In-House Training Report

On

Handwritten Digit Recognition on MNIST dataset using Python

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Vaibhav Bhargava

Under the guidance of

Mr Sanjiv Kumar Tomar

Asst. Professor (Grade - III)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY
AMITY UNIVERSITY UTTAR PRADESH, NOIDA

June 2022

DECLARATION

I Vaibhav Bhargava of B.Tech.(CSE) hereby declare that the in-house project report titled " Handwritten Digit Recognition on MNIST dataset using Python" which is submitted by me to Department of Computer Science & Engineering, Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, in partial fulfillment of requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

Place: NOIDA

Vaibhav Bhargava

A2305220585

CERTIFICATE

On the basis of declaration submitted by Vaibhav Bhargava, student of B.Tech. CSE, I hereby certify that the in-house project titled "Handwritten Digit Recognition on MNIST dataset using Python", submitted to Department of Computer Science & Engineering, Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering, is an original contribution with existing knowledge and faithful record of work carried out by him under my guidance and supervision.

Place: NOIDA

Mr. Sanjiv Kumar Tomar (Guide)
Department of Computer Science and Engineering
Amity School of Engineering and Technology
Amity University Uttar Pradesh, Noida

ACKNOWLEDGEMENT

Inspiration and motivation have always played a key role in success of any venture and right guidance, assistance and encouragement of other people have played an essential part.

I am grateful to my faculty guide **Mr Sanjiv Kumar Tomar,** Associate Professor, Amity School of Engineering and Technology (ASET) for his able guidance and support. His guidance helped me in every aspect for writing this report. I could not have imagined having a better advisor and mentor for my report.

I am also grateful to Amity School of Engineering and Technology for providing me with this opportunity and also to our respectful head of department **Prof. (Dr.) Sanjeev Thakur** department of Computer Science & Engineering for the regular encouragement.

And lastly I would like to acknowledge the main support I had that made me complete this report on time and that is my family. They have helped me throughout and supported me. It would have been unimaginable without their support.

Vaibhav Bhargava

A2305220585

ABSTRACT

The problem of handwritten number recognition has long been an open problem under the view of pattern bracket. plenty of studies have resulted that Neural networks, machine literacy have great and effective performance. In data bracket Deep literacy and Algorithms of NN are a branch of Machine literacy which will automatically identify patterns within the data, and also use the uncovered patterns to prognosticate unborn data, or to perform other indispensable sorts of deciding under unreliability Deep literacy algorithms are accustomed model high position abstractions in data. number Recognition may be a combination of Deep literacy and Neural Network algorithms, which uses TensorFlow tool to hang together to develop a model. This paper describes the popularity of handwritten scrutinized integers where the input is given by the stoner and displays the affair as digital figures pertaining because the input handed consequently by using Machine literacy styles with the assistance of TensorFlow, MNIST Database, python therefore the image is also tasted by the system because the stoner provides bare handed input thereto and the system could also show the separate recognition consequently.

CONTENTS

CHAPTER-1

.INTRODUCTION.

ML and Deep Learning plays a crucial role in sciences of computer, its equipment and computer science. the utilization of machine learning, deep learning , common principles have decreased the human practices in industry. For experts recognition of Handwritten digits has been popular from the very beginning of ML and deep learning. Handwriting recognition is the ability to recognize handwritten text from a scanned file, image, touch-screen or other tools and converting it to a text which can be edited. It is a grueling task in the view of series recognition. Every person in this world has their own way of jotting. In real- time operations like the conversion of handwritten information into digital format, verification of autographs, number plate recognition, this kind recognition is needed. Developing program for it needs proper understanding of classification of digits and the subtraction between every major and minor points to properly differentiate between different digits which can be only done by proper training and testing on predefined datasets. The goal to design a handwriting recognition system with 100% accuracy is not a possible thing and can't be achieved as even humans do not have the ability of recognizing any handwritten text without any doubt. Reasons that can cause inaccuracy are pattern twist, reality of unwanted objects, confused patterns, and dwindling of paper.

This report presents the algorithm recognizing the handwritten digits (0 to 9) from the famous MNIST dataset using TensorFlow framework(library) and python as language and its libraries as user enters the respective digit the algorithm would recognize and show the results with accuracy percentage. MNIST is the most extensively used database of marked handwritten digits, with different trainings and test sets, and hence is an

easily understandable domain that secures a fast comparison between various techniques. In contrast, MNIST is very commonly and universally used that it is even called as “hello earth(refers to world)” of various deep learning and ML frameworks, like TensorFlow.

