Intelligent Handwritten Digit Identification System For Computer Applications Using IBM Watson Studio

Mini Project Report

Submitted By

(Batch No: CSE_005)

GANTA SUCHENDAR REDDY(18UK1A0516)

KANKATI PRANITHA(18UK1A0525)

MUSHIKA SAIRAM(18UK1A0535)

JANAGANI VAMSHI KRISHNA(18UK1A0576)

INDEX

1. INTRODUCTION

1. 1 OVERVIEW

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI

1.2 PURPOSE:

Recognizing handwritten numbers is a piece of cake for humans, but it's a non-trivial task for machines. Nowadays, with the advancement of machine learning, people have made machines more and more capable of performing this task. The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image. In order to analyze the performance of the methods, data is needed to be used for training using machine learning methods.

2. Literature Survey

2.1 EXISTING SOLUTION

Automatic handwritten digits extracts from images is a crucial role for creating documents and processing the systems. The main purpose is to find out the rules to be used in the AHDR for document images using machine learning methods. The field studied in this work is to recognize the corrupted handwritten digits and increase the reliability of the result of the recognition process and to speed up the collecting training and test data from handwritten digit strings. The overall recognition process consists of preprocessing, segmentation, classification and finally recognition of given input data.

In the paper, the authors have conducted research related to "Handwritten Word Recognition Using Multi-view Analysis". The major contribution of this research is a solution to the problem of efficiently recognizing handwritten words from a limited size lexicon.

The authors of another paper have conducted research related to "Handwriting Recognition On Form Document". The author used Freeman Chain Code, with the division of a region into nine sub-regions, histogram normalization of chain code as feature extraction and Artificial Neural Networks, to classify the characters on the form document.

PROPOSED SOLUTION:

Artificial Neural Networks:

Neural networks can be known as a set of algorithms, loosely modeled after the brain of humans that are designed in a way to recognize patterns. They are capable of interpreting data using a type of machine perception, clustering or labeling raw input.

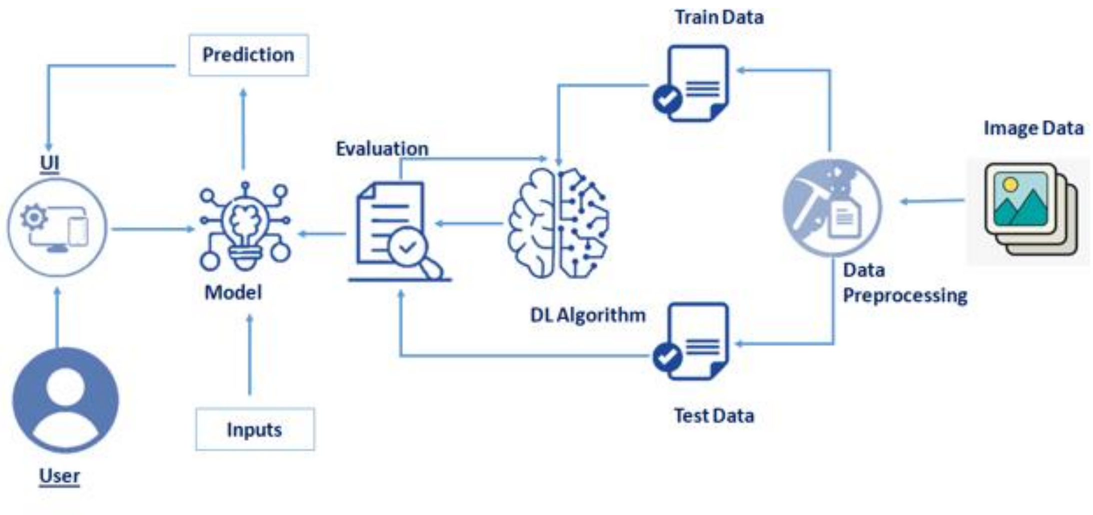
The patterns recognized by them are contained in vectors, numerical, into which all the data of the real-world, be it text, sound, time series, or images, are supposed to be translated. Along with various advantages of neural networks, the most common ones are that they help us classify and cluster. They can be considered as a classification of the clustering layer maintained above the data that you store and manage. They allow you to group the data that is unlabeled based on similarities between example inputs, and they are responsible for the classification of data when the dataset is labeled by them to train on. To be more precise, neural networks can be considered as components of larger applications of machine learning as a service that involve algorithms for classification, regression, and reinforcement learning.

