

Handwritten Digit Recognizer



Prabhakar Maity

Hrishikesh Kukarni,

Deepak Pawade

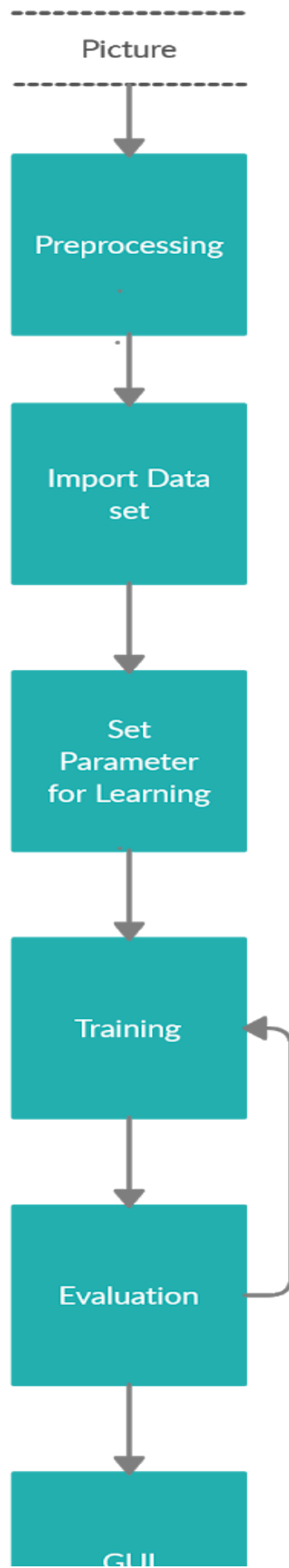
Akshay Agarwal

Digit Recognition System

Digit recognition system is the working of a machine to train itself or recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (say — tax forms) and so on.

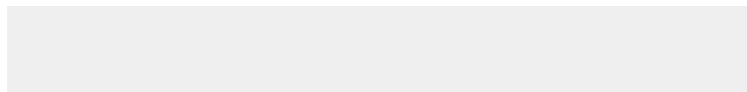
Problems with handwritten digits

The handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person, so the general problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. This problem is faced more when many people write a single digit with a variety of different handwritings. Lastly, the uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits. Now we introduce the concepts and algorithms of deep learning and machine learning.



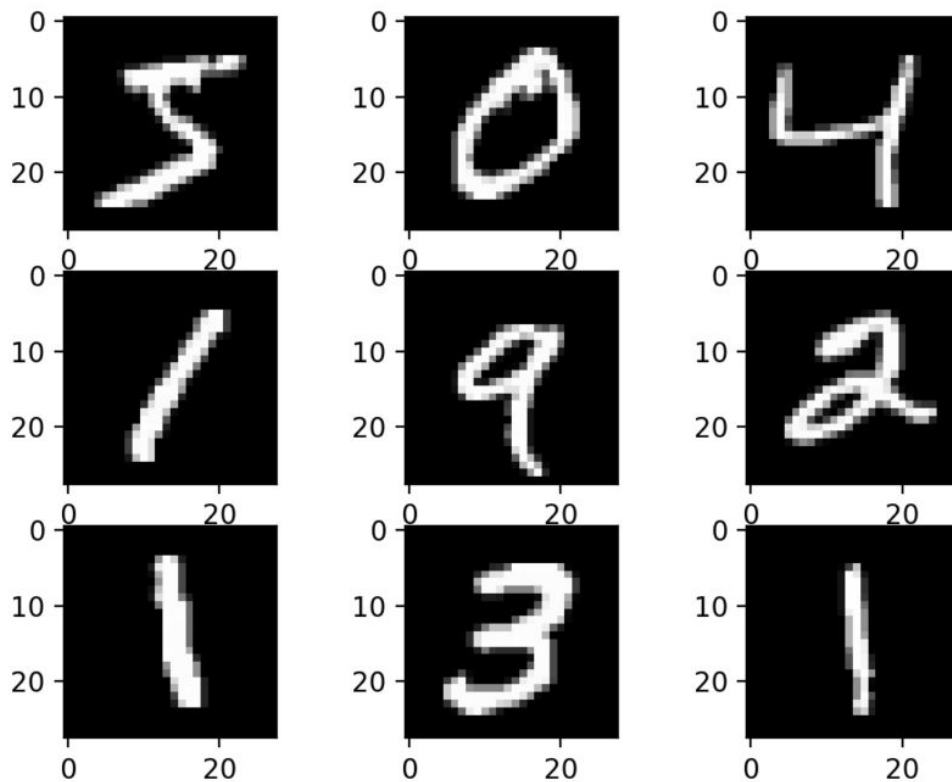
NIST National Institute of Standards and Technology.

