An adequate classification must be simultaneously mutually exclusive and exhaustive. In other words, the classification must provide a place (but only one place) for every individual in the sample [25].

This process, as well as the extraction of features, are regarded as key factors in the recognition processes. The retrieved features are compared with other attributes of previously known characters (model) to determine the closest match (58 classes).

We recall that our dataset was separated into three subsets: training, validation, and testing. First, we ran our proposed CNN model 1 on the training set for 10, 30, 70, 150, and 300 epochs with batch sizes of 128. Then, using the same batch size, we ran CNN model 2 on the training set for 10, 30, 70, 150, and 250 epochs. The inclusion of convolutional layers speeds up training and boosts accuracy dramatically.

The Table 4-3 summarizes the accuracy and loss values for you from CNN Model 1 and CNN Model 2.

| Model | Training Accuracy | Validation Accuracy | Testing Accuracy | Training Loss | Validation Loss | Testing Loss |
|---|---|---|---|---|---|---|
| **CNN Model 1** | 99.94% | 98.86% | 99.53% | 0.22% | 7.98% | 0.46% |
| **CNN Model 2** | 99.99% | 99.47% | 99.86% | 0.0026% | 4.43% | 0.13% |

**Table 4-3:** Accuracy and Loss

In comparison between CNN model 1 and the CNN model 2, we find that the CNN model 2 is more accurate, as it outperforms the other in training accuracy, verification accuracy, and testing accuracy, and the lowest percentage of training Loss, validation loss and testing loss of the CNN model 1.

Figure 4-29 presents a detailed diagram of the use of the printed Arabic character recognition system CNN model in order to improve the performance of the classification task.

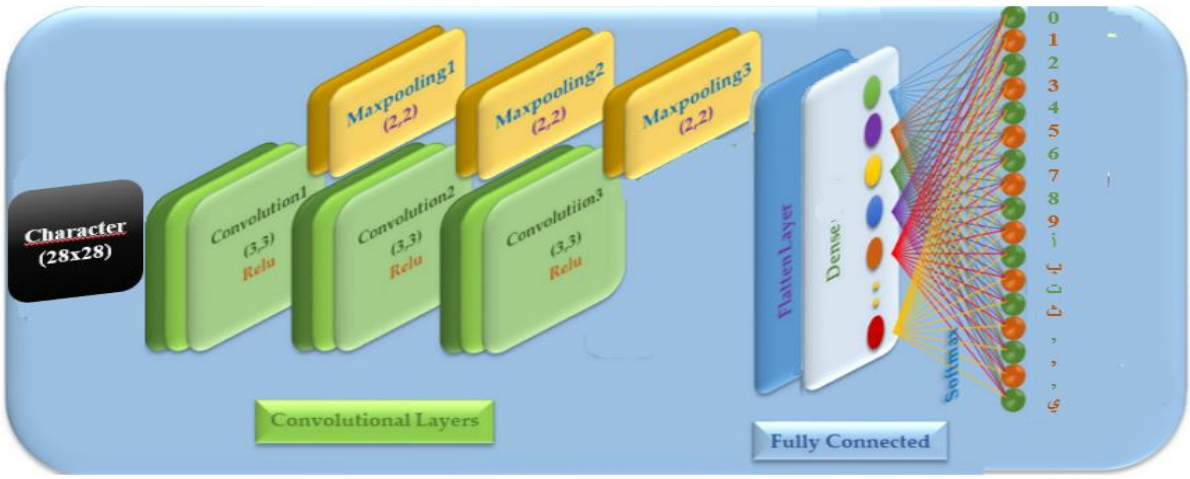



**Figure 4-29:** Used CNN Model

## 3. System Interface

Figure 4-30 shows the design of the system interface. The interface contains the following areas:

- New: is Button opens a new screen to file dialog to choose the image of the character to be recognized.
- Character Recognition: Recognize an image and display the result in Label2.
- Label1: is Label Shows the image of the character being selected.
- Label2: is Label Shows character classification.
- In addition to six other buttons for operations in the processing stage.
- grayscale image: is a button to convert the image to a gray image.
- Noise Removal and Binarization
- Invert image
- Remove Borders
- Character Enhancement
- Resize(28, 28).

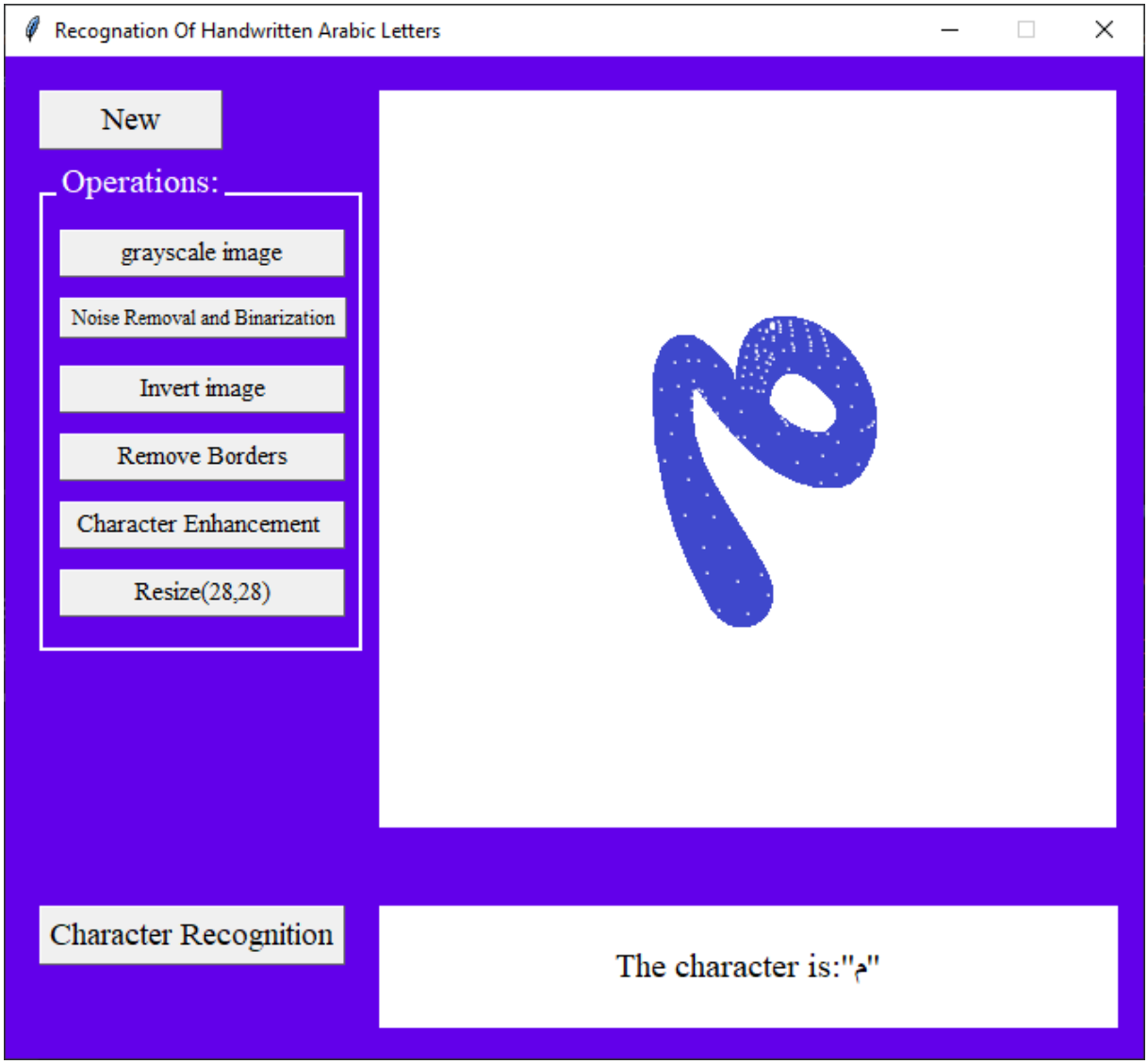



**Figure 4-30:** Interface.

## 4. Conclusion

In this chapter, we have presented a detailed description of the printed Arabic characters recognition system.

We developed a recognition system that has three stages: preprocessing, feature extraction, and classification.

