Journal of Informatics
Electrical and Electronics Engineering, 2022,
Vol. 03, Iss. 02, S. No. 003, pp. 1-11
ISSN (Online): 2582-7006

Handwritten Digit Image Recognition Using Machine Learning

Aditya Srivastava[1], Pawan Singh[2]

[1,2]Department of Computer Science and Technology, Amity University, AUUP, India
adityasrivastava2002@gmail.com[1], pawansingh51279@gmail.com[2]

How to cite this paper: A. Srivastava and P. Singh, “Handwritten Digit Image Recognition Using Machine Learning,” *Journal of Informatics Electrical and Electronics Engineering* (JIEEE), Vol. 03, Iss. 02, S No. 003, pp. 1–11, 2022.

https://doi.org/10.54060/JIEEE/003.02.003

Received: 12/10/2022
Accepted: 16/11/2022
Published: 25/11/2022

Abstract

Machine Learning is a type AI (also known as Artificial Intelligence) that makes the pc or computer to act like individuals and learn more as they experience additional information from their client or user. So here in this report we got basic introduction about machine learning like actually what is it, what are its use, how it works, languages for coding, value of python for machine learning, and many more things. As python is majorly used for machine learning, so we discussed about its use and its libraries. Thereafter, we discussed about the categories in machine learning, i.e., supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. After this discussion we got a glimpse of some basic pros and cons of machine learning in artificial intelligence. After that we discussed about basic implementation formats of the algorithm in machine learning. Then we discussed about some majorly used applications of machine learning which are trending nowadays and has a great demand in today’s market. At the end we came to know about the future scopes in machine learning and concluded it.

Keywords

Machine Learning, Artificial Intelligence, Convolutional neural network, MNIST

1. Introduction

Artificial intelligence has number of subdivisions, which includes even Machine Learning. Machine Learning can be characterized as a subtype of science innovation that instructs computers to learn things on their own without any human intervention. It's a type of AI (also known as Artificial Intelligence) that makes the pc or computer to act like individuals and learn more as they experience additional information from their client or user. To analyze or understand circumstances and get conditions and simply decide, AI frameworks as often as possible uses the information or a wide range of data sources given by the client

user of that system. Machine learning software has a good power to read all the data in a very short span of time or we can say it is really fast. The most important and the main goal of machine learning is to enable the system to make decisions automatically without any human intervention, guidance or instruction to the system to make correct decisions Fundamentally, it assists a system with expanding work productivity, thinking capacity, dynamic capacity, just as it assists a system with behaving like a human brain utilizing machine learning. Different algorithms are used to create a model of AI. These algorithms 'learn' to optimize their operations as they get new data, which means they improve performance and grow more intelligent.
There is no single first-rate language for getting to know machine learning. However, there are only a few programming languages which can be suitable for machine learning than other languages. Many machine learning engineers choose a programming language like Python or R [1].

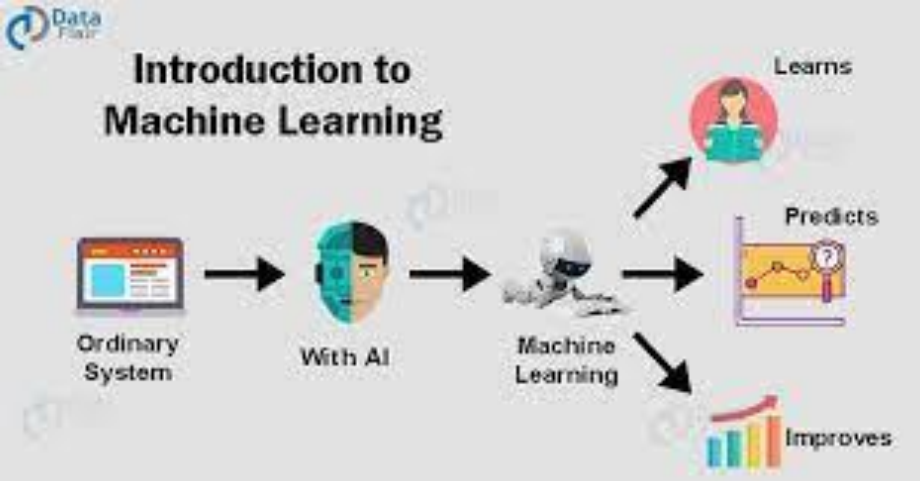


Figure 1. Introduction to MACHINE LEARNING

2. Importance of Python for Machine Learning

Python makes it simpler for developers to experience more confident and productive while growing, deploying and maintaining programs. Python's accessibility to strong AI and MACHINE LEARNING libraries and frameworks, versatility, platform freedom, simplicity and consistency, and enormous groups make it the ideal preference to start with it in the coding field of machine learning and AI.

