People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University Larbi Tébessi - Tébessa

Faculty of Exact Sciences and Natural and Life Sciences

Department: Mathematics and Computer Science

Final thesis

For obtaining of the MASTER diploma

Domain: Mathematics and Computer Science

Field: Computer Science

Specialty: Systems and Multimedia

Handwritten digit recognition

Using encryption methods

Presented by:

Hamla Djalal eddine

In front of the jury:

Dr. Bennour. A	MCA Larbi Tebessi University	President
Dr. Gattal. A	MCA Larbi Tebessi University	Thesis Supervisor
Mr. Zebdi. A	MAA Larbi Tebessi University	assistant supervisor
Dr. Merzoug. S	MCB Larbi Tebessi University	Examiner

College year:

2022/2021

Thank

First of all, I would like to express my gratitude to the editors of this dissertation, Pr. Gattal. A, and Mr. Zebdi. A, for their patience, their availability and above all their sound advice. And especially their understanding which allowed us to move forward and do this work so well during these months.

I would also like to thank the members of the jury:

- Dr: Bennour. A
- Dr: Merzug. S

I would also like to thank all the students with whom I have been happy to study during these years of university.

I thank all the teachers that I have been able to meet and benefit from.

I would like to thank my colleagues in the Mathematics and Computer Science department and all those who have helped me directly or indirectly in the realization and accomplishment of this work.

Dedication

I dedicate this modest work to: To my parents. No tribute could match the love They keep showering on me. May God grant them good health and long life.

To the one I love very much and who has supported me throughout this project: my brothers and sisters, To all my family, and my friends.

And of course my supervisor Pr. Gattal. HAS.

And to all those who have contributed from near or far to make this project possible, I say thank you.

Table of Contents

List of Figures

List of Tables

Abstract

Over the past years, deep learning, a field of study of multi-layer technical neural networks, has had a strong impact on many areas of technical intelligence, including handwriting recognition. Handwriting recognition is a major research problem in the fields of image analysis and pattern recognition. Even today, digit recognition plays an important role in many areas, such as authenticating bank checks, exchanging remote computer files, and recognizing student notes. In this thesis, we present several contributions from the fields of deep learning and handwritten digit recognition, where we used encryption methods to encrypt images in order to improve the performance of recognizing handwritten digits. Each encryption method was studied separately. After that, several proposed protocols were used to monitor the accuracy of handwritten digit recognition.

In this study, we used Convolutional Neural Networks CNN to train our model, where we conducted an empirical study on the CVL dataset and achieved great results and high accuracy rates that were compared to the state of the art in this subject.

ملخص

على مدى السنوات الماضية ، كان للتعلم العميق ، وهو مجال دراسة الشبكات العصبية التقنية متعددة الطبقات ، تأثير قوي على العديد من مجالات الذكاء التقني ، بما في ذلك التعرف على خط اليد. يعد التعرف على خط اليد مشكلة بحث رئيسية في مجالات تحليل الصور والتعرف على الأنماط. حتى اليوم ، يلعب التعرف على الأرقام دورًا مهمًا في العديد من المجالات ، مثل مصادقة الشيكات المصرفية وتبادل ملفات الكمبيوتر عن بُعد والتعرف على ملاحظات الطلاب. نقدم في هذه الاطروحة عدة مساهمات من مجالات التعلم العميق والتعرف على الأرقام المكتوبة بخط اليد ، حيث استخدمنا طرق التشفير لتشفير الصور من أجل تحسين أداء التعرف على الأرقام المكتوبة بخط اليد. تمت دراسة كل طريقة تشفير على حدة. بعد ذلك ، تم استخدام العديد من البروتوكولات المقترحة لمراقبة دقة التعرف على الأرقام المكتوبة بخط اليد.

في هذه الدراسة ، استخدمنا الشبكات العصبية التلافيفية CNN لتدريب نموذجنا ، حيث أجرينا دراسة تجريبية على مجموعة بيانات CVL وحققنا نتائج رائعة ومعدلات دقة عالية تمت مقارنتها بأحدث التقنيات في هذا الموضوع .

Résumé

Au cours des dernières années, l'apprentissage en profondeur, un domaine d'étude des réseaux de neurones techniques multicouches, a eu un fort impact sur de nombreux domaines de l'intelligence technique, y compris la reconnaissance de l'écriture manuscrite. La reconnaissance de l'écriture manuscrite est un problème de recherche majeur dans les domaines de l'analyse d'images et de la reconnaissance de formes. Aujourd'hui encore, la reconnaissance des chiffres joue un rôle important dans de nombreux domaines, tels que l'authentification des chèques bancaires, l'échange de fichiers informatiques à distance et la reconnaissance des notes des étudiants. Dans cette thèse, nous présentons plusieurs contributions issues des domaines de l'apprentissage profond et de la reconnaissance des chiffres manuscrits, où nous avons utilisé des méthodes de chiffrement pour chiffrer des images afin d'améliorer les performances de reconnaissance des nombres manuscrits. Chaque méthode de cryptage a été étudiée séparément. Après cela, plusieurs protocoles proposés ont été utilisés pour surveiller la précision de la reconnaissance des chiffres manuscrits.

Dans cette étude, nous avons utilisé les réseaux de neurones convolutifs CNN pour former notre modèle, où nous avons mené une étude empirique sur l'ensemble de données CVL et obtenu d'excellents résultats et des taux de précision élevés qui ont été comparés à l'état de l'art dans ce domaine.

General Introduction

In the field of artificial intelligence, deep learning has had a strong impact in many areas. Among these areas, handwritten digit recognition covers a very large area, which is one of the most difficult problems. In fact, because it is used in the field of document analysis and recognition, it

In general, two types of handwriting recognition systems are known: offline recognition and online recognition. The first works on images scanned using a scanner or camera. The second takes data directly from an electronic pen or digitizing tablet and transforms it into digitized text.

In this thesis, we use encrypted images to improve the performance of handwritten digit recognition. Another purpose is to perform computations on the encrypted images, preserving the features of the original image. The relationship between HDR systems and cryptosystems is an inverse relationship; the more performance of an HDR system by using only encrypted images provides a bad cryptosystem. By improving the quantity and diversity of training data, we proposed a new data augmentation method by using different encrypted digit images. This process allows us to increase the size of the training data using images that already exist in the training set.

The work will be divided into 3 main chapters. In the first chapter, we will present approximately the most relevant methods and techniques of deep learning for the work described, as well as the dataset used and the deep learning technology used in this thesis.

In the second chapter, we will present the methods of image encryption and their application to our dataset.

In the third chapter, we will talk about the presentation of our model, the operations performed, and the results obtained.

In conclusion, we recall the objective of the work and the experimental results obtained.

Chapter 1

Handwritten Digit Recognition Using Deep Learning

1. Introduction

Handwritten recognition has been a classic topic that has caught the interest of researchers for several decades and continues to be a relatively open field of research due to its wide range of practical applications. Advances in this field are now evident in a variety of applications, including the automatic reading of bank checks, postal addresses, and form addresses. The Handwritten Digit Recognition (HDR) system is a prominent research issue in the realm of pattern recognition.

In this chapter we present the techniques used in this thesis and also how the HDR system works.

2. Handwritten Digit Recognition

The first step in a handwritten digit recognition system that makes it possible to recognize any number in any format is to convert the written numbers into numeric values suitable for the processing system with the least possible degradation.

There are two types of handwriting recognition systems:

1. On-line handwriting recognition.

2. Off-line handwriting recognition.

Online recognition entails real-time conversion of a user's handwriting on a tablet or smartphone and is the more difficult of the two.

Offline recognition is much easier because it requires machine translation of scanned images or images into computer-readable text [1].

3. State of the art

In this section, we will present some work in the same field to recognize handwritten digits, where we chose:

1. Isolated Handwritten Digit Recognition Using oBIFs and Background Features (2016) [2]:

This study presents how a combination of oriented basic image features (oBIFs) with background concavity features can be used effectively to improve the performance of handwritten digit recognition systems. Using a one-for-all support vector machine (SVM) while the pilot study is being conducted on the CVL database, a series of evaluations using different configurations and feature sets achieved high recognition rates with a score of 95.21.