The National Institute of Standards and Technology (NIST) is a physical sciences laboratory and a non-regulatory agency of the [United States Department of Commerce](#). Its mission is to promote innovation and industrial competitiveness. NIST's activities are organized into laboratory programs that include nanoscale science and technology, engineering, [information technology](#), neutron research, material measurement, and physical measurement. From 1901–1988, the agency was named the National Bureau of Standards.



MNIST Dataset

Samples provided from MNIST (Modified National Institute of Standards and Technology) dataset includes handwritten digits total of 70,000 images consisting of 60,000 examples in training set and 10,000 examples in testing set, both with labeled images from 10 digits (0 to 9). This is a small segment from the wide set from NIST where size was normalized to fit a 28*28 pixel box and not altering the aspect ratio. Handwritten digits are images in the form of 28*28 gray scale intensities of images representing an image along with the first column to be a label (0 to 9) for every image. The same has opted for the case of the testing set as 10,000 images with a label of 0 to 9.



Plot of a Subset of Images From the MNIST Dataset

Available files in the dataset

So, before starting further deep in this topic, the better point should be to get familiar with the provided dataset. Following points are same in training and testing set along with the set of the images and labels files –

1. Pixels are arranged row-wise, ranging from 0 to 255, as from RGB color code.
2. Background as white (0 value from RGB) and foreground as black (255 value from RGB).
3. Labels of digits classified from 0 to 9.

There are four files of training and testing are:

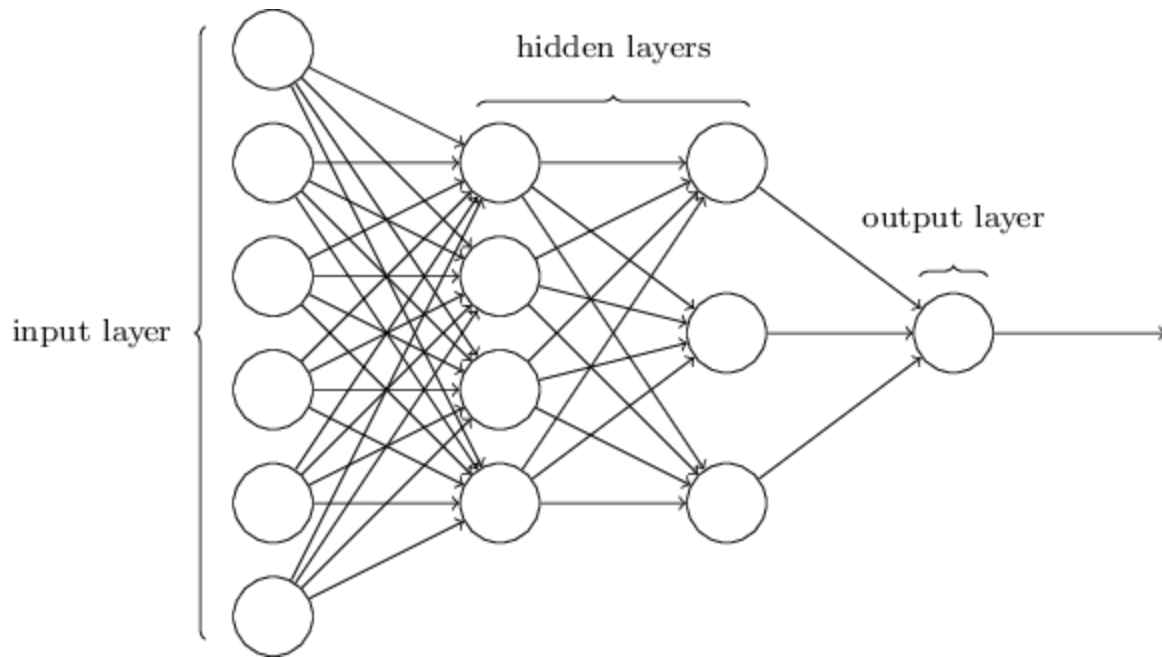
1. Training set images files (train-images-idx3-ubyte)
2. Training set labels file (train-labels-idx1-ubyte)
3. Test set images files (t10k-images-idx3-ubyte)
4. Test set label files (t10k-labels-idx1-ubyte)