2.1. Availability of many libraries

Libraries includes a pre-written segments of some code that lets in a user to use a particular function or carry out numerous operations. The Python library gives base-stage items so that coders do not have to write code from scratch every time. Machine Learning calls for non-stop information processing, and Python libraries can help you get admission to process and remodel or change your data. Some of the most tremendous and popular libraries for AI and MACHINE LEARNING are Scikit-learn, Pandas, Numpy, TensorFlow, Keras and many others.

2.2. Platform independence

Python is simple to use, research, and it's far flexible too. It implies that Python, that's used to expand machine learning, can run on all systems, together with Windows, Linux, macOS, and many other platforms. To shift the procedure from one platform to every other, developers will have to perform a few minor changes and revise some lines of coding to create executable code for executing it.

2.3. Readability

Python is straightforward to read and recognize, so Python developers don't have any hassle knowledge of editing, copying, or pasting any type of code. There isn't any confusion, inconsistency, or errors while the usage of Python. That facilitates the efficient exchange of algorithms, ideas and other things that gears up between AI and machine learning specialists.

2.4. A low entry barrier

There is a lack of coders in the world. Python is easy for practicing and learning a language – the boundaries to enter are very low. Numerous data scientists can examine Python speedy to participate in system of machine learning projects.

3. Python Libraries

Some of the majorly used libraries in python for machine learning are discussed below.

3.1. NumPy

Created right after the top of Numeric, an older library, the NumPy is used for dealing with multi-dimensional information and complex mathematical capabilities. NumPy is a computational library which is very fast and that can handle obligations and features ranging from fundamental algebra to even Fourier transforms, shape manipulations and, random simulations.

3.2. Pandas

Python data analysis library is the full form of Pandas. Using this we can make multidimensional structures, revising records for numerical information and time series. It makes use of data frames and collection to outline 3-dimensional and 2-dimensional data simultaneously. It additionally gives options for indexing large statistics for brief indexing in large datasets. It is well known for the talent of records reshaping, dealing with lacking records, merging datasets, and the alternative for information filtrations. Pandas could be very beneficial and rapid with huge datasets.

3.3. Keras

Keras is a high stage deep learning API written in python for neural networks. It helps with multiple backend neural network computations and makes implementing neural networking smooth. Keras is powerful and clean to use open-source python library for developing deep learning models and is an excessive level API of TensorFlow.

3.4. Matplotlib

Matplotlib is a library utilized in Python for graphical illustration to recognize the records before shifting it to data-processing and training it for Machine learning functions. It makes use of python GUI toolkits for getting graphs and their plots using object orientated APIs. The Matplotlib also offers a MATLAB-like UI so that a consumer can perform similar tasks as in MATLAB. This library is unfastened and open-source and has many extension interfaces that increase matplotlib API to diverse other libraries. In machine learning, there are four primary types: supervised learning, unsupervised learning, semi-supervised learning, and reinforcing learning.

4. Types of Machine Learning Algorithm

4.1. Supervised learning

It is simply a sort of ML which uses past data and this analysis to predict future values or events. The use of labelled datasets to train algorithms that reliably classify data or predict outcomes is characterized as supervised learning, often known as supervised machine learning. The training data presented to the computers acts as a supervisor, teaching the machines how to correctly forecast the output.

4.2. Unsupervised learning

In contrast to supervised learning, it utilizes information that isn't labelled or marked in any way. The principal objective of unsupervised learning is to find covered up and obscure patterns from the information(data) of user. Unsupervised learning, in other terms, allows the system to recognize patterns in data sets on its own.

4.3. Semi-supervised learning

Between both types of above-mentioned learning, semi-supervised learning is a good compromise against them. It guides categorization and feature extraction from a larger unlabeled data set using a smaller labelled data set during training.
As long as we are short of enough labelled data for supervised learning, we can sum up all unlabeled type of data to increment the size of the information that trains the software or system

4.4. Semi-supervised learning

These algorithms perform dynamic evaluation and implementation concurrently during the learning process – that is, learning takes place in an interactive, trial-and-error environment, using feedback of inputs and outputs. It interacts with the environment by generating actions and results and in the same case, it detects errors and corrects them.

5. Pros and Cons of Machine Learning

5.1. Pros

1. It can easily identify the trends and learning.
2. Interference of human is not needed.
3. As the amount of data increases, the accuracy of prediction becomes more better.
4. It can handle multi-dimensional and multi-variety data easily.

5.2. Cons

1. A lot of data is required for precision.
2. High error chances.
3. Machine Learning requires adequate opportunity of time to learn and offer output to its capacity with a significant degree of exactness. It also requires a lot of resources to process.
4. Machine learning is very prone to produce errors as it works on its own.

