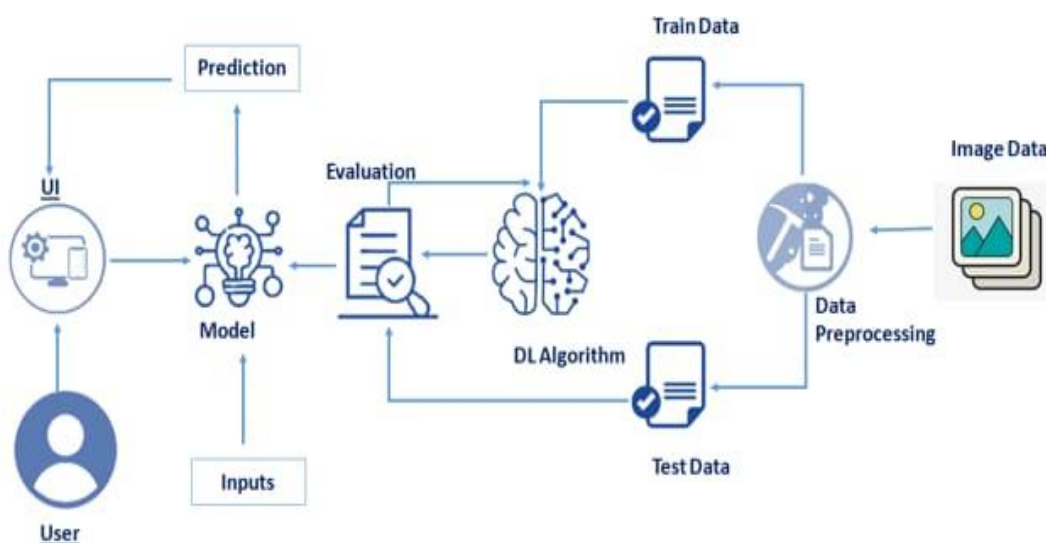


A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

ABSTRACT

In this digital world, everything including documents, notes is kept in digital form. The requirement of converting these digital documents into processed information is in demand. 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. We create a model that recognize multiple-digits present in the image in offline mode. The future scope of this method is, we can recognize number in License plate, bank cheques and postal mail sorting. We get a predicted result in two manner one is, the recognized digits is showed in the interface and the another manner is, we can get a predicted result through voice mode. This means the model tell the multiple-digit in voice mode

