Handwritten Digit Recognition

Internship Project Report

Ankit

Internship Domain: Data Science ankitthakur592000@gmail.com

Abstract: - This project focuses on developing a handwritten digit recognition system using the MNIST dataset. The MNIST dataset is a widely used benchmark dataset in the field of machine learning and computer vision. The primary goal of this project is to build and train a neural network model that can accurately classify handwritten digits from 0 to 9.

Introduction: -

Background: Handwritten digit recognition is a fundamental problem in the field of computer vision and pattern recognition. It has various applications, including postal code recognition, check processing, and digitized document analysis. The MNIST dataset is a collection of 28x28 grayscale images of handwritten digits and has been a standard benchmark for evaluating machine learning algorithms.

Objective: The main objective of this project is to develop a machine learning model capable of accurately recognizing handwritten digits from the MNIST dataset. By employing neural networks, I aim to achieve high accuracy in classification.

Dataset Overview: -

MNIST dataset: The MNIST dataset (Modified National Institute of Standards and Technology database) is a collection of 28x28 grayscale images of handwritten digits from 0 to 9. It's commonly used for training and testing machine learning models, especially in image classification. The dataset consists of 60,000 training images and 10,000 test images, each labeled with the corresponding digit. It's a foundational dataset in the field of machine learning and computer vision. While the MNIST dataset has been valuable for its simplicity and ease of use, it's worth noting that it might not fully reflect the challenges posed by real-world data. For instance, the images are relatively small and the digits are centered and well-structured. As a result, models that perform well on MNIST might not necessarily generalize well to more complex datasets.

Fig 02: Sample images from MNIST test dataset

Methodology of handwritten digit recognition: -

Convolutional Neural Networks (CNNs) have emerged as a breakthrough in the realm of image and digit recognition, predominantly owing to their intrinsic ability to autonomously extract essential features. This attribute eliminates the necessity for human intervention, setting CNNs apart from their predecessors. Specifically tailored for image and digit classification tasks, CNNs hold a distinct advantage.

In comparison to conventional neural networks, CNNs mirror the neural architecture found in the frontal lobe—the region responsible for processing visual inputs in both humans and animals. This arrangement obviates the need to process images in fragmented portions, which had been a challenge for earlier networks. By spanning the entire visual field through the layering of neurons, CNNs effectively address this limitation.

The proposed system employs a data flow diagram that caters to two modes of input: user-provided digit images or data from the MNIST dataset. Following preprocessing, the accuracy of identified digits is benchmarked across various classifiers to yield comprehensive results. This accuracy is not only quantified but also accompanied by a measure of correctness.

Integral to the system is the integration of the MNIST dataset, facilitated by the Keras programming interface. Comprising 28x28 matrices of pixel values and

corresponding labels, this dataset serves as a foundation for training and testing. The pixel values span a range of 0 to 255, symbolizing black and white, respectively.

Python, as a high-level programming language, synergizes seamlessly with this system. Its concise syntax facilitates effective communication of concepts, and its focus on code readability enhances development speed. Through Python's frameworks, the system operates with efficiency and flexibility, aligning well with the demands of image and digit recognition tasks.

To encapsulate, Convolutional Neural Networks (CNNs) offer an automated and highly accurate solution for image and digit recognition. The fusion of brain-inspired architecture, MNIST dataset utilization, and Python's prowess culminates in a robust system that deftly tackles these classification challenges.

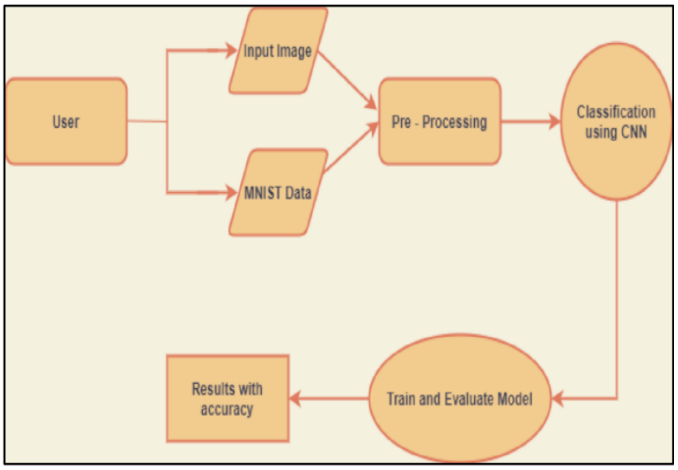
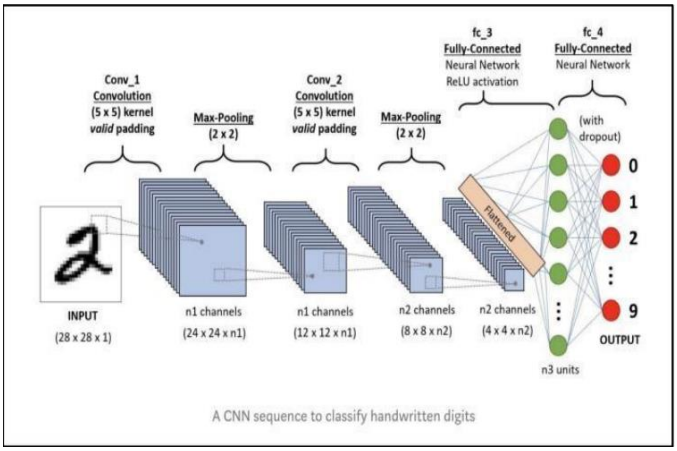