2. Handwritten Digit String Recognition Using Convolutional Neural Network (2018) [3]:

This study presents a new CNN-only architecture applied to handwritten digit strings, conducted on the CVL and ORAND-CAR datasets. The proposed study obtained 42.69% for the "CVL" dataset and for the "ORAND-CAR-A" and "ORAND-CAR-B" data sets, 92.2% and 94.02%, respectively.

3. Handwritten Digit Recognition Using Image Encryption (2020) [4]:

In this study, work was done on a system for recognizing handwritten digits in an encrypted manner, where machine learning and support vector machine (SVM) were used in this study. Work was done on the CVL data set, and good results were achieved. 85.99% for the unencrypted data set, 61.15% for the RSA encrypted dataset, 85.99% for the Arnold Map encrypted dataset, and 85.92% for the Henon encoded data set.

4. Handwritten Digit Recognition using Machine and Deep Learning Algorithms [5]:

In this paper, we have performed handwritten digit recognition with the help of MNIST datasets using Support Vector Machines (SVM), Multi-Layer Perceptron (MLP) and Convolution Neural Network (CNN) models. Our main objective is to compare the accuracy of the models stated above along with their execution time to get the best possible model for digit recognition.

5. Développement d'un système de reconnaissance de chiffres manuscrits [6]:

We conducted a group of tests for a set of machine learning and deep learning algorithms on a Mnist database. We noticed that the CNN algorithm is efficient in the recognition process, achieving 96 % accuracy and the results obtained have been very convincing and satisfactory. And so we opted for the Convolutional Neural Network method to decide the membership class for each digit. The proposed approach has been tested on two cases of the models and all the experiments carried out have shown very encouraging results. For the first Model we obtained 95.81 % and for the second Model we obtained 93.94 %.

6. Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) [7]:

His objective is to achieve comparable accuracy by using a pure CNN architecture without ensemble architecture, as ensemble architectures introduce increased computational cost and high testing complexity. Thus, CNN architecture is proposed in order to achieve accuracy even better than that of ensemble architectures, along with reduced operational complexity and cost. Moreover, we also present an appropriate combination of learning parameters in designing a CNN that leads us to reach a new absolute record in classifying MNIST handwritten digits. We carried out extensive experiments and achieved a recognition accuracy of 99.87% for a MNIST dataset.

	dataset	approach	Recall/ accuracy
State.1 [2]	CVL	oBIFs + SVM	95.21%
State.2 [3]	CVL	CNN	42.69%
	ORAND-CAR-A		92.2%
	ORAND-CAR-B		94.02%
State.3 [4]	CVL	SVM	85.99%
State.4 [5]	MNIST	SVM	99.98%
		MLP	99.92%
		CNN	99.53%
State.5 [6]	MNIST	CNN	96.00%
state.6 [7]	MNIST	CNN	99.87%

Table 1.1: state of the art Comparison

4. Deep learning

Deep learning is a type of artificial intelligence derived from machine learning where the machine is able to learn on its own. Deep Learning is based on a network of artificial neurons inspired by the human brain. This network is made up of tens or even hundreds of "layers" of neurons, each receiving and interpreting information from the previous layer. Deep Learning has been applied to areas such as image recognition, natural language processing, machine translation, and many others [8].

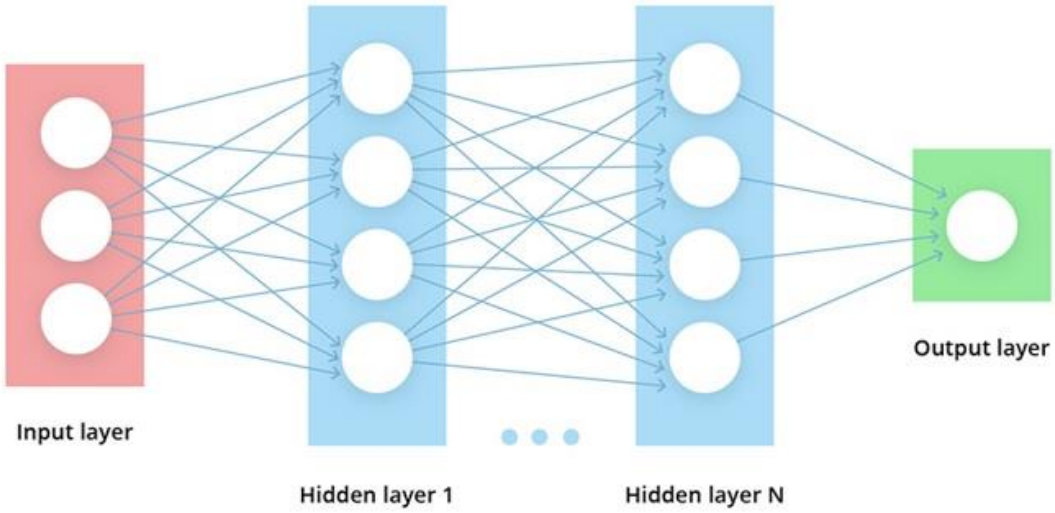

Figure 1.1: Deep learning architecture [9]

5. Convolution Neural Network

Convolutional neural networks (CNNs) are feed-forward artificial neural networks that are used in deep learning. is a type of multilayer preceptor that is supervised. Convolutional neural networks (CNNs) are a type of deep model inspired by the way the human brain processes information. Each neuron has a receptive field capturing data from a certain local neighborhood in visual space. They're made to detect multi-dimensional data that's resistant to distortion and shift scaling [7].

5.1. CNN model

The CNN architecture is made up of one input layer and multiple types of hidden layers as well as one output layer. The first kind of hidden layer is responsible for convolution, and the other one is responsible for local averaging, sub sampling, and resolution reduction. The third hidden layer acts as a traditional multi-layer perceptron classifier [10].

5.2. Convolution layer

CNN's core is the convolution layer. It performs a convolution operation on an image represented as a matrix of pixels, with a learnable kernel (filter) doing a calculation and producing a result that is often smaller, resulting in a reduction in the number of features. [8].

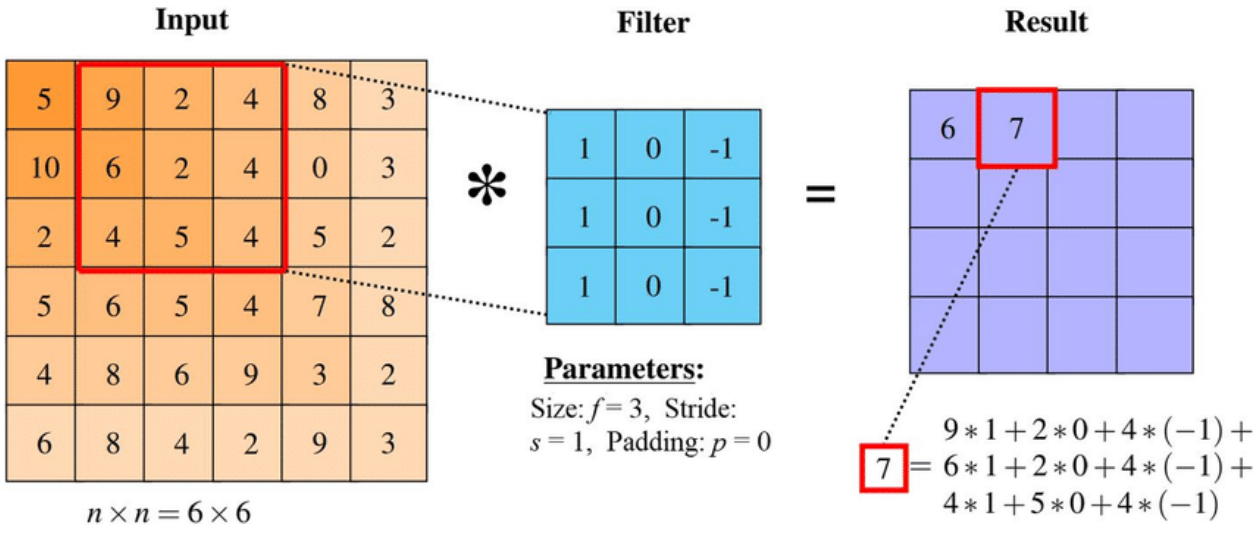

Figure 1.2: Convolution layer example [11]

