T.C.

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

DEVELOPING A BASIC NEURAL NETWORK TO CLASSIFY IMAGES FROM THE MNIST DATASET

MASTER'S THESIS

Hamza Basil Hasan Ben TARİF

Department of Computer Engineering

Department Of Computer Engineering Program

NOVEMBER, 2023

T.C.
ISTANBUL AYDIN UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

DEVELOPING A BASIC NEURAL NETWORK TO CLASSIFY IMAGES FROM THE MNIST DATASET

MASTER'S THESIS

Hamza Basil Hasan Ben TARİF
(Y2113.011002)

Department of Computer Engineering
Department Of Computer Engineering Program

Thesis Advisor: Prof. Dr. ALI OKATAN

NOVEMBER, 2023

APPROVAL PAGE

DECLARATION

I hereby declare with the respect that the study “Developing a basic neural network to classify images from the MNIST dataset”, which I submitted as a Master thesis, is written without any assistance in violation of scientific ethics and traditions in all the processes from the project phase to the conclusion of the thesis and that the works I have benefited are from those shown in the References. (22/11/2023)

Hamza Basil BenTarif

FOREWORD

First, I would like to express my endless gratitude to God for being who I am right now and helping me to find patience, strength within myself to complete this thesis, “Developing a basic neural network to classify images from the MNIST dataset”.

I would also like to thank my family not only for encouraging me to go abroad for a master’s degree but also for teaching me to chase my dreams and never give up.

I feel very fortunate to have Prof. Dr. ALI OKATAN as my supervisor and want to express my appreciation for guiding me within the whole research process in a patient and effective manner.

Prof. Dr. ALI OKATAN is not only professional in her field, but a person with a great heart that keeps encouraging me.

Finally, I would like to acknowledge the important contribution of Istanbul Aydin University to my life, not only from an academic perspective but helping to meet great people that inspire, challenge, support and motivate me.

November, 2023 Hamza Basil BenTarif

DEVELOPING A BASIC NEURAL NETWORK TO CLASSIFY IMAGES FROM THE MNIST DATASET

ABSTRACT

This research discusses one of the important topics, which is entitled " Developing a basic neural Network to classify images from the MNIST dataset". The research aims mainly to talk about the concept of MNIST dataset. It also aims to reveal the importance of developing a basic neural network in classifying images from the MNIST dataset and discuss how to create a neural network to recognize handwritten digits from the famous MNIST dataset.

The process of collecting data is based on the descriptive study by collecting and measuring the required information, which can be obtained using many variables. So, and this process is very important for a number of reasons, which are: the most important that It helps in obtaining answers and solutions of many problems and questions, it helps in facilitating the decision-making and increases the quality of decisions that were taken, and also it helps in improving the quality of different outputs,the findings of this paper are that the MNIST dataset consists of a total of 70,000 images, with 60,000 images designated for training and 10,000 images for testing purposes. İt was found out that the MNIST dataset consists of 10 classes, enabling us to classify numbers ranging from 0 to 9. İt was concluded that the MNIST dataset is widely recognized as a popular benchmark for learning the usage of neural networks.

Key words: neural network, MNIST dataset, images.

DEVELOPING A BASIC NEURAL NETWORK TO CLASSIFY IMAGES FROM THE MNIST DATASET

ÖZET

Bu araştırmada önemli konulardan biri olan "MNIST veri kümesindeki görüntüleri sınıflandırmak için temel bir sinir ağının geliştirilmesi" başlıklı konu tartışılmaktadır. Araştırma esas olarak MNIST veri seti kavramından bahsetmeyi amaçlamaktadır. Aynı zamanda, MNIST veri kümesindeki görüntülerin sınıflandırılmasında temel bir sinir ağı geliştirmenin önemini ortaya çıkarmayı ve ünlü MNIST veri kümesindeki el yazısıyla yazılmış rakamları tanıyacak bir sinir ağı oluşturmaya yönelik gow'u tartışmayı amaçlamaktadır.

Veri toplama süreci, birçok değişken kullanılarak elde edilebilecek gerekli bilgilerin toplanması ve ölçülmesi yoluyla betimsel çalışmaya dayanmaktadır. Dolayısıyla bu süreç birçok nedenden dolayı çok önemlidir: En önemlisi Birçok sorun ve sorunun yanıtlarına ve çözümlerine ulaşmaya yardımcı olur, Karar almayı kolaylaştırmaya yardımcı olur ve alınan kararların kalitesini artırır. alınır ve aynı zamanda farklı çıktıların kalitesinin artırılmasına da yardımcı olur.

Bu makalenin bulguları, MNIST veri setinin 60.000'i eğitim amaçlı ve 10.000'i test amaçlı olmak üzere toplam 70.000 görüntüden oluştuğu yönündedir. MNIST veri setinin 10 sınıftan oluştuğu ve bu sayede 0'dan 9'a kadar olan sayıları sınıflandırmamıza olanak sağladığı tespit edildi. MNIST veri setinin sinir ağlarının kullanımını öğrenmek için yaygın olarak popüler bir kıyaslama olarak kabul edildiği sonucuna varıldı.

Anahtar kelimeler: sinir ağı, MNIST veri seti, görüntüler.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AI : Artificial intelligence

ANN : Artificial neural network

CNN : Convolutional Neural Network

Conv : Convolution

ConvNet : Convolutional Network

EMINIST : Extended MINIST

MINIST : Modified National Institute of Standards and Technology

MLP : Multi-layer perceptron

NN : Neural network

OCR : Optical character recognition

ReLU : Rectified Linear Unit

RNN : Recurrent Neural Networks

SNN : Simulated neural network

USPS : United States Postal Service

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

They are interesting tools for creating excellent AI systems that are also enjoyable to learn and use because they can be created and trained using a number of libraries. To put it another way, creating your own neural network from the start will enable you to comprehend the basic concepts and inner operations of neural networks, a well-known dataset for computer vision and machine learning, the MNIST dataset, contains 70,000 grayscale pictures of apparel and accessories categorized into 10 different categories. Despite having square formats that are relatively small—28x28 pixels for each image compared to other enormous image datasets, the variations among the images make it difficult for machine learning models to correctly identify the images,even if the problem is solved, convolutional deep-learning neural networks can be trained and educated to recognize images using the dataset. To accomplish this, a trustworthy testing tool must be developed to evaluate model performance. In addition, model changes must be corrected, and the model must be saved and loaded later in order to produce projections based on recent data, the handwritten digits in the MNIST dataset are part of a well-known public dataset.10,000 test cases are included in the test set, while 60,000 examples are used in the training set. The voices of American high school students and Census Bureau personnel were combined to create it. The training set comprises 30,000 images from both students and staff, as opposed to the test set's 5,000 employee photos and 5,000 student photos, a dataset named MNIST is a collection that appears as tables. Knowing that all rows and columns of a table represent separate variables in the data set, these tables contain each variable in that element, for example, as follows: It might give us, the length and width of a given object, and there is one or more row elements, depending on the number of rows in data set,a beloved dataset for computer vision and deep learning is the MNIST handwritten number classification problem,and this group consists of 60,000 square and small images in order to obtain grayscale, each one measuring 28 x 28, in the form of handwritten numbers from 0 to 9.we shall allocate a group out of ten groups, each representing a handwritten

picture, each of which corresponds to an integer between 0 and 9, inclusive.We can use the data set to start learning and practicing how to create, evaluate, and apply deep learning convolutional neural networks to classify images even after they have been successfully resolved. This section explains how to build a new, reliable testing tool to evaluate how the model works, check for modifications to the model, and save the model before loading it to make new predictions about the data,neural networks are used for deep learning because it is one of the many sub-fields of the method of developing artificial intelligence. It was announced for the first time seventy years ago to be an attempt to simulate the principle and method of working of the human brain, but it turned out to be simpler than the real neuron, where each cell is Neurons are associated with a group of layers, each of which has certain weights that express the importance of these layers, and in order to discover how neurons respond when we publish data and information through them.

In fact, neural networks have been rare and common for many years. Statistics and data scientists believed that the model family had at least one important flaw whose results were difficult to understand. Also, the difficulty of visualizing the structure of any neural network is one of the reasons why many people see it as a puzzle.

Many neural networks exhibit a multitude of weights ranging from tens of thousands to millions, necessitating manual adjustments during the training process in order to minimize errors. Identifying the precise factors that contribute to the superior performance of one neural network over another is a formidable task, given the intricate and interconnected nature of the numerous variables involved. Furthermore, the design of high-quality neural network architectures poses significant challenges due to the complex interplay of these factors.

A lot of advanced studies are being done on neural network architecture. It is unusual for a new architecture to perform better than the existing one for a given task. Many practitioners of architecture build on recent publications, mass-produce it for new tasks or make a small change for Gradual improvement.

The advancement of the learning process was previously limited by the capacity of neural networks to simulate a limited number of neurons. However, recent years have witnessed remarkable progress in hardware development, enabling the construction of deep neural networks capable of effectively processing extensive

datasets. This significant achievement has led to groundbreaking advancements in the field of artificial intelligence, specifically.

These qualitative leaps allowed machines to approach human capabilities, and even exceed them in performing some limited tasks. Among these tasks is its ability to recognize objects. However, new developments in deep learning have made it feasible to create neural networks that can recognize objects, faces, texts, and even emotions! Historically, robots have been unable to match the power of human vision,shows us the meaning of the neural network to a group of different elements, for example, the number of layers within the network, the number of units in all layers, and how to connect the units between each different layer to conduct the process of communication between them because neural networks take their idea from the principle of the human brain, so we use the term module to represent what is called cell biology.

The function of the units for some values is to take the previous units and put them as inputs to them, as they are similar to the neurons that send their signals around the brain, and then perform mathematical operations and send the new values as outputs to other units. We put these units in the form of connected layers on top of each other to make a neural network, so that this network can consist of at least two layers: a layer for input values and a layer for outputting values, while the hidden layer is for all layers that are between the input and output layers. Knowing that it is hidden from the real world.

Performance can be used as a criterion to compare various network architectures, while other factors like medians, data, and training time can also be used as a criterion to compare various network architectures.

A. Background Studıes And Related Works:

Image classification has previously been researched using MNIST using a variety of techniques and methodologies, some of the more recent of which are described below:

In this comprehensive review Written by Pedro Isasi (4 August 2019)

This paper summarizes the top state-of-the-art contributions reported on the MNIST dataset for handwritten digit recognition. This dataset has been extensively

used to validate novel techniques in computer vision, and in recent years, many authors have explored the performance of convolutional neural networks (CNNs) and other deep learning techniques over this dataset. By mid-2017, a new dataset was introduced: EMNIST, which involves both digits and letters, with a larger amount of data acquired from a database different than MNIST's. In this paper, EMNIST is explained and some results are surveyed.

A research conducted by Md. Anwar Hossain & Md. Mohon Ali (2019) Humans can see and visually sense the world around them by using their eyes and brains. Computer vision works to enable computers to see and process images in the same way that human vision does. Several algorithms developed in the area of computer vision to recognize images. The goal of our work will be to create a model that will be able to identify and determine the handwritten digit from its image with better accuracy. We aim to complete this by using the concepts of Convolutional Neural Networks and the MNIST dataset. We will also show how MatConvNet can be used to implement our model with CPU training as well as less training time. Though the goal is to create a model that can recognize the digits, we can extend it to letters and a person's handwriting. Through this work, we aim to learn and practically apply the concepts of Convolutional Neural Networks.

A research conducted by (Sérgio D. Correia (2021)) Online fashion market is constantly growing, and an algorithm capable of identifying garments can help companies in the clothing sales sector to understand the profile of potential buyers and focus on sales targeting specific niches, as well as developing campaigns based on the taste of customers and improve user experience. Artificial Intelligence approaches able to understand and label humans' clothes are necessary, and can be used to improve sales, or better understanding users. Convolutional Neural Network models have been shown efficiency in image c1assification. This paper presents four different Convolutional Neural Networks models that used Fashion-MNIST dataset. Fashion-MNIST is a dataset made to help researchers finding models to classify this kind of product such as clothes, and the paper that describes it presents a comparison between the main classification methods to find the one that better label this kind of data. The main goal of this project is to provide future research with better comparisons between classification methods. This paper presents a Convolutional Neural Network approach for this problem and compare the classification results

with the original ones .

A research conducted by (Vinay Kumar V 2019) Recognizing handwritten digits is a challenging task primarily due to the diversity of writing styles and the presence of noisy images. The widely used MNIST dataset, which is commonly employed as a benchmark for this task, includes distorted digits with irregular shapes, incomplete strokes, and varying skew in both the training and testing datasets. Consequently, these factors contribute to reduced accuracy in digit recognition. To overcome this challenge, we propose a two-stage deep Learning approach. In the first stage, we create a simple neural network to identify distorted digits within the training set. This model serves to detect and filter out such distorted and ambiguous Images. In the second stage, we exclude these identified images from the training dataset and Proceed to retrain the model using the filtered dataset. This process aims to improve the Classification accuracy and confidence levels while mitigating issues of underfitting and overfitting.

Table 1 A summary of Developing a basic neural network to classify images from the MNIST dataset:

Study	**Study**	**Study**
(4 August 2019)	K-NN	89.51%
Ali (2019)	MatConvNet; ReLu	99.15%
Co (2021)	SVM, CNN	96.46%
Vinay Kumar V	RNN,CNN	99.44%

I want to talk about **“Developing a basic neural network to classify images from the MNIST dataset”.**

I wish this thesis to be useful for all people who are interested in the MNIST dataset.

II. BACKGROUND INFORMATION

A. An Introduction to Neural Network

A computational model known as a neural network is based somewhat on the cortical architecture of the brain. It's also known as a parallel distributed processing network. The interconnected processing units known as ganglia or neurons that make up this structure work together to generate the output function. The output of a neural network is based on how cooperatively any individual neuron in the network can act. As opposed to earlier sequential (or sequential) information processing devices like Von Neumann or binary computers, a neural network has the ability to process information and data in parallel. The distinguishing feature by which they are distinguished from others is their overall ability to maintain general functions even if some of the individual neurons within the networks are not working. This flexibility arises from the interconnection between neurons and organs, allowing them to perform their functions faithfully.

With deep learning in still-under-researched methodologies, amazing outcomes for image processing can be obtained. The method of building an image using a neural network is very important, and in order to do this, we need a little knowledge about the workings of CNN from the inside, we use the phrase neural network theory to describe it sometimes as a branch of computational science.

Uses neural networks to simulate complex events and/or conducts analytical research on the underlying ideas of neural networks. In contrast to artificial intelligence (AI), which solves problems using traditional computational techniques, A neural network uses a network called an agent network, which is an entity connected to programs and hardware connected to each other as a computational architecture. Neural networks are systems that accept training in order to be able to effectively address complex or unexpected problems and difficulties using different models that we can call acquired knowledge. Such as those found in some environmental forecasts and the stock market. In other words, they are self-adapting

systems.