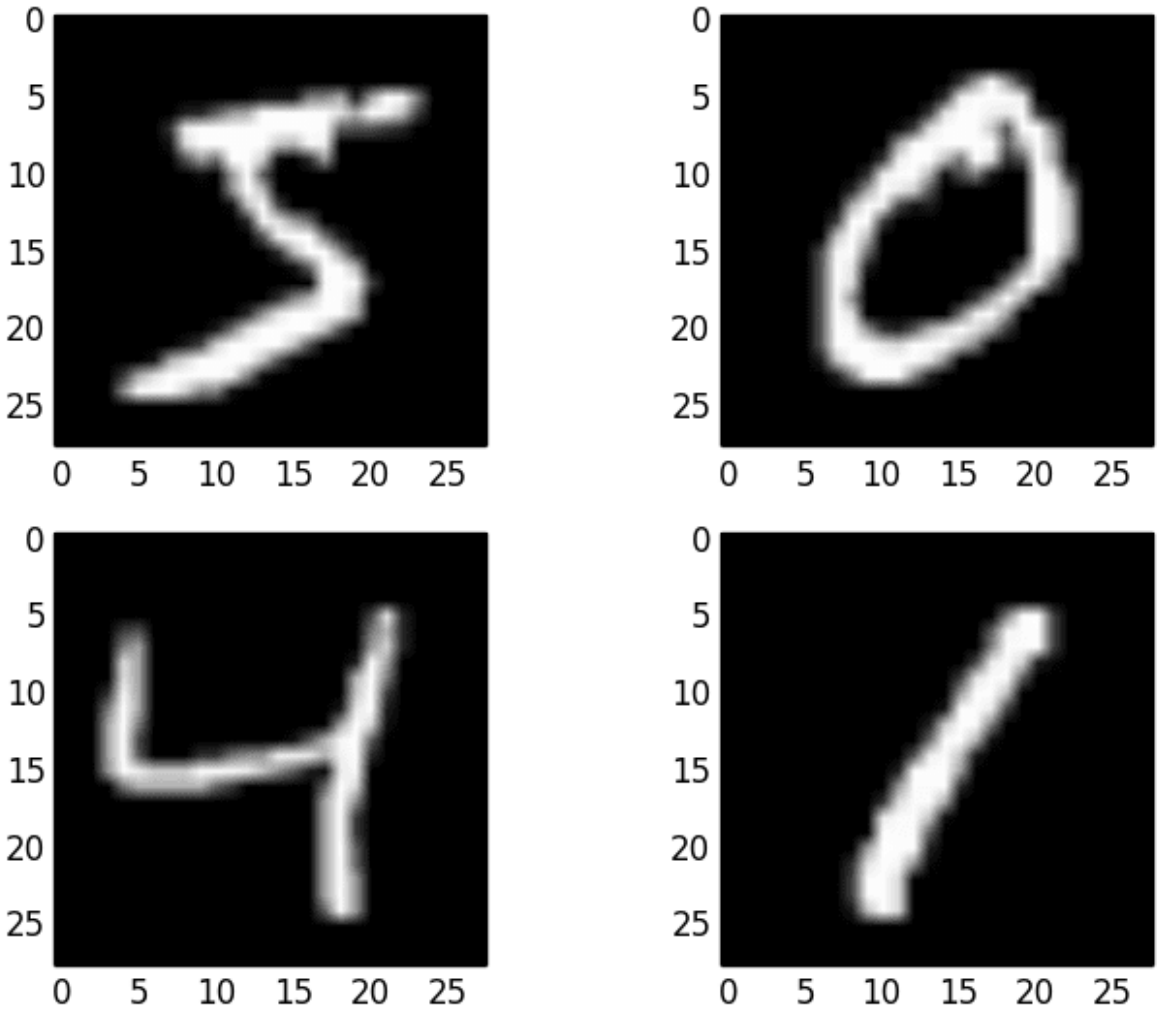


Fig 1 shows the various digits outputs that can be recognized.