5.3.1. Activation functions

The activation function in a neural network is responsible for converting the node's summed weighted input into the node's activation or output for that input. The following are some of the most prevalent forms of activation functions [12]:

1. Sigmoid function

2. Tanh function

3. Leaky ReLU function

4. ReLU function

5. Maxout function

6. ELU function

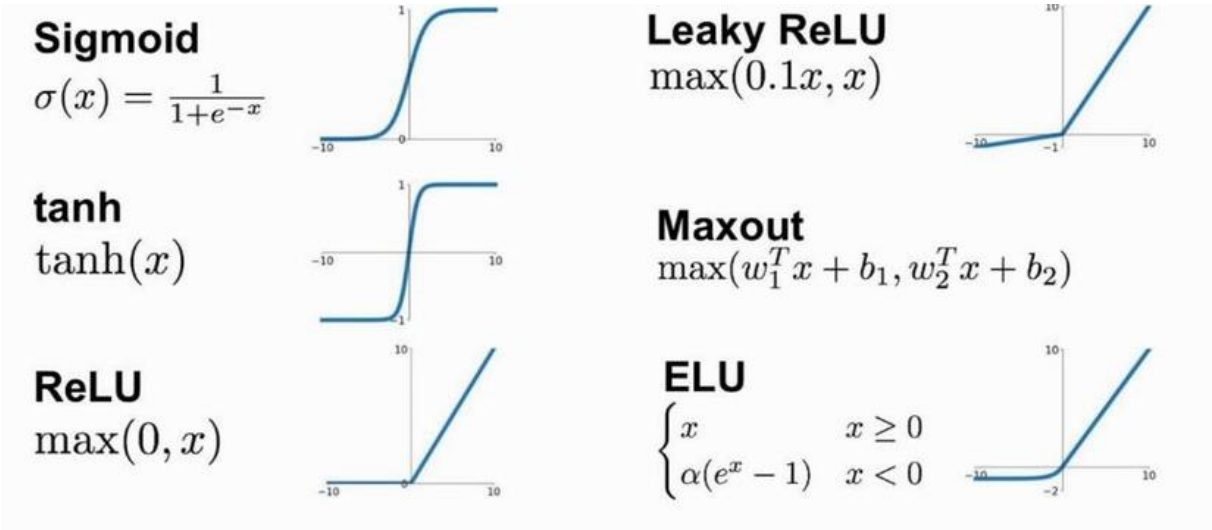

Figure 1.3: Activation functions example [13]

5.3.1.1. RELU Layer

In deep learning models, the Rectified Linear Unit is the most widely employed activation function. If the function receives any negative input, it returns 0, but if it receives any positive value x, it returns that value. As a result, it can be written as $R(z) = \max(0, z)$

Graphically it looks like this: $y = Activation(\sum(weigth \times x) + bias)$ [14]

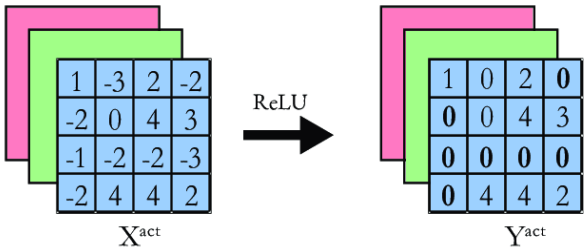

Figure 1.4: Relu Layer example [15]

5.3.2. Strides

the stride is the number of pixels that the analysis window travels on Each iteration. Each kernel is displaced by 2 pixels from its predecessor with a stride of 2 [16].

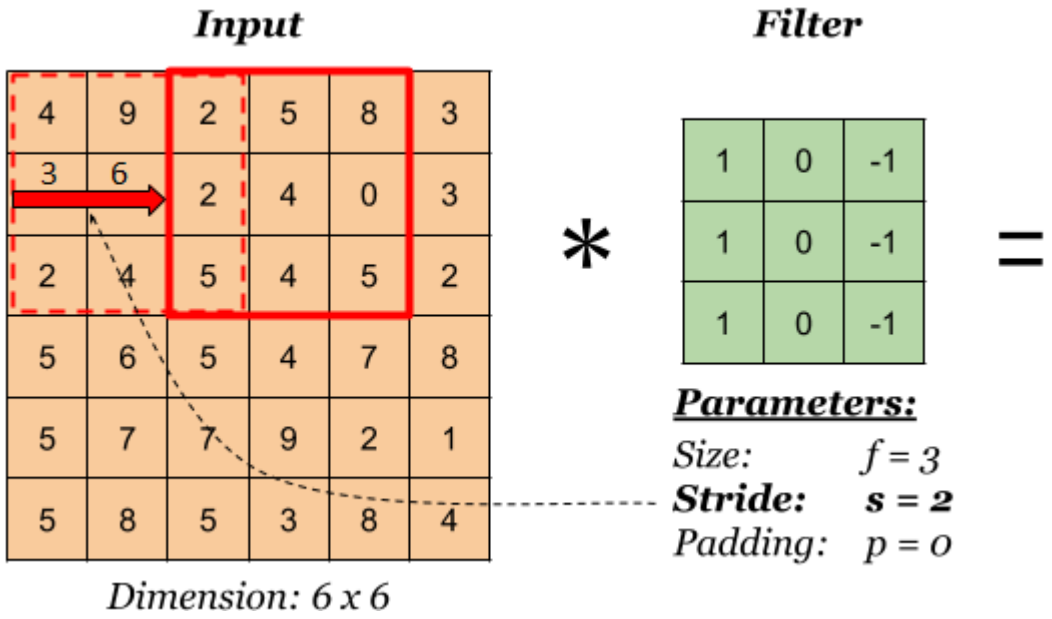

Figure 1.5: Strides example [16]

5.3.3. Padding

Padding is the insertion of (mostly) 0-valued pixels to an image's borders. Because the border pixels would normally only participate in a single receptive field instance, this ensures that they are not devalued (lost) in the output. Padding is usually applied with one less dimension than the kernel dimension [16].

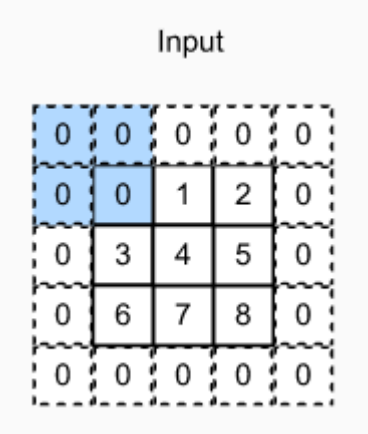

Figure 1.6: padding example [16]

5.4.Pooling layer

A down sampling operation will be performed by the pooling layer along the spatial dimensions (width and height). Pooling is a basic procedure that involves replacing a square of pixels with a single value (typically 2*2 or 3*3). There are three forms of pooling to consider [17]:

1. Average pooling: calculate the average value of the pixels in the chosen square.

2. Max pooling: take the pixel with the highest value in the selected square.

3. Average pooling: takes the average value of the pixels in the selected square.

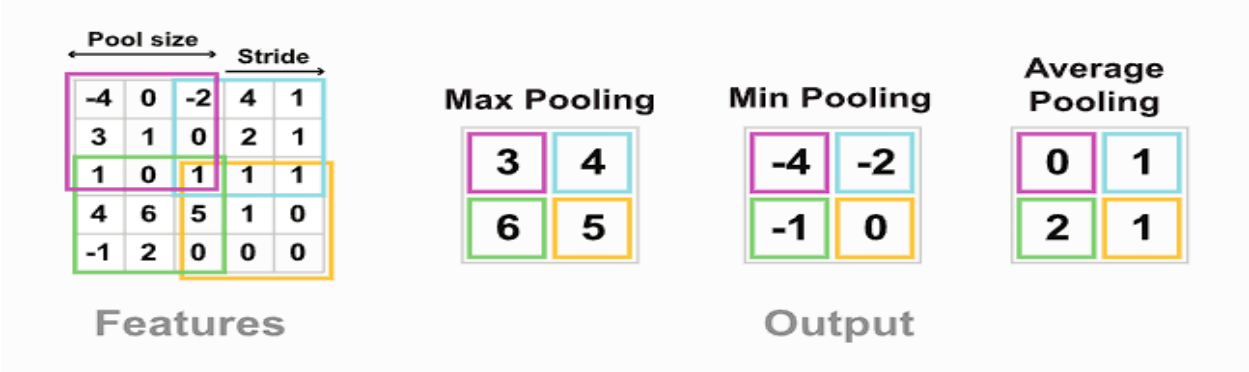

Figure 1.7: pooling layer example [18]

5.4.Dropout

Dropout is a regularization approach that prevents complex co-adaptations on training data, thereby decreasing overfitting in artificial neural networks. It's a quick and easy technique to use on average models [16].