6. Pros and Cons of Machine Learning

Handwritten digit recognition is nothing but just the computer systems skill that have the capability to understand human handwritten digits. It is a hard venture for the machine because handwritten digits aren't ideal and may be made with many one-of-a-kind flavors. The handwritten digit recognition helps to remove this hassle which uses the photograph of a digit and acknowledges the digit present inside the photo. We're going to put in force a handwritten digit recognition interface using the MNIST dataset. We are going to have the use of a special sort of deep neural network which is called as Convolutional Neural Networks.

6.1. MNIST

MNIST is nothing but a massive set of handwritten digits. It is a completely famous dataset inside the area of processing an image. MNIST incorporates a set of 70,000, 28 x 28 snap shots of handwritten digits from zero to 9. The dataset of MNIST has been already divided into two sets, that is, training and testing sets.

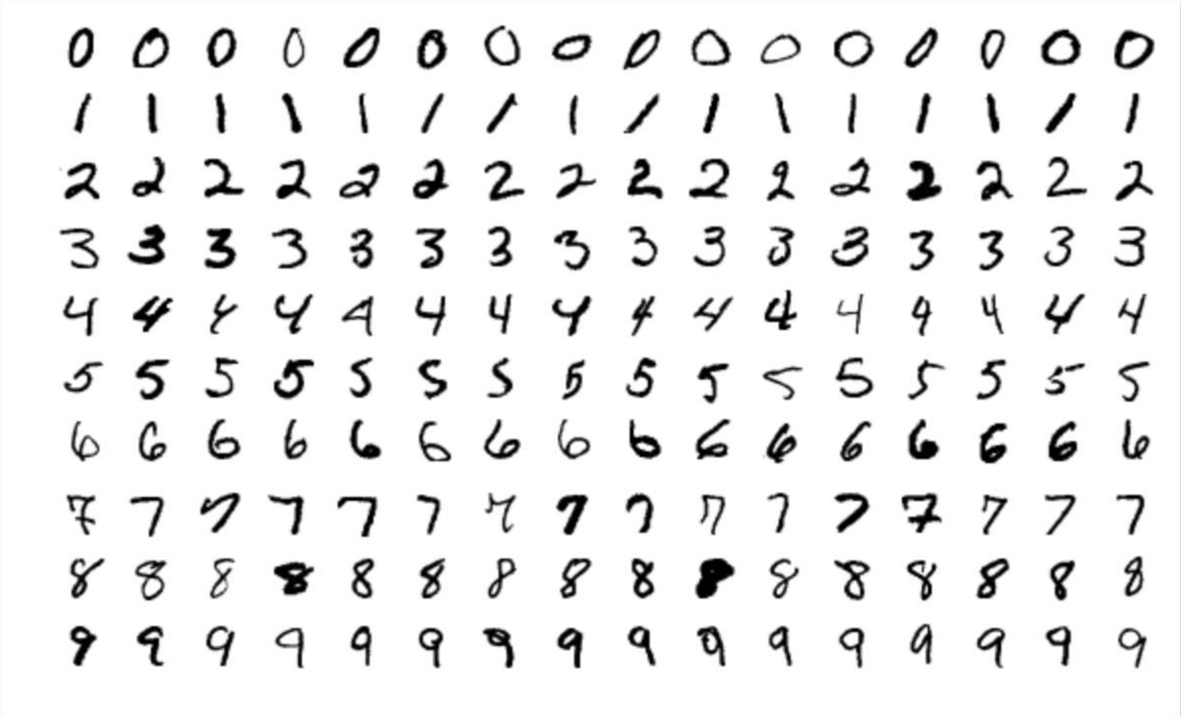


Figure 2. MNIST Dataset

6.2. Convolutional Neural Network

Convolutional Neural Network is one of the types of Deep Learning Algorithm that has an input as an image and study the diverse functions of the picture through filters. This permits them to examine the critical items present in the picture, letting them parent one photograph from the alternative.