CHAPTER-2

MNIST

The MNIST which stands for (Modified. National .Institute .of Standards. and Technology) database is a widely used digits database of which are handwritten which is generally used for training different image processing devices. Under the field of ML, this database is extensively used for training and testing. It was formualated by using NIST's original datasets and "re-mixing" its samples. MNIST consists of samples of handwritten digits, it contain total 70,000 images from which 60,000 are used in training set and 10,000 are used in testing set and both are labelled appropriately with 10 digits(0-9). Handwritten digits are images of the form 28*28 gray scale intensities of images representing an image with the first column to be labelled as (0 to 9) for every image. The sets which are to be tested and involved in training are chosen as with same writer should not involve in both sets. Samples more than 250 writers are there in training set. MNIST is also the computer science and vision database that consist of digits that are handwritten along with the labels identifying the digits properly, every MNIST data point has two parts: an image of a handwritten digit and its corresponding label. Original images were submitted for preprocessing. In this procedure firstly we involve smoothning images to fit into a 20 × 20 box of pixel while maintaining and safeguarding the aspect ratio. Then, we applied an anti-aliasing filter, and as a result b and w images were effectively transformed into grayscale color scheme. Finally, blank padding was introduced so that the image fit in a larger 28 × 28 box of pixel, and the center of mass of the digit matched with its center. Examples of digit are shown in figure 1

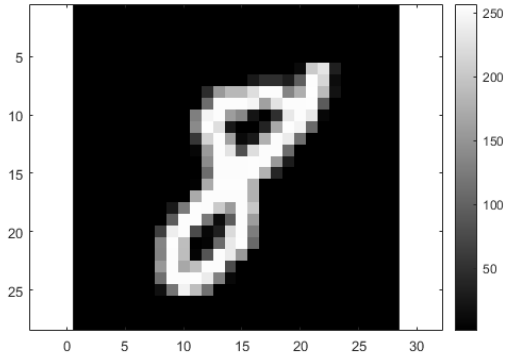


Fig 2. Shows the eight that is to be recognized

Also MNIST has been extensively used to test and check the behavior of various implementations of classifiers with the help of MNIST as a benchmark, there's been a push to publish various ranks in the past. Many of those rankings and established literature that too uses the "rate of test error" metric whenever we have to ponit to the show presented over MNIST. The metric shows the percentage of wrongly instances that were being categorized. One of the oldest ranks that was published and drafted by Mr. LeCun et al, which includes hints and supports up to 2012. The website which is here provides an anatomy of group dividers and points out whether each and every work perfectly performed data pre-processing and augmentation by adding new instances to the training set which results from distortions or other modifications of the original data).

From the various completions and sources accumulated in Mr. LeCun et al.'s ranks, those dependent upon various NN that overcomes the various techniques. Thereafter, there are various ML techniques which are still able to provide competitive error rates.

Fig 3 shows the MNIST Digits

ML->Machine Learning , b and w-> black and white

CHAPTER-3

Tensor Flow

TensorFlow was introduced and made by the team of Google Brain for Google internal uses which was used in development and the main production purpose of TensorFlow, which can be used in a most variety of programming languages like most notably Python, Javascript, C++, and as well as Java. This wide usability lends itself to a range of applications in many different sectors. TensorFlow is a software library or framework, used to implement ML* and deep learning concepts in the easiest manner which was designed by the Google team. It combines the computational algebra of optimization techniques which is used for easy calculation of many mathematical expressions. It is a library with open source i.e. created by the Brain of Google Trust which is used in heavy computational work, geared towards ML and deep learning tasks. Tensor Flow is built on c, c++ making it very fast while it is available for use for platform like Python, C++, Haskell, Java and Go API depending upon the type of work. It created data graph flows for each model, where a graph contains two units – first a tensor and second a node.

Fig 3.1 Tensor Flow Logo

- Tensor: A tensor is a multi-dimensional array.
- Node: A node is a mathematical computation that is being worked at the moment to give the desired result.

A data graph flow substantially maps the flow of information via an reciprocate between these two components. As a graph gets completed, the model gets executed for the output.

TensorFlow is the latest addition to this toolbox. It provides many improvements, like graphical visualization and improved compilation time. The frameworks of deep learning which are most extensively used are Torch7, Caffe, and Theano, which is mostly usable for convolutional NN*. In addition, different start-up companies have also open sourced their deep learning tools—for example, neon (a Python-based framework developed by Nervana), Deeplearning4J (a Java tool by Sky-mind), and H2O-3 (a Java-based machine learning toolkit with multiple interfaces to Python, R, Scala and others, developed by H2O).