CLASSIFIERS

In this section, we will be discussing various algorithms of machine learning and deep learning for making predictions and accuracy. Classifiers in machine learning –

NN (Neural networks)

Neural Networks mimics the working of how our brain works. They have emerged a lot in the era of advancements in computational power.



Neural networks with input, output, and hidden layers

Deep learning is the acronym for Neural Networks, the network connected with multilayers. The layers are composed of nodes. A node is just a perception which takes an input, performs some computation, and then passes through a node's activation function, to show that up to what context signal progress proceeds through the network to perform classification.

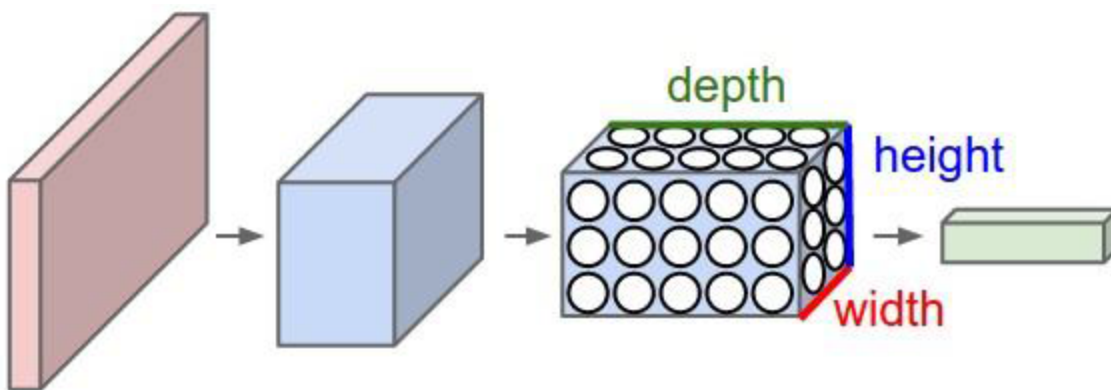
Algorithm

1. Initializing the weights randomly (not by keeping them zero)
2. Implementing forward propagation to achieve $h\theta(x^{(i)})$.
3. Compute cost

4. Now evaluate backpropagation to compute partial derivatives and use gradient checking to confirm that backpropagation is working fine. Then disable gradient checking.
5. Use gradient descent or any built-in optimization function to minimize the cost function with weights of theta Θ .

CNN (Convolutional Neural Network)

Convolutional Neural Networks, CNN has become famous among the recent times. CNN is part of deep, feed forward artificial neural networks that can perform variety of task with even better time and accuracy than other classifiers, in different applications of image and video recognition, recommender system and natural language processing.



Arrangements of neurons in CNN

Use of CNN have spread as Facebook uses neural nets for their automatic tagging algorithms, google for photo search Amazon for their product recommendations, Pinterest for their home feed personalization and Instagram for search infrastructure.

Image classification or object recognition is a problem is passing an image as a parameter and predicting whether a condition is satisfied or not (cat or not, dot or not), or the probability or most satisfying condition for an image. We are able to quickly recognize patterns, generalize from previous information and knowledge.