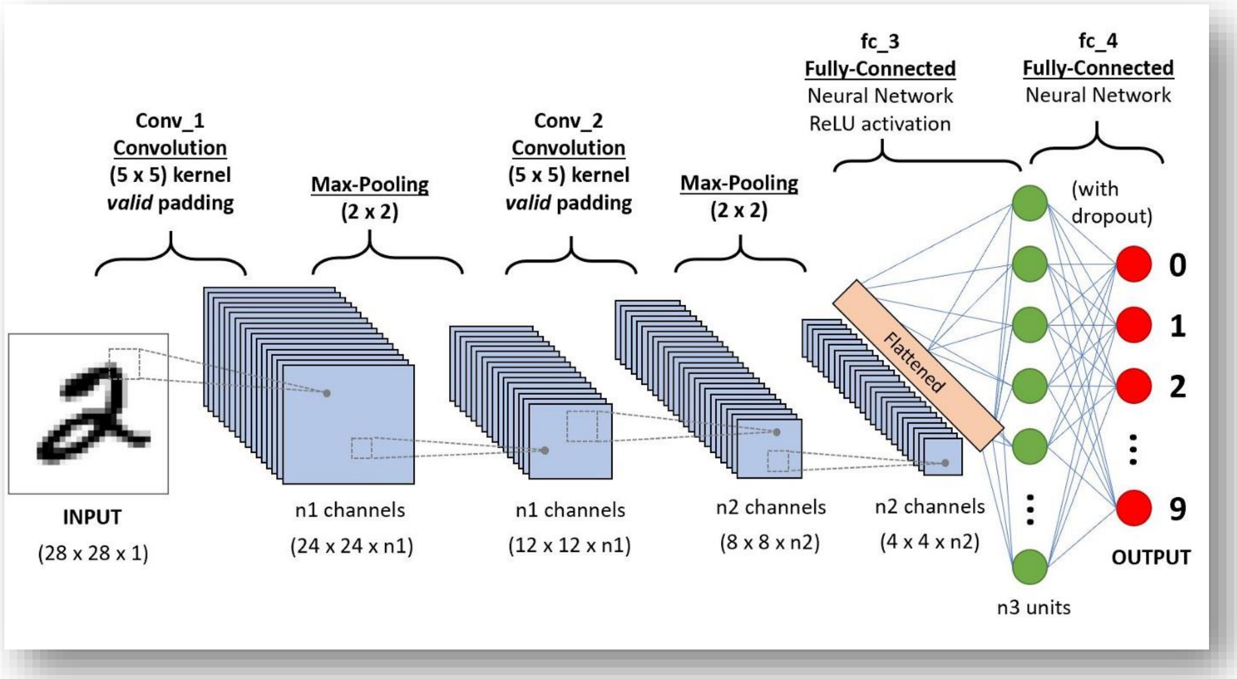
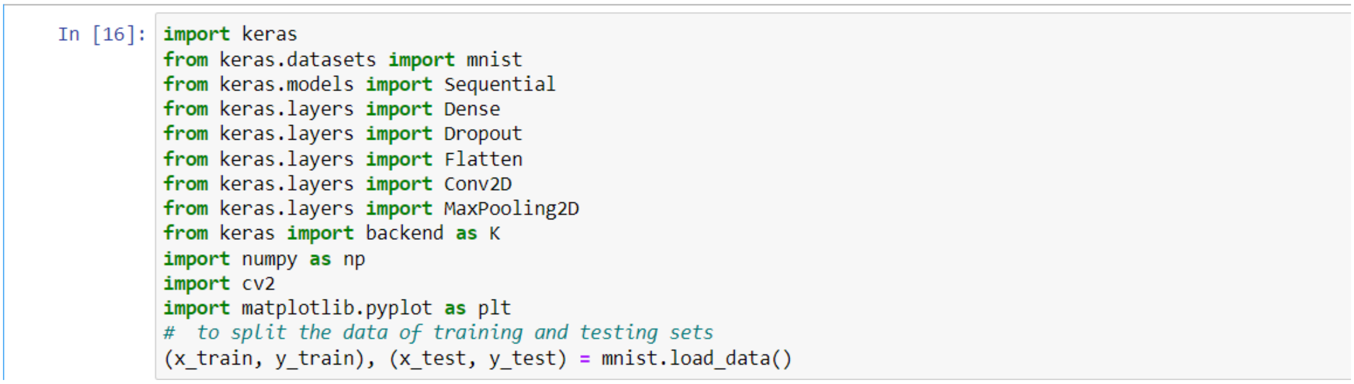


Figure 3. Convolutional Neural Network

7. Methodology of Machine Learning

7.1. Collecting Data

As you already know, machines have to begin analyzing from the facts which you deliver them. It is of the maximum significance to acquire reliable records so that your machine learning models can locate the right styles. The first-class of the matter that you give to the machine will decide how accurate the model is going to be. If you have got wrong or old information, you'll have wrong outcomes or predictions which aren't applicable.