Basically, Convolutional Neural Network was used for feature extraction and classification which has proven its efficiency in the recognition process.

# CHAPTER 5: EXPERIMENTAL RESULTS

## 1. Introduction

After continuous training of the Convolutional Neural Network models, starting with the first stage where the number of epochs is 10, and testing the models after each training. The number of times reached by the training of the two models was five, until the network reached a relatively high recognition accuracy, as the two models recognized a large number of letters, estimated by accuracy of 99.53% and an error of 0.46% for the first model, and an accuracy of 99.86% and an error of 0.13% for CNNs Model 2. The two Convolutional Neural Network models were tested after the training phase on a database consisting of 67,542 images (20% - 80%).

When training a convolutional neural network, we keep the generalization of the bypass neural network in mind. Generalization refers to the network's ability to learn from the data supplied to it and apply what it has learnt to different types of input. When training a convolutional neural network, some data is used to train the network and some data is saved to test its performance. If the convolutional neural network performs well on the untrained data, we may conclude that the network has generalized well on the data.

generalization improvement methods, in general, there are several ways to improve generalization ability:

- Increasing the number of Epochs
- Increasing/decreasing the number of filters for a convolutional layer.
- Use the correct cost function.
- Use good weight conditioning techniques.
- Increasing the population of learning samples, based on the augmentation of data.
- The regularization of the architecture such as weight decay and dropout.
- An important method for increasing generalization capability is the modification of network structures.

Below we present an intuitive result for the theories of parameter optimization through the experimental results of two methods: increasing the number of epochs and increasing the number of convolutional layers of a neural network.

## 2. Testing

The purpose of testing is to compare the outputs from the neural network against targets in an independent set (the testing instances). Note that the testing methods are subject to classification Printed Arabic Characters. The model was tested from 16,620 images (80%-20%) and the recognition accuracy was calculated help us diagnose if the model has ffi-learned, under-learned, or fits the learning set.

## 3. CNN Model 1

When building a bypass neural network model, we set the number of epochs parameters before starting training. However, at first, we cannot know how many tenses are suitable for the form. Depending on the structure of the neural network and the data set, we need to determine when the neural network weights converge. Variables identified in or training (stage 1):

batch size = 128.

epoch=10

batch size: Represents the number of images that will be sent to the network at a given time.

epochs: The number of times the training data is entered into the network.

The number of epochs in the model training was increased in order to improve the model's performance.

These variables are considered the basic variables that control the training of the network, in each stage Training for the network only number epochs are changed. Five experiments were conducted to train the CNN 1 model as follows:

| | Epochs | DataSet | | | Testing | |
|---|---|---|---|---|---|---|
| | | Testing DataSet | Recognized DataSet | Unrecognized DataSet | Accuracy | Loss |
| Stage 1 | 10 | 16620 | 16491 | 129 | 99.22% | 0.77% |
| Stage 2 | 30 | | 16522 | 98 | 99.41% | 0.34% |
| Stage 3 | 70 | | 16561 | 59 | 99.64% | 0.35% |
| Stage 4 | 150 | | 16555 | 65 | 99.60% | 0.39% |
| Stage 5 | 300 | | 16542 | 78 | 99.53% | 0.46% |

**Table 5-1:** Training stages and the testing results by CNN Model 1.

Table 5-1 shows the training stages that occurred and the test results after each training of the Convolutional Neural Network Model 1, where the model was trained in stage 1 by specifying the number of epochs 10, so the model was able to recognize 16,491 characters out of 16,620 characters, but it did not recognize the remaining 129 characters, meaning that the accuracy is 99.22% and the error rate is 0.77%. Testing it on the same set of data by changing the number of epochs to 30. At this stage, the model was able to recognize 16522 characters out of 16620 characters. Failed to recognize 98 characters. In stage 3, in which the number of epochs was determined at 70, the performance of the model improved better than the previous times, as it was able to recognize 16,561 characters, but failed to recognize 59 characters out of 16,620 characters. As for the last two stages, when the number of epochs was raised to 150 and 300, the model's accuracy decreased to 99.53% after it rose to 99.64%, and the error rate increased to 0.46% after it reached 0.35%.

In stage 1, 2 and 3: The reason for the failure of the model in recognizing some letters is due to the lack of learning of the model well, due to the similarity of the Arabic letters among themselves.

In stage 4 and 5, the so-called overfitting has occurred. When training a convolutional neural network model, a sample dataset is used to train the model. However, if the model trains on sample data for too long, it may begin to learn the "noise," or irrelevant information, inside the dataset. The model gets "overfitted" when it memorizes the noise and fits too closely to the training set, and it is unable to generalize adequately to new data. If a model is unable to generalize adequately to new data, it will be unable to accomplish the classification or prediction tasks for which it was designed.

This model fails to recognize some characters at every stage of training. Samples of these characters are shown in Figures 5-1, 5-2, 5-3, 5-4, and 5-5.