Fig 02: Block diagram

Architecture:

The image is compressed to 28 by 28 pixels in the first step. This compression is required in order to obtain quick results. Convolution 1 is the second step, which is kernel valid padding, which means there is no padding at all. The supplied image is preserved in its original state. The image is compressed to 24 by 24 pixels once again in this step. The image is further compressed to 12 by 12 pixels using max pooling. After that, the image is reduced to 4 x 4 pixels using a convolution filter with maximum pooling. Finally, the rectified linear unit is activated before CNN. Every pixel in the rectified linear activation process is given a 0 or 1 value based on the preceding filters procedure. These values are then flattened. Before being transmitted to the MNIST training dataset. We acquire our result after comparing it to its data set.



System Architecture:

```
from tensorflow.keras import layers
from tensorflow.keras import models
from keras.datasets import mnist
from keras.utils import to_categoricalA
```

```
from tkinter import *
import cv2
import numpy as np
from PIL import ImageGrab
from tensorflow.keras.models import load_model
```

- **tensorflow. keras. layers:** These are modules from the TensorFlow library's Keras API, which is used for building and training neural networks. The layers module provides classes to define the layers of a neural network, such as dense (fully connected) layers, convolutional layers, etc. The model's module provides tools to define and compile neural network models using these layers.

- **keras.datasets.mnist:** This module from Keras provides access to the MNIST dataset, a widely used dataset of handwritten digits. The dataset includes both training and testing sets of images of digits 0 to 9. It's a common benchmark for testing the performance of machine learning models, particularly in the field of computer vision.

- **keras.utils.to_categorical:** This function from Keras is used to perform one-hot encoding. One-hot encoding is a technique used to convert categorical data (like class labels) into a binary matrix format, which is often required when training neural networks. It's useful for dealing with multi-class classification problems.

- **tkinter:** Tkinter is a standard GUI (Graphical User Interface) library for Python. It provides tools for creating windows, dialogs, buttons, labels, and other GUI elements. It's often used to develop desktop applications with graphical interfaces.

- **cv2 (OpenCV):** OpenCV is an open-source computer vision library that provides tools for image and video analysis, processing, and computer vision tasks. It's commonly used for tasks like image filtering, edge detection, object detection, and more.

- **NumPy**: NumPy is a fundamental package for numerical computations in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays. NumPy is extensively used in scientific computing, data analysis, and machine learning.

- **PIL (Python Imaging Library):**The Python Imaging Library is a library for opening, manipulating, and saving various image file formats. It provides a range of image processing capabilities, such as resizing, cropping, rotating, and more. The Image Grab module within PIL allows you to capture screenshots of the screen.

- **tensorflow.keras.models.load_model:**This function from TensorFlow's Keras module is used to load a pre-trained neural network model that was previously saved. It's useful for reusing a trained model to make predictions on new data without having to retrain the model.

Result: - Before the input is fed into the model for processing, analyze and get the output, the model needs to be trained first. In the program, the epochs must be set to 10. By setting this, the model will be trained 10 times. It can be changed to how many times you want the model to be trained depending on the error rate of the prediction. After executing and training the model, the GUI program will be executed to allow users to input the number (draw).

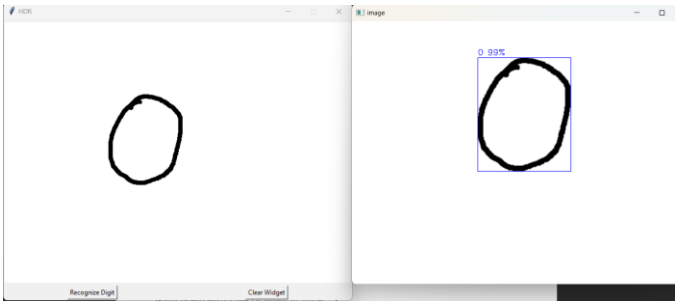


Fig 03 Recognizing the number 0.