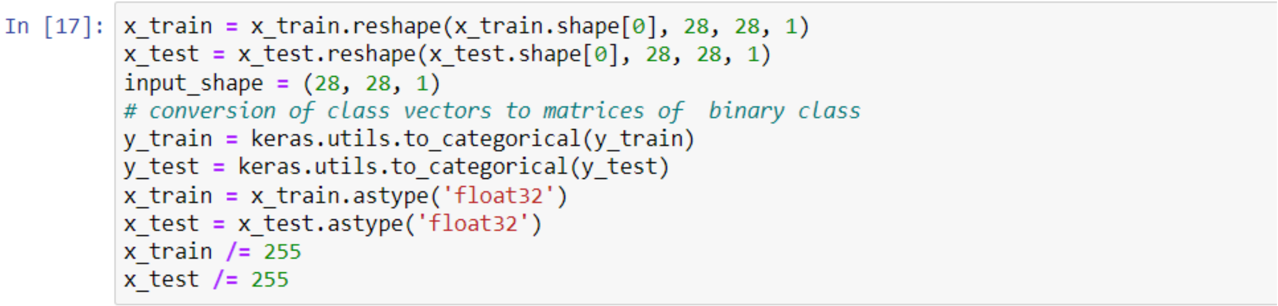


```
In [16]: import keras
         from keras.datasets import mnist
         from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import Dropout
         from keras.layers import Flatten
         from keras.layers import Conv2D
         from keras.layers import MaxPooling2D
         from keras import backend as K
         import numpy as np
         import cv2
         import matplotlib.pyplot as plt
         #  to split the data of training and testing sets
         (x_train, y_train), (x_test, y_test) = mnist.load_data()
```

Figure 4. Importing libraries

7.2. Preparing Data

Preparing of data includes collecting all the data together that we have got and putting it in random sequence; cleaning the information to remove unwanted values, duplicate digits, lacking values, conversion of data type, rows, columns, and many others; splitting the cleaned information into units, that is, a training and a testing outset. Our model learns from the training set that we provide. We can get the accuracy of our model by using the testing set.

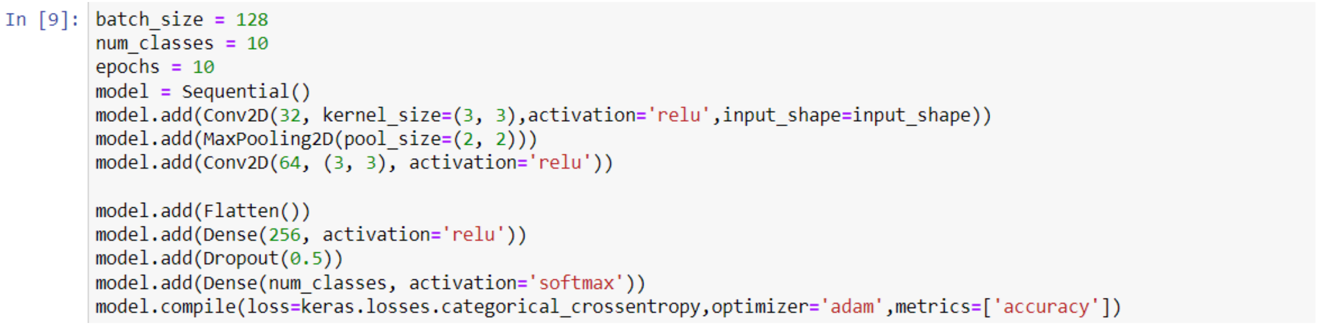


```
In [17]: x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
         x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
         input_shape = (28, 28, 1)
         # conversion of class vectors to matrices of  binary class
         y_train = keras.utils.to_categorical(y_train)
         y_test = keras.utils.to_categorical(y_test)
         x_train = x_train.astype('float32')
         x_test = x_test.astype('float32')
         x_train /= 255
         x_test /= 255
```

Figure 5. Pre-processing data

7.3. Choosing a Model

Machine learning model also examines the output that one gets on the gathered data by implementing a machine learning algorithm. It is essential to pick a model which is handy and applicable to the mission. Researchers and engineers developed numerous models desirable for unique tasks to be performed like speech recognition, prediction, etc.