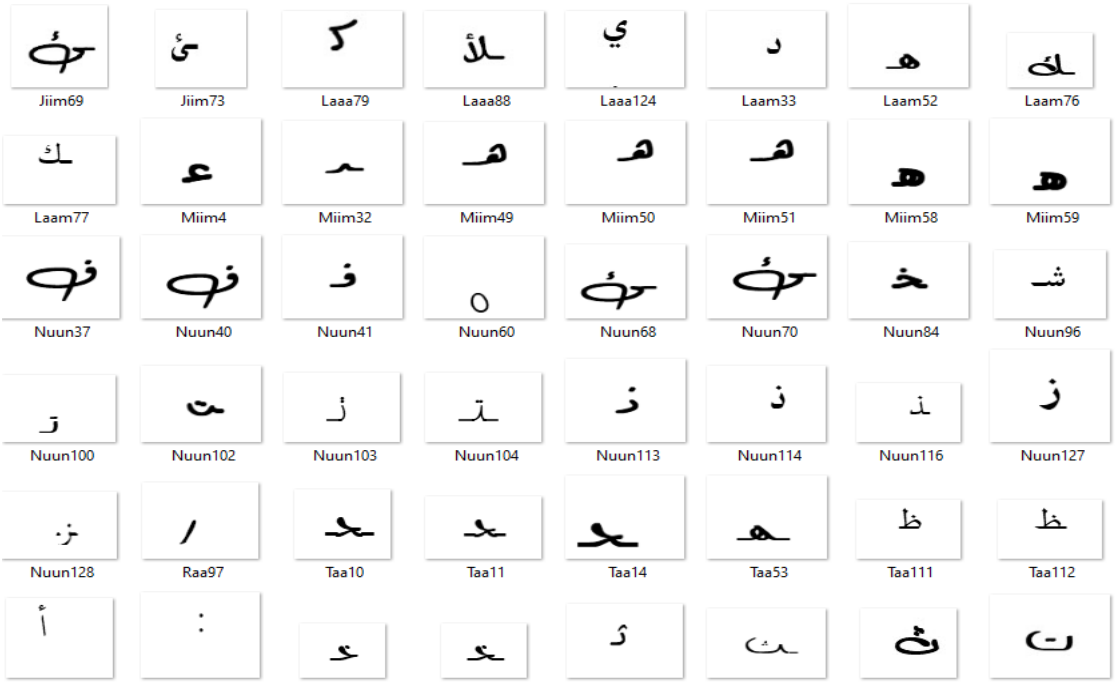



**Figure 5-1:** Samples of unrecognized characters using CNN mode11 in the stage 1.

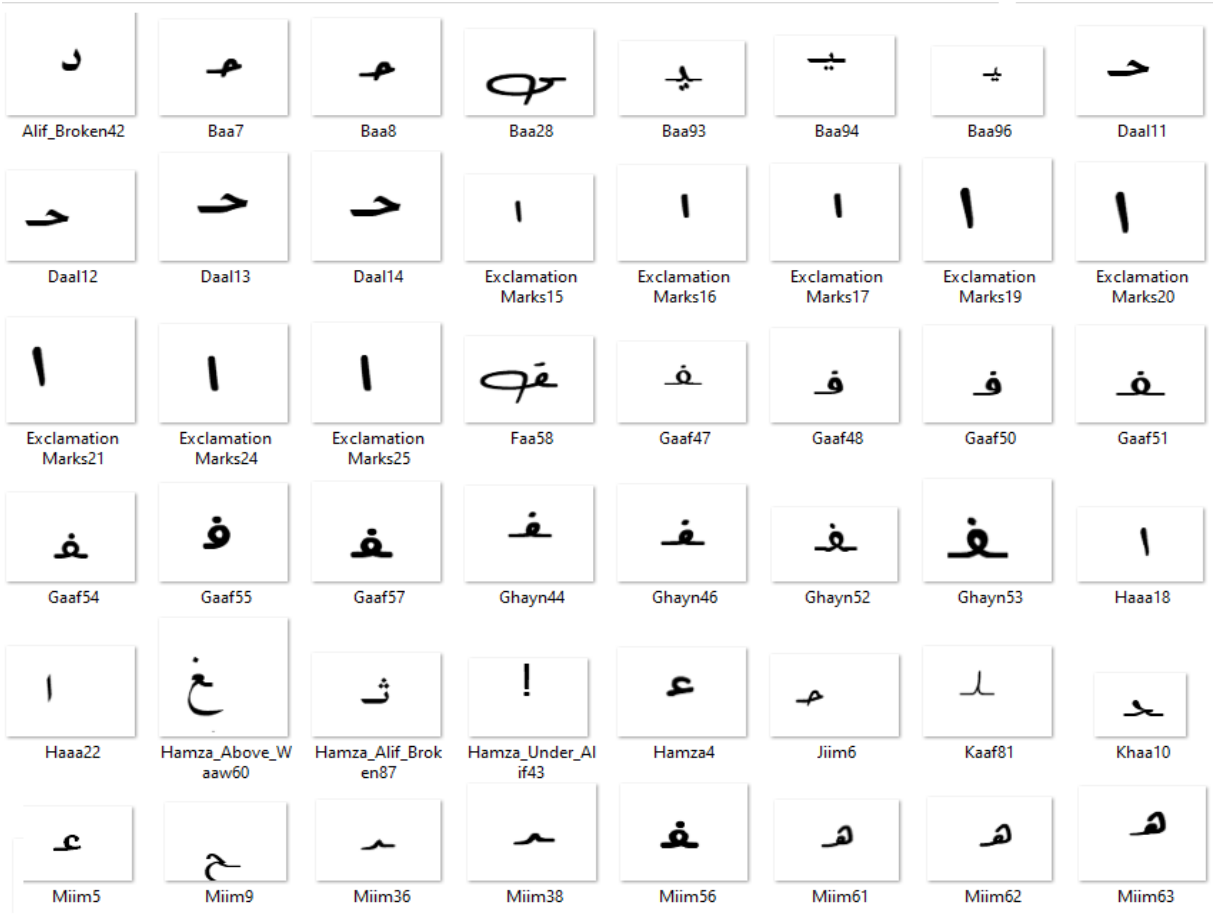



**Figure 5-2:** Samples of Unrecognized characters using CNN mode11 in the stage 2.


**Figure 5-3:** Samples of Unrecognized characters using CNN mode11 in the stage 3.

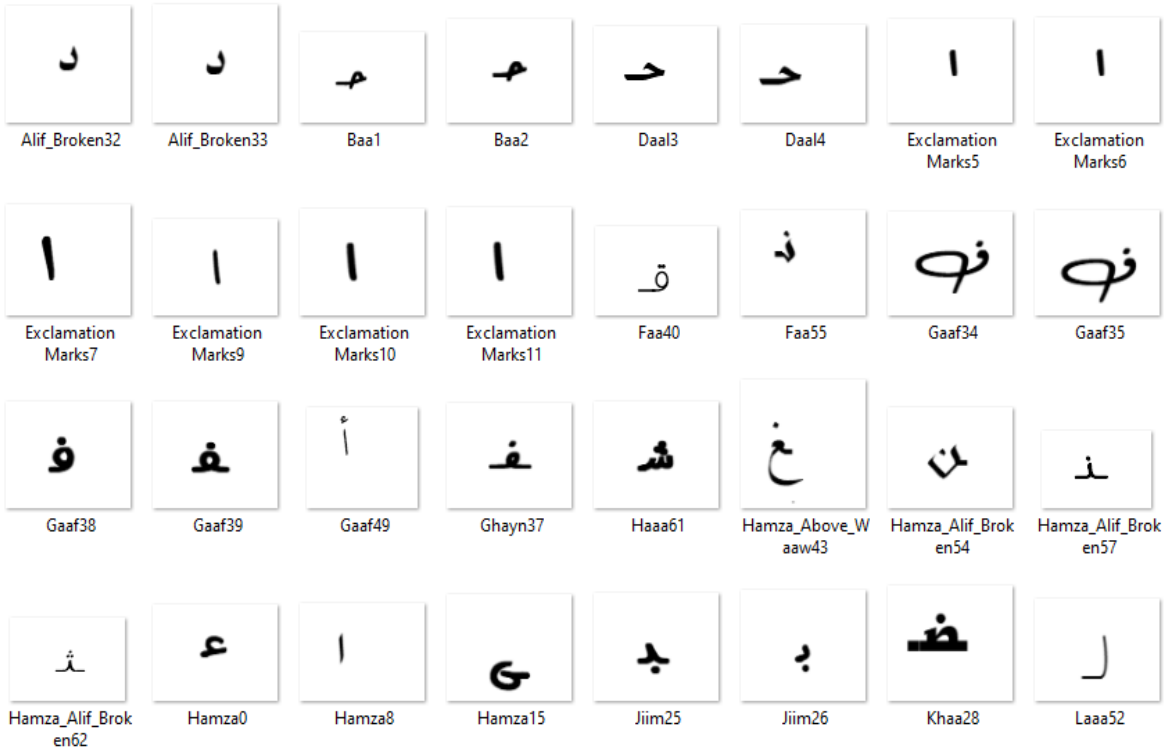



**Figure 5-4:** Unrecognized Character Samples using CNN Model 1 in Stage 4.

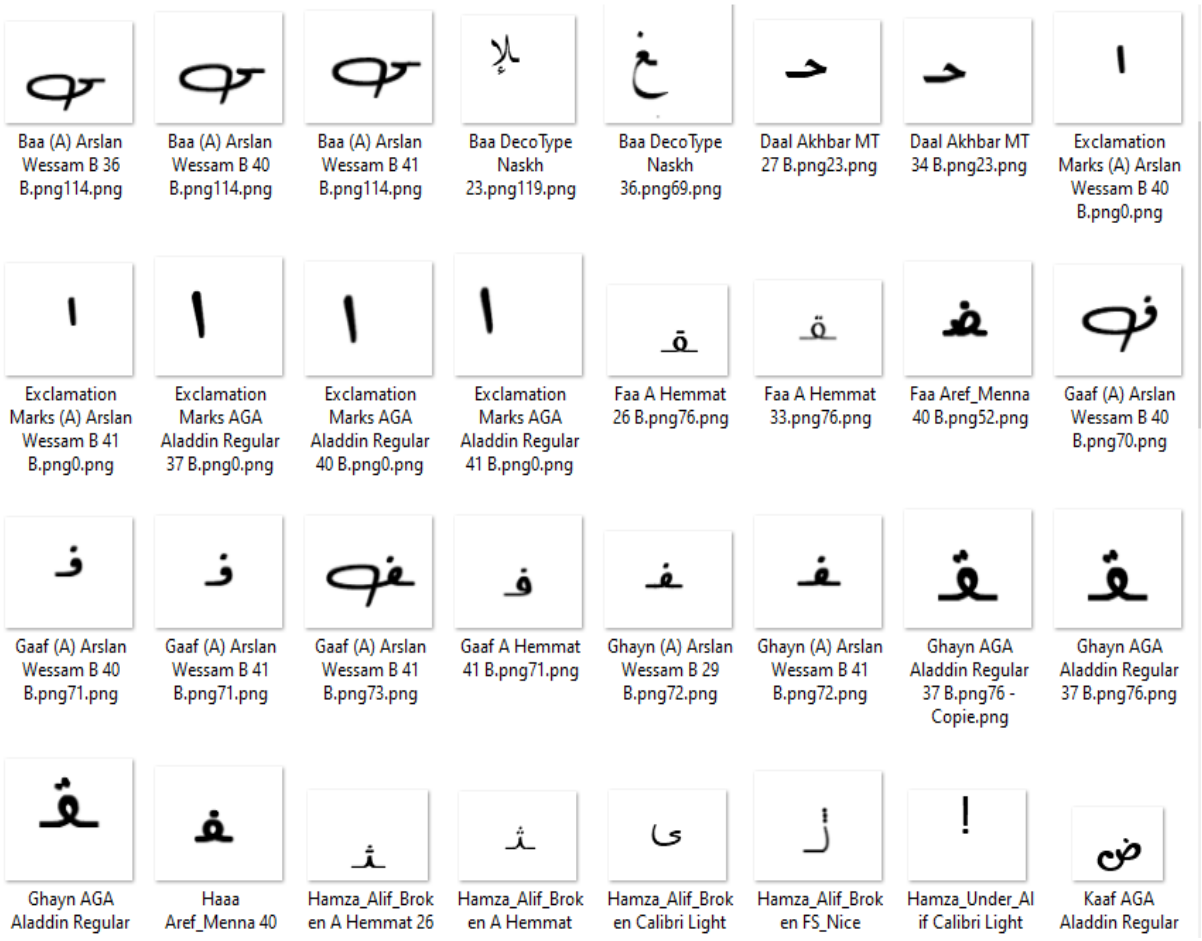



**Figure 5-5:** Unrecognized Character Samples using CNN Model 1 in Stage 5.

Table 5-2 shows the number of samples and the recognition accuracy of each class. The total number of samples is 16620. Samples were collected from different types of fonts (21 fonts).

Table 5-2 summarizes the final results of the recognition of printed Arabic characters of the CNN model 1 for each class, where the first column represents the character (class), the second column represents the number of the testing set for each class, and the third column represents the number of characters that the CNN model 1 could not recognize, the fourth column represents the recognition accuracy achieved by the CNN model 1 from the testing process for each class, the last column represents the loss rate for each class of the CNN model 1, where the total number of testing set is 16620 images.

| Chara-ctrer | Total number of characters | Unrecog-nized character | Accuracy | Loss | Chara-ctrer | Total number of characters | Unrecog-nized character | Accuracy | Loss |
|---|---|---|---|---|---|---|---|---|---|
| ا | 228 | 5 | 97.80% | 2.19% | أ | 228 | 0 | 100% | 0% |
| ب | 456 | 5 | 98.90% | 1.09% | إ | 228 | 1 | 99.56% | 0.43% |
| ت | 456 | 0 | 100% | 0% | ئ | 456 | 0 | 100% | 0% |
| ث | 456 | 4 | 99.12% | 0.43% | ؤ | 228 | 0 | 100% | 0% |
| ج | 456 | 0 | 100% | 0% | لا | 228 | 0 | 100% | 0% |
| ح | 456 | 7 | 98.46% | 1.53% | لأ | 228 | 0 | 100% | 0% |
| خ | 456 | 1 | 99.78% | 0.21% | لإ | 228 | 1 | 99.56% | 0.43% |
| د | 228 | 8 | 96.49% | 3.5% | لحـ | 45 | 0 | 100% | 0% |
| ذ | 228 | 1 | 99.56% | 0.43% | محـ | 45 | 0 | 100% | 0% |
| ر | 228 | 1 | 99.56% | 0.43% | ى | 114 | 7 | 93.85% | 6.14% |
| ز | 228 | 1 | 99.56% | 0.43% | ة | 114 | 0 | 100% | 0% |
| س | 456 | 0 | 100% | 0% | 0 | 217 | 0 | 100% | 0% |
| ش | 456 | 0 | 100% | 0% | 1 | 217 | 0 | 100% | 0% |
| ص | 456 | 0 | 100% | 0% | 2 | 217 | 0 | 100% | 0% |
| ض | 456 | 2 | 99.56% | 0.43% | 3 | 217 | 0 | 100% | 0% |
| ط | 456 | 0 | 100% | 0% | 4 | 217 | 0 | 100% | 0% |
| ظ | 456 | 2 | 99.56% | 0.43% | 5 | 217 | 0 | 100% | 0% |
| ع | 456 | 5 | 98.90% | 1.09% | 6 | 217 | 0 | 100% | 0% |
| غ | 456 | 2 | 99.56% | 0.43% | 7 | 217 | 0 | 100% | 0% |
| ف | 456 | 8 | 98.24% | 1.75% | 8 | 217 | 0 | 100% | 0% |
| ق | 456 | 6 | 98.68% | 1.31% | 9 | 217 | 0 | 100% | 0% |
| ك | 456 | 2 | 99.56% | 0.43% | ! | 114 | 1 | 99.12% | 0.87% |
| ل | 456 | 0 | 100% | 0% | ؟ | 114 | 0 | 100% | 0% |
| م | 456 | 0 | 100% | 0% | < | 114 | 0 | 100% | 0% |
| ن | 456 | 3 | 99.34% | 0.65% | > | 114 | 0 | 100% | 0% |