What We See

```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 1
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 4
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 0
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 7
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 2
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 2
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 6
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 9
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 8
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 9
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 1
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 2
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 5
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 7
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 7
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 3
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 3
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 8
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 0
```

What Computers See

Difference what we see vs what system see

Inputs and output

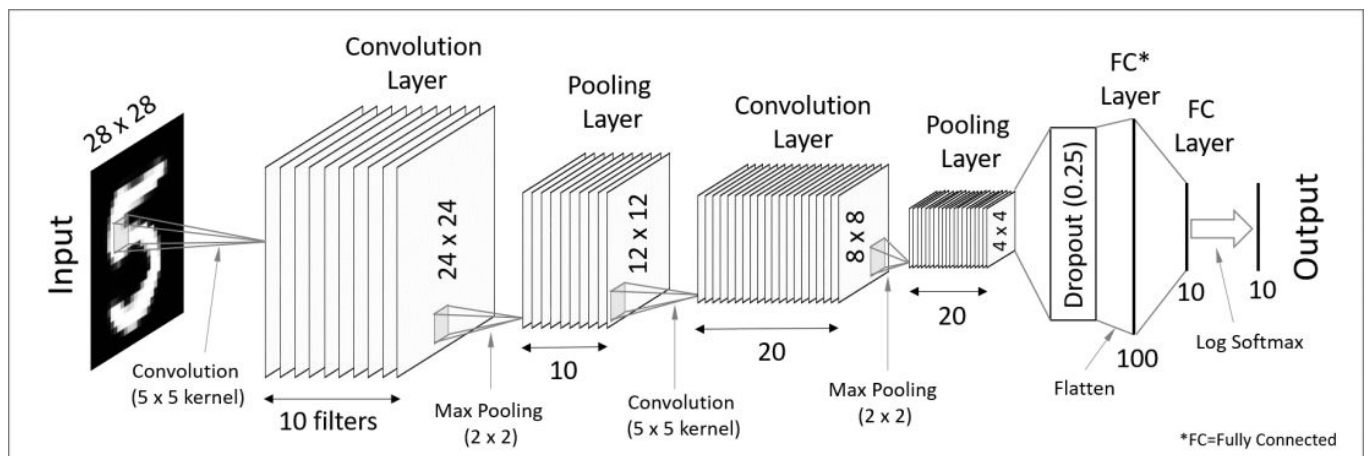
When a computer or system takes an image, it just sees an array of pixel values. Suppose $480 \times 480 \times 3$ where 480×480 is size, 3 refers to RGB values. Each of these numbers is assigned with a value of 0 to 255 as pixel intensities at that point. The key point is that based on taking the image as an input, computer system predicts and make an assumption as output

for describing the probability of the image being a said or certain class (say 0.90 for class 1, 0.96 for class 2, 0.4 for class 3).

Algorithm

To see the performing steps for a system to predict, we can define algorithms as –

1. Break the image into small image tiles — Similar to sliding window, we can pass sliding window over the entire large image and each result is saved as separate, as a segment of large image as tiny picture tile.
2. Feeding each tiny tile into the smaller size neural network — we rarely initialize the parameters with the same values and if not so, then we mark that tile as interesting.
3. Save the results from each small tile into a new array — we would not like to misplace the index of the original file. So we place the results in a grid of the same arrangement as an original image.
4. Downsampling — to reduce the size of a newer array, downsampling is used by max-pooling.