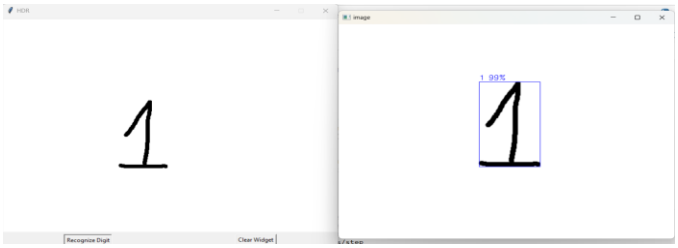


Fig 04 Recognizing the number 1.

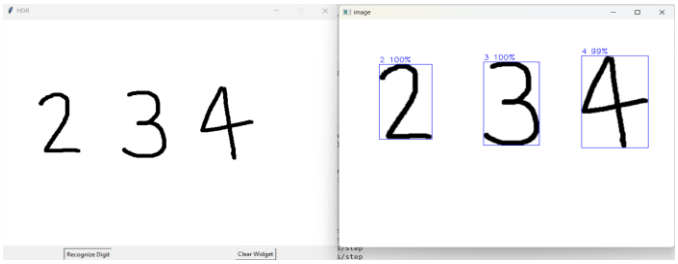


Fig 04 Recognizing the number 2,3,4

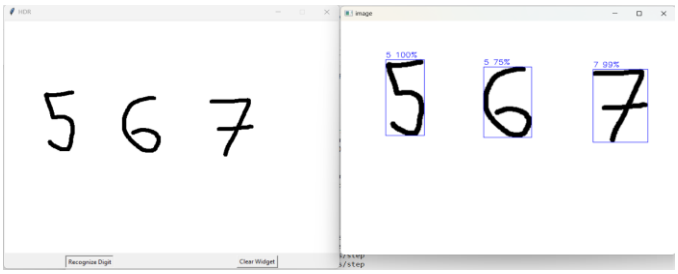


Fig 05 Recognizing the number 5,6,7

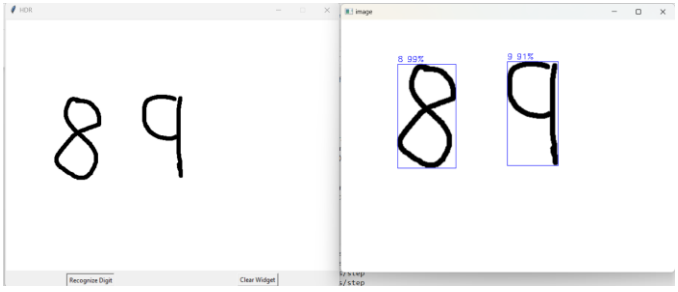


Fig 06 Recognizing the number 8,9

Conclusion: - In this endeavor, we have successfully implemented a Handwritten Digit Recognizer, enabling the identification of numerical digits across diverse handwriting styles. Utilizing Convolutional Neural Networks (CNN), a prominent machine learning technique, we have meticulously trained and evaluated the model using the provided dataset for comprehensive comparison and insightful analysis.