5.5.Fully connected layer

Fully-connected layer (FC) will compute the class scores, resulting in volume of size [1×1×N], Where each of the N numbers correspond to a class score, such as among the N categories.

The flattened matrix goes through a fully connected layer to classify the images [16].

5.6.Classification Layer

The classification layer is the last layer in the CNN architecture. It is a fully connected feed-forward network, mainly used as a classifier. The neurons in the fully connected layers are

connected to all the neurons of the previous layer. This layer calculates predicted classes by identifying the input image, which is done by combining all the features learned by previous layers. The number of output classes depends on the number of classes present in the target dataset. In the present work, the classification layer uses the "softmax" activation function for classifying the generated features of the input image received from the previous layer into various classes based on the training data [19].

$$softmax(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

5.7.RMSProp optimizer

Root Mean Squared Propagation, or RMSProp for short, is an optimization approach that extends the gradient descent process. It's a technique in which the learning rate is customized for each parameter. The idea is to divide a weight's learning rate by a running average of recent gradient magnitudes for that weight. RMSProp has demonstrated high learning rate adaptation in a variety of situations [16].

5.9.Early stopping

Early stopping is a type of regularization that is used to avoid overfitting when using an iterative method like gradient descent to train a learner. With each repetition, such strategies update the learner to make it better fit the training data [20].

5.10. ReduceLROnPlateau

ReduceLROnPlateau is a scheduling strategy that keeps track of a quantity and slows down the learning rate when it stops improving. The quantity's improvement is determined by whether it grows or drops by a specific amount. The threshold is set at this amount [21].

6. CVL Dataset

The CVL Single Digit dataset is part of the CVL Handwritten Digit database (CVL HDdb), which contains samples from 303 writers and was collected mostly among students at Vienna University of Technology and an Austrian secondary school.

There are 10 classifications in the entire single-digit CVL data set (0–9). The train set consisted of 7000 numbers (700 numbers per class) from 67 writers. A different group of 60 writers was used to create an equal-sized verification group. The validation set can be used to estimate and validate parameters, but not for supervised training. There were 2,178 numbers per class in the testing set, resulting in 21,780 assessment samples from the remaining 176 writers [2].

7. Conclusion

In this chapter, we have introduced a general introduction to the handwritten digit recognition system. We also explained the various deep learning techniques and the dataset used. In the second chapter, we will cover the main methods of image encryption.

Chapter 2

Image Encryption

1. Introduction

Cryptography is the science of converting uncoded data into coded or encrypted data. It is an important field that handles many algorithms that can be used to secure data. In this chapter, we are going to give a big picture of image processing and explain how to apply the security mechanisms to images.

2. Cryptography

Cryptography is a science that allows you to change information that is "in clear" into coded or encrypted information, i.e. not comprehensible, and then restore the original information from the coded information (information in clear) [22].

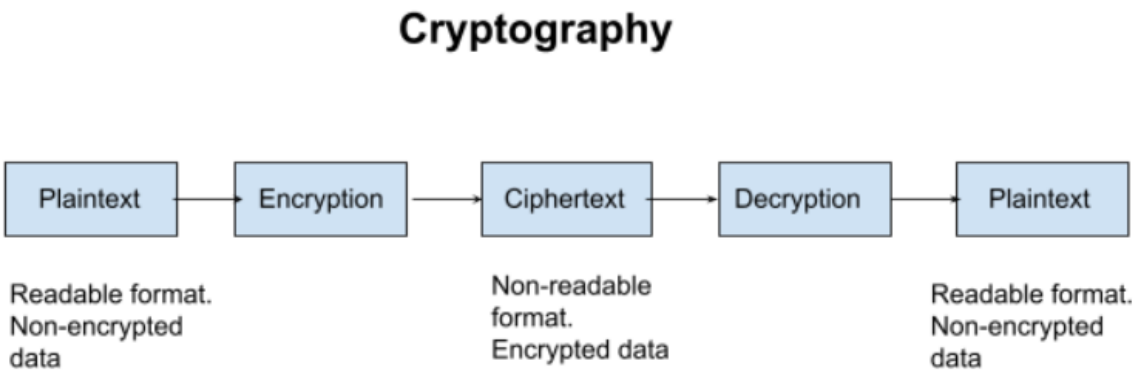

Figure 2.1: cryptography [23]

2.1.Symmetric cryptography

Symmetric cryptography is a type of cryptography that employs the same cryptographic keys for plaintext encryption and ciphertext decoding (AES, DES, RC4, 3DES). The keys might be the same or there could be a simple change between them.

In reality, the keys constitute a shared secret between two or more parties that may be utilized to keep a private data link open [24].

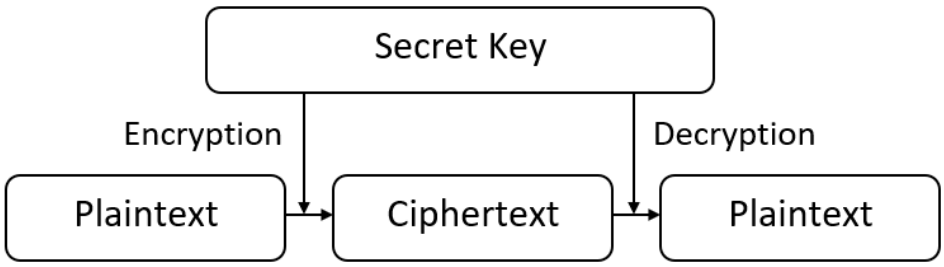

Figure 2.2: symmetric cryptography [25]

2.2. Asymmetric cryptography

Asymmetric cryptography (RSA, ECC) is a cryptographic technique that employs two types of keys: public keys that may be shared widely and private keys that are only known by the owner. To build one-way functions, cryptographic procedures based on mathematical problems are used to generate such keys. The sole requirement for effective security is that the private key be kept secret; the public key can be openly distributed without compromising security [26].

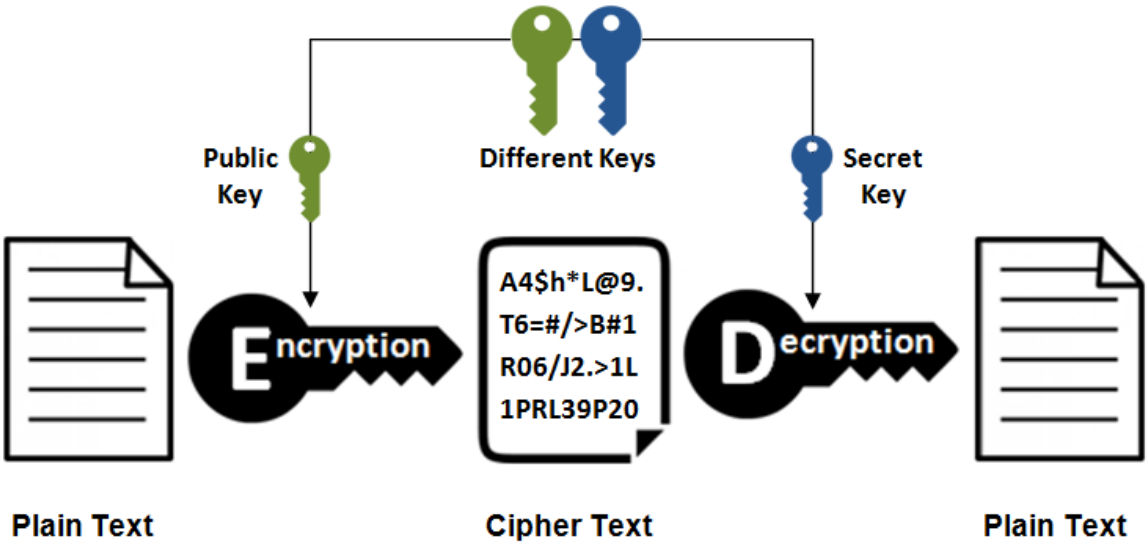

Figure 2.3: asymmetric cryptography [27]

2.3. Chaotic cryptography

Chaos theory is a branch of mathematics that analyzes the behavior and conditions of dynamic systems that are very sensitive to their beginning conditions. A little change in the initial conditions causes the output to diverge substantially. In other words, if chaotic cryptography is utilized for image scrambling, one may not be able to descramble the picture from the input scrambled image without knowing the beginning condition (here, the scrambling key). There are several chaotic maps that may be used to create chaotic sequences [28].