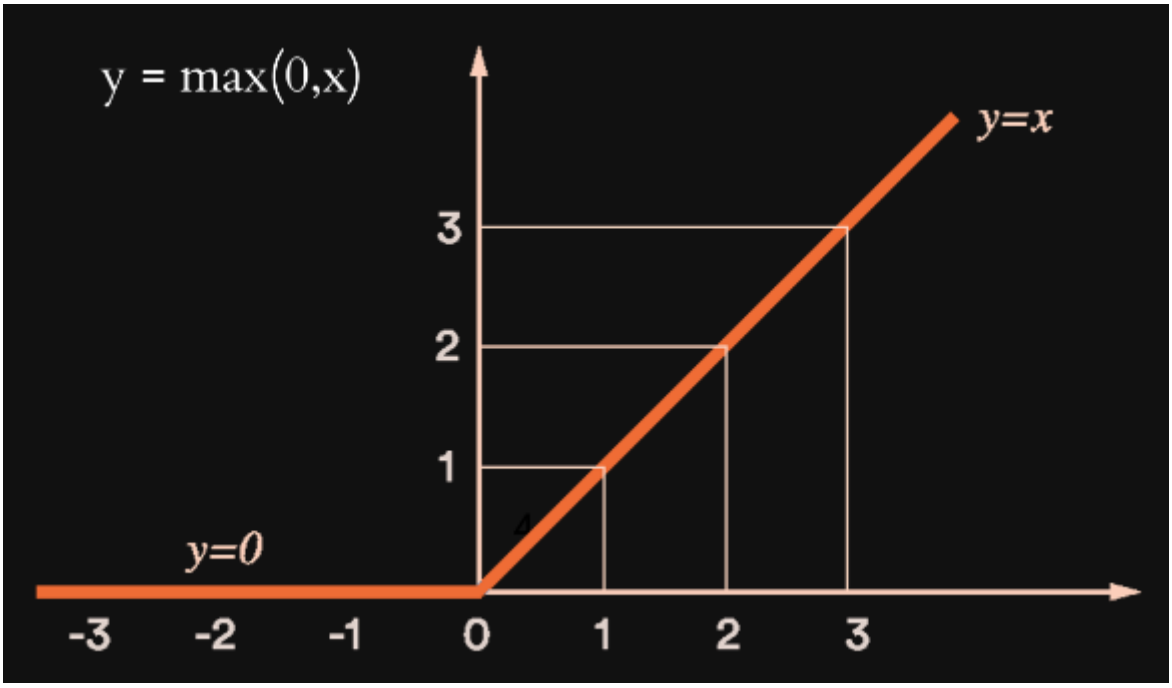


Fig 3.2

(*ML-> Machine Learning ,*NN-> Neural Networks)

Chapter – 4

Implementing the handwritten digit's recognition Model

Below is the code snippet of the system and their uses

To import the needful python packages, we need to import all these packages which include tensor flow and metrics as well.

```
import pandas as pd
import tensorflow as tf
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
from tensorflow.python.keras.metrics import accuracy
```

To take a MNIST dataset as reference, the MNIST database full formed as Modified National Institute of Standards and Technology database is a huge database of digit that are handwritten that is primarily used for the training of different image systems that are being processed.

```
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test,y_test)=mnist.load_data()
# split the data in training set as tuple
```

Normalizing generally means to again scalling the real-valued numbers which has arguments from 00 to 11.

Data normalization mainly used in ML to make model training less delicate to overall scale the main features. It allows our Code to work efficiently and smoothly and, in result gives a more accurate code and output.

```
x_train = tf.keras.utils.normalize(x_train , axis = 1)
x_test = tf.keras.utils.normalize(x_test , axis = 1)
```

```
model= tf.keras.models.Sequential()
model.add(tf.keras.layers.Flatten(input_shape=(28,28)))
model.add(tf.keras.layers.Dense(units=128,activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(units=128,activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(units=10,activation=tf.nn.softmax))
model.compile(optimizer='adam' , loss='sparse_categorical_crossentropy',metrics=['accuracy'])
model.fit(x_train,y_train, epochs=3)
#As the number of epochs increases beyond 11,chance of overfitting of the model on training data

loss , accuracy  =model.evaluate(x_test,y_test)
print(accuracy)
print(loss)



for x in range(1,5):
    # now we are going to read images it with open cv

    img=cv.imread(f'{x}.png')[:,:,0]#all of it and 1st and last one
    img=np.invert(np.array([img]))#invert black to white in images so that model wont get confues
    prediction=model.predict(img)
    print("----------------")
    print("The predicted value is : ",np.argmax(prediction))
    print("----------------")
    plt.imshow(img[0],cmap=plt.cm.binary)#change the color in black and white
    plt.show()
```