An artificial neural network ANN, also referred to as a simulated neural network SNN or simply a neural network NN, is a network comprising interconnected artificial neurons. It employs a mathematical or computational model to process data and relies on a communication-based computing strategy.

Usually, the structure of the neural network called ANN undergoes some modifications to respond to the flow of information within the network, whether it is internal or external. In practical applications, we find that the neural network is useful in modeling non-linear statistical data. They can be used to spot trends in data or to simulate complex interactions between inputs and output.

To put it another way, it is challenging to accept failure or a mistake. Furthermore, neural networks are more appropriate for fuzzy logic computing applications than von Neumann computers.

The term neural network has been used throughout history to describe a network or group of biological neurons. And at the present time, the term is used to describe artificial neural networks, which contain neurons or artificial nodes, and accordingly, the meaning of the term neural network has several different interpretations:

Biological neural networks contain neurons whose function is connected to the central nervous system. Neuroscientists typically recognize them as groups of neurons that, when observed in a lab setting, perform a certain physiological function.

Deep learning neural networks include convolutional neural networks CNN, which are a subset of them. The advancement of CNNs has greatly enhanced image recognition. They function in the background of image categorization and are mostly employed to assess visual images.

Images are categorized using deep neural networks, which have a large number of intermediate (hidden) layers. The networks can thus extract complex components from the images. The following layers output a vector, in this case, one with ten components, whose elements each represent the likelihood that the image belongs to a particular class.

The neural network (NN) processes input images and extracts relevant features required for its task. Within the network architecture, there exist multiple layers, each designed with specific objectives that facilitate the transformation from input to output. There are, however, a few core components that each CNN must possess. Contemporary CNNs are extremely complex and include a wide range of layers or blocks of layers that improve performance.

Neural networks are a common tool for image processing in the current world. It can produce images as the output layer in a machine-learning model. The output layer contains each of the desired properties for your image. If you know how to use a convolutional neural network (CNN), there are limitless ways to create a picture.

The Neural Network Up scaling Tool, a piece of software, uses deep learning image processing to upscale pictures beyond its initial resolution number of pixels.

For the neuron or nerve ganglion that is interconnected in multi-layered frameworks, we may resort to using the computer to build a system that allows them to learn permanently and continuously from errors in order to improve their results, and here artificial neural networks seek to deal with difficult problems, such as summarizing some documents and recognizing faces.

Neural networks empower computers to make intelligent decisions with limited human involvement, enabling them to learn and represent intricate, nonlinear connections between input and output data. As a result, they possess the ability to perform various tasks, such as making overarching generalizations and deductions, even without explicit training. Furthermore, neural networks exhibit the capacity to comprehend unstructured information and derive extensive inferences. In the past, despite the fact that neural networks received a lot of attention, many data scientists believed that the model family has at least one flaw, results that we can't crack, and it's one of the reasons why people think neural networks are confusing.

Neural networks commonly comprise a substantial number of weights, ranging from tens of thousands to millions, where each weight is individually adjusted during the training process to minimize error. Determining the precise reasons behind the superior performance of one neural network compared to others is a formidable task, given the intricate interplay of numerous tightly interconnected

variables. Designing advanced neural network topologies that exhibit high performance presents challenges due to their inherent complexity.

In the event that you are unfamiliar with them, the following machine-learning terms are included:

• When we talk about labels, we refer to them as 'y' and input data is represented as 'x'. The model's predictions are denoted by 'y-hat' (pronounced as y-hat).

• To train our model, we utilize training data, which comprises datasets.

• Even after training the model, test data is kept confidential. We evaluate the performance of our model using test data.

• The loss function is employed to measure the accuracy of the model's future predictions.

• The optimization algorithm has precise control over the training process of the weights in the computational graph.

The field of computer vision has undergone a significant transformation due to the advancements in neural networks, enabling machines to detect and interpret images. Their increasing popularity can be attributed to their remarkable capability to understand complex patterns and intricate features. Convolutional neural networks (CNNs) have emerged as the most widely utilized type of neural network for image processing.

Recently, with the rise in popularity in the realm of natural language processing, there has been a focus on exploring alternative topologies based on adapters and generative pre-trained adapters GPT. These approaches aim to enhance the capabilities of language models and advance the field of natural language processing, has extended to the domain of vision. As a result, vision adapters ViT have gained significant traction due to their groundbreaking achievements.

The field of computer vision has undergone a significant transformation due to the advancements in neural networks, enabling machines to detect and interpret images. Their increasing popularity can be attributed to their remarkable capability to understand complex patterns and intricate features. Convolutional neural networks (CNNs) have emerged as the most widely utilized type of neural network for image

processing.

In general, neural networks can evaluate and recognize images in a variety of ways. This will depend on the network architecture and the problem we're trying to address. Images are widely used by neural networks to address the following problems:

- **Image classification:** Image categorization involves assigning labels or categories to an image, such as distinguishing if a dog or lion is shown in the image.
- **Object detection:** Locate the image and identify its elements.
- **Image_ segmentation:** Divide the image into groups of pixels, each of which is represented by a mask.
- **Image generation:** Image design is the process of designing a new image with specific specifications.

In addition to the aforementioned tasks, we won't discuss the other image-related problems that neural networks might be able to solve, pattern transmission, feature recognition, human posture analysis, image annotation, and image retrieval encompass a range of applications within the domain of interest, in the field of machine learning algorithms known as neural networks, there are some different examples, including perceptual network, Hopfeld network, Boltzmann machines, convolutional, recurrent, long-term, and deep memory networks, and generative adversarial networks the back propagation algorithm is predominantly employed for training the majority of these neural network architectures.

B. Background information of the MNIST dataset

The MNIST dataset, consisting of a vast collection of handwritten numbers, is regularly employed to enhance different image-processing techniques. This database is widely utilized to train and evaluate machine-learning algorithms. To create it, samples from the original MNIST databases were combined in a process known as "remixing." The MNIST creators recognized that this dataset was unsuitable for machine learning research due to its sources:

The training data set was created by US Census Bureau employees, while the

test data was collected by US school students. Accordingly, grayscale levels were extracted from black-and-white MNIST images, which were adjusted to fit a 28 x 28-pixel bounding box and anti-aliasing.

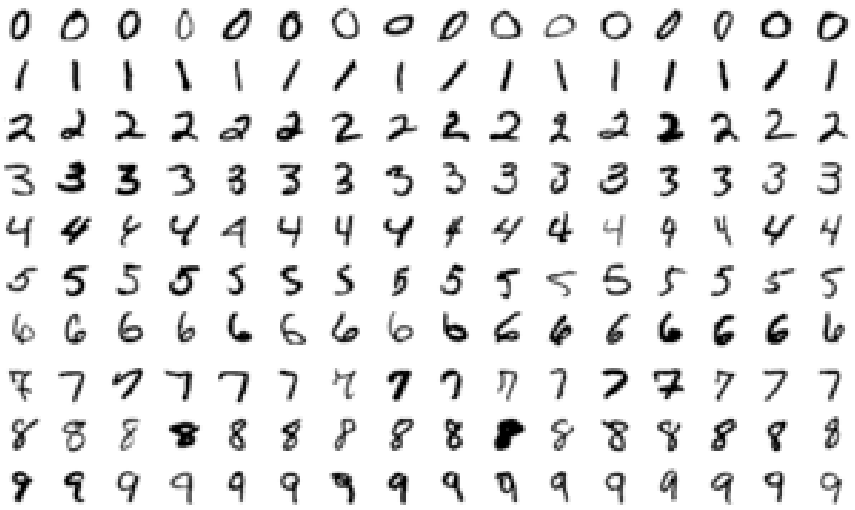


Figure1. Sample images from MNIST test dataset

The MNIST dataset consists of a total of 70,000 images, with 60,000 images designated for training and 10,000 images for testing purposes. To construct the test and training sets, the MNIST training data was used, i.e., the sum of half the data for training and the other half for the test, and similarly, the MNIST test data set was used for the other half of the test and training. The database designers kept a record of the methods that were tested with the training group, and in the first study, they used a supporting machine and got an error rate of 0.8%.

Table 2 MNIST dataset consists of total images:

MNIST IMAGE	Total Images	Training	Testing	Error rate
	70,000	10,000	60,000	0.8%

NIST developed the Extended MNIST EMNIST dataset as. More recent, comprehensive replacement for MNIST. While MNIST only contained handwritten number images, EMNIST includes all 19 classes from NIST's own database, encompassing uppercase letters, lowercase letters, and digits. The photos in EMNIST underwent the same transformation process as the MNIST images, resulting in a consistent 28x28 pixel format. Consequently, tools that are compatible with the original, unmodified MNIST dataset are also likely to work seamlessly with EMNIST.

The MNIST dataset, which consists of handwritten digits, is widely employed for image classification tasks. When it was introduced in the 1990s, the dataset included 60,000 photos, posing a considerable challenge for most machine learning algorithms.

When considering pixel resolution, the MNIST dataset includes a test data collection of 10,000 images. To provide some context, back in 1995 at AT&T Bell Labs, the cutting-edge machine learning hardware was represented by a Sun SPARCstation 5, equipped with a substantial 64MB of RAM and performing at 5 MFLOPs. During the 1990s, the United States Postal Service (USPS) focused on achieving high accuracy in number recognition for automated letter sorting. Deep networks like LeNet-5, SVMs with constants as described by Schölkopf et al., 1996, and, tangent distance classifiers have achieved error rates below 1% in this domain.

For over a decade, MNIST has served as a benchmark for comparing various machine-learning approaches. Even by today's standards, even simplistic models achieve classification accuracy exceeding 95%. Consequently, they are not suitable for distinguishing between stronger and weaker models, despite their effectiveness as a standard dataset. Additionally, the dataset utilized in this context exhibits an exceptional level of precision, surpassing what is typically encountered in most classification scenarios. This characteristic has prompted algorithmic research to concentrate on specific families of algorithms, including active set methods and boundary-seeking active set algorithms which can benefit from clean datasets.

MNIST has evolved into a means of ensuring safety rather than merely a standard for comparison. The challenges posed by Image Net are much more significant. However, due to the extensive size of Image Net, incorporating interactive examples will require extensive expertise. Hence, our focus in the subsequent sections will shift toward the Fashion-MNIST dataset, which is qualitatively comparable to Image Net but considerably smaller in scale. This dataset was released in 2017.

The MNIST collection has thousands of examples of handwritten numbers from 0 to 9. Over the decades, the information has been used in machine learning research and also with regard to the (OCR) optical character recognition system, whose function is to convert handwritten addresses into computer text. For example, postal services help in mail delivery systems.

In the MNIST database, researchers achieved a level of performance comparable to human capabilities by employing a set of neural networks. Additionally, the authors of the study outperformed humans by achieving twice the accuracy in other recognition tests. A simple linear classifier can be used to reach the initial database's maximum error rate of 12 percent without the use of any preprocessing techniques. Researchers unveiled the LIRA classifier in 2004, which featured three brain layers influenced by Rosenblatt's perceptual theories. The database's best-case error rate using this method was an excellent 0.42 percent. Some researchers have deliberately distorted a database for testing purposes in order to assess AI systems. (Wong, 2018)

In such cases, neural networks are commonly used, and the distortions often involve affine or elastic distortions. These systems have shown promising effectiveness, with one particular system achieving a database error rate of 0.39 %.

The MNIST dataset consists of a substantial collection of manually entered digits. This dataset is highly regarded and widely used in the field of image processing. It serves as a popular choice for evaluating the performance of various machine-learning algorithms.

Data scientists often resort to employing the MNIST digits dataset when they want to experiment with machine learning and pattern recognition techniques using real-world data, without investing much time and effort in data preprocessing and formatting.

The MNIST dataset consists of 70,000 images, each measuring 28 × 28 pixels, depicting handwritten digits ranging from 0 to 9.

The field of computer vision has undergone a significant transformation due to the advancements in neural networks, enabling machines to detect and interpret images. Their increasing popularity can be attributed to their remarkable capability to understand complex patterns and intricate features. Convolutional neural networks (CNNs) have emerged as the most widely utilized type of neural network for image processing.

Table 3 MNIST database, researchers achieved a level:

Dataset	Total images	Image dimesions	Content	Accuracy	database error rate
MNIST	70,000	28*28 pixels	Handwritten (0,9)	99.58%	0.42%

The dataset has already been divided into training and testing sets, which we will explore further in the tutorial. MNIST contains a substantial collection of handwritten numbers and is highly popular in the field of image processing. It is commonly employed to assess the effectiveness of various machine-learning techniques. The MNIST database, which stands for Modified National Institute of Standards and Technology database, is widely utilized to train various image processing algorithms. The MINST database is widely used as a repository of handwritten numbers and as a training and testing resource for a machine learning algorithm. To design this data set, the NIST researchers, in contrast, used a process that involved mixing samples from the original NIST operating process. The choice to remix the dataset stemmed from the recognition that the original NIST dataset, collected from American high school students and American Census Bureau employees, lacked suitability for machine learning research. Additionally, the NIST grayscale images underwent adjustments to conform to a standardized 28 x 28-pixel format and were further refined to include a range of gray shades. The MNIST database comprises a collection of 60,000 training images and 10,000 test images. The test set and training set were derived from both halves of the NIST dataset, with an equal distribution, the original designers of the database keep track of some of the methods used on it.Using a direction support machine, they found that their initial study's error rate was 0.8%. For the MNIST data set of handwritten digits and letters, the EMNIST extended data set, which is comparable to MNIST and contains 240,000 training images and 40,000 test images. If you will, it is comparable to the "hello world" of machine learning. A new architecture or framework will be tested and ensured to perform properly using an algorithm that data scientists trained using the MNIST dataset.

Table 4 summarizes the details of the MNIST dataset, including the original dataset

Dataset	Training images (original)	Test original images	Study error rate	Training images (Extended)	Test images (Extended)
MNIST	60,000	10,000	0.8%	240,000	40,000

The MNIST dataset is widely utilized for supervised learning, particularly in training classifiers, as it comprises images of handwritten numbers paired with corresponding labels. This dataset, along with Fei Fei Li's Image Net, is a notable example of high-quality labeled datasets available for research purposes. It contributes to the broader field of machine learning, offering valuable instances of publicly accessible datasets. To create the MNIST database, binary representations of handwritten digits were combined from two NIST sources: Database 3 and Special Database 1. Initially, NIST designated Database 3 as the training set and Special Database 1 as the test set. However, due to its broader recognition and representativeness, Database 3 is preferred over Special Database 1 as a training set, belonging to the training and testing group,between possible samples in order to obtain correct conclusions and answers, and therefore there was an agreement to build a new database that was created by rearranging the NIST data,the MNIST handwritten digit classification issue is a typical data set for computer vision and deep learning.