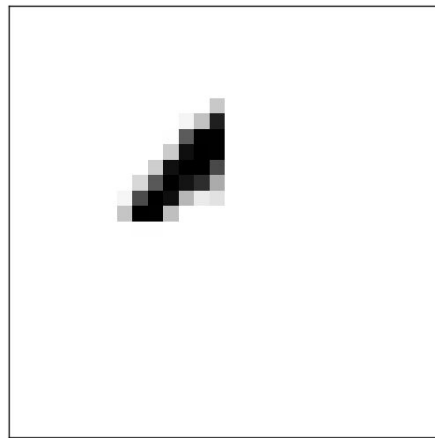
CNN architecture in MNIST dataset

Layers of Convolutional neural network

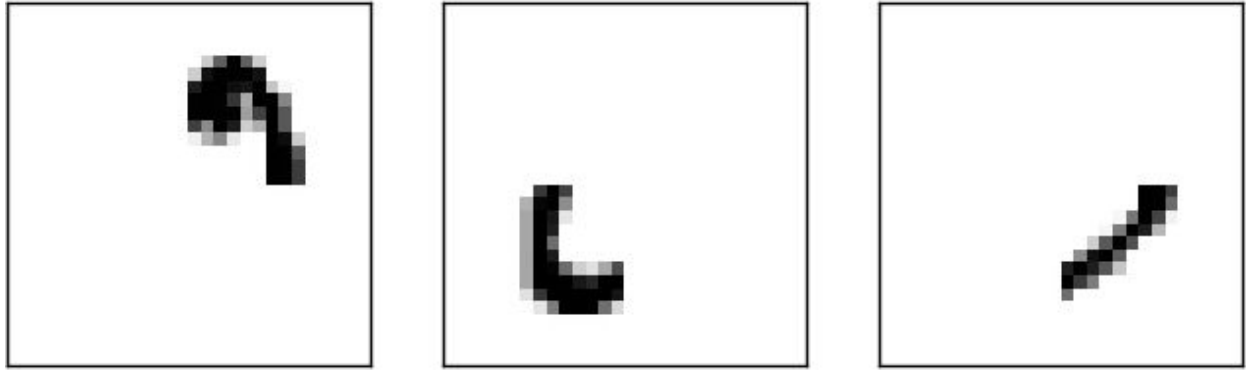
The multiple occurring of these layers shows how deep our network is, and this formation is known as the deep neural network.

1. Input: raw pixel values are provided as input.
2. Convolutional layer: Input layers translates the results of neuron layer. There is need to specify the filter to be used. Each filter can only be a 5*5 window that slider over input data and get pixels with maximum intensities.
3. Rectified linear unit [ReLU] layer: provided activation function on the data taken as an image. In the case of back propagation, ReLU function is used which prevents the values of pixels form changing.
4. Pooling layer: Performs a down-sampling operation in volume along the dimensions (width, height).
5. Fully connected layer: score class is focused, and a maximum score of the input digits is found.

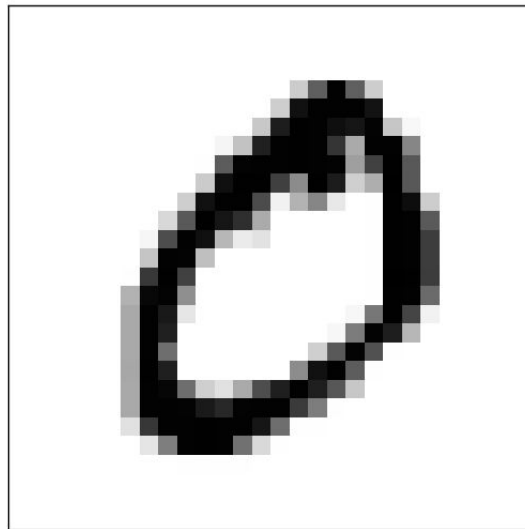
As we go deeper and deeper in the layers, the complexity is increased a lot. But it might worth going as accuracy may increase but unfortunately, time consumption also increases.



It can do this by heavily weighting input pixels which overlap with the image, and only lightly weighting the other inputs. In a similar way, let's suppose for the sake of argument that the second, third, and fourth neurons in the hidden layer detect whether or not the following images are present:



