# 1.1. Introduction

 Artificial neural networks (ANN) are one of the main tools used in machine learning. As the “neural” part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize. While neural networks  [have been around since the 1940s](https://www.digitaltrends.com/cool-tech/history-of-ai-milestones/), it is only in the last several decades where they have become a major part of artificial intelligence. This is due to the arrival of a technique called “back propagation,” which allows networks to adjust their hidden layers of neurons in situations where the outcome doesn’t match what the creator is hoping for like a network designed to recognize dogs, which misidentifies a cat, for example. Another important advance has been the arrival of deep learning neural networks, in which different layers of a multilayer network extract different features until it can recognize what it is looking for.

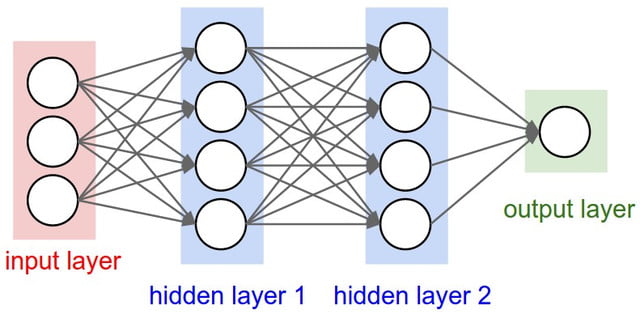


Fig.1: Process running behind

**1.2. Objective of Research**

The main objectives of handwritten digit recognition system are to achieve accuracy and high recognition rate, so for achieving this a number of steps are involved in this system like digitization, preprocessing, feature extraction and classification, etc each step have their own significance in the system.

# 1.3. Problem Statement

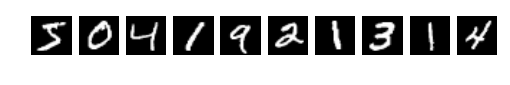
It is a real time application. So, there is no problem statement.

# 1.4. Industry profile

Some of these used in the Amazon just walk out technology, the Tesla autopilot car, Spaceships and more.

# 2. Data Collection

A neural network is used to recognize handwritten digits from the famous MNIST dataset. MNIST is a widely used dataset for the hand-written digit classification task. It consists of 70,000 labeled 28x28 pixel grayscale images of hand-written digits. The dataset is split into 60,000 training images and 10,000 test images. There are 10 classes (one for each of the 10 digits). The task at hand is to train a model using the 60,000 training images and subsequently test its classification accuracy on the 10,000 test images.



**Fig.2:** Sample images from the MNIST dataset.

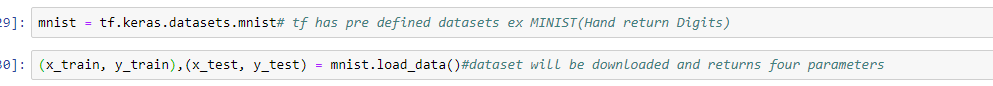
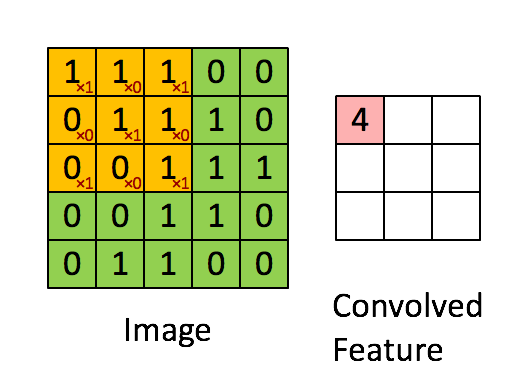


Fig.3: The data set from MNIST is downloaded which is the backend of tensor flow by using keras.

# 3.Methodology

**Convolution Operation:**

In this project, CNN method is used. Convolutional Neural Networks (CNN) orConvNet are popular neural network architectures commonly used in Computer Vision problems like Image Classification & Object Detection. Consider a colour image of 1000x1000 pixels or 3 million inputs, using a normal neural network with 1000 hidden units in first layer will generate a weight matrix of 3 billion parameters! CNN uses set of Convolution & Pooling operations to deal with this complexity.

Convolution operation involves overlapping of a filter/kernel of fixed size over the input image matrix. Then sliding across pixel-by-pixel to cover the entire image/matrix. As shown below as 3x3 filter(yellow) window slides over 5x5 image matrix all values within yellow matrix are added & stored in new convolved matrix.

**Pooling Operation:**

Along with Convolution layers CNN also use pooling layers to reduce the size of the representation, to speed the computation, as well as make some of the features detected a bit more robust. Pooling is of 2 types: Max Pooling & Average Pooling. We will be using Max Pooling in our ConvNet. Max Pooling operation simply find maximum number within sliding filter window over image matrix and return it new matrix as shown below. So maximum numbers 6,8,3,4 are selected from each 2x2 window from a 4x4 image matrix.



**Typical CNN Architecture:**

Combination of all these & fully connected layers results in various ConvNet architectures used today for various computer vision tasks. Below is an example of this architecture:

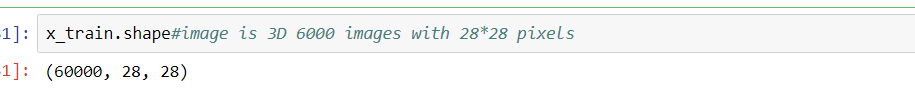


Fig.3: Typical CNN Architecture

# 3.1. Exploratory of data analysis

**Importing Tensor flow:**





Reshaping the size of image into 28\*28 pixels. Each image is **28** pixels by **28** pixels. We can interpret this as a big array of numbers. We can flatten this array into a vector of 28×28 = 784 numbers. It doesn’t matter how we flatten the array, as long as we’re consistent between images. From this perspective, the MNIST images are just a bunch of points in a **784**-dimentional vector space.



Fig. Sample from MNIST dataset

**Findings:**

1. Kindle services
2. Electrical slates

**Conclusion:**

The sequential API of Keras was also discussed. In the next chapter, the Functional API will be presented, which will enable us to build more complex models specifically for advanced deep neural networks.

This project reviewed the important concepts of the sequential API of Keras was discussed. For ease of understanding, these concepts were presented in the context of the MNIST digit classification. Different solutions to the MNIST digit classification using artificial neural networks, specifically MLPs, CNNs, which are important building blocks of deep neural networks, were also discussed together with their performance measures.