| ه | 456 | 5 | 98.90% | 1.09% | ( | 114 | 0 | 100% | 0% |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| و | 114 | 1 | 99.12% | 0.87% | ) | 114 | 0 | 100% | 0% |
| ي | 456 | 0 | 100% | 0% | / | 114 | 0 | 100% | 0% |
| ء | 114 | 0 | 100% | 0% | : | 114 | 0 | 100% | 0% |

**Table 5-2:** Recognition accuracy and Loss rate of the CNN Model1 form for each class.

The Table 5-3 shows the types of fonts that do not achieve 100% of recognition rate:

| Fonts Type | Accuracy | Fonts Type | Accuracy |
|---|---|---|---|
| (A) Arslan Wessam B | 98,14% | Calibri Light | 99.36% |
| A Hemmat | 98,92% | DecoType Naskh | 99.53% |
| AGA Aladdin Regular | 99.08% | FS_Nice | 99.21% |
| Akhbar MT | 99.01% | Hayah | 98.15% |
| Andalus | 99.55% | Kamran | 99.79% |
| Arabic Typesetting | 99.86% | Sakkal Majalla | 99.60% |
| Aref_Menna | 99.34% | | |

**Table 5-3:** Types of fonts that do not achieve 100% of recognition rate by CNN Model1.

The model CNN1 was able to recognize some types of fonts with 100% accuracy. However, most other types of fonts could not be recognized with 100% accuracy. The reason for the low recognition accuracy is due to the similar structure of the characters.

## 4. CNN Model 2

It may help to add more hidden layers to see the abstract features of the input image. Increase character recognition accuracy and reduce error. Therefore, a second convolutional neural network model was created by adding a third convolutional layer. so that we can generalize data outside of the model accurately.

Table 5-4 displays the Convolutional Neural Network Model 2 training stages and testing results after each training, where the first column denotes the stage number. The number of eras in each stage is shown by the second column. The third column shows how many images were used to test the model. The fourth column shows how many images the model was able to recognize. The fifth column shows how many images the model did not recognize. The sixth column summarizes the accuracy for each level. The error rate for each stage is shown in the last column.

| | Epochs | DataSet | | | Testing | |
|---|---|---|---|---|---|---|
| | | Testing DataSet | Recognized DataSet | Unrecognized DataSet | Accuracy | Loss |
| Stage 1 | 10 | 16620 | 16420 | 200 | 98.79% | 1.20% |
| Stage 2 | 30 | | 16542 | 78 | 99.53% | 0.47% |
| Stage 3 | 70 | | 16551 | 69 | 99. 58% | 0.41% |
| Stage 4 | 150 | | 16597 | 23 | 99.86% | 0.13% |
| Stage 5 | 250 | | 16597 | 23 | 99.86% | 0.13% |

**Table 5-4:** Training stages and testing results using CNN Model 2.

To improve the model's performance, the number of epochs in the model training was increased. The failure of the model to recognize some letters is due to the lack of model learning due to the similarity of the Arabic characters among themselves.

By following the same training method for the CNN model 1 as for the CNN model 2, we get a test accuracy of 99.89%, which is a good recognition rate.

This model fails to recognize some characters at every stage of training. Samples of these characters are shown in Figures 5-6, 5-7, 5-8, 5-9, and 5-10.