Convolutional Neural Networks :

The field of machine learning has taken a dramatic twist in recent times, with the rise of the Artificial Neural Network (ANN). These biologically inspired computational models are able to far exceed the performance of previous forms of artificial intelligence in common machine learning tasks. One of the most impressive forms of ANN architecture is that of the Convolutional Neural Network (CNN). CNNs are primarily used to solve difficult image-driven pattern recognition tasks and with their precise yet simple architecture, offers a simplified method of getting started with ANNs.

THEORITICAL ANALYSIS :

ARCHITECTURE :



REQUIREMENTS :

Software specifications :

REQUIREMENT	SPECIFICATION
Anaconda Navigator	You must have anaconda installed in your device prior to begin.
Spyder, Jupyter Notebook, Flask Frame work	1. One should have Spyder and Jupyter notebook. 2. One should install flask framework through anaconda prompt for running their web application 3. We need to build the model using jupyter notebook with all the imported packages.
Web browser	For all Web browsers, the following must be enabled: • cookies • JavaScript

Hardware Specifications :

REQUIREMENT	**SPECIFICATION**
Operating system	Microsoft Windows UNIX Linux®
Processing	Minimum: 4 CPU cores for one user. For each deployment, a sizing exercise is highly recommended.
RAM	Minimum 8 GB.
Operating system specifications	File descriptor limit set to 8192 on UNIX and Linux
Disk space	A minimum of 7 GB of free space is required to install the software.

EXPERIMENTAL INVESTIGATIONS :

The result of this recognizes the digit. Recognizing handwritten numbers is a piece of cake for humans, but it's a non-trivial task for machines. Nowadays, with the advancement of machine learning, people have made machines more and more capable of performing this task. The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

CONTROL FLOW OF THE SOLUTION :

Project Flow:

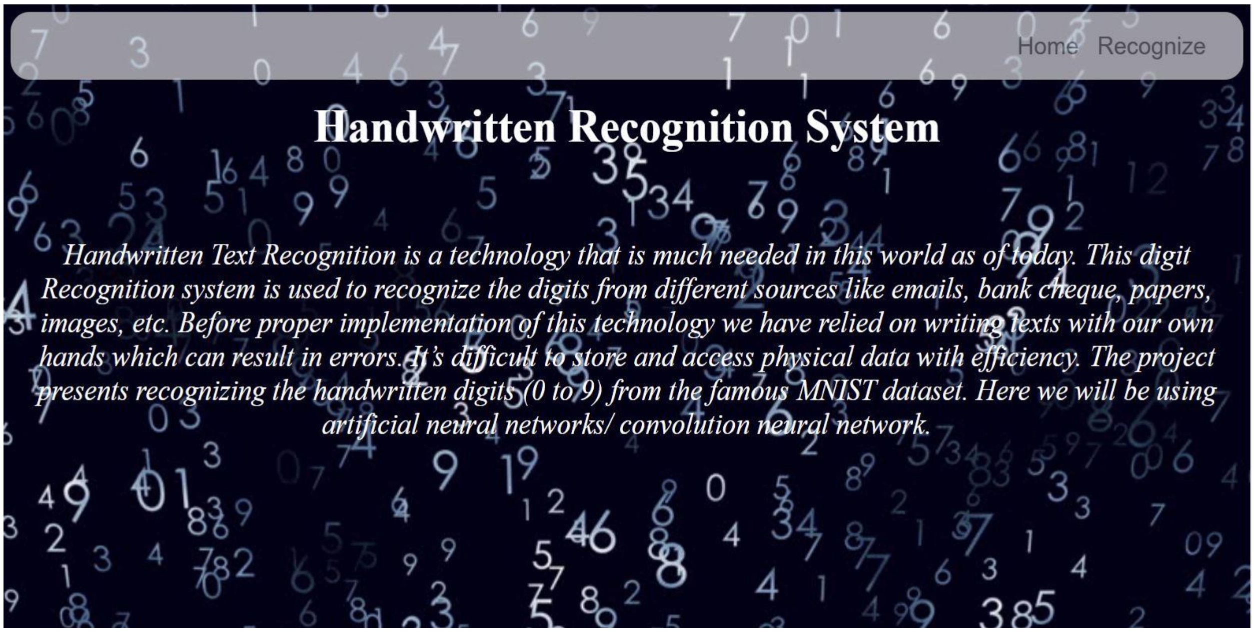
- The user interacts with the UI (User Interface) to upload the image as input
- The uploaded image is analyzed by the model which is integrated
- Once the model analyses the uploaded image, the prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below:

- Understanding the data.
  - Importing the required libraries
  - Loading the data
  - Analyzing the data
  - Reshaping the data.
  - Applying One Hot Encoding
- Model Building
  - Creating the model and adding the input, hidden and output layers to it
  - Compiling the model
  - Training the model
  - Predicting the result
  - Testing the model by taking image inputs
  - Saving the model
- Application Building
  - Create an HTML file
  - Build Python Code

RESULT :

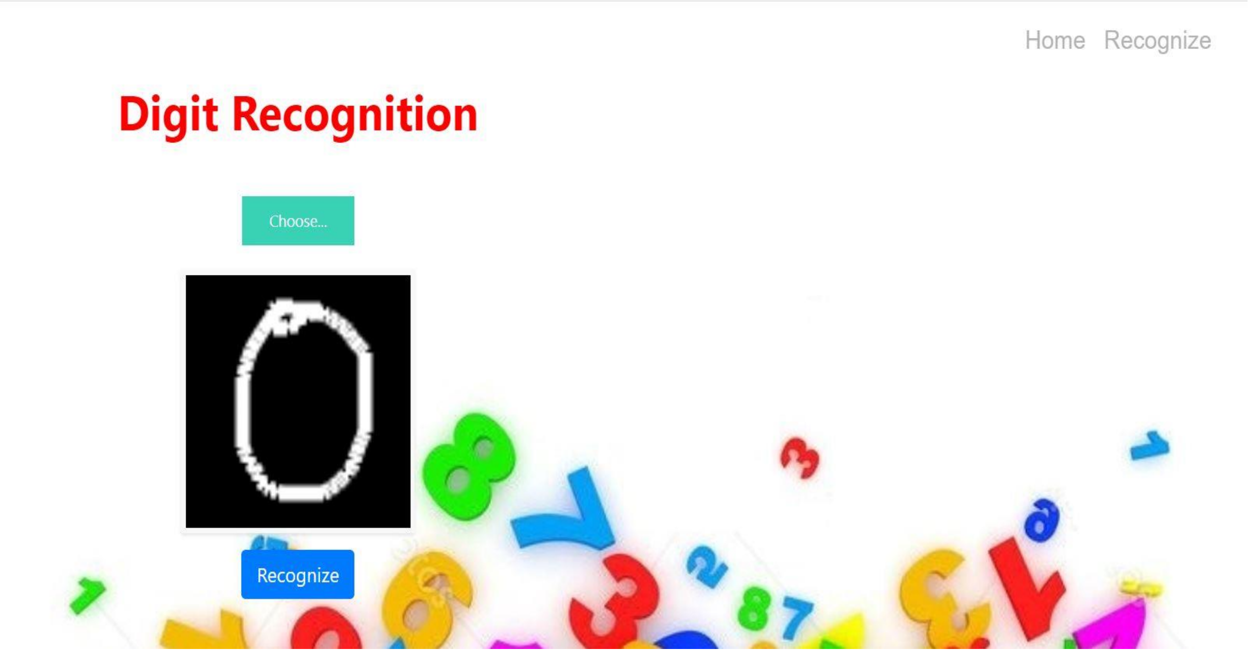
There comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI.