3. The used methods

In this section, we describe the three encryption methods selected from various techniques in cryptography. We used AES (symmetric encryption method), RSA (asymmetric encryption method), and Arnold and Henon as the cipher methods from Chaos Maps.

3.1. AES encryption

The Advanced Encryption Standard (AES) algorithm is known for its security as well as its speed. NIST recommends it as a new encryption standard to replace DES. It encrypts 128-bit data blocks in 10, 12, or 14 rounds, depending on the key size [29].

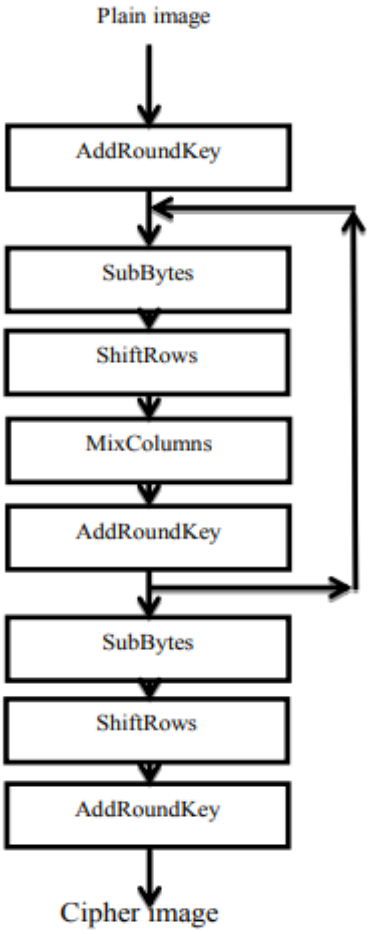

Figure 2.4: AES encryption method [29]

Each round consists of the following four steps:

3.1.1. Substitute Bytes:

The Sub Bytes transformation includes non-linear byte substitution, operating on each of the state bytes independently. This is done by using a once-precalculated substitution table called S-box. S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values [29].

3.1.2. Shift Row:

Shift Rows transformation includes the rows of the state being cyclically left shifted. Row 0 is unchanged; row 1 shifts one byte to the left; row 2 shifts two bytes to the left; and row 3 shifts three bytes to the left [29].

3.1.3. Mix Columns:

In Mix Columns transformation, the columns of the state are considered as polynomials over GF (2^8) and multiplied by modulo $x^4+ 1$ with a fixed polynomial c(x), given by [29]:

$$c(x)=\{03\}x^3 + \{01\}x^2 + \{01\}x+ \{02\}.$$

3.1.4. Add Round Key:

In the Add Round Key transformation, a Round Key is added to the State resulted from the operation of the Mix Columns transformation by a simple bitwise XOR operation.

The Round Key of each round is derived from the main key using the Key Expansion algorithm. The encryption and decryption algorithm needs fourteen 256-bit Round Key [29].

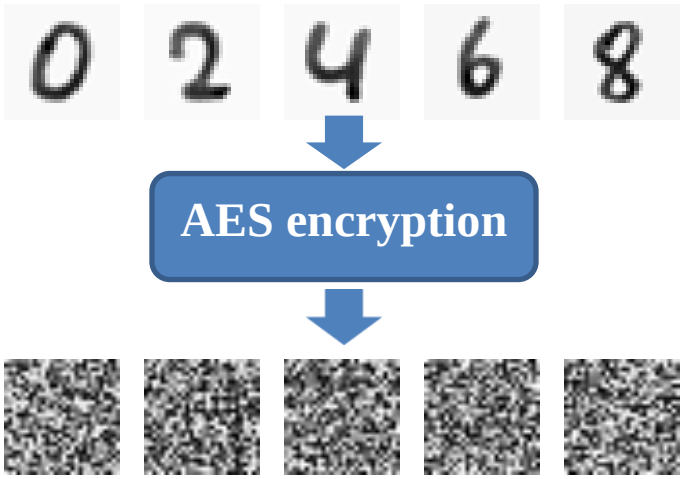

Figure 2.5: AES encryption

3.2.RSA encryption

RSA is one of the first widely used systems for secure data transmission. It is an asymmetric encryption algorithm. It uses two different keys, a public key known to all while the private key is kept secret. Only authorized users know how to open the message. The RSA algorithm has high encryption accuracy and is fast in processing. The key length of this algorithm is more than 1024 bits. The block size of the RSA algorithm is 446 bytes and one round of encryption. There are three different processes used to achieve the encryption process: key generation, encryption and decryption [30].

3.2.1. Key Generation

Step 1: Choose two different prime numbers randomly, name as p and q

Step 2: Multiply these two prime numbers and the results stored in variable n. (n= P*q)

Step 3: Calculate the value of φ (n) = φ (p) φ (q)

Step 4: Select an integer e such that 1<e<φ (n) and calculate the greatest common divisor between the integer e and φ (n). These gcd value is should equal to 1 (gcd (e, φ (n)) =1).

Step 5: Calculate the value of d, such that d= e-1 mod φ (n).

3.2.2. Encryption image:

We used the RSA algorithm to encrypt the images, where the pixel value (v) is encrypted using the keys (n, e) from the equation [30].

$$c = v\,e\,mod\,(n)$$

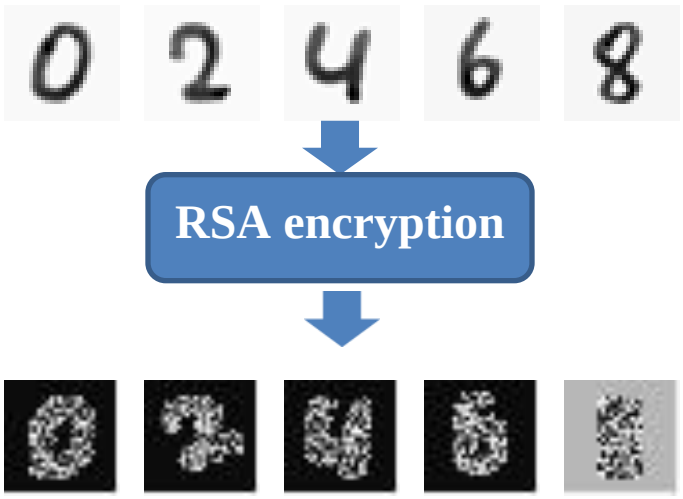

Figure 2.6: RSA encryption

3.3.Arnold cat map encryption

Arnold's Cat Map is a chaotic two-dimensional image that can be used to change the position of a pixel without removing any information from the image. The following equation can be used to write a 2-dimensional image of Arnold's Cat Map [31]:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (mod\ n)$$

Where,

- n = the width and the height of the image.
- x, y = the original image's pixel position
- x', y' = pixel position after the mapping.
- q, p = system parameters.