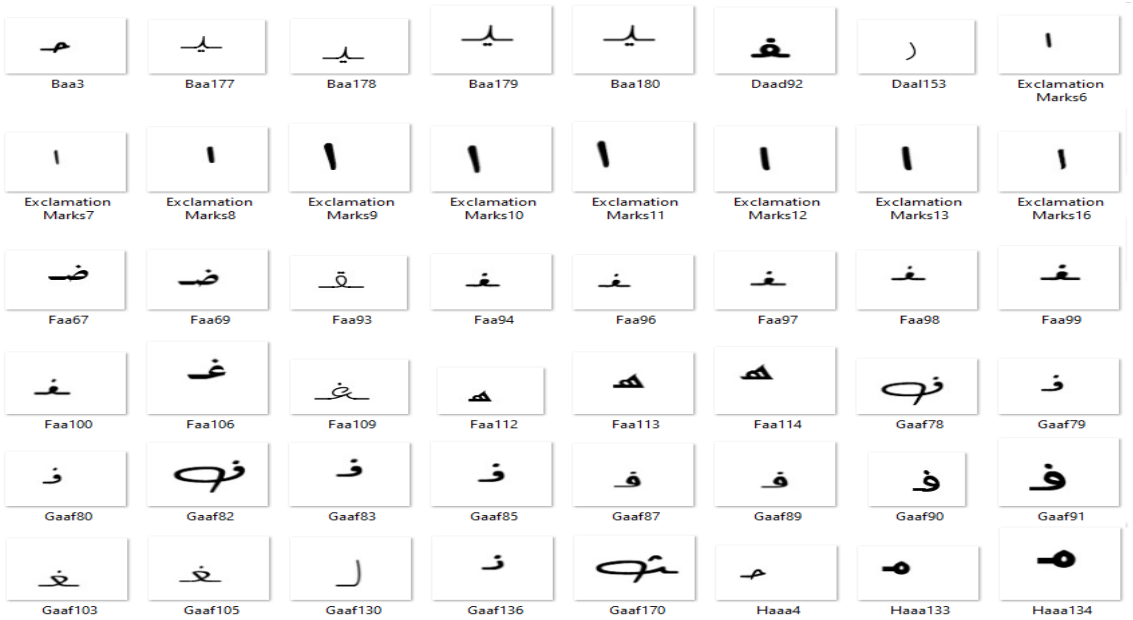



**Figure 5-6:** Unrecognized character samples Model CNNs 2 in the stage 1.

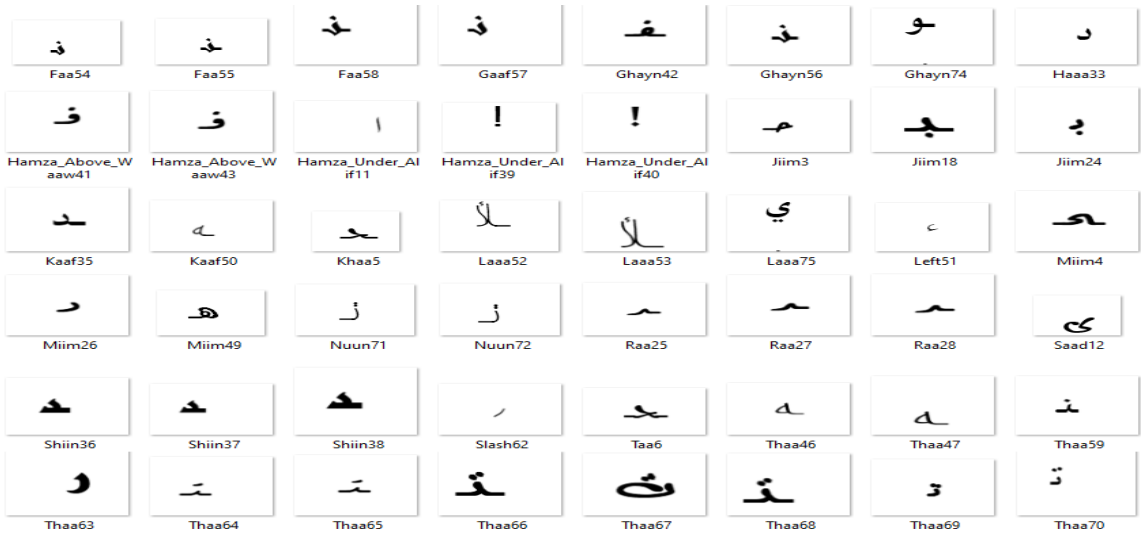



**Figure 5-7:** Unrecognized character samples using CNN model 2 in the stage 2.

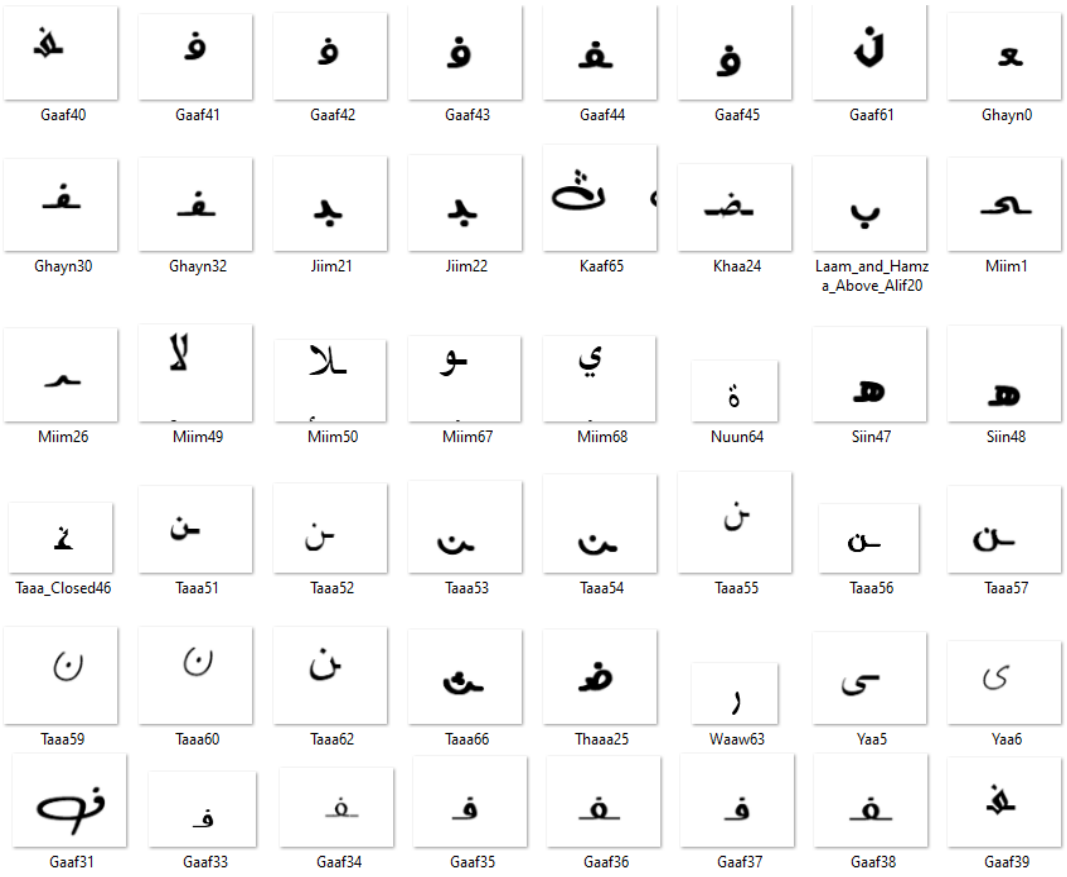



**Figure 5-8:** Unrecognized character samples CNN model 2 in the stage 3.

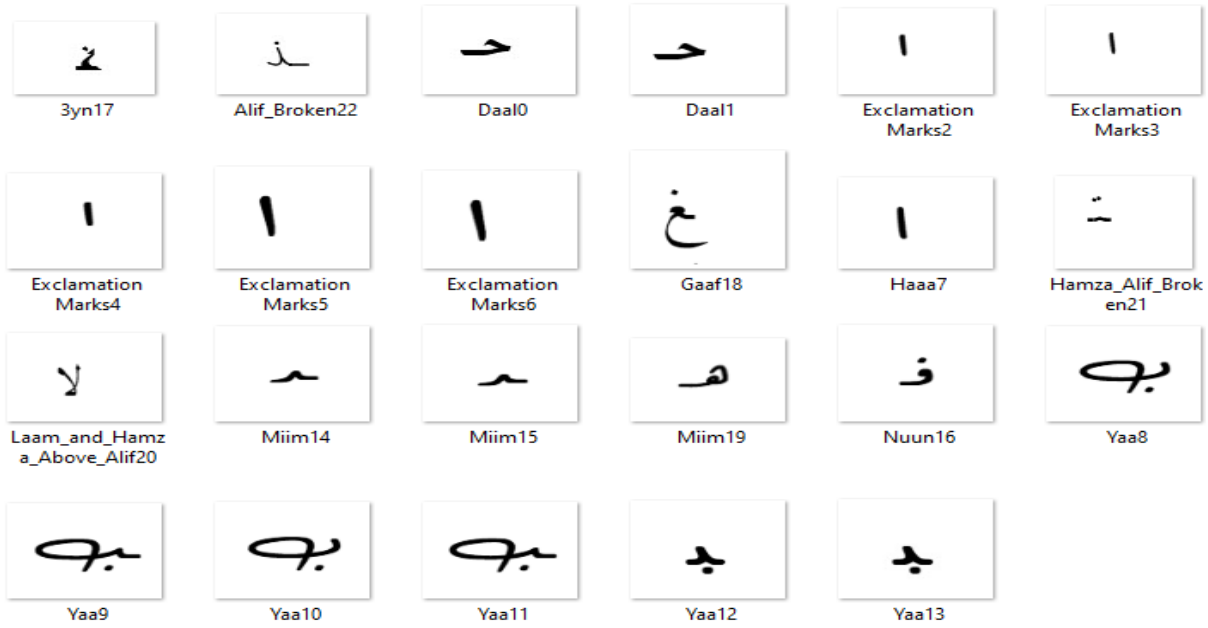



**Figure 5-9:** Unrecognized character samples using CNN model 2 in the stage 4.

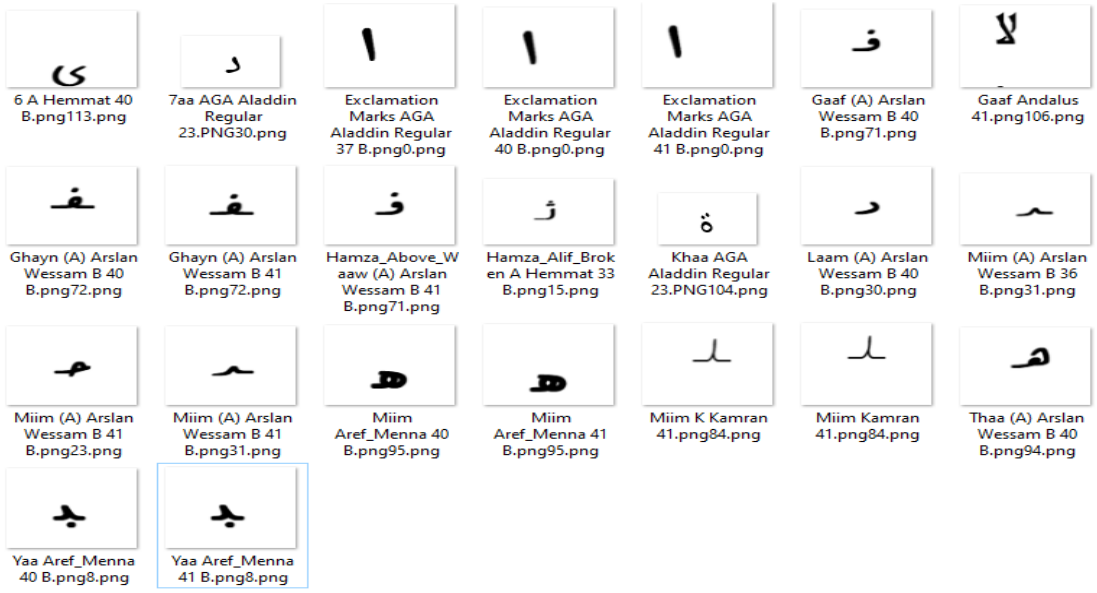



**Figure 5-10:** Unrecognized character samples using CNN model 2 in the stage 5.

Table 5-5 shows the final results in the recognition of printed Arabic characters of the CNN model 2 for each class, where the first column represents the character (class), the second column represents the number of test sets for each class, the third column represents the number of characters that the CNN model 2 could not recognize, the fourth column represents the recognition accuracy achieved by the CNN model 2 from the test process for each class, and the last column represents the recognition accuracy achieved by the CNN model 2 from the test process for each class.

| Chara-ctrer | Total number of charac-ters | Unrecog-nized character | Accuracy | Loss | Cha ra-ctrer | Total number of characters | Unrecog-nized character | Accuracy | Loss |
|---|---|---|---|---|---|---|---|---|---|
| ا | 228 | 3 | 98,68% | 1.31% | أ | 228 | 0 | 100% | 0% |
| ب | 456 | 2 | 99.56% | 0.43% | إ | 228 | 0 | 100% | 0% |
| ت | 456 | 0 | 100% | 0% | ئ | 456 | 0 | 100% | 0% |
| ث | 456 | 1 | 99.78% | 0.21% | ؤ | 228 | 0 | 100% | 0% |
| ج | 456 | 0 | 100% | 0% | لا | 228 | 1 | 99.56% | 0.43% |
| ح | 456 | 1 | 99.78% | 0.21% | لأ | 228 | 0 | 100% | 0% |
| خ | 456 | 0 | 100% | 0% | لإ | 228 | 0 | 100% | 0% |
| د | 228 | 4 | 98.24% | 1.75% | لحـ | 45 | 0 | 100% | 0% |
| ذ | 228 | 0 | 100% | 0% | محـ | 45 | 0 | 100% | 0% |
| ر | 228 | 0 | 100% | 0% | ى | 114 | 1 | 99.12% | 0.87% |