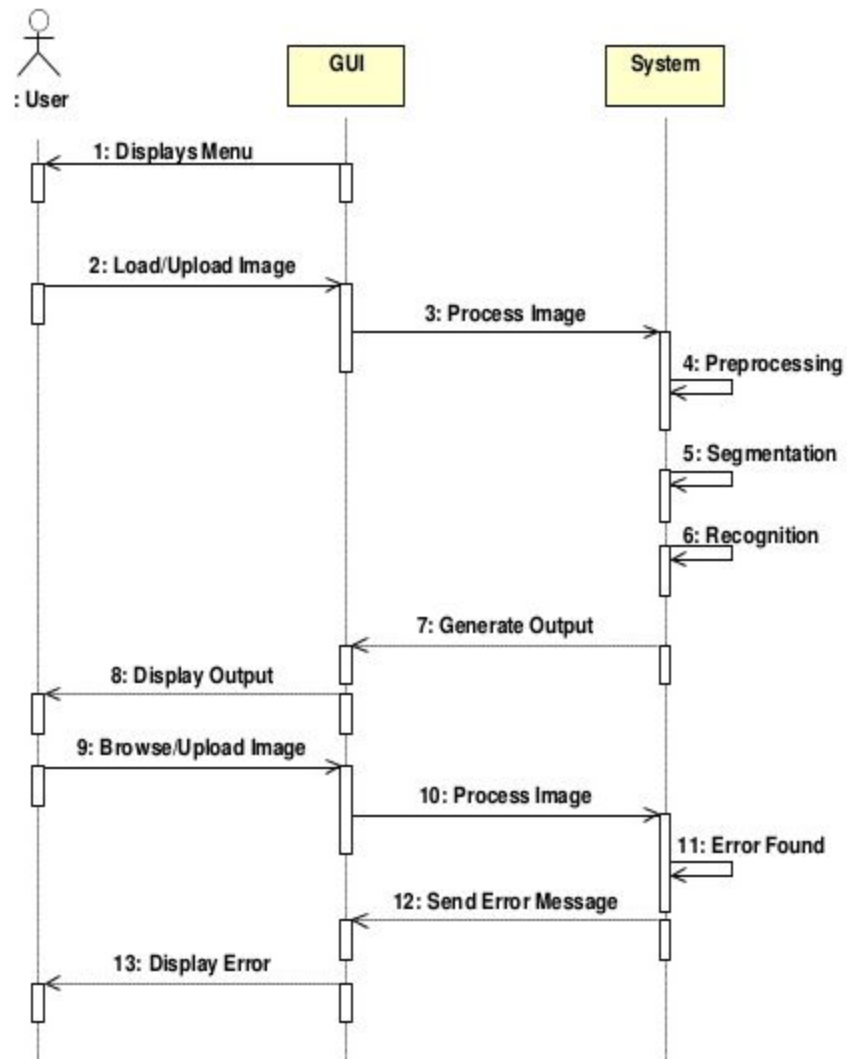
As you may have guessed, these four images together make up the o



UML Diagrams :



DFD Level 0



Extended Features :

1. Train the machine to be able to recognize simple equations.



Sample Handwritten Equation

Separate Contour of each digit/symbol

Though it requires a model of itself to segregate each character and is kind of simple than detection. But the main focus is digit detection and this approach is to be considered after the desired accuracy has been achieved in digit recognition.

2. EXTEND THE MODEL TO WORK ON NIST DATASET.
3. INCREASE THE ACCURACY FURTHER BY IMPLEMENTING MORE NUMBER OF HIDDEN LAYERS AND/OR EPOCHS
4. DEVELOP A SIMILAR SYSTEM TO IDENTIFY ALPHABETS.

Tools :

Python(Programming Language) IDE: IDLE/Spyder/Jupyter



win32gui :Python extensions for Microsoft Windows' Provides access to much of the Win32 API, the ability to create and use COM objects, and the Pythonwin environment

pywin32 : dependency for win32gui.

Pillow : image managing library.

Keras :Keras contains numerous implementations of commonly used neural-network building blocks

TensorFlow is a [free](#) and [open-source software library](#) for [dataflow](#) and [differentiable programming](#) across a range of tasks.