CHAPTER 5

CONCLUSION AND RESULT

This paper has practiced machine learning techniques including use of Tensorflow to obtain a appropriate recognition of digit. This study built handwritten recognizers evaluated their performances on MNIST dataset and hence training speed gets improved as well as the recognition performance. As the rate of error obtained is of 1.25 and training accuracy is 98% and test accuracy 97% therefore demonstrating significant and promising performance. Thus by this model we have achieved success in properly of identifying the digits drawn at different angles and properly displaying the correct digit at a single turn along with accuracy. The property to recognize the introduced digit according to the formations made and according to the values in the MNIST dataset will be shown by system.

RESULT

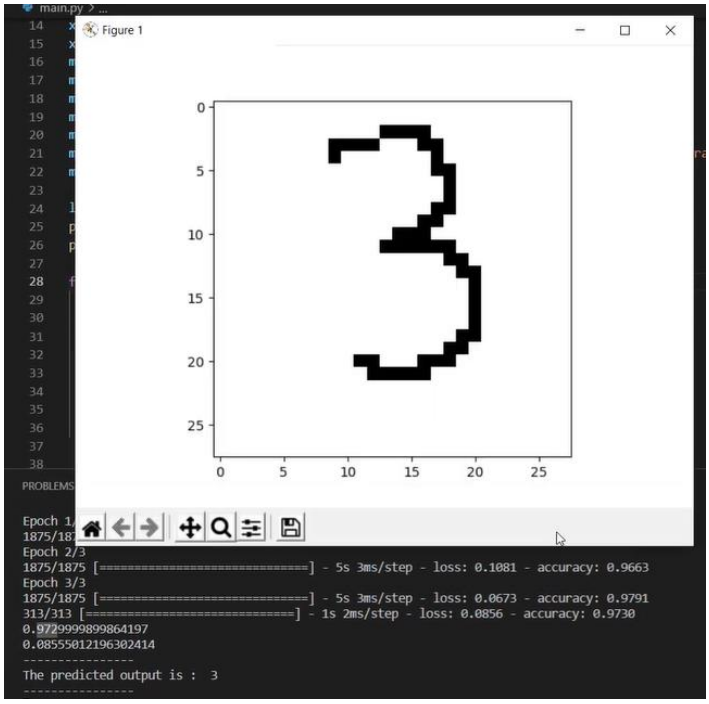


Fig 5.1 Three as the output and the predicted text

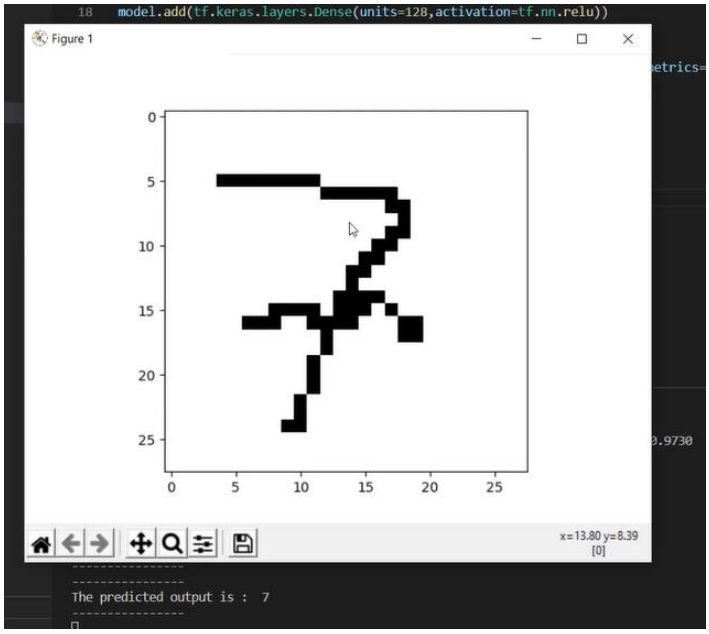


Fig 5.2 Seven as the output and the predicted text

REFERNECES

1. Assegie, Tsehay Admassu, and Pramod Sekharan Nair. "Handwritten digits recognition with decision tree classification: a machine learning approach." International Journal of Electrical and Computer Engineering (IJECE) 9.5 (2019): 4446-4451.

