

Test The Model

Firstly we are slicing the `x_test` data until the first four images. In the next step we the printing the predicted output.

Predicting the output

```
prediction=model.predict(X_test[:4])
print(prediction)

[[5.50544734e-15  7.41999492e-20  5.00876077e-12  1.26642463e-09
  3.52252804e-21  1.54133163e-17  3.15550259e-21  1.00000000e+00
  1.32678888e-13  6.44072333e-14]
 [1.51885260e-08  8.02883537e-09  1.00000000e+00  6.44802788e-13
  6.37117113e-16  3.40490114e-15  2.15804121e-08  2.18907611e-19
  3.38496564e-10  2.07915498e-20]
 [3.14093924e-08  9.99941349e-01  2.01593957e-06  1.45100779e-10
  5.25237965e-06  1.59223120e-07  3.15299786e-08  1.53995302e-07
  5.09846941e-05  1.14552066e-07]
 [1.00000000e+00  1.35018288e-14  2.28308122e-10  1.79766094e-16
  1.28767550e-14  7.12401882e-12  2.92727509e-11  3.52439052e-13
  2.56207252e-12  2.32345068e-12]]
```

```
import numpy as np
print(np.argmax(prediction,axis=1)) #printing our labels from first 4 images
print(y_test[:4]) #printing the actual labels

[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

As we already predicted the input from the `x_test`. According to that by using `argmax` function here we are printing the labels with high prediction values