| ز | 228 | 0 | 100% | 0% | ة | 114 | 1 | 99.12% | 0.87% |
| س | 456 | 0 | 100% | 0% | 0 | 217 | 0 | 100% | 0% |
| ش | 456 | 0 | 100% | 0% | 1 | 217 | 0 | 100% | 0% |
| ص | 456 | 0 | 100% | 0% | 2 | 217 | 0 | 100% | 0% |
| ض | 456 | 0 | 100% | 0% | 3 | 217 | 0 | 100% | 0% |
| ط | 456 | 0 | 100% | 0% | 4 | 217 | 0 | 100% | 0% |
| ظ | 456 | 0 | 100% | 0% | 5 | 217 | 0 | 100% | 0% |
| ع | 456 | 0 | 100% | 0% | 6 | 217 | 0 | 100% | 0% |
| غ | 456 | 0 | 100% | 0% | 7 | 217 | 0 | 100% | 0% |
| ف | 456 | 4 | 99.12% | 0.87% | 8 | 217 | 0 | 100% | 0% |
| ق | 456 | 0 | 100% | 0% | 9 | 217 | 0 | 100% | 0% |
| ك | 456 | 0 | 100% | 0% | ! | 114 | 0 | 100% | 0% |
| ل | 456 | 2 | 100% | 0% | ؟ | 114 | 0 | 100% | 0% |
| م | 456 | 0 | 100% | 0% | < | 114 | 0 | 100% | 0% |
| ن | 456 | 0 | 100% | 0% | > | 114 | 0 | 100% | 0% |
| ه | 456 | 3 | 99.34% | 0.65% | ( | 114 | 0 | 100% | 0% |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| و | 114 | 0 | 100% | 0% | ) | 114 | 0 | 100% | 0% |
| ي | 456 | 0 | 100% | 0% | / | 114 | 0 | 100% | 0% |
| ء | 114 | 0 | 100% | 0% | : | 114 | 0 | 100% | 0% |

**Table 5-5:** Recognition accuracy and Loss rate of the CNN Model 2 form for each class.

The Table 5-6 shows the types of fonts that do not achieve 100% of recognition rate

| Fonts Type | Accuracy | Fonts Type | Accuracy |
|---|---|---|---|
| (A) Arslan Wessam B | 99.16% | Aref_Menna | 99.62% |
| A Hemmat | 99,80% | Kamran | 99.79% |
| AGA Aladdin Regular | 99,61% | K Kamran | 99.80% |
| Andalus | 99.85% | | |

**Table 5-6:** Types of fonts that do not achieve 100% of recognition rate by CNN Model 2.

The CNN model 2 was able to achieve a high accuracy in the recognition of printed Arabic characters. This model was able to recognize 14 out of 21 fonts types with 100% accuracy. At an accuracy of no less than 99.16% in 7 other types of fonts. The failure of the model is due to the method of writing unclear letters in these types of fonts.