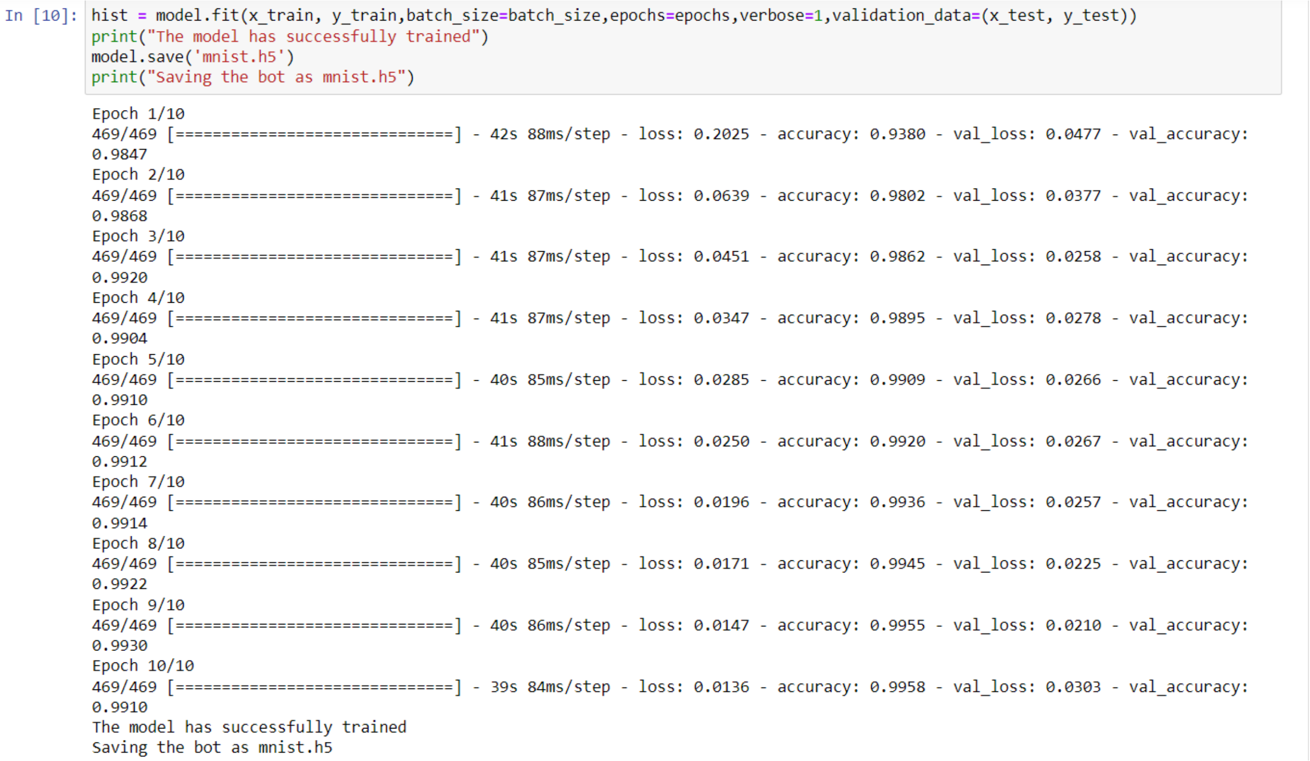
```
In [9]: batch_size = 128
        num_classes = 10
        epochs = 10
        model = Sequential()
        model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',input_shape=input_shape))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Conv2D(64, (3, 3), activation='relu'))

        model.add(Flatten())
        model.add(Dense(256, activation='relu'))
        model.add(Dropout(0.5))
        model.add(Dense(num_classes, activation='softmax'))
        model.compile(loss=keras.losses.categorical_crossentropy,optimizer='adam',metrics=['accuracy'])
```

Figure 6. Making model

7.4. Training the Model

Training is the maximum essential step in machine learning. In training, you transfer the organized data to your machine learning model to find styles and make predictions. It results within the model, gaining knowledge from the records in order that it can accomplish the set. Over the time, with epochs, the version gets more better at the time of predicting.



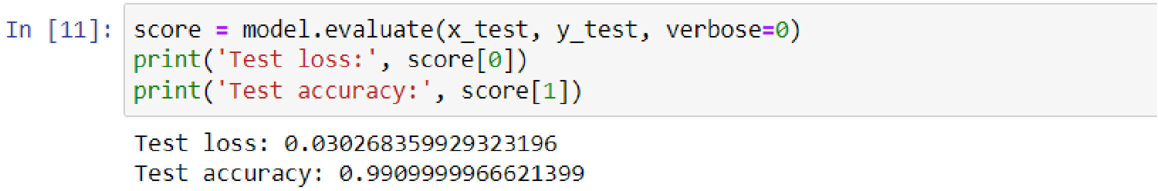
```
In [10]: hist = model.fit(x_train, y_train,batch_size=batch_size,epochs=epochs,verbose=1,validation_data=(x_test, y_test))
         print("The model has successfully trained")
         model.save('mnist.h5')
         print("Saving the bot as mnist.h5")

Epoch 1/10
469/469 [==============================] - 42s 88ms/step - loss: 0.2025 - accuracy: 0.9380 - val_loss: 0.0477 - val_accuracy:
0.9847
Epoch 2/10
469/469 [==============================] - 41s 87ms/step - loss: 0.0639 - accuracy: 0.9802 - val_loss: 0.0377 - val_accuracy:
0.9868
Epoch 3/10
469/469 [==============================] - 41s 87ms/step - loss: 0.0451 - accuracy: 0.9862 - val_loss: 0.0258 - val_accuracy:
0.9920
Epoch 4/10
469/469 [==============================] - 41s 87ms/step - loss: 0.0347 - accuracy: 0.9895 - val_loss: 0.0278 - val_accuracy:
0.9904
Epoch 5/10
469/469 [==============================] - 40s 85ms/step - loss: 0.0285 - accuracy: 0.9909 - val_loss: 0.0266 - val_accuracy:
0.9910
Epoch 6/10
469/469 [==============================] - 41s 88ms/step - loss: 0.0250 - accuracy: 0.9920 - val_loss: 0.0267 - val_accuracy:
0.9912
Epoch 7/10
469/469 [==============================] - 40s 86ms/step - loss: 0.0196 - accuracy: 0.9936 - val_loss: 0.0257 - val_accuracy:
0.9914
Epoch 8/10
469/469 [==============================] - 40s 85ms/step - loss: 0.0171 - accuracy: 0.9945 - val_loss: 0.0225 - val_accuracy:
0.9922
Epoch 9/10
469/469 [==============================] - 40s 86ms/step - loss: 0.0147 - accuracy: 0.9955 - val_loss: 0.0210 - val_accuracy:
0.9930
Epoch 10/10
469/469 [==============================] - 39s 84ms/step - loss: 0.0136 - accuracy: 0.9958 - val_loss: 0.0303 - val_accuracy:
0.9910
The model has successfully trained
Saving the bot as mnist.h5
```