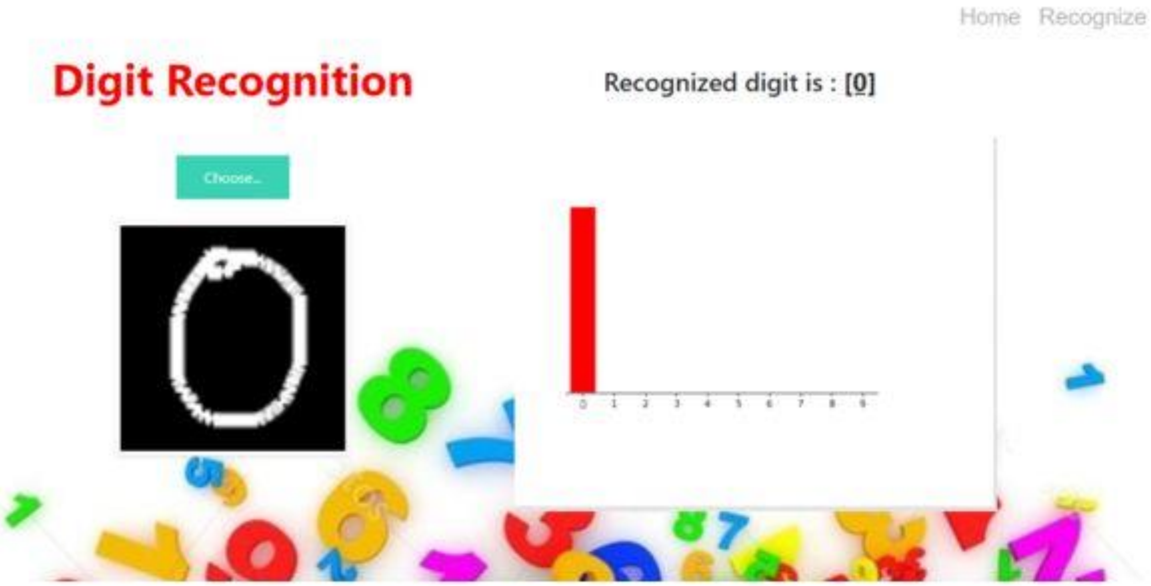
Handwritten Recognition System

Handwritten Text Recognition is a technology that is much needed in this world as of today. This digit Recognition system is used to recognize the digits from different sources like emails, bank cheque, papers, images, etc. Before proper implementation of this technology we have relied on writing texts with our own hands which can result in errors. It's difficult to store and access physical data with efficiency. The project presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset. Here we will be using artificial neural networks/ convolution neural network.

When we click on recognize option, we can upload the image which has to be recognized.



Then the output will be recognized as:



Recognized digit is : [0]

ADVANTAGES :

There are various advantages of neural networks, some of which are discussed below:

1) Store information on the entire network

2) The ability to work with insufficient knowledge:

3) Good falt tolerance:

4) Distributed memory:

5) Gradual Corruption:

6) The ability of parallel processing:

Advantages of Convolutional Neural Networks:

Utilizing intelligent automation in order to boost up the growth process of your business might become your best decision to stay ahead of your competitors.