2. Alsaafin, Areej, and Ashraf Elnagar. "A minimal subset of features using feature selection for handwritten digit recognition." Journal of Intelligent Learning Systems and Applications 9.4 (2017): 55-68.

3. Tandon, Shikhar, et al. "Handwritten Digits Recognition Using Machine learning." International Journal of Information Sciences and Application (IJISA) (2019).

4. Jiang, Weiwei. "MNIST-MIX: a multi-language handwritten digit recognition dataset." IOP SciNotes 1.2 (2020): 025002.

5. Thangamariappan, P., and J. C. Pamila. "Handwritten recognition by using machine learning approach." Int. J. Eng. Appl. Sci. Technol (2020): 564-567

6. Kussul, Ernst, and Tatiana Baidyk. "Improved method of handwritten digit recognition tested on MNIST database." Image and Vision Computing 22.12 (2004): 971-981.

7. Wu, Ming, and Zhen Zhang. "Handwritten digit classification using the mnist data set." Course project CSE802: Pattern Classification & Analysis (2010).

8. El Kessab, B., et al. "Extraction method of handwritten digit recognition tested on the mnist database." International Journal of Advanced Science & Technology 50 (2013): 99-11.

9. Kwon, Joonyub, and Sunhee Kim. "Design of a Low-area Digit Recognition Accelerator Using MNIST Database." JOIV: International Journal on Informatics Visualization 6.1 (2022): 53-59.

10. Jadhav, Rinku, et al. "HANDWRITTEN DIGIT RECOGNITION USING MACHINE LEARNING ALGORITHM."

DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

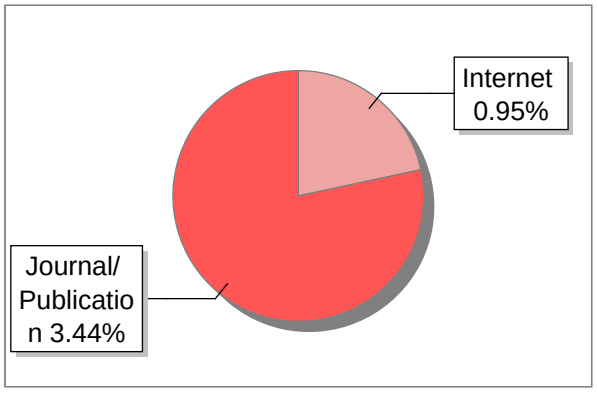
Author Name	vaibhav
Title	report
Paper/Submission ID	565869
Submission Date	2022-07-22 10:44:00
Total Pages	14
Document type	Project Work

Result Information

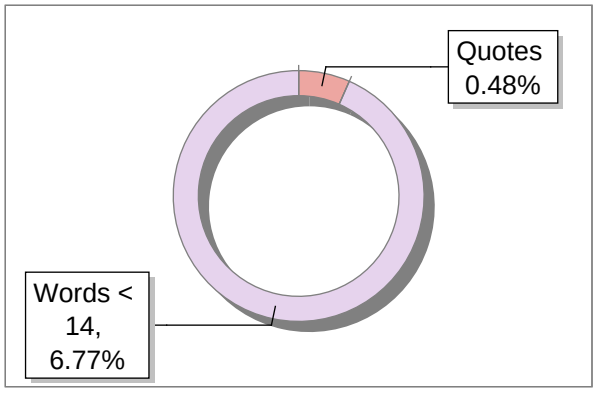
Similarity **7 %**

1 10 20 30 40 50 60 70 80 90

Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

7	10	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	Incremental and Parallel Machine Learning Algorithms With Automated Learning Rat by Hishinuma-2019	1	Publication
2	Analyzing the robustness of an array of wireless access points to mobi by Schweitzer-217	1	Publication
3	IEEE 2013 4th International Conference on Computer and Communication by	1	Publication
4	moam.info	1	Internet Data
5	moam.info	1	Internet Data
6	www.reflectingeducation.net	1	Publication
7	SUITABILITY OF EXTRACTANTS, DETERMINATION OF CRITICAL LIMIT OF NICKEL by Omkar Kumar - 2017 , krishikosh	<1	Publication
8	researches and application of basic education state database of regular colleges and universities	<1	Publication
9	etd.aau.edu.et	<1	Publication
10	IEEE 2018 International Conference on Electrical Engineering and In, by Fauzan, Muhammad Ar- 2018	<1	Publication