Figure 7. Training model

7.5. Evaluating the Model

After all the training of your model, you've got to test to look how it's appearing. This is accomplished via checking out the overall performance of the model on unseen data. The unseen data is one of those records which were split earlier for having a check on trained model. If testing out turned into the identical data that's used for training, you will no longer get a correct wanted result, as the model is already used to the facts, and finds the identical patterns in it, because it formerly did. This will provide you with disproportionately excessive accuracy.



```
In [11]: score = model.evaluate(x_test, y_test, verbose=0)
         print('Test loss:', score[0])
         print('Test accuracy:', score[1])

         Test loss: 0.030268359929323196
         Test accuracy: 0.9909999966621399
```

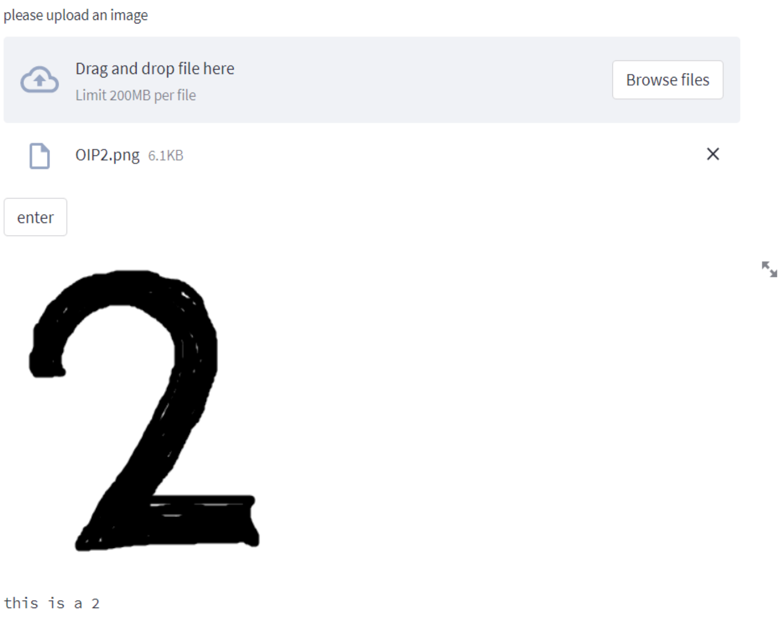
Figure 8. Evaluating Model

7.6. Parameter Tuning

Once we are done with creating and evaluating our model, we will have to see if its accuracy can be stepped forward in any manner. This is executed by improving the parameters present in our model. Parameters are nothing but just the variables within the different models that the programmer generally decides on its will.

7.7. Make Predictions

At last, we may now use our model on the unseen records that were gained at the time of splitting (testing set) to make predictions appropriately.



```
this is a 2
```

Figure 9. Prediction using stream lit

8. Applications of Machine Learning

Some of major applications are listed below:

8.1. Image Recognition

Image recognition is one of the most requested and well-known utilizations of MACHINE LEARNING in AI. It is utilized to distinguish objects, individuals, places, advanced pictures, and so on.

8.2. Speech Recognition

Speech recognition is a technique for detecting words in spoken language. Voice recognition is a biometric innovation used to recognize a particular individual's voice or to distinguish the speaker. It's also known as "speech to text,".