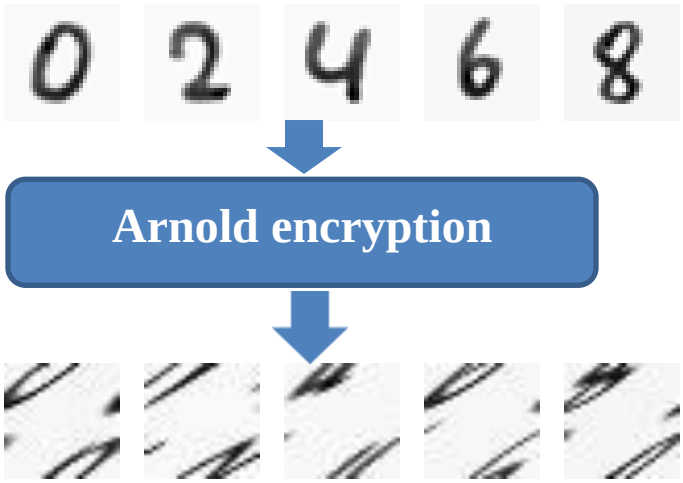

Figure 2.7: Arnold encryption

3.4. Henon map encryption

As developed by Henon in 1976, the Henon map is a two-dimensional iterated discrete-time dynamical system with a chaotic attractor. It can be expressed using the following equations [31]:

$$x' = y' + 1 - ax^2$$

$$y' = bx$$

Where,

- n = the width and the height of the image.
- x, y = the original image's pixel position
- x', y' = pixel position after the mapping.
- a,b = system parameters.

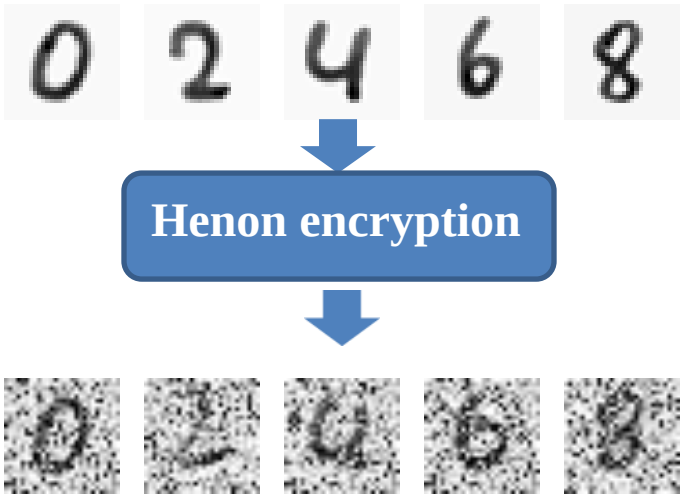

Figure 2.8: Henon encryption

4. Conclusion

In this chapter, we described cryptography and its branches, asymmetric, symmetric, and chaotic with the used methods of image encryption. In the next chapter, we will see the final results of our work.

Chapter 3

Experimental result

1. Introduction

In the previous chapter, we presented the general structure of the system and showed the details of each stage and each step of the treatment. In this chapter, we present the tools used in developing our work and also present the empirical results obtained.

2. Development Tools

2.1. Anaconda Distribution 2.1.1

The open-source Anaconda Distribution is the easiest way to perform Python data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for developing, testing, and training on a single machine [32].

2.2. Spyder IDE 5.1.5

Spyder is a free and open source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package [33].

2.3. Google colab

Google Research's Colaboratory, "Colab" for short, is a product. Anyone may use Colab to write and run Python code directly from their browser. It's a great place to learn about machine learning, data analysis, and education. Colab is a hosted Jupyter notebook service that requires no configuration and gives users free access to computational resources, including GPUs [34].

2.4. Python language

The used versions is 3.9, python is an easy-to-learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms [35].

2.6.The used python libraries

List some used libraries:

2.6.1. Os: Python OS module provides easy functions that allow us to interact and get Operating System information and even control processes up to a limit [36].

2.6.2. TensorFlow 2.7: TensorFlow is an open-source framework compatible with python for machine learning, it brings together a large number of Machine Learning and Deep Learning models and algorithms. This library makes it possible in particular to train and run neural networks for the classification of handwritten digits [37].

2.6.3. Keras 2.7: Keras is a free, powerful, and easy-to-use open-source Python library for developing and evaluating deep learning models [38].

2.6.4. Scikit-learn: is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN [39].

2.6.5. Numpy: NumPy is the most important Python package for scientific computing. It's a Python library that includes a multidimensional array object, derived objects (such as masked arrays and matrices), and a variety of routines for performing fast array operations, such as mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and more [40].

2.6.6. Matplotlib: Matplotlib is a Python package that allows you to create static, animated, and interactive visualizations [41].

2.6.7. Pycrypto: This is a collection of both secure hash functions (such as SHA256 and RIPEMD160), and various encryption algorithms (AES, DES, RSA, ElGamal, etc.) [42].

2.7.Hardware Configuration

This program was created in a type of HP laptop with the following specifications:

- Processor: AMD Ryzen 3 2200U/2.50 GHz
- Installed memory (RAM): 12.0GB.
- System type: 64-bit operating system.
- Operating system: Windows 10 pro.

3. Experimentations and Results

3.1. Used CNN models

After we have prepared our data set, we will explain our training using the Convolution neural network method, where we used the AlexNet model and followed the progress accuracy, and we made adjustments to the settings for our model in an experimental way, and we also made adjustments and augmentations of the data in our data set. The results obtained are represented in (accuracy/loss) and (recall/precision).

3.1.1. Accuracy

Accuracy is a statistical measure which is defined as [43]:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

TP = True positive.

FP = False positive.

TN = True negative.

FN = False negative.

All predictions made by the classifier.

3.1.2. Loss

The loss reduces all the various good and bad aspects of a possibly complex system down to a single number, a scalar value, which allows candidate solutions to be ranked and compared [44].

3.1.3. Recall

Recall, also known as sensitivity, is the ratio of the correctly identified positive cases to all the actual positive cases" [43].

$$precision = \frac{TP}{TP + FN}$$

Where:

TP = True positive.

FN = False negative.

3.1.4. Precision

Precision is the ratio of the correctly identified positive cases to all the predicted positive cases, i.e. the correctly and the incorrectly cases predicted as positive [43].

$$precision = \frac{TP}{TP + FP}$$

Where:

TP = True positive.

FP = False positive.

3.1.5. AlexNet

Is one of the deep convolution nets designed to deal with complex scene classification task on datasete.

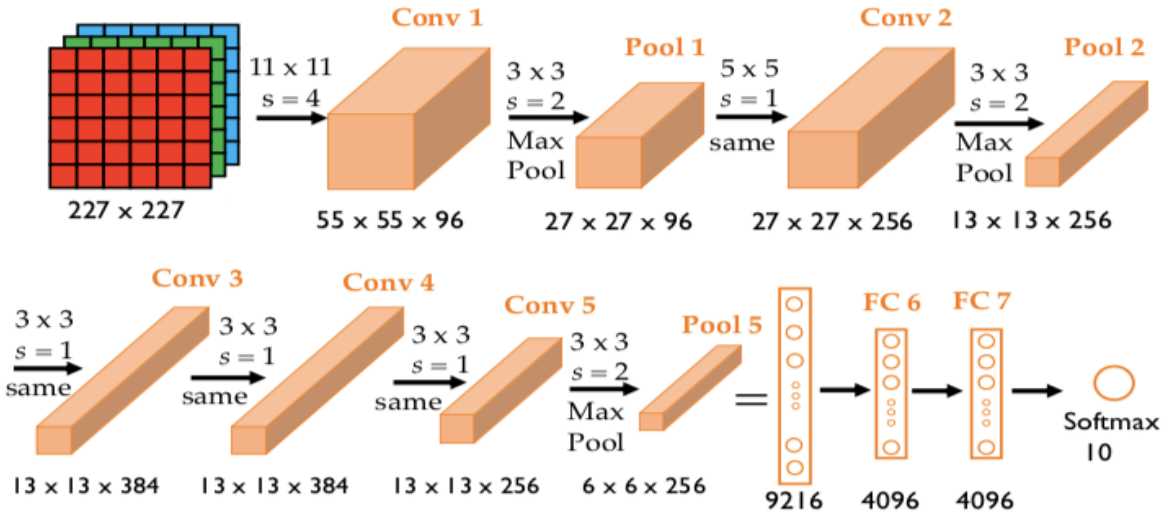

Figure 3.1: AlexNet architecture [45]