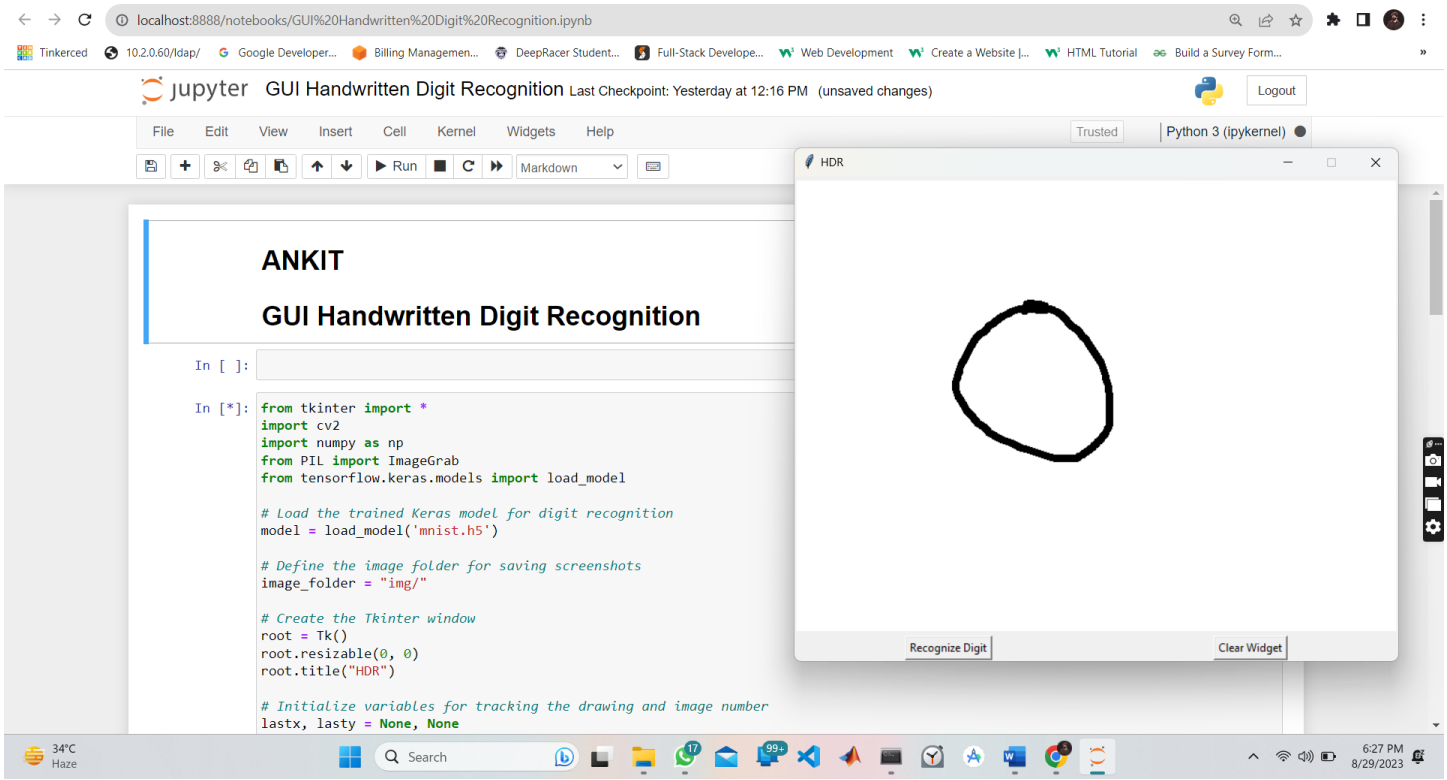
By harnessing the power of deep learning, we've achieved a remarkable level of accuracy. Our implementation employs Keras as the backend framework and leverages TensorFlow for the underlying machine learning operations, ensuring the delivery of robust and precise outcomes.

In a world where data generation is experiencing an exponential surge, the significance of machine learning becomes increasingly pronounced in our daily lives. The implications of this project extend into the future, where machine learning stands poised to play a pivotal role in shaping our interactions with the world. This project underscores the prowess of deep learning and its multifaceted applications. It showcases how this technology can be harnessed to decipher intricate patterns within images, identify ailments, and more. As a result, we affirm that our Handwritten Digit Recognizer exhibits a profound ability to accurately discern various numerical digits. Furthermore, the strides made in Deep Learning pave the way for transformative impacts in the times ahead.

Summary: In this project, we have successfully developed a Handwritten Digit Recognizer. This model effectively identifies numerical digits across different styles of handwriting. By employing Keras as the backend and TensorFlow as the foundational software for machine learning tasks, we have achieved a high level of accuracy in digit recognition.

The project emphasizes the potential of deep learning techniques and their practical applications. As data generation continues to grow exponentially, the role of machine learning becomes increasingly significant in our daily lives. This project serves as a testament to the capabilities of deep learning in deciphering complex patterns within images and potentially diagnosing diseases.

In conclusion, the Handwritten Digit Recognizer showcases its ability to accurately interpret various numerical digits, highlighting the effectiveness of deep learning. Furthermore, the project underscores the transformative power of Deep Learning in shaping our future endeavors.



Video Link:

Real time simulation of this project in the video

References:

[1] https://cse.anits.edu.in/projects/projects2021A10.pdf

[2] https://medium.com/the-andela-way/applying-machine-learning-to-recognize [3] M. Kloss, D. Bern Reuther, M. Paulik, S. Stucker, S. Vogel, and A. Waibel, “Open Domain Speech Recognition & Translation: Lectures and Speeches,” in Proceedings of ICASSP, 2006.

[3] https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/

[4] https://www.tensorflow.org/learn.

[5] https://chat.openai.com/

[6]https://towardsdatascience.com/image-recognition-with-machine-learning-on-python-image-processing-3abe6b158e9a

[7] Handwritten Digit Recognition in www.geeksforgeeks.org