If we can solve the data set completely, we can use it to start learning and practicing how to develop and evaluate a neural network and use this network for deep learning to classify images.

Here we can say that the MNIST dataset has been successfully resolved and may be a useful place to start developing and refining a plan for using a convolutional neural network to address the challenges of image classification.

We can use the dataset to launch and learn in action how to design, evaluate and deploy deep-learning convolutional neural networks for image set classification. Even if it is solved correctly, it is necessary to build a reliable test tool to measure the performance of the model, fix it, update it, and then store and download it to give us predictions on the new data.

To ensure clarity and privacy while preserving the main idea, the images

within the dataset underwent a process of centering. This involved calculating the center of mass for each pixel and shifting the image to position this point at the center of a 28x28 field. Consequently, all images in the dataset possess a uniform size of 28x28 pixels. The dataset is distributed as CSV files named mnist_train.csv and mnist_test.csv, containing the pixel data for each image. Each line within these CSV files represents an image, consisting of a series of numbers ranging from 0 to 255. The first number in each line serves as the label, corresponding to the actual number represented by the image. The subsequent 784 integers denote the pixel values that form the 28x28 picture,the MNIST database was constructed using binary representations of handwritten numbers from NIST's Special Databases 3 and 1, with SD-3 and SD-1 initially designated as the training and test sets, respectively. However, SD-3, sourced from Census Bureau personnel, displayed greater recognition and representativeness compared to SD-1, sourced from high school students. To ensure unbiased learning studies and reliable conclusions, it was crucial to incorporate both NIST datasets into the creation of a new database, the MNIST training set comprises 30,000 patterns from SD-1 and an additional 30,000 patterns from SD-3. The test set consists of 5,000 patterns from SD-3 and 5,000 patterns from SD-1. Within the training set of 60,000 patterns, there were approximately 250 distinct writers. Careful consideration was given to ensure that the writer sets for both the training and test sets remained distinct. (Feiyang Chen&Nan Chen&Hanyang Mao&Hanlin Hu, 2018)

C. Types of MNIST Dataset:

Although the handwritten version of MNIST is the most common, there are six further extended variants as well:

1. Fashion MNIST:

The fashion MNIST dataset developed by Zalando Research replaces handwritten digits with photographs of various clothing and accessory items belonging to ten different categories. The fashion MNIST dataset encompasses a range of categories, such as ankle boots, purses, coats, dresses, pullovers, sandals, shirts, sneakers, and more.

Similar to the original MNIST dataset, the images in fashion MNIST are presented in grayscale.

2. 3D MNIST:

Unlike the initial MNIST dataset, which consists of grayscale images of size 28x28 pixels with a single channel, the 3D MNIST dataset incorporates images with three channels, representing the Red, Green, and Blue components like regular color images. This dataset serves as a valuable starting point for tackling 3D computer vision challenges.

3. E- MNIST:

EMNIST is a distinct dataset that sets itself apart from MNIST by containing handwritten letters instead of numerical digits. Like MNIST, EMNIST follows the convention of utilizing 28x28 grayscale images, resulting in a notable similarity between the two datasets in terms of format and representation.

4. Sign Language MNIST:

The sign language MNIST dataset showcases images of the English alphabet (A-Z) represented through sign language, drawing parallels to EMNIST. This dataset presents a slightly more complex challenge in hand gesture detection, thus offering valuable applications in real-world scenarios.

5. Colorectal Histology MNIST:

The primary focus of this dataset is on histology tiles collected from patients with colorectal cancer, a disease that affects the colon or rectum in humans. This specific dataset poses a more captivating challenge for biologists, resembling the well-known MNIST problem. It contains eight different types of malignant tissue, offering valuable insights for research and analysis.

6. Skin Cancer MNIST:

The medical dataset comprises images of skin lesions and malignancies, accompanied by corresponding labels. It serves as a valuable resource for individuals employing deep learning techniques to address medical classification problems, making it an ideal primary dataset for such endeavors.

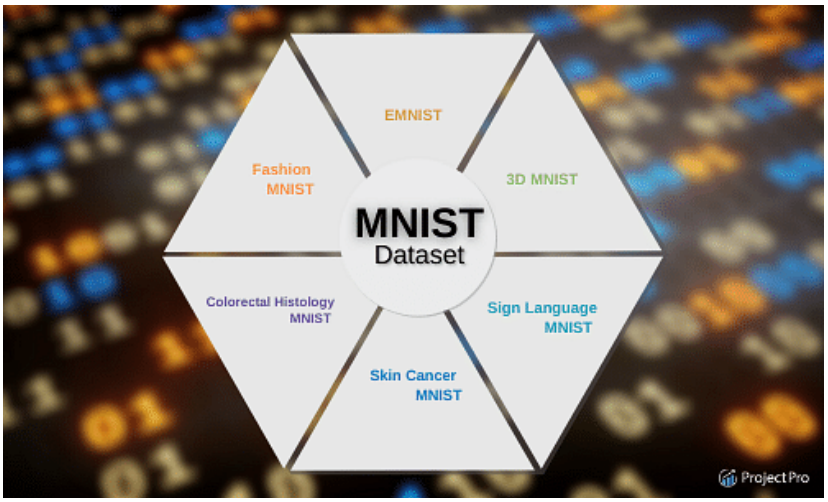


Figure 2. Types of MNIST Dataset

The paper focuses on using modified versions of the MNIST database samples. To train and test the systems discussed in the study, a combined database was created by merging NIST The MNIST database was constructed by merging the binary images of handwritten digits from NIST's Special Databases 3 and 1. Initially, SD-3 and SD-1 were designated as the training and test sets, respectively. However, SD-3 received more recognition and was visually more appealing compared to SD-1,the dissimilarity between SD-1 and SD-3 can be attributed to the fact that they were sourced from different groups of individuals: high school students and Census Bureau employees, respectively. In order to ensure trustworthy and valid conclusions, it is essential to ensure that the results of learning studies remain unbiased by the selection of training sets and test samples.

To create a comprehensive new database, the NIST datasets were combined. SD-1 contained 58,527 digit images contributed by 500 different authors. Unlike SD-3, the data in SD-1 was not organized sequentially, with blocks of data from each writer appearing in order. By utilizing known writer identities, we deciphered the writers associated with the data in SD-1. We then divided SD-1 into two halves: the first 250 writers' characters constituted the new training set, while the remaining 250 writers formed the test group. This division resulted in two sets, each containing approximately 30,000 samples.

Starting with Model #0, we augmented the new training set with sufficient SD-3 samples to achieve a total of 60,000 training patterns. To create a complete test suite comprising 60,000 test patterns, we extended the new test suite by incorporating SD-3 examples from sample #35,000 onwards. While we utilized a

total of 60,000 training samples, we selected a subset of 10,000 test images, specifically 5,000 from SD-1 and 5,000 from SD-3 :

The resulting database was known as the Modified MNIST Data Set. (Hamza BenTarif&Prof. Dr. Ali Okatan, 2023)

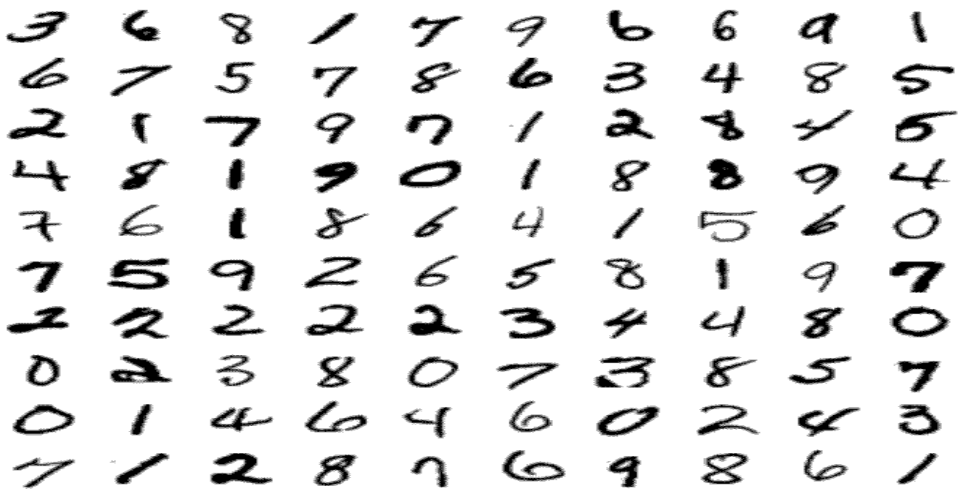


Figure 3. ***MNIST Database Example***

Suppose we have the following figure:

Figure 4. MINIST Picture Example 1

When we observe an image, our eyes and brain work together to distinguish it as the digit eight. Our mind possesses impressive capabilities, allowing it to swiftly categorize the image as an eight. Numbers can exhibit diverse shapes, and although our brains can promptly recognize different shapes and associate them with specific numbers, computers face challenges in accomplishing this task. The only viable approach is to train a computer using a deep neural network to achieve accurate recognition of handwritten numbers. (Hamza BenTarif&Prof. Dr. Ali Okatan, 2023)

Until now, our focus has been on working with straightforward data points on the Cartesian coordinate system. Throughout our work, we have primarily dealt with binary layer datasets. However, as we transition to using multiclass datasets, certain adjustments need to be made. One significant change involves replacing the sigmoid activation function in the output layer by means of the Softmax activation feature. (Hamza BenTarif&Prof. Dr. Ali Okatan, 2023)

The sigmoid activation function has been widely recognized for its effectiveness in assigning probability values ranging from 0 to 1, making it suitable for categorizing binary datasets. However, when dealing with multiclass datasets, the sigmoid function becomes inadequate. Instead, we employ the Softmax activation function, which is capable of effectively handling multiclass classification tasks. (Hamza BenTarif&Prof. Dr. Ali Okatan, 2023)

The MNIST dataset consists of 10 classes, enabling us to classify numbers ranging from 0 to 9. The primary distinction between the previously used data set and the MNIST data is the way MNIST is inserted into the neural network.

Both the perceptron model and linear regression approach describe each data point as a simple

X, y coordinate. This shows that for a single data point to be input, two nodes are needed at the input layer. (Feiyang Chen&Nan Chen&Hanyang Mao&Hanlin Hu, 2018)

In the MNIST collection, a picture corresponds to a single data point. These images are typically the dimensions of the image are 28 pixels by 28 pixels, with 28 pixels intersecting each axis on both sides, and can be found in MNIST datasets. As a result, it is necessary to look at 784 pixels in one image from the MNIST database. One of these photos can be examined by the 784 nodes in our neural network's input layer. (Hamza BenTarif&Prof. Dr. Ali Okatan, 2023)

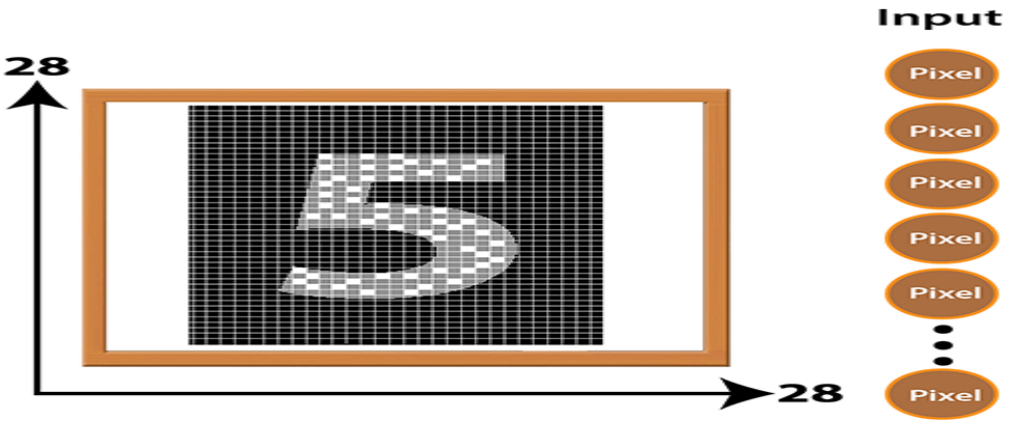


Figure 5. MINIST Picture Example 2

Due to the addition of more input nodes and the expanded range of categories from 0 to 9, our dataset has become significantly more complex compared to the datasets we previously analyzed. To classify this data set correctly and clearly, it is very important to have a deep neural network with a hidden layer. (Hamza BenTarif&Prof. Dr. Ali Okatan, 2023)

Table 5 summarizing the architecture of deep neural network:

Layer	Input layer	Hidden layers	Output layer
Number of Nodes	784	multiple layer	10

Within the framework of our deep neural network, we have an input layer consisting of 784 nodes, which corresponds according to each image's pixel count. This is followed by several hidden layers that process and propagate the input values. Finally, we have 10 nodes in the output layer, each representing one of the handwritten numbers from 0 to 9. During the network's operation, the input values are fed through the layers, and the node in the output layer with the highest activation value is selected, thereby determining the identified character or number.

III. LITERATURE REVIEW

A. Image Classification with MNIST Dataset

RegularNets are characterized by the interconnection of all neurons, which is their main structural feature. In a manageable layer, RegularNets typically consist of 784 neurons 28 x 28 x 1, suitable for grayscale images with 28 by 28 pixels. However, when dealing with color photographs with higher resolutions, such as 4K Ultra HD, the number of unique neurons connected in the first layer becomes significantly larger when dealing with 4K Ultra HD images,

the number of interconnected :

"Total Pixels = Horizontal Pixels × Vertical Pixels × Color Channels "neurons would amount to an impressive 26,542,080. This calculation is derived by multiplying the dimensions of the image, which are 4096 pixels horizontally and 2160 pixels vertically, by 3, representing the three color channels red, green, and blue. Each of these interconnected neurons plays a crucial role in processing and analyzing the vast amount of visual information contained within high-resolution images, contributing to the intricate workings of image recognition and processing algorithms. which is an overwhelming number and not practical to handle. Therefore, it can be concluded that RegularNets are not scalable for image classification due to this limitation.

Table 6 Image Classification with MNIST Dataset:

Property	Number neurons	Neuronst	Total neurons	Dimensions
Value	784	28 x 28 x 1	26,542,080	4096 pix

The layers in a convolutional neural network can vary greatly. However, convolution, pooling, and being totally connected are the most important layers. So let me simply introduce these layers before I begin to develop them.