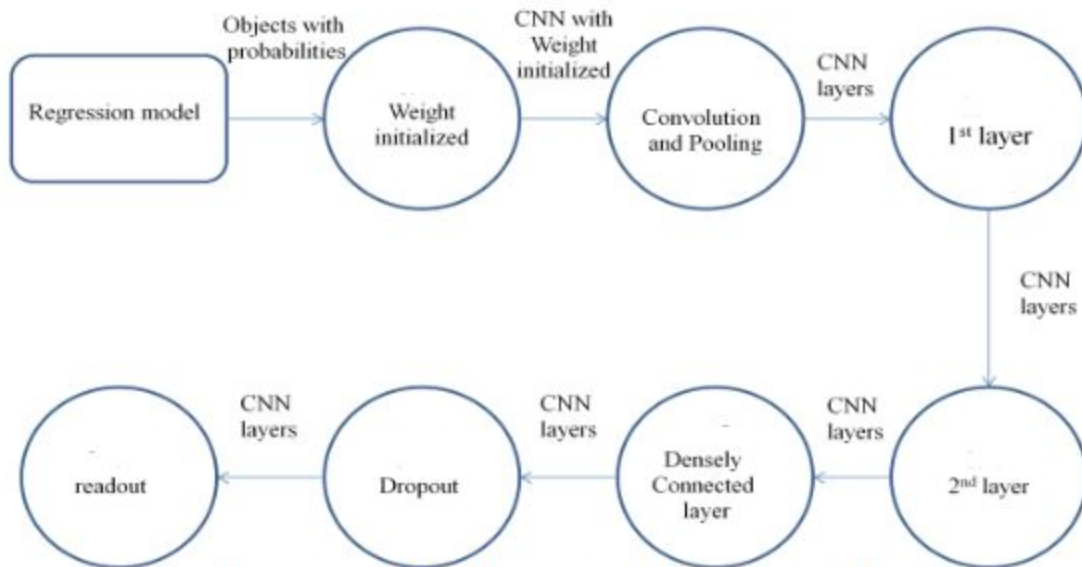


Figure 6 DFD level-2 (Build CNN)

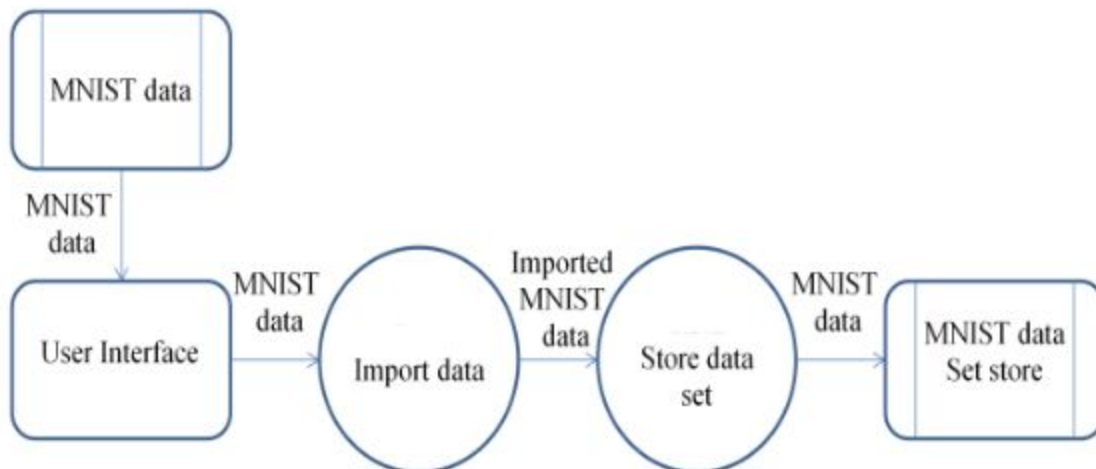


Figure 4 DFD level-2

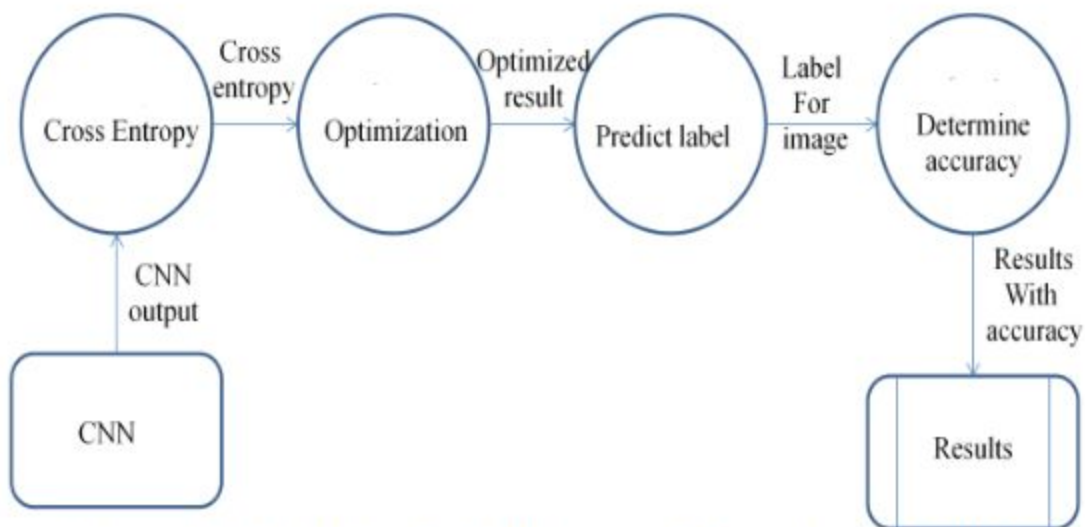


Figure 7 DFD level-2 (Train and Evaluate Model)