8.3. Traffic prediction

It is also a type of machine learning software of artificial intelligence. Google map is widely used application for this purpose. With help of its algorithm, it can get the continuous information of traffic and can make it visible to us

8.4. Stock Market trading

Machine learning is greatly in use for stock market trading. There are possibilities of many ups and downs in the rate to stocks. So here machine learning is used to predict the rates. It takes the previous data of company and calculates ratios of profit and loss and accordingly gives result.

8.5. Product Recommendation

When we shop on some eCommerce brands like big basket, Amazon, etc. or sites the algorithm of machine learning screens the information with which the client connects the most and then in form of ads or suggestion, it shows similar types of products in accordance to the previous search history and more personal data in that particular field.

8.6. Self-driving cars

Machine learning introduced a very useful software in which a car can drive on its own. TESLA is well known company which launched these self-driving cars. It takes help of algorithm which detects the real time data of its surroundings and accordingly drives the car on its own.

9. Conclusion

Today, the discipline of AI is quickly extending, especially in the computer vision fields. In computer vision, the possibilities of a human making an error is about 3%. This recommends that PCs are now better compared to people at distinguishing and investigating photographs. What an incredible achievement it is that computers used to be large pieces of technology the size of a room; now, they can comprehend the world around us in ways we never imagined. Undoubtedly there’s huge scope of machine learning in the coming next years. And as we studied machine learning is a very board topic within itself and there are many languages by which we can code in machine learning and overall, it has taken the world to the next level of smartness.

References

[1]. U. Pal, T. Wakabayashi, and F. Kimura, “Handwritten numeral recognition of six popular scripts,” *Ninth International conference on Document Analysis and Recognition ICDAR 07*, vol. 2, pp. 749–753, 2010.

[2]. J. Brownlee, “Your first deep learning project in Python with keras step-by-step,” *Machine Learning Mastery*, 17-Jun-2022. [Online].

[3]. V. Singh, “Digit recognition using single layer neural network with principal component analysis,” *Asia-Pacific World Congress on Computer Science and Engineering*, pp. 1–7, 2014.

[4]. A. K. Seewald, “Digits-a dataset for handwritten digit recognition,” *Osterreichisches Forschungsinstitut f ¨ ur Artificial ¨ Intelligence TR*, 2005.

[5]. R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[6]. M. Neill, *Neural Network for Recognition of Handwritten Digits, Code Project*. 2006.

[7]. L. Yann and B. Yoshua, “Convolutional networks for images speech and time series,” in *Arbib 4 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) 605 Michael A. The handbook of brain theory and neural networks*, pp. 276–278, 1995.

[8]. M. S, C. N. Vanitha, N. Narayan, R. Kumar and G. R, "An Enhanced Handwritten Digit Recognition Using Convolutional Neural Network," *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1724-1727, 2021. DOI: 10.1109/ICIRCA51532.2021.9544669

[9]. M. Jain, G. Kaur, M. P. Quamar and H. Gupta, "Handwritten Digit Recognition Using CNN," *2021 International Conference on Innovative Practices in Technology and Management (ICIPTM)*, pp. 211-215, 2021. DOI: 10.1109/ICIPTM52218.2021.9388351

[10]. C. Zhang, Z. Zhou and L. Lin, "Handwritten Digit Recognition Based on Convolutional Neural Network," *2020 Chinese Automation Congress (CAC)*, 2020, pp. 7384-7388, 2020. DOI: 10.1109/CAC51589.2020.9326781

[11]. R. Sethi and I. Kaushik, "Hand Written Digit Recognition using Machine Learning," *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 49-54, 2020. DOI: 0.1109/CSNT48778.2020.9115746

[12]. K. Matsuoka, L. Wilen, S. P. Hurley, and C. F. Raymond, “Effects of birefringence within ice sheets on obliquely propagating radio waves,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 5, pp. 1429–1443, 2009.

[13]. A. Shrivastava, I. Jaggi, S. Gupta and D. Gupta, "Handwritten Digit Recognition Using Machine Learning: A Review," 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), pp. 322-326, 2019. DOI: 10.1109/PEEIC47157.2019.8976601

[14]. L. Liu and H. Fujisawa, “Classification and learning for character recognition: comparison of methods and remaining problems,” in *Proceedings of the International Workshop on Neural Networks and Learning in Document Analysis and Recognition*, Seoul, Korea, 2005.

[15]. “A database for handwritten text recognition research,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[16]. K. Gaurav and P. K. Bhatia, “Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition”,” in *2nd International Conference on Emerging Trends in Engineering & Management, ICETEM*, 2014.

[17]. A. Yavariabdi and H. Kusetogullari, “DIDA: A dataset of handwritten digit images in the context of HCR.” IEEE DataPort, 2021. DOI: 10.21227/hv16-0j03