1. Convolutional Layers:

In our data set, we can show the features of the image through the convolutional layer, and through this method, we maintain the relationship between the different elements of the image, as we conclude that the pixels are mainly associated with neighboring pixels. The convolution method includes applying smaller filters to the image, which allows for reducing effective scale while maintaining its relationship to the pixels. Let's consider an example to better understand the concept of convolution in image processing. If we apply a 3x3 filter with a stride of 1x1 (moving one pixel at a time) to convolve a 5x5 image, the resulting output will have a size of 3x3. This means that the complexity of the image is reduced by 64%. The reduction in complexity indicates that the convolution operation effectively captures the essential features of the original image within a smaller representation. This process allows for efficient analysis and manipulation of images while preserving important visual information.

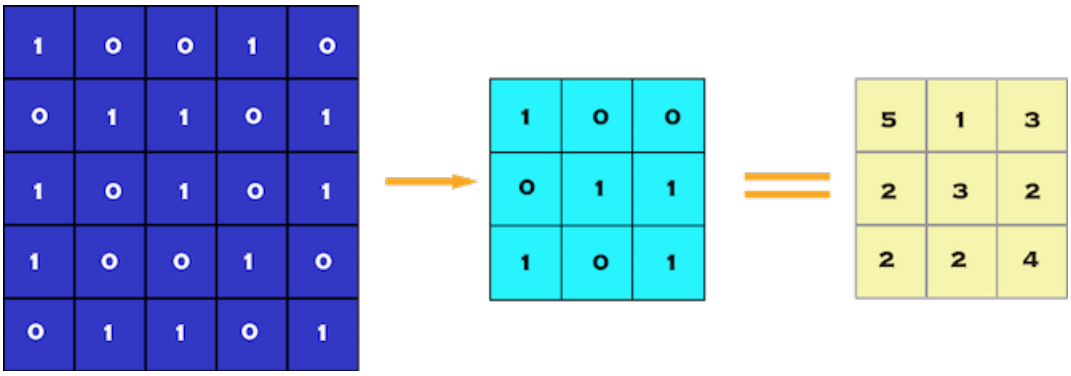


Figure 6.5 x 5-pixel image convolution with a 3 x 3-pixel filter

2. Pooling layer:

Usually, when developing CNN, a pooling layer is placed after each convolution layer. to lessen the computational complexity, the number of parameters, and the spatial dimension of the representation. Additionally, layer pooling helps get rid of overfitting. Basically, we choose the pooling size to reduce the number of parameters by defining the maximum, average, or aggregate values inside these pixels. Max Pooling is one of the most commonly used pooling techniques, as shown in the example below:

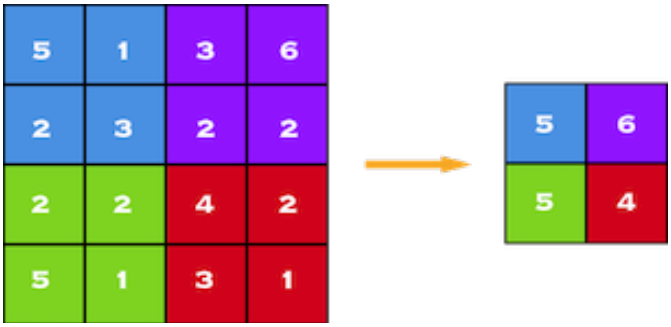


Figure 7. Max Pooling by 2 x 2

3. A group of completely connected layers:

Actually, every variable within our regular network is connected to another variable to establish its actual relationship. and effect on the labels. Since convolution and pooling layers dramatically reduce our time-space complexity, we can eventually construct a fully connected network to classify our photographs.

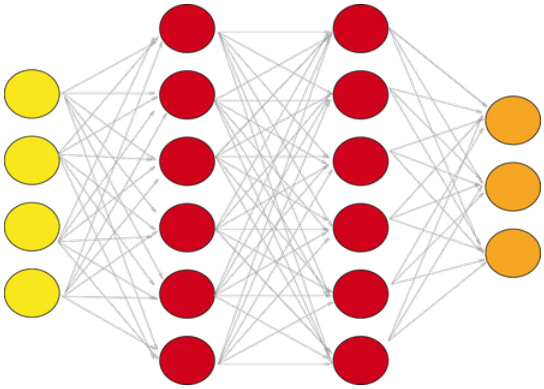


Figure 8. a layer that is entirely connected and has two hidden layers

B. Developing a basic neural network to classify images from the MNIST

It is known that neural networks have received a lot of attention over the past years, but scientists know that the model family has at least one major flaw: results are challenging to comprehend. One of the reasons why many people think of neural networks as mysterious is the difficulty of understanding the structure of any specific neural network.

The MNIST dataset is widely recognized as a popular benchmark for learning the usage of neural networks. Interestingly, even though some of the digits in the dataset can be quite challenging, many basic networks exhibit impressive performance on this task. This has led to a humorous comparison, stating that MNIST is the deep learning equivalent of the "hello world" programming exercise,

highlighting its ubiquity and simplicity in the field. (Bettilyon, 2018)

C. Recognition of Handwritten Digit using Convolutional Neural Network (CNN)

1. The Architecture of the Proposed Model:

Convolutional neural networks CNN, have recently become a popular choice for machine learning tasks, such as image classification, object detection Image Net, and face recognition. This is because CNNs capable of learning features, and images in a manner comparable to how humans do. For example, a CNN can learn to recognize the edges of a face or the curves of a letter.

One rapidly realizes how important CNNs are for photo classification when using a neural network to learn deep learning. A convolutional neural network is a special type of neural network that is layered in order to recognize direct visual patterns from pixel images with very little pre-processing. (Md. Anwar Hossain&Md. Mohon Ali , 2019)

Most convolutional neural network (CNN) architectures share the same basic design principles. These principles include:

1) In the input layer, apply each convolutional layer one by one.
2) Sometimes, samples are taken with a few spatial dimensions, and maximum pooling is used.
3) Increasing the number of feature map.
4) And also the fully connected layer with activation and loss functions, such as discrete entropy and softmax.

The most important work of the CNN layer is the convolutional layer and the aggregation layers, which are fully connected. Therefore, before we define our proposed model, we will explain these layers in a simple way.

Convolutional layers are the foundation of CNNs. They extract features from the input data by applying a filter to each local region of the input. The filter is a small matrix of weights that is learned during training.

Pooling layers are used to reduce the spatial dimensions of the output from

the convolutional layers. This helps to reduce the number of parameters in the network and to make the network more robust to noise.

Fully connected layers are used to classify the input data. They take the output from the convolutional layers and connect it to a set of output nodes, each of which represents a class.

The convolutional layer is the first to be able to extract features from an image. Due to the fact that pixels only have relationships with their close neighbors and immediate neighbors, convolution allows us to keep the relationships between different elements of a picture. Convolution is a technique that allows for reducing the size of an image while preserving the pixel relationships. By applying convolution to a 5x5 image with a 3x3 filter and a stride of 1x1 (shifting one pixel at each step), we can effectively shrink the image while maintaining the essential information encoded in the original pixels. The complexity is decreased by 64%.

Table 7 this table describes the application of convolution to image :

Property	İmage size	Filter size	Stride	Complexity reduction
Value	5x5	3x3	1x1	64%

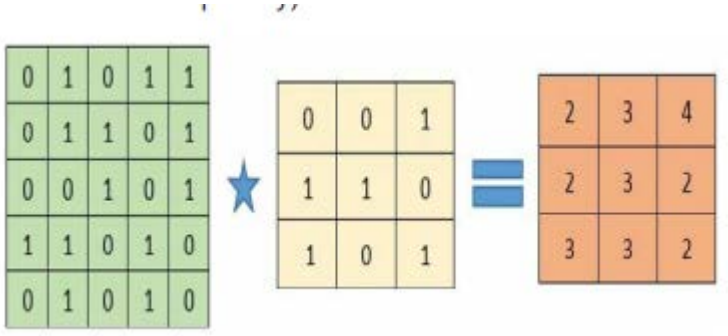


Figure 9. Convolution operation

Pooling layers are commonly used in convolutional neural networks (CNNs) to reduce the size of feature maps. They also assist in mitigating overfitting. To minimize the number of parameters, we select a pooling size and aggregate values within the chosen pixels, either by taking the maximum, average, or sum. Max Pooling is a widely employed pooling technique, and it can be illustrated as follows:

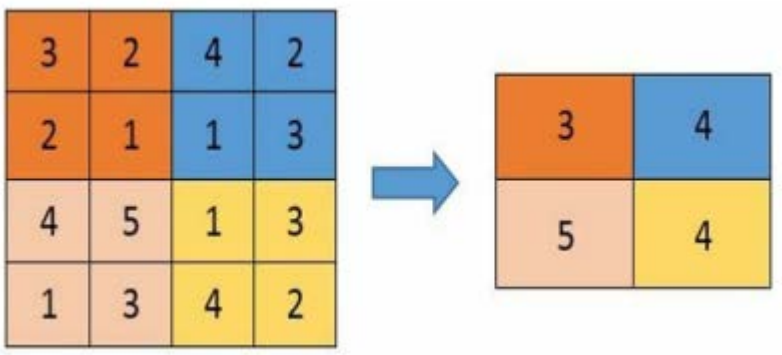


Figure 10. ***Max-pooling operation***

A completely connected network is any architecture where each parameter is connected to another to determine how each parameter relates to and affects the labels. We can ultimately construct a fully linked network to recognize the images due to convolution and pooling layer reduction of time-space complexity.

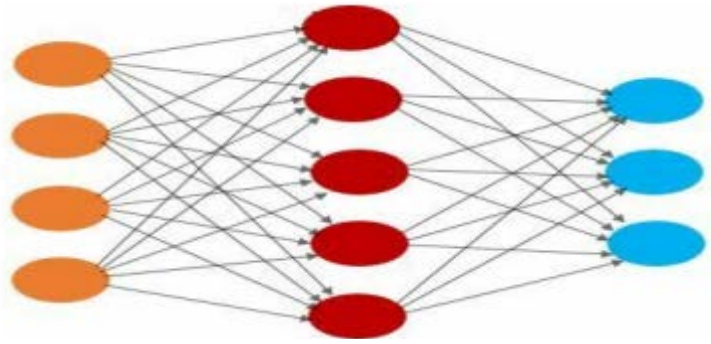


Figure 11. Fully connected layers

We now expect to give an overview of our proposed convolutional network, which has similarities with the handwritten number recognition architecture 1, 6, 8, 10, 1, and from here we made performance improvements for a different set of filters and neurons, as well as seven activation functions.

2. Explanation of the Model:

A differentiable function changes one volume of activations into another. in each layer of a straightforward convolutional neural network. We mainly use three types of layers to build the network. What is being discussed are convolutional, pooling, and fully connected layers. We will stack these layers to create our network design.

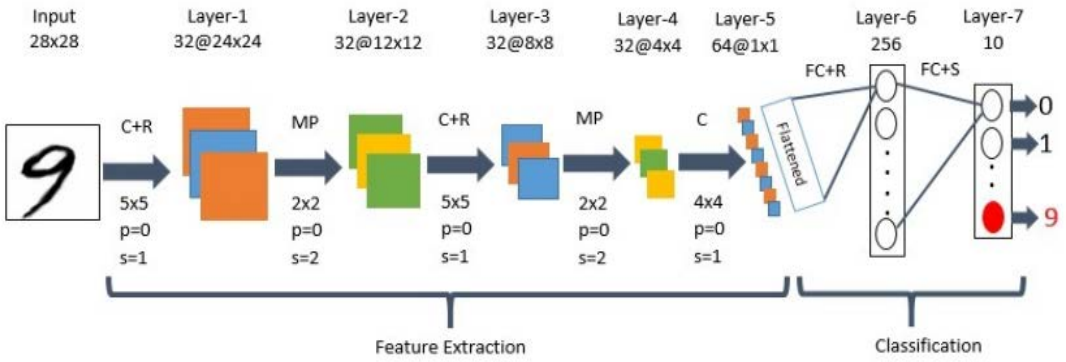


Figure 12. The architecture of our proposed CNN

The proposed CNN model is illustrated in Figure 12. Before providing the data to the model, some preliminary processing, such as shrinking and leveling the pixel values of the images, is required.

In our CNN architecture, the first layer is a convolutional layer that employs the Rectified Linear Unit (ReLU) activation function. This activation function enhances the nonlinearity of the model and aids in learning complex patterns from the input data. The input to this layer is a preprocessed image with dimensions of n * n, specifically 28 * 28 in our case. By applying the ReLU activation function within this convolutional layer, the network can efficiently extract important features and capture meaningful representations from the input image, contributing to the subsequent stages of the convolutional neural network's analysis and classification process.

There are 32 filters, a padding of 0 (around both of the image's edges), a stride of 1, a convolution filter size (f*f) of 5*5, and a total of 32 filters. After completing this convolution, we get feature maps with a size of 32@24*24, which is equivalent to the number of filters used. The size of 24 is obtained by applying the formula ((n + 2p - f) / s) to the given values, where n represents the initial size (28 in this case), p is the padding (0 in this case), f denotes the filter size (5 in this case), and s represents the stride (1 in this case). Applying this formula results in ((28 + 2*0 - 5) / 1) + 1 = 24. After obtaining the size of 24, the Rectified Linear Unit (ReLU) activation function is utilized to activate each feature map. The ReLU activation function introduces non-linearity, allowing the neural network to capture complex patterns and extract meaningful representations from the feature maps obtained from the previous layers.

Layer 2 has the most pooling potential. This layer receives a size 32@24*24 input from the layer above. The pooling size is 2*2, the padding is 0, and the stride is 2. With the help of this maximum pooling technique, 32@12*12-inch feature maps are produced. We obtain the same number of feature maps (12 using the same calculation, ((n+2p-f)/s)+1) by independently performing max pooling on each feature map. This layer doesn't have an activation function.

Layer three is the second convolutional layer with the ReLu activation function. This layer receives an input of size 32@12*12 from the layer above. There are 32 filters; there is no padding; there is one stride; and there is a 5*5 filter. After doing this convolution, we have feature maps that are 32@8*8 in size. Then, ReLu is used to activate each feature map.

Layer 4 is the second max pooling layer. This layer receives a size 32@8*8 input from the layer above. The pooling size is 2*2, the padding is 0, and the stride is 2. We end up with a 32@4*4 feature map after this max pooling procedure.

ReLu activation is not present in Layer 5, the third layer of convolution. The layer above provides an input of size 32@4*4 to this layer.. There are 64 filters, 0 filters for padding, and 1 filter for stride. The filter measures 4 by 4. Following the convolution, we have feature maps that are 64@1*1 in size. This layer becomes a fully connected layer when it is flattened, producing a 64-dimensional vector.

Layer 6 is the highest connected layer. From a 64-dimensional input vector, this layer creates a 256-dimensional, one-dimensional vector. It has the capacity to make ReLu active.