## 5. CNN Model 1 and CNN Model 2

Both models were designed by Keras. We created a CNN 1 model from two convolutional layers. The CNN2 model is constructed from three convolutional layers.

The task of the network is to generalize. And we get to it through training using appropriate algorithms. Representing the training in finding the optimal set of weights to achieve the correct classification.

The method of training the two models is supervised and of the Bath Training type. That is, the weights are modified after all the examples are entered. In contrast to the second type (Online Training), in which the weights are modified after each example is entered.

In comparison between the CNN1 model and the CNN2 model, we find that the CNN model 2 is more accurate, as the CNN-2 model outperforms the other in training accuracy, verification accuracy, and test accuracy. As shown in the figure 5-11.

And the lowest percentage of training Loss, validation loss and test loss of the CNN model 1, as shown in the figure 5-12.

We relied in our system on the CNN model 2 because it is more accurate.


**Figure 5-11:** Accuracy of CNN Model 1 and CNN Model 2.

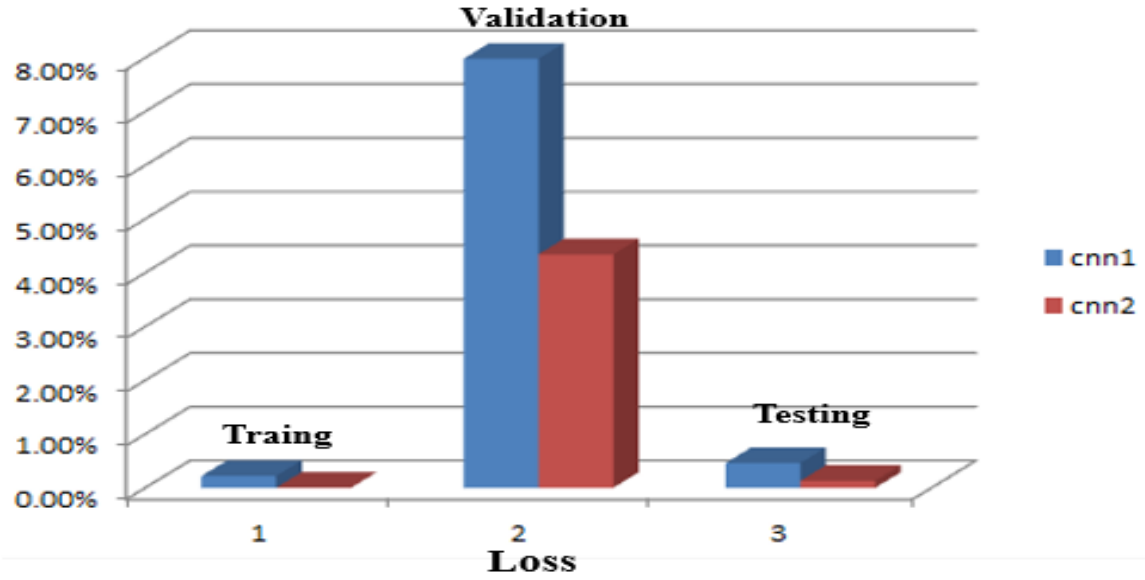



**Figure 5-12:** Loss of CNN Model 1 and CNN Model 2.

## 6. Conclusion

Convolutional Neural Networks (CNNs) are powerful models that achieve high performance in character classification. It managed to achieve high accuracy in Recognition of printed Arabic characters accuracy reached 99.86% in Recognition of printed Arabic characters.

too many epochs may cause model to over-fit the training data. It means that model does not learn the data, it memorizes the data. so, the accuracy of validation data has to be found for each epoch or maybe iteration to investigate whether it over-fits or not.

Increasing the number of convolutional layers of a convolutional neural network model improves network performance to some extent.

# Conclusion and Future Work

Although this research came as a preliminary study in the field of recognition, it has made progress in the right direction. The results were promising and the accuracy rate is high compared to previous research. Therefore, the work we have done is an important step and an effective contribution to the recognition of printed Arabic characters which is different from other languages. As a result, this research can be used as a basis for many future projects.

Undoubtedly, the issue of character recognition is a very important topic because it can provide assistance in various aspects of life, whether it is for those who speak Arabic or for non-Arabs who are interested in the Arabic language. Also, character recognition programs make a significant contribution to exploiting the cultural balance of Arab nations after it was outside the field of exploitation and investment because it was an incomprehensible content that could not be exploited by the computer.

Recognition of printed Arabic characters is an important field due to its different applications. In addition, it is an active field of research that always needs accuracy improvement. Object recognition is based mainly on Machine Learning methods that give high performance especially with the emergence of a new subfield called deep learning where the latter depends mainly on the use of many Convolutional Neural Networks. So, we have developed a system for the recognition of printed Arabic characters using deep learning technique. Our system has succeeded in designing and implementing a CNN that works efficiently. We have used CNN to perform the feature extraction and the classification of the characters.

In this study, convolutional neural networks were used to recognize printed Arabic Characters. The convolutional neural network was trained using Character data set that was compiled using 21 types fonts of Word to process. Characters were divided into 58 classes and a total of 84,162 images were obtained. After the data set was acquired, it was split 80% to 20% as training and testing set. Then, a CNN model with two convolution layers is built to Recognition it 58 classes. The network was trained with 67542 images and tested with 16620 images. The same process was repeated to create a second CNN consisting of three convolutional layers to observe the effect of system depth. on performance recognition. It has

been noticed the best system discrimination result is obtained when the number of convolutional layers is increased.

This study only looked at the recognition of printed Arabic characters; however, it can be broadened by creating a new dataset with examples of handwriting.

Some suggestions for the future work:

- Expand the recognition phase to include different types of Arabic fonts.
- Develop a segmentation algorithm for Arabic ligatures.
- Include punctuation.
- Add special characters like mathematical symbols.

# Bibliography


[1] Wikipédia, https://ar.wikipedia.org/wiki/%D8%A7%D9%84%D9%84%D8%BA%D8%A9_%D8%A7%D9%84%D8%B9%D8%B1%D8%A8%D9%8A%D8%A9 consulté le : 20/04/2022

[2] Khalid Mohammed Musa Yaagoup, Mohamed Elhafiz Mustafa, “Online Arabic Handwriting Characters Recognition using Deep Learning”, Dept of Computer Science, Faculty of Computer Science and Information Technology, Sudan, 2020.

[3]عبدالحفيظ حامد محمد بابكر, " تصميم خوارزمية للتعرف على الكتابة العربية المتصلة بإستخدام [3]الشبكات العصبية الاصطناعية", بحث لنيل درجة الدكتوراة في علوم الحاسوب, جامعة النيلين, كلية الدراسات العليا, 2020.

[4] LABEL YOUR DATA, https://labelyourdata.com/articles/what-is-dataset-in-machine-learning consulté le : 27/04/2022.

[5] Javatpoint, https://www.javatpoint.com/how-to-get-datasets-for-machine-learning consulté le : 27/04/2022.

[6] A. Zoizou, A. Zarghili, and I. Chaker, “A new hybrid method for Arabic multi-font text segmentation, and a reference corpus construction,” J. King Saud Univ. - Comput. Inf. Sci., 2018, DOI: 10.1016/j.jksuci.2018.07.003.

عبدالحفيظ حامد محمد بابكر, " تصميم خوارزمية للتعرف على الكتابة العربية المتصلة بإستخدام [7]الشبكات العصبية الاصطناعية", بحث لنيل درجة الدكتوراة في علوم الحاسوب, جامعة النيلين, كلية الدراسات العليا, 2020.

[8] Tahmi. Hassina, ‘‘developing an efficient CNN model to recognition Handwritten digits’’, M'sila Mohamed Boudiaf University, master, 2020.