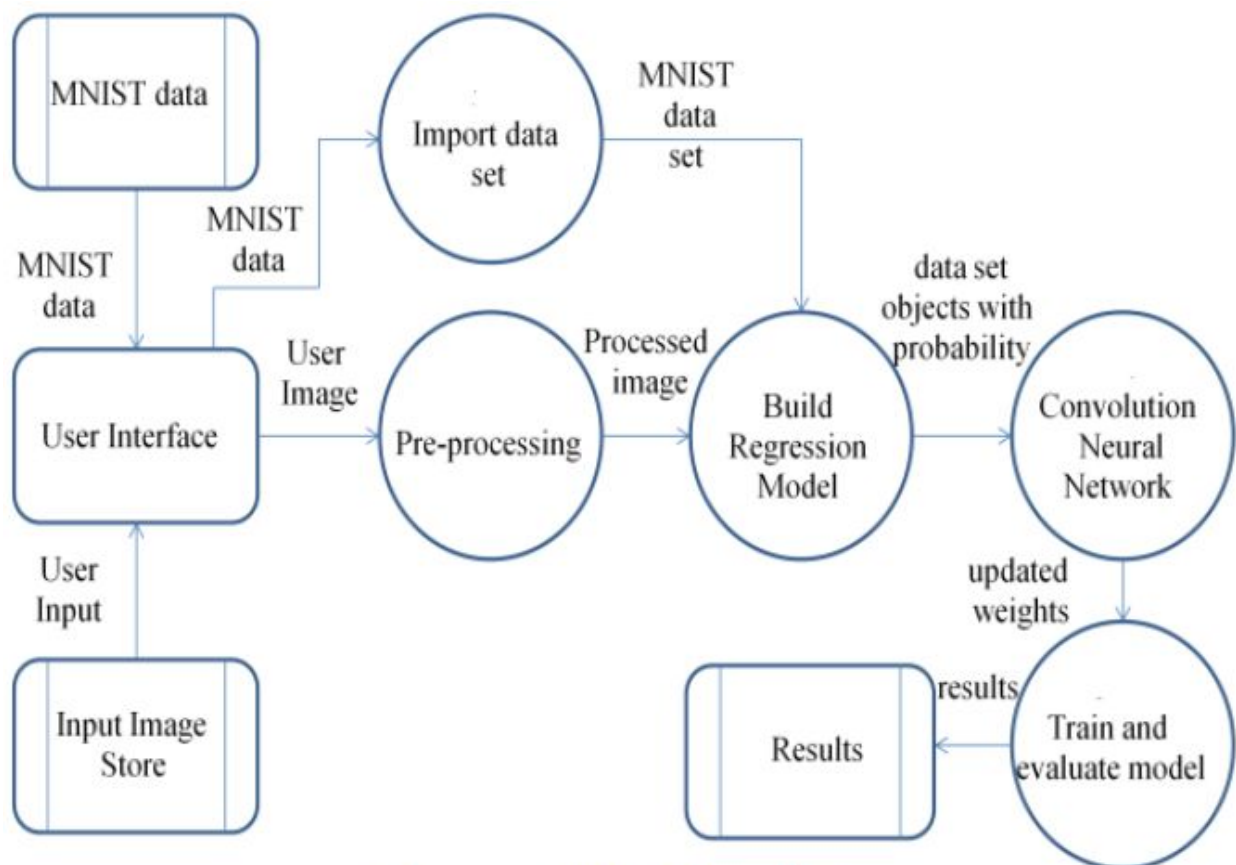


Figure 3 DFD level-1

References :

1. <http://yann.lecun.com/exdb/mnist/>
2. <http://neuralnetworksanddeeplearning.com/chap1.html>
3. Wikipedia.org
4. <http://www.deeplearningbook.org/>.
5. Coursera and Udemy