Layer 7 is the topmost layer of the network. Additionally, it offers total connection. Each of the ten values in the vector of size 10 that this layer produces correlates to a class score, such as one of the ten categories in the MNIST dataset. The class scores will be calculated by this layer. It has a softmax activation function for final outputs.

3. Implementation:

To implement our CNN design, we'll use MatConvNet. In MATLAB, MatConvNet is a convolutional neural network (CNN) implementation. And also by increasing the linear convolution arithmetic technology with filter banks, feature

aggregation, and second operations, which leads to the provision of basic components of CNN as MATLAB functions that are easy to use. By providing efficient CPU and GPU processing in this way, MatConvNet makes it easier to quickly create new CNN architectures. and makes it possible to train complex models on huge datasets like Image Net ILSVRC.

The most complex structure available today, especially for the task of image classification, is the CNN convolutional neural network, whose task is to locate known MNIST numbers. we propose a 2-D Convolutional Neural Network (CNN) model supported by MatConvNet. The complete workflow may consist of getting the data ready, building and training the model, testing and evaluating the model, and saving the model to disk for later use.

Preparing the data is the first stage in our process. The training and test sets of data must be organized, integrated, labeled, and resized to the appropriate dimensions before the network can be built. Labels, extra (meta)data, and normalized data (single precision and zero mean) are all saved in the dataset. The creation and compilation of the model is the second phase. In order to construct the CNN, we must first initialize the MatConvNets DagNN (Directed acyclic graph neural network) network and give essential startup settings.

In this case, the number of samples that we will use in the CNN training phase has been determined. CNN will process all the training data used, but the process will be in the form of batches of the specified size only. Also, the batch size is used for computing efficiency, and its values depend on the devices that enable it. The user can reach it, this epoch knows that it is a successful forward pass and a successful backpass through the network, and sometimes it is good to put high values on it after we can reduce the value at least once if the convergence in a particular case (the chosen epoch) in the network is appropriate.

Finding its best value will be an empirical process until one employs more powerful approaches like batch normalization. In our test, we chose a learning rate of 0.001, a batch size of 40, and an epoch count of 8.

Table 8 These are the hyperparameters chosen for the training process

Hyperparameter	Learning rate	Batch size	Epoch
Chosen Value	0.001	40	8

As shown in Figure 13, we can create each layer of our CNN independently. At the end of each epoch, we will call the objective by log loss and the error layer by classification loss, which will provide a graphical representation of the training approximation and validation. After building for network layers, we initialize the weights using a Gaussian distribution from MatConvNet.

```
% Layer-1 and Layer-2 ( Convolution + ReLu and Max Pooling )
net.addLayer('conv1', dagnn.Conv('size', [5 5 1 32], 'hasBias', true, 'stride', [1 1], 'pad', [0 0 0 0]), {'input'}, {'conv1'},  {'conv1f'  'conv1b'});
net.addLayer('relu1', dagnn.ReLU(), {'conv1'}, {'relu1'}, {});
net.addLayer('pool1', dagnn.Pooling('method', 'max', 'poolSize', [2 2], 'stride', [2 2], 'pad', [0 0 0 0]), {'relu1'}, {'pool1'}, {});

% Layer-3 and Layer-4 ( Convolution + ReLu and Max Pooling )
net.addLayer('conv2', dagnn.Conv('size', [5 5 32 32], 'hasBias', true, 'stride', [1 1], 'pad', [0 0 0 0]), {'pool1'}, {'conv2'},  {'conv2f'  'conv2b'});
net.addLayer('relu2', dagnn.ReLU(), {'conv2'}, {'relu2'}, {});
net.addLayer('pool2', dagnn.Pooling('method', 'max', 'poolSize', [2 2], 'stride', [2 2], 'pad', [0 0 0 0]), {'relu2'}, {'pool2'}, {});

% Layer-5 ( Convolution acts as Fully Connected and Flattened )
net.addLayer('conv3', dagnn.Conv('size', [4 4 32 64], 'hasBias', true, 'stride', [1 1], 'pad', [0 0 0 0]), {'pool2'}, {'conv3'},  {'conv3f'  'conv3b'});

% Layer-6 ( Fully Connected + ReLu )
net.addLayer('fc1', dagnn.Conv('size', [1 1 64 256], 'hasBias', true, 'stride', [1 1], 'pad', [0 0 0 0]), {'conv3'}, {'fc1'},  {'conv5f'  'conv5b'});
net.addLayer('relu5', dagnn.ReLU(), {'fc1'}, {'relu5'}, {});

% Layer-7 ( Fully Connected + Softmax )
net.addLayer('classifier', dagnn.Conv('size', [1 1 256 10], 'hasBias', true, 'stride', [1 1], 'pad', [0 0 0 0]), {'relu5'}, {'classifier'},  {'conv6f'  'conv6b'});
net.addLayer('prediction', dagnn.SoftMax(), {'classifier'}, {'prediction'}, {});

% Loss
net.addLayer('objective', dagnn.Loss('loss', 'log'), {'prediction', 'label'}, {'objective'}, {});
net.addLayer('error', dagnn.Loss('loss', 'classerror'), {'prediction','label'}, 'error') ;
```

Figure 13. CNN layers in MatConvNe

Validation data is only used to test the CNN's performance on new, similar input. After the CNN is trained, it is saved and can be used for testing.

Each CNN training session produces two plots: the error plot and the objective plot. The error plot shows the CNN model's validation error during training, the objective plot shows the loss of the DagNN network. MatConvNet minimizes the loss of the DagNN network. The error plot can be used to compute statistical inference.

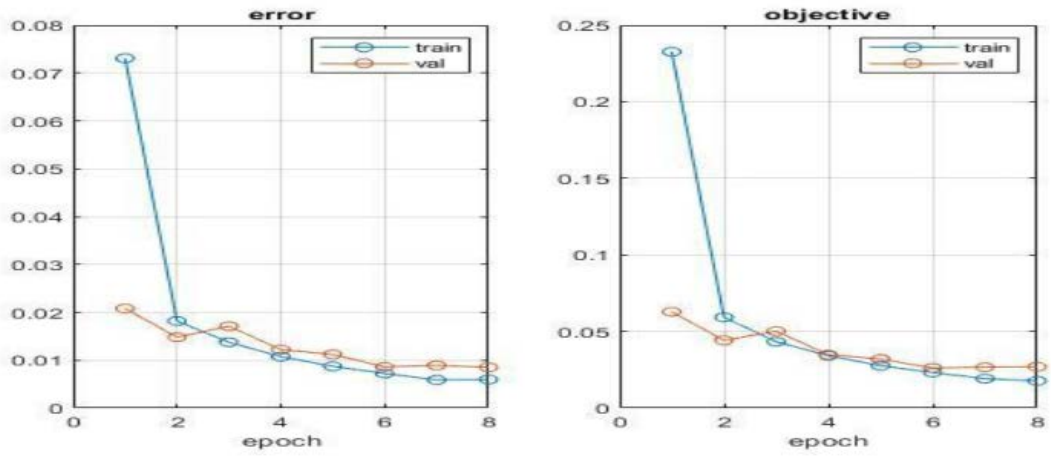


Figure 14. Classification loss (error) and log loss (objective) during training

Finally, using the testing data, we can evaluate our model. The samples below illustrate the categorization outputs from our model during testing.

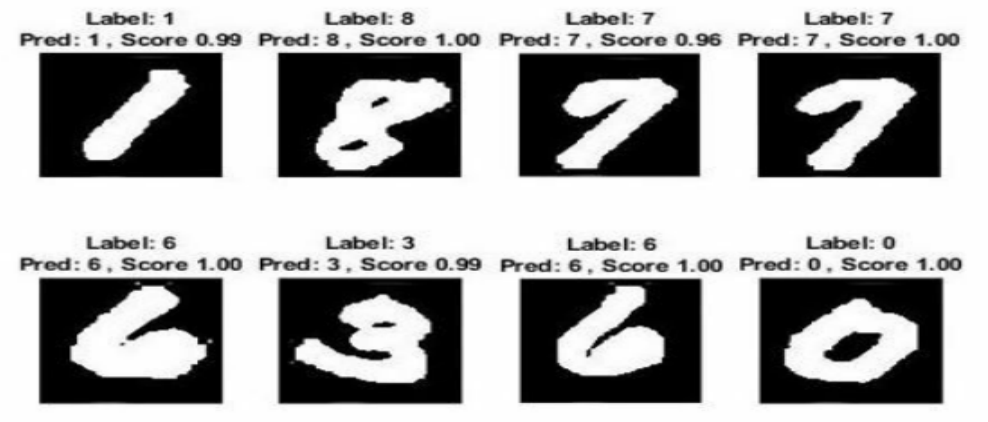


Figure 15. Some correct recognized output

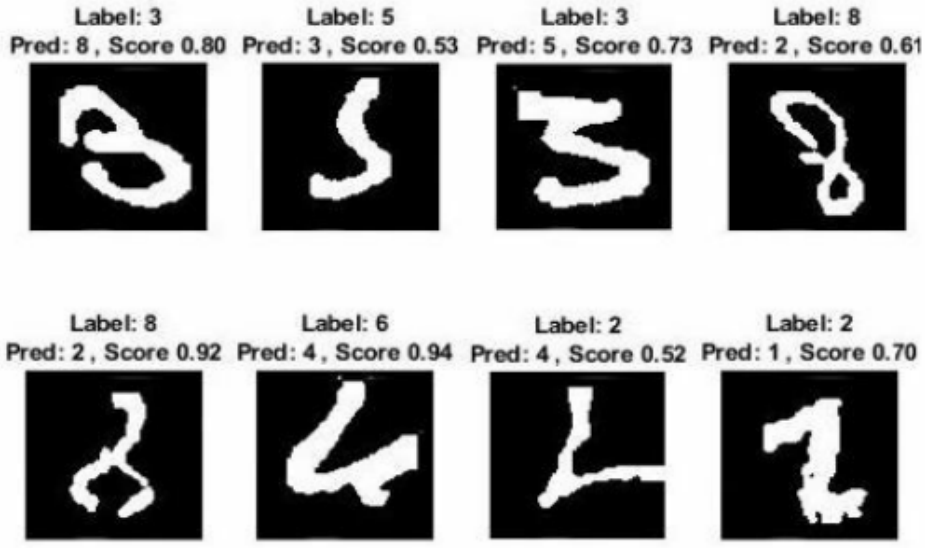


Figure 16. Some wrong-recognized output

It is possible to identify test cases with incorrect classifications. However, we believe that the majority of these cases can be resolved by expanding the training set and the normative number pattern. Additionally, image compression and image sharpness issues can cause pixel loss, which can lead to misdiagnosis.

The final step in the process is to save the trained model to a file so that it can be used again in the future. The model can be saved in a Matlab file format, which makes it easy to reuse in other applications. (Martin, 2019)

D. Creating a neural network to recognize handwritten digits from the famous MNIST dataset

In this study, I will build and develop a neural network. That can identify handwritten digits from the popular MNIST dataset.

I'll test out two different networks for this assignment. The first will be a multi-layer perceptron (MLP), a typical kind of feed-forward neural network with fully connected layers of weights, and the second will be a convolutional neural network (CNN), which makes use of the input images' intrinsically two-dimensional spatial architecture.

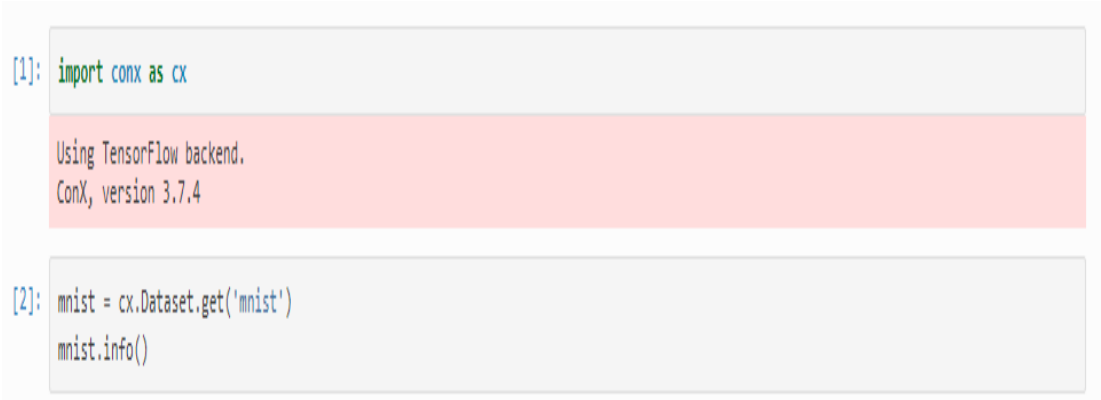


Figure 17. Dataset, MNIST

The MNIST dataset contains 70,000 handwritten digits (from 0 to 9) that have been size-normalized and centered on a pixel grid. Each image is represented by a 28 by 28 by 1 array of floating-point values, where 0 (black) and 1 (white) are the grayscale intensities. The target data consists of one-hot binary vectors of size 10 corresponding to the digit classification groups 0 through 9. Here are a few illustrations using MNIST images:

Table 9 this table describes the dataset :

Dataset Description	Number handwritten	Digit range	Image Dimensions	Grayscale range	Target Data
Value	70.000	0 to 9	28*28*1	0(Black)to1(white)	1-hot binary

Table 10 this table summarizes the information about the "MNIST" dataset:

Information name	Dataset name	Length	Input summary	Target summary
Dataset	MINIST	70,000	Shape(28x28x1) range (0.0,1.0)	Shape (10,) range (0.0,1.0)

A more condensed, concise summary is also available to print:

```
MNIST:
Patterns    Shape                Range
=================================================================
inputs      (28, 28, 1)          (0.0, 1.0)
targets     (10,)                (0.0, 1.0)
=================================================================
Total patterns: 70000
   Training patterns: 70000
   Testing patterns: 0
```

Figure 18. MNIST Summary

Let's now examine some input patterns. When you type (mnist. view (mnist. inputs [0])) a huge array of nested numbers is returned, it is not an ideal way to see patterns, and here we must use CoX's feature instead: (Birbeck, 2018)

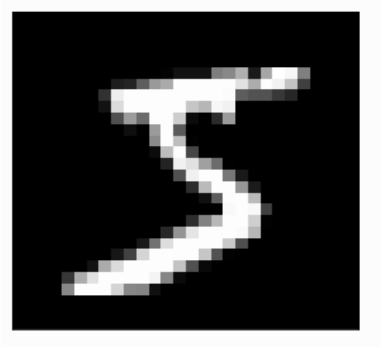


Figure 19. cx. view (mnist. inputs [0])

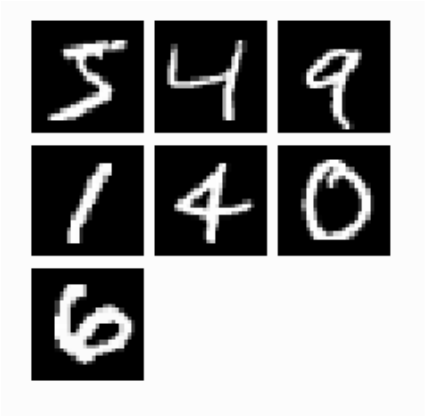


Figure 20. cx.view (mnist.inputs [0, 2, 4, 77, 150, 88, 9000])

In addition to inputs and targets, each dataset also has a third property called labels that maintains a collection of strings indicating the classification category of each input:

cx.view(mnist.inputs[0:5]) print(mnist.labels[0

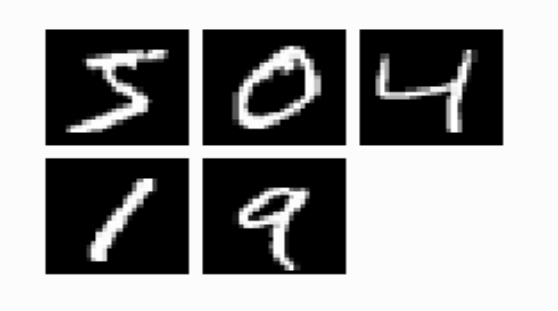


Figure 21. mnist.inputs[0:5]) print(mnist.labels[0:5])

You can use the labels property to choose particular input patterns. For instance, we could design a list that contains all the index numbers, for example, twos index, and then we see the first twenty of that list to display only the two in the data set.