AlexNet includes five convolutional layers, three maxpooling layers, three fully connected layers, and a linear layer with softmax activation in the output. Moreover, it uses the dropout regularization method to reduce overfitting in the fully connected layers and applies Rectified Linear Units (ReLUs) for the activation of those and the convolutional layer function to speed up the learning process.

```
input_shape = (227,227,1)
img_imput = Input(shape=input_shape, name='img_input')

x = Conv2D(96, (11, 11),strides=(4,4), padding='valid',activation='relu', name='layer_1')(img_imput)
x = MaxPooling2D(pool_size=(3,3), strides=(2,2), padding='valid', name='layer_2')(x)

x = Conv2D(256, (5,5),strides=(1,1), padding='same', activation='relu', name='layer_3')(x)
x = MaxPooling2D(pool_size=(3,3), strides=(2,2), padding='valid', name='layer_4')(x)

x = Conv2D(384, (3, 3),strides=(1,1), padding='same', activation='relu', name='layer_5')(x)

x = Conv2D(384, (3, 3),strides=(1,1), padding='same', activation='relu', name='layer_6')(x)

x = Conv2D(256, (3, 3),strides=(1,1), padding='same', activation='relu', name='layer_7')(x)
x = MaxPooling2D(pool_size=(3,3), strides=(2,2), padding='valid', name='layer_8')(x)

x = Flatten()(x)
x = Dense(units=4096, activation="relu", name='layer_9')(x)
x = Dense(units=4096, activation="relu", name='layer_10')(x)
x = Dense(units=10, activation="softmax")(x)
```

Figure 3.2: AlexNet model

In our first experiment, we changed the image size from (28 * 28) to (227 * 227) for the normal data set, then we encrypted it with the encryption methods mentioned in the last chapter, and then we trained the data set separately. The result was as follows:

Encryption method	Accuracy	Loss
CVL Gray scale image	81.87 %	71.04 %
RSA	64.95 %	99.99 %
AES	10.00 %	99.99 %
Arnold cat map	66.52 %	99.99 %
Henon map	64.04 %	99.99 %

Table 3.1: AlexNet model result

We note from the table the accuracy and loss for each data set, as in the normal CVL data set that did not encrypt, the accuracy was 81.87%, the loss was 71.04%, and in the encrypted data set it was RSA 64.95%, AES 10%, Arnold 66.52% and Henon 64.04 %, while their loss for all of them was 99%, which is completely unsatisfactory.

To improve the recognition accuracy and recall, we conducted several experimental methods (protocol):

1. Protocol 1: Made settings adjustments in AlexNet model.
2. Protocol 2: Made data augmentation using encrypted image.
3. Protocol 3: Create new image based on the concatenation between the original image and encrypted image

3.1.6. Protocol 1: Settings adjustments in AlexNet model:

We changed the size of the input image from (227x227) to (28x28), which is the same of the original images in CVL dataset, and we fixed all kernels size to (3x3) and the all strides to (1x1) according the used images dimension in data set.

```
x = Conv2D(96, (3, 3),strides=(1,1), padding='valid',activation='relu', name='layer_1')(img_imput)
x = MaxPooling2D(pool_size=(3,3), strides=(1, 1), padding='valid', name='layer_2')(x)

x = Conv2D(256, (3,3),strides=(1,1), padding='same', activation='relu', name='layer_3')(x)
x = MaxPooling2D(pool_size=(3,3), strides=(1, 1), padding='valid', name='layer_4')(x)

x = Conv2D(354, (3, 3),strides=(1,1), padding='same', activation='relu', name='layer_5')(x)

x = Conv2D(354, (3, 3),strides=(1,1), padding='same', activation='relu', name='layer_6')(x)

x = Conv2D(256, (3, 3),strides=(1,1), padding='same', activation='relu', name='layer_7')(x)
x = MaxPooling2D(pool_size=(3,3), strides=(1, 1), padding='valid', name='layer_8')(x)

x = Flatten()(x)
x = Dense(units=4096, activation="relu", name='layer_9')(x)
x = Dense(units=4096, activation="relu", name='layer_10')(x)
x = Dense(units=10, activation="softmax")(x)
```

Figure 3.3: our model

3.1.7. Protocol 2: Data augmentation using encrypted image:

To improve the results and to achieve the objective of this study to improve the accuracy of recognizing handwritten digits, we made data augmantation by merging the CVL data set with the encrypted data set. Each encrypting method was combined with the CVL data set separately.

3.1.8. Protocol 3: Create new dataset based on the concatenation between the original image and encrypted image

As a proposed protocol, we made a new database in which we concatenated the image with its counterpart and it is encrypted, so that each encoding method is separate, so the new image size becomes (28*56). We entered the new data set for training so that we changed the input size in our model to match the new data set.

3.2. The results

In this section, we will present the results of the proposed protocols, and we will compare the protocols between them.

	Protocol 1		Protocol 2		Protocol 3	
Encryption method	**accuracy**	**Loss**	**accuracy**	**Loss**	**accuracy**	**Loss**
CVL Gray scale image	96.77%	12.28%	/	/	/	/
RSA	96.70%	09.02%	97.07%	10.28%	95.53%	15.15%
AES	10.01%	99%	52.38%	99%	95.60%	13.86%
Arnold cat map	94.58%	19.08%	96.55%	12.81%	98.21%	06.41%
Henon map	97.16%	09.92%	91.02%	10.41%	97.69%	08.55%

Table 3.2: Results of protocols accuracy/Loss

	Protocol 1		Protocol 2		Protocol 3	
Encryption method	**Recall**	**precision**	**Recall**	**precision**	**Recall**	**precision**
CVL Gray scale image	95.00%	95.00%	/	/	/	/
RSA	88.00%	88.00%	80.00%	82.00%	10.00%	13.00
AES	10.00%	01.00%	51.00%	85.00%	83.00%	87.00%
Arnold cat map	92.00%	92.00%	92.00%	92.00%	91.00%	92.00%
Henon map	75.00%	75.00%	85.00%	85.00%	94.00%	94.00%

Table 3.3: Results of protocols Recall/Precision

Table 3.2 and Table 3.3 represent the results obtained from each protocol, where we find that the results of the first protocol are great because they are a great improvement over the results obtained in Table 3.1, which represents the results obtained from the alexnet model.

In the second protocol, after we make data augmantation, the results improved from the first protocol, which achieved one of the objectives, as we were able to improve the recognition of handwritten digits through RSA coding, which recorded more accuracy than the accuracy of the CVL data set. 97.07% For the rsa dataset we augmented the data with the cvl dataset and 96.77 for the cvl dataset.

In the third protocol, the results improved more than in both the first and second protocols, and we also achieved our objective, as we were also able to improve the recognition of handwritten digits using the proposed protocol and encryption methods Arnold Cat Map and Henon Map, which scored 98.21% and 97.69% on the respectively, which is greater than the accuracy of the CVL dataset at 96.77%.

3.3.Comparison

In this section, we will compare the best results obtained from the proposed protocols with other methods from the "state of the art" work in the same field and dataset as us.

	dataset	approach	Recall/accuracy
State.1 [2]	CVL	oBIFs + SVM	95.21%
State.2 [3]	CVL	CNN	42.69%
	ORAND-CAR-A		92.2%
	ORAND-CAR-B		94.02%
State.3 [4]	CVL	SVM	85.99%
Our work	CVL	Our CNN model	96.77%
	CVL data augmentation with RSA		97.07%
	CVL concatenated with Arnold		98.21%
	CVL concatenated with Henon		97.69%

Table 3.4: comparison our results with other results from state of the art

From Table 3.4, we find that the results obtained are great compared to the other states, and from this, we say that our objective was achieved with the proposed protocols and helped improve the recognition of handwritten digits.

4. Conclusion

In this chapter, we presented the results obtained using a group of deep learning algorithms "CNN", where we obtained very encouraging results using CNN models, and we also presented the results obtained and compared them, and we also compared our results with other work from the state of the art our results were great.

General Conclusion

General Conclusion

In this thesis, we used deep learning, specifically convolutional neural networks (CNN) where we used the AlexNet model to recognize handwritten digits, where we used encrypted images to improve performance while preserving the features of the original images.