1) Cost and Time Benefits

2) Quality and Accuracy In Results

3) Job Growth

Disadvantages of ANN:

- Hardware Dependence
- Unexplained functioning of the network
- Assurance of proper network structure
- The difficulty of showing the problem to the network
- The duration of the network is unknown

Disadvantages of CNN :

- Classification of Images with different Positions
- Adversarial examples
- Coordinate Frame
- Other minor disadvantages like performance

Applications:

- ✓ Text Classification and Categorization
- ✓ Named Entity Recognition(NER)
- ✓ Part of Speech Tagging
- ✓ Semantic Parsing and Question Answering
- ✓ Paraphrase Detection
- ✓ Language Generation and Multi-document Summarization
- ✓ Machine Translation
- ✓ Speech Recognition
- ✓ Character Recognition
- ✓ Spell Checking

CONCLUSION :

The primary focus of this project is to build an Automatic Handwritten Digit Recognition on document images. So, we used Artificial Neural Networks and Convolutional Neural Networks to recognize the digits. There are also many wide range of applications which can be built using the above model. Although there are some limitations which may arise when certain constraints are involved.

We have found that CNN gave the most accurate results for handwritten digit recognition. So, this makes us conclude that CNN is best suitable for any type of prediction problem including image data as an input. Next, by comparing execution time of the algorithms we have concluded that increasing the number of epochs without changing the configuration of the algorithm is useless because of the limitation of a certain model and we have noticed that after a certain number of epochs the model starts overfitting the dataset and give us the biased prediction.

FUTURE SCOPE :

A new method can be proposed where improvements has to be made. Thus, there is a place for some future work such as:

● To reduce the complexity of the algorithm, it's better to reduce the number of hypothesis to function the algorithm faster.

The future development of the applications based on algorithms of deep and machine learning is practically boundless. In the future, we can work on a denser or hybrid algorithm than the current set of algorithms with more manifold data to achieve the solutions to many problems. In future, the application of these algorithms lies from the public to high-level authorities, as from the differentiation of the algorithms above and with future development we can attain high-level functioning applications which can be used in the classified or government agencies as well as for the common people.

BIBILOGRAPHY :

1. “Handwritten Digit Recognition using Machine Learning Algorithms”, S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana & Abdullah Al Jobair.
2. “Handwritten Digit Recognition Using Deep Learning”, Anuj Dutt and Aashi Dutt.
3. “Handwritten recognition using SVM, KNN, and Neural networks”, Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sharif.

APPENDIX:

SOURCE CODE :

Python app.py

```
from flask import Flask, render_template, request
from PIL import Image
import numpy as np
from tensorflow.keras.models import load_model
import tensorflow as tf
app = Flask(__name__)
model = load_model("C:/Users/Shravan/HRS/HRS/models/mnistCNN.h5")
@app.route('/')
def upload_file():
    return render_template('main.html')
@app.route('/about')
def upload_file1():
     return render_template('main.html')
@app.route('/upload')
def upload_file2():
    return render_template('index6.html')
@app.route('/predict', methods = ['POST'])
def upload_image_file():
    if request.method == 'POST’:
        img = Image.open(request.files['file'].stream).convert("L")
        img = img.resize((28,28))
        im2arr = np.array(img)
       im2arr = im2arr.reshape(1,28,28,1)
       y_pred = model.predict_classes(im2arr)
           print(y_pred)
       if(y_pred == 0) :
                  return render_template("0.html",showcase =  str(y_pred))
          elif(y_pred == 1) :
           return render_template("1.html",showcase =  str(y_pred))
       elif(y_pred == 2) :
           return render_template("2.html",showcase =  str(y_pred))
       elif(y_pred == 3) :
           return render_template("3.html",showcase =  str(y_pred))
       elif(y_pred == 4) :
           return render_template("4.html",showcase =  str(y_pred))
       elif(y_pred == 5) :
           return render_template("5.html",showcase =  str(y_pred))
       elif(y_pred == 6) :
           return render_template("6.html",showcase =  str(y_pred))
       elif(y_pred == 7) :
           return render_template("7.html",showcase =  str(y_pred))
       elif(y_pred == 8) :
           return render_template("8.html",showcase =  str(y_pred))
       else:
            return render_template("9.html",showcase =  str(y_pred))
    else:
           return None
if __name__ == '__main__':
    app.run(host='127.0.0.1', port=5000,debug=True)
```

GITHUB LINK : https://github.com/smartinternz02/SI-GuidedProject-5649-1634041746.git