Let's shuffle the dataset at random to get it ready for network training: (Staff, 2020)

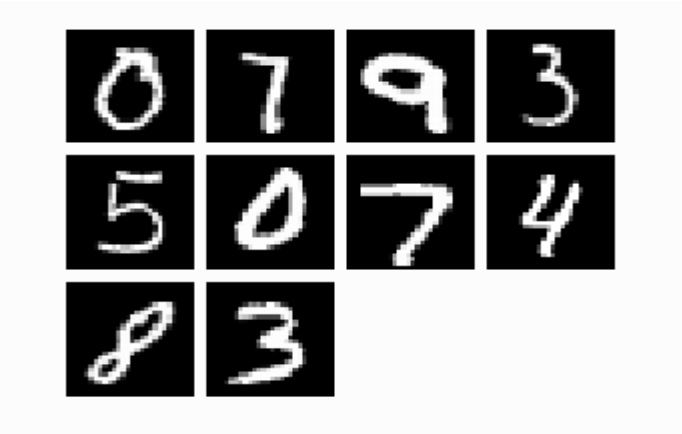


Figure 22. mnist.shuffle() cx.view(mnist.inputs[0:10])

the dataset's total number of images and the number of splits are the two numbers that make up the output of the(MNIST. split())function. In this instance, the output is (70000, 0), indicating that 70,000 photos are included in the dataset and that no splits have been performed.

The code MNIST.split(10000) splits the MNIST dataset into two parts: The data set contains seventy thousand images, of which sixty thousand images are within the training set and ten thousand images are testing set, The code MNIST.split() then verifies the split by printing the number of images in each part. The output of MNIST.split() confirms that the dataset has been split correctly, with 60,000 images in the training set and 10,000 images in the testing set.

10,000 pictures from the MNIST dataset are set aside for testing. Starting with index 0, we can refer to the training and testing sets independently using the attributes' train_inputs, train_targets, train_labels, and test_inputs, test_targets, and test_labels'. The properties inputs, targets, and labels refer to all 70,000 input images, regardless of the current split.

train_inputs: The 60,000 photos that were used to train the machine-learning model are referred to as training inputs.

train_targets: The labels for the training inputs are the training targets. The digit that the image symbolizes is represented by the index of the one in the labels, which are one-hot vectors.

train_labels: The string representations of the numbers that the training images represent are known as training labels.

10,000 photos are utilized as the test inputs to evaluate the machine learning model.

test_targets: The labels for the test inputs and targets. The digit that the image symbolizes is represented by the index of the one in the labels, which are one-hot vectors. (Staff, 2020)

targets: The input labels are the targets. The digit that the image symbolizes is represented by the index of the one in the labels, which are one-hot vectors. the string representations of the digits that the test labels are based on. digits that the images depict are represented by strings known as labels.

For consistency, the grayscale MNIST images are treated as images of depth 1, with a shape of (rows, columns, 1). This can be verified by using the cx. shape function on the first input image.

1. A multi-layer perceptron network for MNIST classification:

We are now ready to build a simple feedforward neural network to learn the MNIST data. The network will have an input layer of size 28x28x1, two hidden layers of size 30 units each, and an output classification layer that matches the geometry of the input patterns.

Before we can feed a 2D input image into the hidden layers, we must first flatten it into a linear vector of size 784 using a special FlattenLayer. The softmax activation function will be used in our output layer, which will contain 10 units total – one for each classification of the digits (from 0 to 9).

In order to utilize ConX, it is necessary to begin by creating a Network object and assigning it the arbitrary name "MNIST_MLP". This should be done before

adding individual Layers to the network.

Subsequently, the connect() command can be employed to establish the connections between the layers:

In the code you provided, the following layers are added to the network:

Input: An input layer with shape (28, 28, 1).

flat input: A Flatten Layer that flattens the input layer into a 1D vector.

hidden1: A hidden layer with 30 units and the ReLU activation function.

hidden2: A hidden layer with 30 units and the ReLU activation function.

Output: An output layer with 10 units and the Softmax activation function.

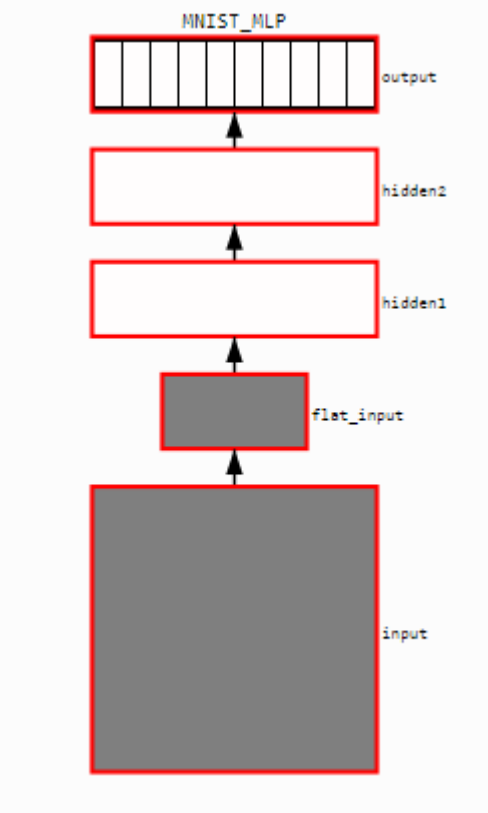
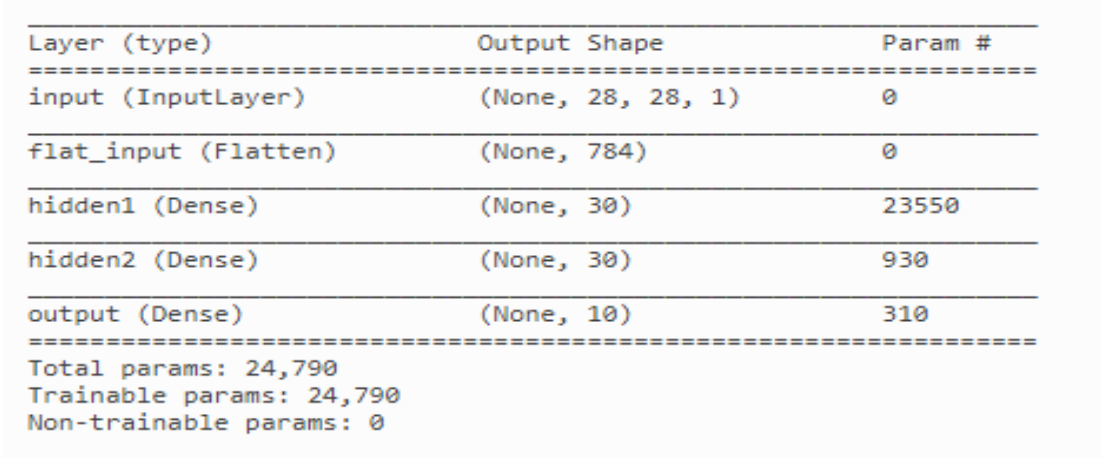


Figure 23. MINIST MLP

The network must then be compiled and trained, with a learning algorithm (the optimizer) The error function known as the finite loss function, usually the error function named categorical_crossentropy is an optimal choice for the classification problem and the SoftMax function. Here are learning algorithm is originally a standard stochastic gradient, and it can also optionally specify the momentum and learning rates to be applied to apply.



Layer (type)	Output Shape	Param #
input (InputLayer)	(None, 28, 28, 1)	0
flat_input (Flatten)	(None, 784)	0
hidden1 (Dense)	(None, 30)	23550
hidden2 (Dense)	(None, 30)	930
output (Dense)	(None, 10)	310

Total params: 24,790
Trainable params: 24,790
Non-trainable params: 0

Figure 24. net.compile(error='categorical_crossentropy', optimizer='SGD', lr=0.3, momentum=0.1)

net.summary()

network summary displays the details for each layer, as well as the total number of network parameters. The network's underlying Keras model is directly accessible through the model property, although ConX users typically do not need to concern themselves with the lower Keras level. (Neela Chattyopadhyay&Hrittika Maity&Bipsa Debnath,, 2013)

The propagate()method can be used to manually broadcast an input pattern across the network. For example, to ask the network to classify input pattern #0,this code will first display the input pattern, and then it will propagate the pattern through the network. The output of the network will be a 10-dimensional vector each element can represent or contain the possibility of a particular input pattern, Because the network has not yet been trained, it is good to note that the output values for each class are close to zero to ten, and after the training process, we expect that at least one of the output values corresponding to the output classification class will outperform the rest of the values.

To gain clearer insights into the activation of each layer, an interactive dashboard can be utilized, offering a user-friendly approach to observing the behavior of the network. This dashboard simplifies the process of examining by examining the network's output and comparing it to the corresponding target classification for various input patterns from either the training or testing set, we gain insights into the performance and accuracy of the model. This evaluation process allows us to assess how well the network is able to correctly classify different input

patterns. To access the detailed analysis, you can click on MNIST_MLP located at the top of the dashboard. This feature provides a comprehensive overview of the network's predictions and enables further exploration of its behavior and predictive capabilities.

, a panel of settings can be accessed, allowing customization of the network display's appearance. For example, the Dataset pull-down menu enables the selection of either the training set or the testing set, labeled as Train or Test, respectively, to analyze the corresponding images.

By employing the plot_layer_weights function, we can visually examine the weights associated with specific units in the neural network. To illustrate, the following command showcases the weights originating from the input layer and connected to units 0, 1, and 2 in the first hidden layer. The weights displayed take the form of a 28 by 28-pixel array, where each pixel represents a different weight from the input layer. The wrange keyword is utilized to define the minimum and maximum weight values for the color coding scheme. Since the network has not undergone training, the weights are initially assigned as modest random numbers near zero.

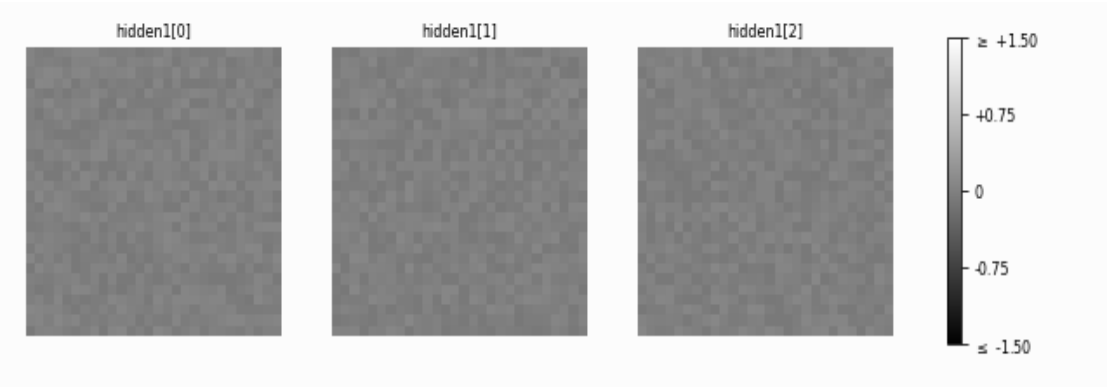


Figure 25. net.plot_layer_weights('hidden1', units=[0,1,2], vshape=(28,28), wrange=(-1.5, 1.5))

Here we will start training the network for thirty elements of size thirty-two, which means that for each batch of thirty-two images from the training set, images from the network are fed.

The SGD approach will appropriately update the weights of the network. Following the processing of all 60,000 training images, in the first training period, a second batch of thirty-two images is provided. This full cycle will be repeated for 30

epochs.

Table 11 this table outlines the training process:

Training Configuration	Number of training elements	Number of training image	SGD	Epochs
Value	32	60.000	yes	30

As training occurs, the accuracy on each of these sets is shown on the right graph, while the left graph displays the network's error (loss) on the training and testing/validation sets. Simply expressed, accuracy is the proportion of input photographs that the network correctly recognizes. If the output value with the greatest value on the output layer corresponds to the goal categorization, the classification is considered accurate.