TECHINICAL ARICHTECTURE



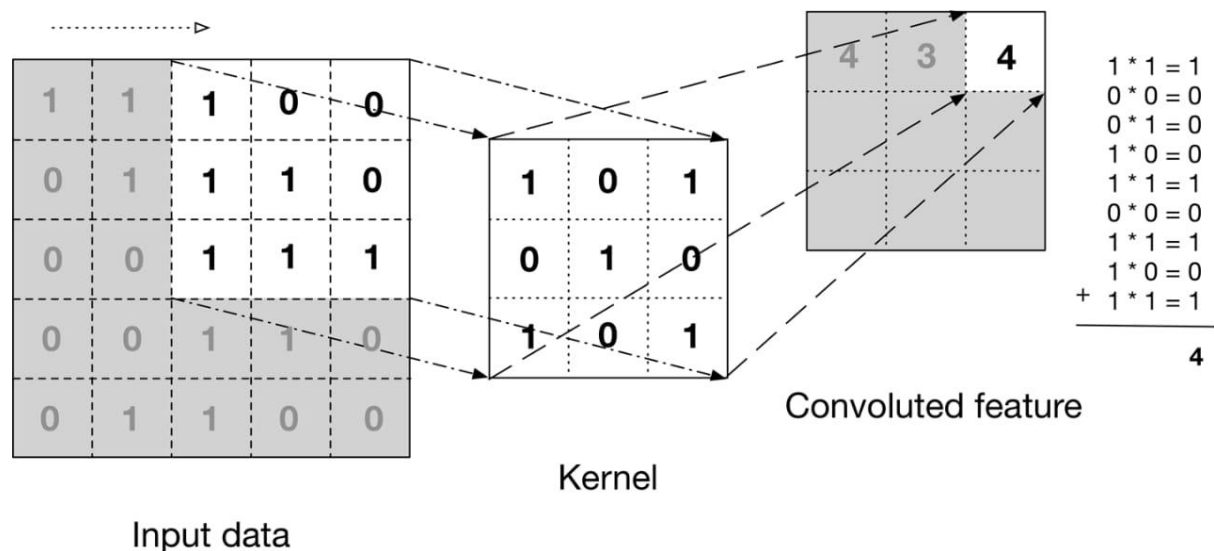
METHODOLOGY

Deep Learning has emerged as a central tool for self-perception problems like understanding images, a voice from humans, robots exploring the world. We aim to implement the concept of Convolutional Neural Network for digit recognition. Understanding CNN and applying it to the handwritten digit recognition system is the target of the proposed model. Convolutional Neural Network extracts the features maps from the 2D images. Then it can classify the images using the features maps. The convolutional neural network considers the mapping of image pixels with the neighbourhood space rather than having a fully connected layer of neurons. The convolutional neural network is a powerful tool in signal and image processing. Even in the fields of computer vision such as handwriting recognition, natural object classification, and segmentation, CNN has been much better tool compared to all other previously implemented tools. The broader aim may be to develop a machine learning model that could recognize people's handwriting.

CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks are deep artificial neural networks. We can use it to classify images (e.g., name what they see), cluster them by similarity (photo search) and perform object recognition within scenes. It can be used to identify faces, individuals, street signs, platypuses and many other aspects of visual data. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels) which have a small receptive field but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product, and producing a 2-dimensional activation map of that filter. As a result, the network learns when they see some specific type of feature at some spatial position in the input. Then the activation maps are fed into a down sampling layer, and like convolutions, this method is applied one patch at a time. CNN has also fully connected layer that classifies output with one label per node. Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a Conv Net, each layer generates several activation functions that are passed on to the next layer.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.



STEPS TO IMPLEMENT HANDWRITTEN DIGIT RECOGNITION

Import the libraries and load the dataset

First, we are going to import all the modules that we are going to need for training our model. The Keras library already contains some datasets and MNIST is one of them. So we can easily import the dataset and start working with it. The `mnist.load_data()` method returns us the training data, its labels and also the testing data and its labels.

Pre-process the data

The image data cannot be fed directly into the model so we need to **perform some operations and process the data** to make it ready for our neural network. The dimension of the training data is (60000,28,28). The CNN model will require one more dimension so we reshape the matrix to shape (60000,28,28,1).

Create the model

Now we will **create our CNN model** in Python data science project. A CNN model generally consists of convolutional and pooling layers. It works better for data that are represented as grid structures, this is the reason why CNN works well for image classification problems. The dropout layer is used to deactivate some of the neurons and while training, it reduces over fitting of the model. We will then compile the model with the Adadelta optimizer.

Train the model

The **model.fit() function** of Keras will start the training of the model. It **takes the training data, validation data, epochs, and batch size**. It takes some time to train the model. After training, we save the weights and model definition in the 'mnist.h5' file.

Evaluate the model

We have 10,000 images in our dataset which will be used to **evaluate how good our model works**. The testing data was not involved in the training of the data therefore, it is new data for our model. The MNIST dataset is well balanced so we can get around 99% accuracy.

Create GUI to predict digits

Now for the GUI, we have created a new file in which we **build an interactive window to draw digits on canvas** and with a button, we can recognize the digit. The Tkinter library comes in the Python standard library. We have created a function **predict_digit()** that takes the image as input and then uses the trained model to predict the digit. Then we **create the App class** which is responsible for building the GUI for our app. We create a canvas where we can draw by capturing the mouse event and with a button, we trigger the predict_digit() function and display the results.

DATASET

In this paper, we used the MNIST database consisting of offline handwritten digits ranging from 0-9. The database was constructed from Special Database 3 (SD-3) and Special Database 1 (SD-1) that contain binary images of handwritten digits. SD-3 was collected among Census Bureau employees, while SD-1 was collected among highschool students. For the results to be independent of both datasets, MNIST dataset was built by mixing NIST SD-1 and

SD-3. The total number of digit image samples (70,000), the total number for training (60,000) and testing (10,000), and the subtotal number for each digit are shown in table 1. Each digit is a gray-level fixed-size image with a size of 28 x 28 (or 784 pixels) in total as the features.



RESULT

We have building a website that recognition multiple digits in a image in a offline mode. Shows the result in two manner one is display the predicted digits in interface and another one is tell the predicted result in voice mode.