[9] Amani Ali Ahmed Ali and M. Suresha, ‘‘A novel features and classifiers fusion technique for Recognition of Arabic handwritten character script,’’, Taiz University, Department of Computer Science, Taiz, Yemen, Kuvempu University, Department of Computer Science and MCA, Shimoga, India, Oct. 2019.

[10] Said Gadri, ‘‘Efficient Arabic Handwritten Character Recognition based on Machine Learning and Deep Learning Approaches,’’ Department of Computer Science, Faculty of Mathematics and Computer Sciences, University of M’sila, M’sila, 2020.

[11] A. Yousaf, M. J. Khan, M. J. Khan, N. Javed, H. Ibrahim, K. Khurshid, and K. Khurshid, ‘‘Size invariant handwritten character recognition using single layer feedforward

backpropagation neural networks,'' in Proc. ational University of Science and Technology, Rawalpindi, Pakistan,2nd Int. Conf. Comput., Math. Eng. Technol. (iCoMET), Jan. 2019, pp. 1–7.

[12] A. Manzoor, E. Shaikh, G. Latif, I. Mohiuddin, and N. Mohammad, ''Automated grading for handwritten answer sheets using convolutional neural networks,'' in Proc. Prince Mohammad bin Fahd University, 2nd Int. Conf. New Trends Comput. Sci. (ICTCS), Oct. 2019, pp. 1–6.

[13] A. I. Abidi and M. Ahmed, ''Performance comparison of ANN and template matching on English character recognition,'' Int. J. Advance Res., Ideas Innov. Technol., sharda university, greater Noida, Uttar Pradesh, vol. 5, no. 4, pp. 367–372, 2019.

[14] R. Ptucha, F. P. Such, S. Pillai, F. Brockler, V. Singh, and P. Hutkowski, ''Intelligent character recognition using fully convolutional neural networks,'' Pattern Recognit., Rochester institute of technology, vol. 88, pp. 604–613, Apr. 2019.

[15] A. Sagheer, S. S. R. Rizvi, K. Adnan, and A. Muhammad, ''Optical character recognition system for nastalique Urdu-like script languages using supervised learning,'', Department of Computer Science and IT The University of Lahore 54000, PakistanInt. J. Pattern Recognit. Artif. Intell., vol. 33, no. 10, Sep. 2019, Art. no. 1953004.

[16] Amir Hussain and Najoua Rahal, "Deep sparse auto-en code features learning for Arabic text recognition", Faculty of Sciences of Tunis, Tunis El Manar Universit,2018.

[17] M. Betke, R. Elanwar, and W. Qin, ''Making scanned Arabic documents machine accessible using an ensemble of SVM classifiers,'' Germany, Int. J. Document Anal. Recognit., vol. 21, nos. 1–2, pp. 59–75, Jun. 2018.

[18] B. Su and S. Lu, ''Accurate recognition of words in scenes without character segmentation using recurrent neural network,'' Pattern Recognit., vol. 63, pp. 397–405, Mar. 2017.

[19] J. Sun, L. Jin, S. Huang, Z. Feng, Z. Yang, ''Robust shared feature learning for script and handwritten/machine-printed identification,'' School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China b Fujitsu R&D Center Co., Ltd., Beijing, China, Pattern Recognit. Lett., vol. 100, pp. 6–13, Dec. 2017.

[20] "التعرف على السماء العربية المكتوبة بخط اليد باستخدام التعلم , ميادة حمد العبيد,سمر فتح الرحمن بابكر آدم العميق. ", بحث مقدم كأحد متطلبات الحصول على درجة البكالوريوس في علوم الحاسوب، جامعة السودان للعلوم والتكنولوجيا، كلية علوم الحاسوب وتقانة المعلومات، قسم علوم الحاسوب 2016.,

[21] Dr. Ahmed Fouad Ali Suez Canal University, Dept. of Computer Science, Faculty of Computers and informatics Member of the Scientific Research Group in Egypt. Presentation slides on Cuckoo search algorithm.

[22] Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). Deep Learning. MIT Press. p. 326.

[23] keras, https://keras.io/about/ consulté le : 02/05/2022.

[24] Investopedia, https://www.investopedia.com/terms/p/paretoprinciple.asp consulté le : 10/05/2022.

[25] ScienceDirect, https://www.sciencedirect.com/topics/computer-science/classification consulté le : 03/05/2022.

## الملخص:

بلغت الكثير من اللغات تقدما كبيرا في مجال التعرف على الحروف منها: اللاتينية، الصينية، اليابانية ... حيث بلغت نسب مرتفعة في التعرف تصل إلى 100% بينما يشهد التعرف على الحروف العربية نسب منخفضة، يعود سببها إلى بعض الخصائص في اللغة العربية التي تعيق وتصعب عملية التعرف. لذا كان العمل المنجز في إطار هذه المذكرة يتعلق بتطوير نظام التعرف على الحروف العربية المطبوعة. في هذا الصدد تم عرض دراسة عن البنية اللغة العربية متبوعا بعرض تقنية الشبكات العصبية الالتفافية التي اثبتت كفاءتها في التعرف السريع والحصول على نتائج أكثر موثوقية. حيث تم اقتراح نموذجين CNN1وCNN2 لقياس دقة التعرف ومراقبة تأثير عمق الطبقات الالتفافية على أداء التعرف. يتكونCNN1 من طبقتين تلافيفيه بحجم (3 X3) أما CNN2 من ثلاث طبقات تلافيفيه. حيث يتم تعديل الاوزان بعد ادخال جميع الأمثلة، على عكس يتم تعديل الاوزان بعد ادخال كل مثال. أظهرت النتائج تميز النموذجCNN2 لتحصل على دقة أفضل وخسارة اقل.



**الكلمات المفتاحية:** التعرف على الحروف العربية المطبوعة؛ الشبكات العصبية الالتفافية؛ المعالجة؛ استخراج السمات؛ التصنيف؛



## Summary:

Many languages have made great progress in the field of character recognition, including: Latin, Chinese, Japanese ... where high rates of recognition reach 100%, while the recognition of Arabic letters is witnessing low rates, due to some characteristics in the Arabic language that hinder It is difficult to identify. Therefore, the work carried out in the framework of this memorandum was related to the development of the system for the recognition of printed Arabic characters. In this regard, a study on the structure of the Arabic language was presented, followed by a presentation of the convolutional neural network technique, which proved its efficiency in rapid recognition and obtaining more reliable results. Where two models CNN1 and CNN2 are proposed to measure the recognition accuracy and monitor the effect of the depth of convolution layers on the recognition performance. CNN1 consists of two convolutional layers (3x3) and CNN2 consists of three convolutional layers. Where the weights are adjusted after entering all the examples (Bath training), in contrast to the weights are adjusted after each example is entered (online training). The results showed that the CNN2 model was distinguished to get better accuracy and loss Less.

**Keywords:** recognition of printed Arabic characters; neural networks convolutional; processing; feature extraction; classification;

## Résumé :

De nombreuses langues ont fait de grands progrès dans le domaine de la reconnaissance des caractères, notamment : le latin, le chinois, le japonais... où les taux de reconnaissance élevés atteignent 100 %, tandis que la reconnaissance des lettres arabes connaît des taux faibles, en raison de certaines caractéristiques de l'arabe langue qui entrave Il est difficile à identifier. Ainsi, les travaux menés dans le cadre de ce mémorandum ont porté sur le développement du système de reconnaissance des caractères arabes imprimés. À cet égard, une étude sur la structure de la langue arabe a été présentée, suivie d'une présentation de la technique des réseaux de neurones convolutifs, qui a prouvé son efficacité dans la reconnaissance rapide et l'obtention de résultats plus fiables. Où deux modèles CNN1 et CNN2 sont proposés pour mesurer la précision de la reconnaissance et surveiller l'effet de la profondeur des couches de convolution sur les performances de reconnaissance. CNN1 se compose de deux couches convolutives (3x3) et CNN2 se compose de trois couches convolutives. Là où les poids sont ajustés après la saisie de tous les exemples (formation au bain), contrairement aux poids sont ajustés après la saisie de chaque exemple (formation en ligne). Les résultats ont montré que le modèle CNN2 se distinguait pour obtenir une meilleure précision et moins de perte.

**Mots clés :** reconnaissance des caractères arabes imprimés ; réseaux de neurones convolutionnels ; En traitement; extraction de caractéristiques; classification;