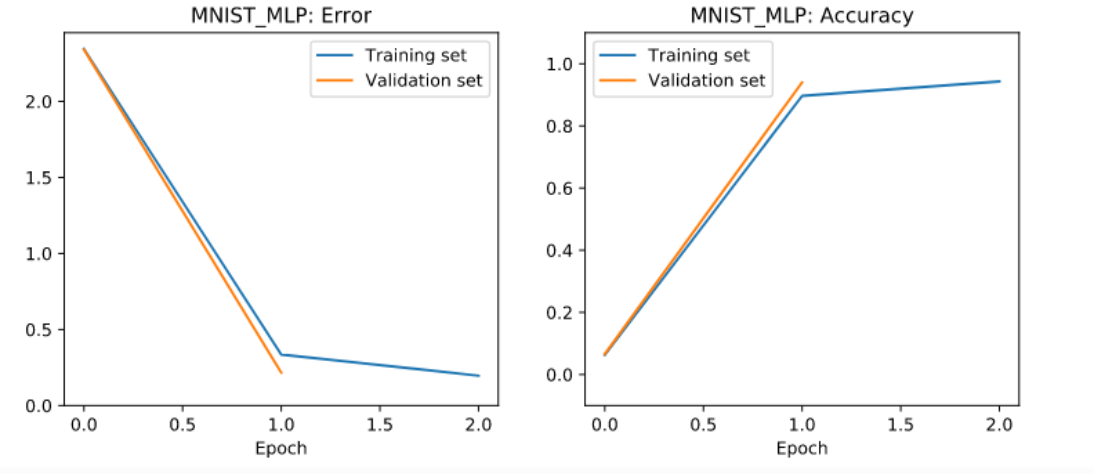


Figure 26. MINIST MLP

Table 12 Training results

Epochs	2
Training error	0.19710
Accuracy	0.94321

2. A Convolutional Network for MNIST Classification:

It is interesting to note that visual system neuroscience serves as a broad source of inspiration. This influence can be observed in convolutional networks (CNNs), where each unit within the network establishes connections with a small subset of preceding units, Referred to as the unit's optical field, these connections are made in a two-dimensional topology. In order to leverage spatial information effectively, the kernel size, or size of the visual field, and the number of independent characteristics that each convolutional layer needs to learn to play crucial roles. In

comparison to networks with fully connected layers, the units within the convolutional layer are responsible for learning features that are shared recursively across the entire layer.

The pooling layer, or partial downsampling, is used to reduce the amount of data that is sent over the network and is a type of second layer found in the CNN network. Each aggregation layer receives input from a certain number of units in the layer below, and the results are recorded after it has completed its function, such as the maximum or average input. A coarser resolution of the information from the preceding layer is created. the network as a whole becomes less sensitive to slight changes in location. can be used to manually broadcast an input pattern across the network. For example, to ask the network to classify input pattern #0,this code will first display the input pattern, and then it will propagate the pattern through the network. The output of the network will be a 10-dimensional vector each element can represent or contain the possibility of a particular input pattern, Because the network has not yet been trained, it is good to note that the output values for each class are close to zero to ten, and after the training process, we expect that at least one of the output values corresponding to the output classification class will outperform the rest of the values.

To create a CNN for MNIST classification, we will utilize two convolutional layers, each consisting of 5x5 kernels. These layers will be succeeded by a pooling layer employing 2x2 kernels, which computes the maximum value of the inputs. The second convolutional layer will be responsible while the first convolutional layer will concentrate on learning 16 very low-level characteristics. for learning 32 higher-level features. These traits will be passed into a hidden layer once they have been flattened. Finally, an output classification layer utilizing softmax will be employed for the final classification process.

Table 13 This table outlines the CNN architecture for MNIST classification

Layer	Input	Convolution1	max pooling1	convolution2	max pooling2
kernel size	-	5*5	2*2	5*5	2*2
Number of kernel	-	16	-	32	-
purpose	28*28*1	low level	down sample	higher level	down sample

Keep in mind that the convolutional layers of the network each have a dropout value of 20%. Dropout is a strategy that aids in enhancing a network's capacity to generalize what it has learned by decreasing a network's sensitivity to noise and insignificant correlations that may be present in the training data. Each training cycle will have a new random subset of units in a dropout layer being turned off (set to zero activation). In this case, about twenty percent of the units will be disconnected. As for leakage, it only occurs during the training process once the network learns and each unit participates in the input data classification process. Finally, an output classification layer utilizing softmax will be employed for the final classification process.

IV. METHOD AND LIMITATIONS

A. Method:

When we do any project, we must use at least one methodology to search for data and information that is relevant to a particular topic.

In my research, I will use the qualitative method, or secondary research, in order to be able to collect the pieces of information that I already have, but the main goal of this method is to find different pieces of information and examine them by collecting them from various and different sources, including websites, articles, and magazines. Everything is available, of course, compared to other methods, and this data collection process is very important because it helps us obtain answers and solutions to many problems and questions that enhance the quality of many products.

When statistical data is solid, quantitative procedures are applied. These are typically utilized in economics essays where the descriptive method can be used. The social sciences and humanities also make extensive use of quantitative techniques. It's difficult to conceive of a sociological conversation that won't include some statistics. Popular quantitative techniques in philology, political science, and history include content analysis. By using content analysis, you may track how frequently different issues are linked together in political speeches and get some intriguing conclusions.

In many different study fields, qualitative approaches are frequently used. Case studies are the most popular qualitative method, but surveys and context analysis are also well-liked.

Data were analyzed based on quantitative and quanitative methods in analyzing scientific research depending on as digital library, internet, books and publications, etc.

The aim of data collection is to collect the necessary information and measure it in order to obtain it through the use of many variables, and this process is

important for several reasons, the most important of which is that it helps us to obtain solutions and answers to many problems and questions and also helps in the decision-making process.

It also increases the quality and improves the decisions that have been taken. As for the tool used in this paper, it is qualitative data that uses websites to obtain part of the information about how to Develop a basic neural network to classify images from the MNIST dataset.

With regard to the size of the sample used in this thesis, it was concluded that the default sample for this research is to try to reach the largest possible number of people interested in this topic.

The methodology section reiterates the research questions by clarifying the definition of neural network, clarifying the MINIST dataset as well as how to create a neural network to recognize handwritten digits from the famous MNIST dataset, clarifying how to develop a basic neural network to classify images from the MNIST and other else.

This research paper explains the neural network which it is can create an image by training and then un-training an image recognition network. Additionally, it has the capacity to combine the aesthetic qualities of two distinct photos into a single output with the same features. The optimum option for image output is a CNN with low complexity and high output.

By making many hypotheses, we note that We note that many neural networks contain many weights, ranging from tens of thousands to millions. Through the training process, these weights are adjusted to reduce the error rate. At the same time, it may be difficult to understand the exact reasons for the superiority of one neural network over another because of the complex interactions between many In addition; the nature of these complex interactions poses a challenge for designing high-quality neural network topologies.

Depending on many previous studies, we note that CNNs can be used for a variety of tasks, including handwriting digit recognition. Handwriting digit recognition is the ability of a machine to identify a handwritten digit from an image. This is a challenging task, but CNNs have been shown to be very effective at it.

Handwritten numbers have many practical applications; for example, we can

use them in the bank to read checks and also in the post office to classify mail. And in other settings where it is important to identify handwritten digits.

This method clarify that layer by layer, CNN converts the pixel values of the source image into the final class rankings. It's important to understand that While some layers do not have parameters, others do. Particularly, the convolution / fully linked layers make adjustments based on the parameters and input volume activations. (the weights and biases of the neurons). On the other hand, the ReLu/pooling layers will implement a fixed function. The parameters in the convolutional / fully connected layers will be trained using the stochastic gradient descent method to make sure that the class scores for each image match the labels in the training set.

Regarding the respondents for this message, there are no participants other than faculty members, and no questionnaires were conducted.

By using this method, we explain well how to create a neural network to recognize handwritten digits from the famous MNIST dataset by demonstrate that for neural network training, it is often preferable to divide the dataset into two subsets: the training set and the testing (or validation) set. The validation set is used to assess the network's performance and see how well it handles unique patterns. We only use training patterns to train the network The split() function returns a tuple with the current sizes of the training and testing sets:

This method clarifies that the MNIST dataset is a collection of grayscale images of handwritten digits. Each image is a 28x28 grid of pixels, where each pixel has one intensity value that ranges from 0 black to 1 white. As a result, each image can be considered to be a collection of 784 numbers.

Table 14 This table provides a clear description of the MNIST dataset:

Dataset Description	Dataset Name	image type	image dimension	pixel range	number of pixel
Value	MNIST	Grayscale	28*28 pixel	o to 1	784

In additin, the method used in this paper clarify that data preparation serves as the initial and critical phase in every machine-learning endeavor. This stage entails obtaining, cleansing, organizing, and converting data into a suitable digital format, often comprising a substantial portion of the efforts invested by scientists and industry professionals. Furthermore, any errors or oversights during this stage can

lead to the learning process identifying incorrect patterns. As the popular saying goes, "garbage in, garbage out," emphasizing the importance of ensuring high-quality input data for obtaining reliable and meaningful outcomes.

The hardest parts have been finished for us by The National Center for Technology and Standards, which is known as MNIST, NIST is a library lover's Keras. The data has been collected and is ready to be processed.

Again, Keras provides a simple tool to help us convert the 28x28-pixel image to a vector:

```
from keras.datasets import mnist

# Setup train and test splits
(x_train, y_train), (x_test, y_test) = mnist.load_data()
print("Training data shape: ", x_train.shape) # (60000, 28, 28) -- 60000 images, each 28x28 pixe
print("Test data shape", x_test.shape) # (10000, 28, 28) -- 10000 images, each 28x28

# Flatten the images
image_vector_size = 28*28
x_train = x_train.reshape(x_train.shape[0], image_vector_size)
x_test = x_test.reshape(x_test.shape[0], image_vector_size)
```

import_and_flatten.py hosted with ❤ by GitHub view raw

Figure 27. Import and flatten

We must first prepare this dataset before we can test neural network models on it. there are an additional task that needs to be completed. It is essential for our algorithm to treat the labels in this dataset as members of a set rather than as sequential values, even if they are integers ranging from 0 to 9. It is important to note that the numbers "0" and "9" represent just two potential classifications within our dataset, without any inherent hierarchy between them.

In the context of evaluating a model's performance, it would be incorrect to conclude that the model is off by 8 unless our algorithm was specifically designed to expect the number 0 but mistakenly predicted the number 8 instead. This scenario highlights the distinction between the expected category and the model's prediction, emphasizing that an incorrect category prediction does not necessarily imply a deviation of 8 units. It is crucial to assess the model's accuracy by considering the intended target category and comparing it to the actual prediction made by the algorithm.

, in the same way, that the expectation of the number 7 when it should We say that the number "8" is not a preference to expect the number "0" when we say the number 8," knowing that both are incorrect.

the convolution / fully linked layers make adjustments based on the parameters and input volume activations. (the weights and biases of the neurons). On the other hand, the ReLu/pooling layers will implement a fixed function. The parameters in the convolutional / fully connected layers will be trained using the stochastic gradient descent method to make sure that the class scores for each image match the labels in the training set.

Regarding the respondents for this message, there are no participants other than faculty members, and no questionnaires were conducted. To address this issue, the recommended approach is to make predictions on categorical data using a one-hot encoded vector rather than continuous values. This involves creating a vector with, a length equal to the number of categories present in the dataset, where only one position in the vector is set to 1 while all other positions are set to 0. This specific position with the value of 1 is referred to as the hot value.

The networks in this article will all have the same input and output layers. The vector we previously identified as the input layer contains the information from the flattened 28x28 image and comprises 784 components. The output layer was also implicitly established When we designed one hot-encoded vector from some of the earlier labels, the ten labels corresponded to the ten nodes at this layer. Our SoftMax output layer is also used, which is a unique activation method. In the other, the ten output values are normalized as follows:

- Each number has a range from 0 to 1.
- The sum of all ten values equals 1.

The prediction for the one-hot vector is determined by selecting the highest value among the ten outputs, enabling us to interpret them as probabilities. In machine learning, it is common to utilize the softmax function when the model generates a one-hot encoded vector as its output.

In this model, there is a hidden layer with 32 nodes that employs the sigmoid activation function. The final design consists of a total of 25,450 configurable parameters. Specifically, 25,088 weights exist. connecting the input layer that comes

before the hidden layer (calculated as 784 multiplied by 32), and 32 biases corresponding to the 32 nodes in the hidden layer. Therefore, the overall parameter count is 25,088 + 32, resulting in 25,120 parameters.

Table 15 describes the final design:

Component	Weights(input hidden)	Biases(Hidden layer)	Total parameters
Parameter Count	25.088	32	25.120

In the path connecting the hidden layer to the output layer, there are 320 weights (32 multiplied by 10). Additionally, each of the ten nodes in the output layer contributes one bias, leading to a total parameter count of 25,120 + 320 + 10 = 25,450. To easily determine the parameter count of a model, the ".summary ()" method in Keras can be employed.

B. Limitations:

While working on my thesis, I encountered several obstacles, including:

- Time constraints due to heavy work demands: My schedule was exceptionally tight as a result of juggling work and family responsibilities, which presented a formidable challenge. Nevertheless, I managed to successfully navigate this constraint.
- Limited availability of references: Sourcing relevant references specifically related to the subject matter of my thesis proved to be a challenging endeavor. This scarcity of resources made it considerably more arduous to find substantial information to bolster my arguments.
- Survey distribution complexities: With a sizable target population, I faced obstacles in terms of both time and resources when it came to distributing the survey to every individual. Consequently, the results obtained may not comprehensively represent the entire community.

Despite these hurdles, I ultimately triumphed in completing my thesis. The experience has been immensely enlightening, and I am confident that my thesis serves as a significant and valuable contribution to the field.

V. DISCUSSION AND FINDINGS

A. Discussion:

There have been numerous talks on this subject; some of these discussions showed that the MNIST dataset, also known as the MNIST dataset, is a significant collection of handwritten digits that is commonly The MNIST database, a widely-used benchmark in image processing and machine learning, was developed by "re-mixing" sections of the NIST datasets. Introduced by LeCun in 1998, this subset of the NIST database specifically focuses on handwritten digits. The MNIST database contains samples from two distinct sources, providing a diverse range of handwritten digits for training and testing image categorization systems. Due to its well-defined and standardized nature, the MNIST dataset has become a standard measure for evaluating the performance of various image classification algorithms and models. Its widespread adoption has facilitated advancements in the field and allowed researchers to compare and benchmark the effectiveness of different methodologies:

The MNIST database is part of a large data set that was provided within the NIST database, which he submitted for the first time in the nineties of the last century. LeCun combined the handwritten numbers of American school students into one of the NIST databases, as well as the written numbers. Handwritten by US Census Bureau employees from the other to create image data within the MNIST database.

In short, CNN, or what is called the Convolutional Network, is a model that has become popular in recent years because its utility uses a multi-layered CNN perceptron to perform a computational function. Also, compared to other classification methods, CNN uses a small amount of pre-processing, which indicates that the network we have to learn uses filters that were done manually by the traditional algorithm. Therefore, CNN is an excellent choice for tasks involving image processing.

The MNIST dataset has gained popularity in the machine learning community

as a valuable resource for constructing and testing neural networks, similar to the well-known MNIST'S digit classification dataset. While MNIST serves as a relatively straightforward dataset that can achieve high accuracy with neural networks, experts like Ian Goodfellow and François Chollet caution against solely relying on MNIST for model benchmarking and validation. Fashion-MNIST, on the other hand, offers a more challenging and sophisticated dataset that provides a better analysis of models compared to MNIST. Its inclusion of various fashion items introduces complexity and diversity, making it a suitable choice for evaluating and validating the performance of different models in image classification tasks.