We chose the CVL data set consisting of images of handwritten digits to train our CNN model, which consists of 10 categories from 0 to 9. Each class contains 700 test images and 2178 training images, i.e., in total 7000 images for the test and 21,780 for training. All images are gradient gray. Our experiments aimed to study the effectiveness of the proposed system on encrypted images. Therefore, as a first step, we encrypted our dataset using two encryption methods, where we chose AES as the symmetric encryption method and RSA as the asymmetric encryption method, and used Arnold and Henon as the encryption method from Chaos Maps. Each method of encryption is considered a new data set,Then we trained them in an alexnet model.

In order to improve the accuracy of recognizing handwritten digits, we proposed protocols. First, we made modifications to the AlexNet model and trained our dataset. Then we made data augmentation by merging the cvl data set with the encrypted data set, each encrypting method separately, and we followed the accuracy of the obtained recognition. And as a last protocol, we concatenated the images from the CVL dataset with their encrypted counterparts, each encoding method separately, and trained the model on them.

The results were great so that the accuracy of recognition of the data set was 96.77%, and through the proposed protocols it reached 97.07% by the second protocol, and also achieved 97.69% and 98.21% using the second protocol, and thus we achieved our objective of this study, which is to use encryption methods to improve the accurate recognition of handwritten digits.

We used google colab and Anaconda Distribution as a platform and Spyder IDE for writing code to implement our work, and we used Python language to implement our system, which has also been used in the data science field in recent years.

Based on the results obtained empirically, our system provided impressive and promising results as well, with the possibility of developing it further by suggesting a better model.

Bibliographies

[1] Qi Wu, and Siyuan Zhang,Stanford University Shaohan Xu, Application of Neural Network In Handwriting Recognition.

[2] Chawki Djeddi,Youcef Chibani,Imran Siddiqi Abdeljalil Gattal, Isolated Handwritten Digit Recognition Using oBIFs and Background Features, 2016.

[3] Shujing Lyu, Yue Lu, Hongjian Zhan, Handwritten Digit String Recognition using Convolutional Neural Network, 2018.

[4] Bouchoucha Mohammed Tayeb, HANDWRITTEN DIGIT RECOGNITION USING IMAGE ENCRYPTION, 2020.

[5] Rishika Kushwah,Samay Pashine Ritik Dixit, Handwritten Digit Recognition using Machine and Deep Learning Algorithms, 2021.

[6] Hala Djerouni, Développement d'un système de reconnaissance de chiffres manuscrits, 2021, Mémoire de fin d'étude, UNIVERSITÉ ECHAHID HAMMA LAKHDAR - D'EL OUED.

[7] Amit Choudhary, Anand Nayyar, Saurabh Singh and Byungun Yoon Savita Ahlawat, Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN), Received: 25 May 2020; Accepted: 9 June 2020; Published: 12 June 2020.

[8] Bogdan-Ionut Cirstea. Contributions to handwriting recognition using deep neural networks and.

[9] Maria Teresa & Coppola, Gerardo & Zangari, Lorenzo & Curcio, Stefano & Greco, Sergio & Chakraborty, Sudip. (2021). Artificial Intelligence-Based Optimization of Industrial Membrane Processes. Earth Systems and Environment. Gaudio,.

[10] EL-Bakry Hazem, and Mohamed Loey Ahmed El-Sawy, Cnn for handwritten arabic digits recognition based on lenet-5, 2016, In International Conference on Advanced Intelligent Systems and Informatics, pages 566-575, Springer.

[11] Chiranjit & Nandi, Utpal & Pal, Rajat. (2021). Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling. Multimedia

Tools and Applications. 10.1007/s11042-021-11595-4. Changdar,.

Murilo Gustineli, A survey on recently proposed activation functions for Deep.
12]

Rahul & Bandaranayake, Thusitha. (2021). ANALYSIS OF OPTIMIZING NEURAL
13] NETWORKS AND ARTIFICIAL INTELLIGENT MODELS FOR GUIDANCE, CONTROL, AND NAVIGATION SYSTEMS. Jayawardana,.

Jason Brownlee. (January 9, 2019) A Gentle Introduction to the Rectified Linear Unit
14] (ReLU). [Online]. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

Kamel. (2018). Reconfigurable hardware acceleration of CNNs on FPGA-based smart
15] cameras. Abdelouahab,.

Dive into Deep Learning. Dive into Deep Learning. [Online].
16] https://d2l.ai/chapter_convolutional-neural-networks

Jason Brownlee. (April 22, 2019) A Gentle Introduction to Pooling Layers for
17] Convolutional Neural Networks. [Online]. https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/

[Online]. https://epynn.net/Pooling.html
18]

Amit Choudhary , Anand Nayyar , Saurabh Singh , Byungun Yoon Savita Ahlawat,
19] Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN), Received: 25 May 2020; Accepted: 9 June 2020; Published: 12 June 2020.

machinelearningmastery. [Online]. https://machinelearningmastery.com/early-
20] stopping-to-avoid-overtraining-neural-network-models/

hasty visionAI wiki. [Online]. https://wiki.hasty.ai/scheduler/reducelronplateau
21]

Scott Vanstone, Paul van Oorschot Alfred Menezes, Handbook of applied
22] cryptography, 1996.

https://www.bartleby.com/subject/engineering/computer-
23] science/concepts/cryptography.

Hans &Knebl, Helmut Delfs, "Symmetric-key encryption". Introduction to
24] cryptography: principles and applications. Springer. ISBN 9783540492436., 2007.

Jan Carlo & Delima, Allemar Jhone. (2020). Caesar Cipher With Goldbach Code
25] Compression For Efficient Cryptography. 8. 2992-2998. 10.30534/ijeter/2020/19872020. Arroyo,.

William Stallings, Cryptography and Network Security: Principles and Practice, 1994.
26]

gamze & Sönmez, Ferdi & Zontul, Metin & Kaynar, Oguz. (2018). Comparison of
27] Symmetric and Asymmetric Cryptography Algorithms and A Better Solution: Hybrid Algorithm. maden,.

L.Kocarev,J.Szczepanski J.M.Amigó, Theory and practice of chaotic cryptography.
28]

Abhishek Sachdeva Dr. Prerna Mahajan, A Study of Encryption Algorithms AES,
29] DES and RSA for Security.

Atheer Sultan Almutiri, Bashaier Alqahtani, Rahaf Mohammed Alamri, Hanan
30] Fahhad Alqahtani, Nada Nasser Alqahtani, Ghadeer Mohammed alshammari, and Azza. A. Ali Dalia Mubarak Alsaffar, Image Encryption Based on AES and RSA Algorithms, March 24, 2020.

Niraj San,Baddigam Asha,Savvy Prasad,S.S.Sahu Rupesh Kumar Sinha, Chaotic
31] Image Encryption Scheme Based on Modified Arnold Cat Map and Henon Map.

[Online]. https://www.anaconda.com/distribution/
32]

[Online]. https://www.spyder-ide.org/
33]

[Online]. https://research.google.com/colaboratory/faq.html
34]

Guido Van Rossum and Fred L Drake Jr., Python tutorial. Centrum voor Wiskunde en
35] Informatica Amsterdam, Netherlands, 1995.

[Online]. https://docs.python.org/3/library/os.html
36]

[Online]. https://www.tensorflow.org/
37]

[Online]. https://keras.io/
38]

Fabian Pedregosa et al., ""Scikit-learn: Machine Learning in Python".," *Journal of*
39] *Machine Learning Research. 12: 2825–2830.*

[Online]. https://numpy.org/doc/stable/
40]

[Online]. https://matplotlib.org/
41]

[Online]. https://pypi.org/project/pycrypto/
42]

[Online]. https://python-course.eu/machine-learning/evaluation-metrics.php
43]

[Online]. https://machinelearningmastery.com/loss-and-loss-functions-for-training-
44] deep-learning-neural-networks/

https://blog.devgenius.io/alexnet-the-net-that-surpassed-cnns-5d551ba1b901.
45]

Stefan Fiel, Angelika Garz, Manuel Keglevic, Florian Kleber, Robert Sablatnig
46] Markus Diem, ICDAR2013 Competition on Handwritten Digit Recognition (HDRC 2013).