Experiment-6

Aim:- Implementation of CNN(Convolution Neural Network).

It is a type of deep learning model specifically designed for processing and analyzing visual data, such as images and video. It uses layer of filters to automatically learn and extract features from the input data, enabling tasks like image recognition, object detection etc. CNN has revolutionized computer vision and are widely used in applications like facial reorganization, self-driving cars and medical image analysis.



Credit:- Geeksforgeeks.

Code:-

|  |
| --- |
| #importing the required libraries  from tensorflow.keras.datasets import mnist  from tensorflow.keras.models import Sequential  from tensorflow.keras.layers import Conv2D  from tensorflow.keras.layers import MaxPool2D  from tensorflow.keras.layers import Flatten  from tensorflow.keras.layers import Dropout  from tensorflow.keras.layers import Dense  #loading data  (X\_train,y\_train) , (X\_test,y\_test)=mnist.load\_data()  #reshaping data  X\_train = X\_train.reshape((X\_train.shape[0], X\_train.shape[1], X\_train.shape[2], 1))  X\_test = X\_test.reshape((X\_test.shape[0],X\_test.shape[1],X\_test.shape[2],1))  #checking the shape after reshaping  print(X\_train.shape)  print(X\_test.shape)  #normalizing the pixel values  X\_train=X\_train/255  X\_test=X\_test/255  #defining model  model=Sequential()  #adding convolution layer  model.add(Conv2D(32,(3,3),activation= "relu",input\_shape=(28,28,1)))  #adding pooling layer  model.add(MaxPool2D(2,2))  #adding fully connected layer  model.add(Flatten())  model.add(Dense(100,activation="relu"))  #adding output layer  model.add(Dense(10,activation= "softmax" ))  #compiling the model  model.compile(loss= "sparse\_categorical\_crossentropy",optimizer= "adam",metrics=["accuracy"])  #fitting the model  model.fit(X\_train,y\_train,epochs=10) |

Output:-