Discussions have highlighted the role of computer vision in image classification, where trained algorithms aid in processing and categorizing objects. A significant breakthrough in the field occurred in 2012 when Alexnet surpassed ImageNet, marking a pivotal moment. Since then, the field of image classification has experienced exponential growth. While humans effortlessly categorize objects around us, the identical job is more difficult for robots to complete since our brains have been organically taught to recognize a collection of familiar images. Machines encounter challenges in accurately classifying images due to factors such as variations in viewpoint, size fluctuations, occlusion (when objects overlap in an image), and changes in lighting direction and source. Mastering the fundamentals of image classification often involves working with the MNIST dataset, which provides a valuable means to identify and classify images. Nonetheless, it remains an intriguing and evolving field. . Also, compared to other classification methods, CNN uses a small amount of pre-processing, which indicates that the network we have to learn uses filters that were done manually by the traditional algorithm. Therefore, CNN is an excellent choice for tasks involving image processing.

The MNIST dataset has gained popularity in the machine learning community as a valuable resource for constructing and testing neural networks, similar to the well-known MNIST'S digit classification dataset. While MNIST serves as a relatively straightforward dataset that can achieve high accuracy with neural networks, experts like Ian Goodfellow and François Chollet caution against solely relying on MNIST for model benchmarking and validation. Fashion-MNIST, on the other hand, offers a more challenging and sophisticated dataset that provides a better analysis of models compared to MNIST. Its inclusion of various fashion items

introduces complexity and diversity, making it a suitable choice for evaluating and validating the performance of different models in image classification tasks.

Several research studies have provided evidence that the MNIST dataset plays a vital role in training and evaluating deep learning models, such as Convolutional Neural Networks CNN, Support Vector Machines, and other machine learning techniques, for image classification tasks. The dataset's simplicity and well-structured nature make it an essential resource for researchers and practitioners in the fields of computer vision and machine learning.

The current research in image classification focuses on the Convolutional Neural Network (CNN), aiming to achieve accurate and efficient classification of visual data. However, traditional CNN architectures often struggle with large datasets, leading to suboptimal results. To address this, a modified CNN architecture that incorporates batch normalization and data augmentation is proposed. In today's complex world, accurately classifying or identifying digits across various modes presents significant challenges. The proposed method demonstrates its advantages when applied to the widely recognized MNIST dataset.

The MNIST dataset comprises approximately 70,000 images. Among these, 48,000 images were utilized for training the model, We use twelve thousand images to verify the validity of the model and ten thousand images to test this model, and from here we can say that we obtained the first half of the training data contained within the NIST group, as for the second half of the NEST test data during the model development process Hence, part of the training set has been allocated so that we can verify its validity.

Table one provides a breakdown of the train-validation-test split used in creating the model.

Split category	No. of images
Train set	48000
Validation set	12000
Test set	10000

Figure 28. Train-Validation-Test Split

The MNIST dataset consists of 28x28 grayscale representations of the numbers 0 through 9 for each image. Each image is represented by a number between 0 and 1. This means that each image can be represented as a vector of 784 numbers, where each number represents the intensity of a single pixel.

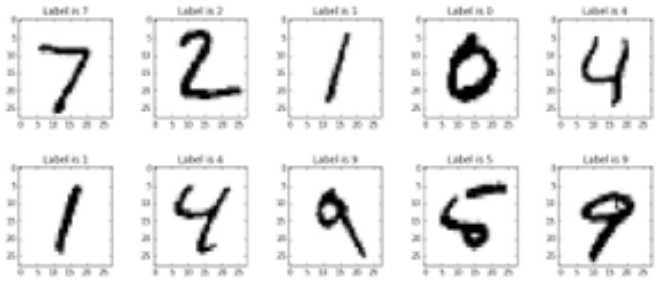


Figure 29. Image samples from the MINIST database

Handwritten digit classification holds great importance in the rapidly advancing field of technology, and Convolutional Neural Networks (CNNs) within deep learning offer effective solutions. This study focused on devising an efficient architecture for categorizing the MNIST dataset, resulting in a successful outcome. The findings indicate that incorporating the best results come from including batch normalization and data augmentation in the model. The model has four convolution layers, the last layer has sixty-four filters while the first two layers have thirty-two filters each. Impressively, the model outperformed many methods used in the field by achieving a high-test accuracy of 99.68%.

Here it is easy to check several patterns at once thanks to the layout keyword and the Python operator.

f ***Fig (20): cx. view (mnist. inputs [0:20], layout= (2, 10))***

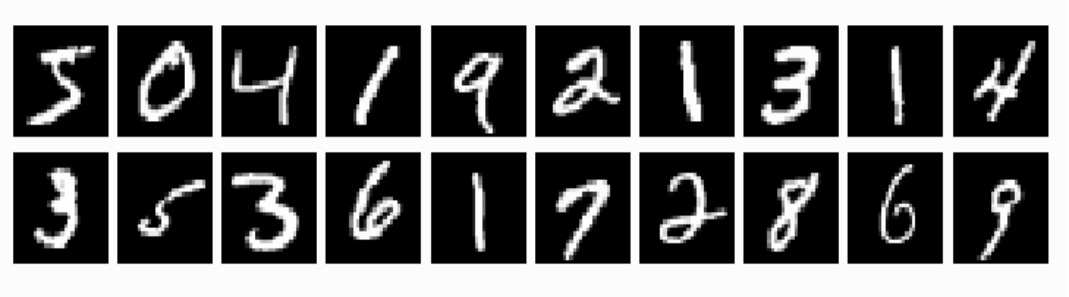


Figure 30. conx images

ConX allows us to refer to a variety of input patterns using arbitrary indices: (Staff, 2020)

Table 16 provides an overview of your study:

Study Details	Focus	Deep Learning	Enhanceme	Number of Convolution	Last Convolution	First Two	Accuracy
Value	Digit	CNN	batch normalization	4	64	32	99.68%

B. Findings:

At the end of my career in writing this scientific thesis, I must mention the most important results that have been reached from the various previous studies conducted on this subject and from the scientific experiments that have been applied. These findings are the following:

- I found out that neural networks are a common image-processing tool. As the output layer in a machine-learning model, it can generate images.
- I concluded that there are six types of MNIST Datasets; these types are 3D MNIST, Fashion MNIST, Skin Cancer MNIST, Sign Language MNIST, EMNIST, and MNIST Colorectal Histology.
- I found out that the MNIST database, Modified National Institute of Standards and Technology Database) is usually used to train image processing algorithms. It is used to test and train machine-learning algorithms and serves as a large number of repository of handwritten numbers.
- It was found out that a convolutional neural network can have a big variety of

layers such as: pooling layer, a Convolutional Layer, and many completely connected layers.

- I reached that the MNIST numbers are grayscale pictures with a single intensity value ranging from zero to one for each pixel. The entire image is composed of 784 integers organized in a plane with twenty-eight rows and twenty-eight columns. In color (RGB) photos, however, each pixel contains three numbers: one for the intensity of red, one for the intensity of green, and one for the intensity of blue.
- I reached that in the rapidly growing realm of technology, handwritten digit classification is a critical task, and CNN from deep learning is one of the best solutions.
- I found out that neural networks frequently employ images to solve the following issues: image segmentation, image classification, image generation, and image detection.
- I concluded that the first and most important stage in each process of machine learning is data preparation. Most of industry practitioners and scientists spend the majority of their time obtaining, cleaning, categorizing, and converting data into a useful digital format.
- I found out that depending on the learned algorithms, the image classification sector of computer vision assists in the processing and categorization of objects.

VI. CONCLUSION

In conclusion, I reached to the fact that neural networks are a common tool for image processing in the current world, which empower computers to make intelligent decisions with limited human involvement, i concluded that the MNIST dataset contains of a total of 70,000 images, with 60,000 images designated for training and 10,000 images for testing purposes and widely utilized for supervised learning, particularly in training classifiers, as it comprises images of handwritten numbers paired with corresponding labels, I reached out that Convolutional Network is a model that has become popular in recent years because its utility uses a multi-layered CNN perceptron to perform a computational function, I reached that a lot of neural networks contain many weights, ranging from tens of thousands to millions. I found out that it is preferable to divide the dataset into two subsets: the training set and the testing (or validation) set. The validation set is used to assess the network's performance and see how well it handles unique patterns. We only use training patterns to train the network, From this thesis, I recognised that Convolutional Network uses a small amount of pre-processing, which indicates that the network we have to learn uses filters that were done manually by the traditional algorithm, After writing this thesis, I proposed the following: I proposed all the specialist of this field to work hard on developing an easy basic neural network to enable all the students to use it in classifying images based on the MNIST dataset. In addition, I proposed all the people, scientists and researchers to conduct more studies on this subject, because of its great importance now and in the future. Finally yet importantly, I proposed the specialists to train the students on the way they follow to use neural network on the image classification.

Finally, I wish this thesis to be useful for all

VII. REFERENCES

INTERNET SOURCE

ANAND, V. (2022). Introduction to Neural Networks. Retrieved from https://www.analyticsvidhya.com/blog/2022/01/introduction-to-neural-networks/

BETTILYON, T. E. (2018). How to classify MNIST digits with different neural network architectures. Retrieved from https://medium.com/tebs-lab/how-to-classify-mnist-digits-with-different-neural-network-architectures-39c75a0f03e3

BHATTACHARYYA, J. (2020). 6 MNIST Image Datasets That Data Scientists Should Be Aware Of (With Python Implementation). Retrieved from https://analyticsindiamag.com/mnist/

BIRBECK, E. (2018). How To Build a Neural Network to Recognize Handwritten Digits with TensorFlow. Retrieved from https://www.digitalocean.com/community/tutorials/how-to-build-a-neural-network-to-recognize-handwritten-digits-with-tensorflow

BROWNLEE, J. (2019). How to Develop a CNN for MNIST Handwritten Digit Classification. Retrieved from https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/

FATAHI, M. (2014). MNIST handwritten digits Description and using. Retrieved from https://www.researchgate.net/publication/273124795_MNIST_handwritten_digits_Description_and_using

FEIYANG CHEN&NAN CHEN&HANYANG MAO&HANLIN HU. (2018). Assessing Four Neural Networks on Handwritten Digit Recognition Dataset (MNIST). Retrieved from https://arxiv.org/pdf/1811.08278

FUJISAWA, H. (2016). Sample images of MNIST data. Retrieved from https://www.researchgate.net/figure/Sample-images-of-MNIST-data_fig3_222834590

HAMZA BENTARIFAND OKATAN.A. (2023). Developing a basic neural network to classify images from the MNIST dataset. Retrieved from https://www.hnjournal.net/4-7-9/

JAREDMCMULLEN. (2022). Developing a simple CNN for MNIST. Retrieved from https://medium.com/@jaredmcmullen1/developing-a-simple-cnn-for-mnist-f98c38f0d38d

LATEEF, Z. (2023). What Is A Neural Network? Introduction To Artificial Neural Networks. Retrieved from https://www.edureka.co/blog/what-is-a-neural-network/

LECUN, Y.; CORTES, C.; BURGES, C.J.C. The MNIST Database of Handwritten Digits. 2012. Available online: **http://yann.lecun.com/exdb/mnist/** (accessed on 25 April 2018).

LIU, Y. (2021). The image classification of MNIST dataset by using machine learning techniques. Retrieved from https://escholarship.org/content/qt81b0z2hh/qt81b0z2hh.pdf?t=r46499

MARTIN, R. (2019). A fully connected layered neural network . Retrieved from https://www.researchgate.net/figure/A-fully-connected-layered-neural-network-with-two-hidden-layers-The-term-layered-is_fig1_355840800

MD. ANWAR HOSSAIN&MD. MOHON ALI . (2019). Recognition of Handwritten Digit using Convolutional Neural Network (CNN). Retrieved from https://globaljournals.org/GJCST_Volume19/4-Recognition-of-Handwritten-Digit.pdf

NDUATI, J. (2020). Introduction to Neural Networks. Retrieved from https://www.section.io/engineering-education/introduction-to-neural-networks/

NEELA CHATTYOPADHYAY&HRITTIKA MAITY&BIPSA DEBNATH,. (2013). Classification of MNIST Image Dataset Using Improved Convolutional Neural Network. Retrieved from

https://www.ijraset.com/research-paper/classification-of-mnist-image-dataset#introduction

RANJAN, S. (2019). Image Classification with MNIST Dataset. Retrieved from https://debuggercafe.com/image-classification-with-mnist-dataset/

RIZALPUTRI, L. N. (2019). Colorectal Histology CSV Multi-classification Accuracy Comparison using Various Machine Learning Models. Retrieved from https://ieeexplore.ieee.org/document/8988846

SALAKHUTDINOV, R.; HINTON, G. 2009 Deep Boltzmann Machines. In Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, Clearwater Beach , FL, USA, 16–18 April 2009; Volume 5, pp. 448–455. [**Google Scholar**]

SINGH, A. (2020). Sign-Language MNIST Problem(American Sign Language). Retrieved from https://medium.com/@abhkmr30/sign-language-mnist-problem-american-sign-language-48896ea960e0

SRIVASTAVA, N. (2022). A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. Retrieved from https://www.e2enetworks.com/blog/a-gentle-introduction-to-pooling-layers-for-convolutional-neural-networks

STAFF. (2020). The MNIST Dataset. Retrieved from https://conx.readthedocs.io/en/latest/MNIST.html

TALHA MAHBOOB ALAM&KAMRAN SHAUKAT&ET AL. (2022). An Efficient Deep Learning-Based Skin Cancer Classifier for an Imbalanced Dataset. Retrieved from https://pubmed.ncbi.nlm.nih.gov/36140516/

TENSORFLOW. MNIST for ML Beginners. 2017. Available online: https://www.tensorflow.org/get_started/mnist/beginners (accessed on 20 April 2018).

WONG, E. (2018). Investigation on the Effects of Curriculum Learning through a Simulation Study. Retrieved from https://escholarship.org/content/qt2d5972zq/qt2d5972zq_noSplash_721a4a4e93321818e62120851f0c97dd.pdf

ZHOU, V. (2019). Machine Learning for Beginners: An Introduction to Neural

Networks. Retrieved from https://victorzhou.com/blog/intro-to-neural-networks/

RESUME

Name Surname: Hamza Basil Hassan BenTarif

EDUCATION:

- **Bachelor**:

2017, Jordan, Amman, AL-Zaytoonah University, Faculty of Computer Sciences, Department of Software Engineering.

- **M.S**:

2023, Istanbul Aydin University, Department of Software Engineering, Computer Engineering (English)(Thesis) Program.