# Assignment -1

**Prajakta Mane (BE/A/B1821016)**

**Title**: Recognize Optical Character using ANN

**Objective**:

To recognize the optical characters using ANN

**Theory**:

**What is Optical Character Recognition?**

**Optical character recognition** or **optical character reader** (**OCR**) is the [electronic](https://en.wikipedia.org/wiki/Electronics) or [mechanical](https://en.wikipedia.org/wiki/Machine) conversion of [images](https://en.wikipedia.org/wiki/Image) of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example: from a television broadcast).[[1]](https://en.wikipedia.org/wiki/Optical_character_recognition#cite_note-1)

Widely used as a form of [data entry](https://en.wikipedia.org/wiki/Data_entry) from printed paper data records – whether passport documents, invoices, [bank statements](https://en.wikipedia.org/wiki/Bank_statement), computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as [cognitive computing](https://en.wikipedia.org/wiki/Cognitive_computing), [machine translation](https://en.wikipedia.org/wiki/Machine_translation), (extracted) [text-to-speech](https://en.wikipedia.org/wiki/Text-to-speech), key data and [text mining](https://en.wikipedia.org/wiki/Text_mining). OCR is a field of research in [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition), [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision).

Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs.[[2]](https://en.wikipedia.org/wiki/Optical_character_recognition#cite_note-2) Some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components.

**What is keras\_ocr?**

keras-ocr provides out-of-the-box OCR models and an end-to-end training pipeline to build new OCR models. Using this we get pre trained data and weights so our time and effort is saved.

OCR USING ANN CODE:-

# -\*- coding: utf-8 -\*-

"""Assignment 2 - OCR.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1yatXVuu-sZvitB2zUh0SpAvb8NkNLfAt

"""

from google.colab import drive

drive.mount('/content/gdrive')

!unzip gdrive/My\ Drive/Datasets/Handwritten.csv.zip

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

from sklearn.preprocessing import MinMaxScaler

from sklearn.model\_selection import train\_test\_split

from keras.utils import np\_utils

df = pd.read\_csv("A\_Z Handwritten Data.csv")

df.head()

df.rename(columns={'0':'target'}, inplace=True)

df.head()

df['target'].value\_counts()

X = df.drop('target',axis = 1)

y = df['target']

print(X.shape)

print(y.shape)

x\_train,x\_test,y\_train,y\_test = train\_test\_split(X,y, test\_size=0.2)

standard\_scaler = MinMaxScaler()

standard\_scaler.fit(x\_train)

# scaling data

x\_train = standard\_scaler.transform(x\_train)

x\_test = standard\_scaler.transform(x\_test)

# One hot encoding targets

y\_train = np\_utils.to\_categorical(y\_train)

y\_test = np\_utils.to\_categorical(y\_test)

print(x\_train.shape)

print(x\_test.shape)

print(y\_train.shape)

print(y\_test.shape)

#no. of features per data sample

x\_train.shape[1]

#total targets

len(y.unique())

# creating ANN model

model = keras.Sequential()

model.add(layers.Dense(500, activation="relu" , input\_dim = x\_train.shape[1]))

model.add(layers.Dense(400, activation="relu"))

model.add(layers.Dense(300, activation="relu"))

model.add(layers.Dense(200, activation="relu"))

model.add(layers.Dense(100, activation="relu"))

model.add(layers.Dense(len(y.unique()), activation="softmax"))

model.compile(loss='categorical\_crossentropy', optimizer="adam" , metrics = ['acc'])

model.summary()

model.fit(x\_train, y\_train, epochs=10, validation\_data = (x\_test, y\_test))

model.evaluate(x\_test,y\_test)

# original target

plt.imshow(x\_test[150].reshape(28,28), cmap='Greys')

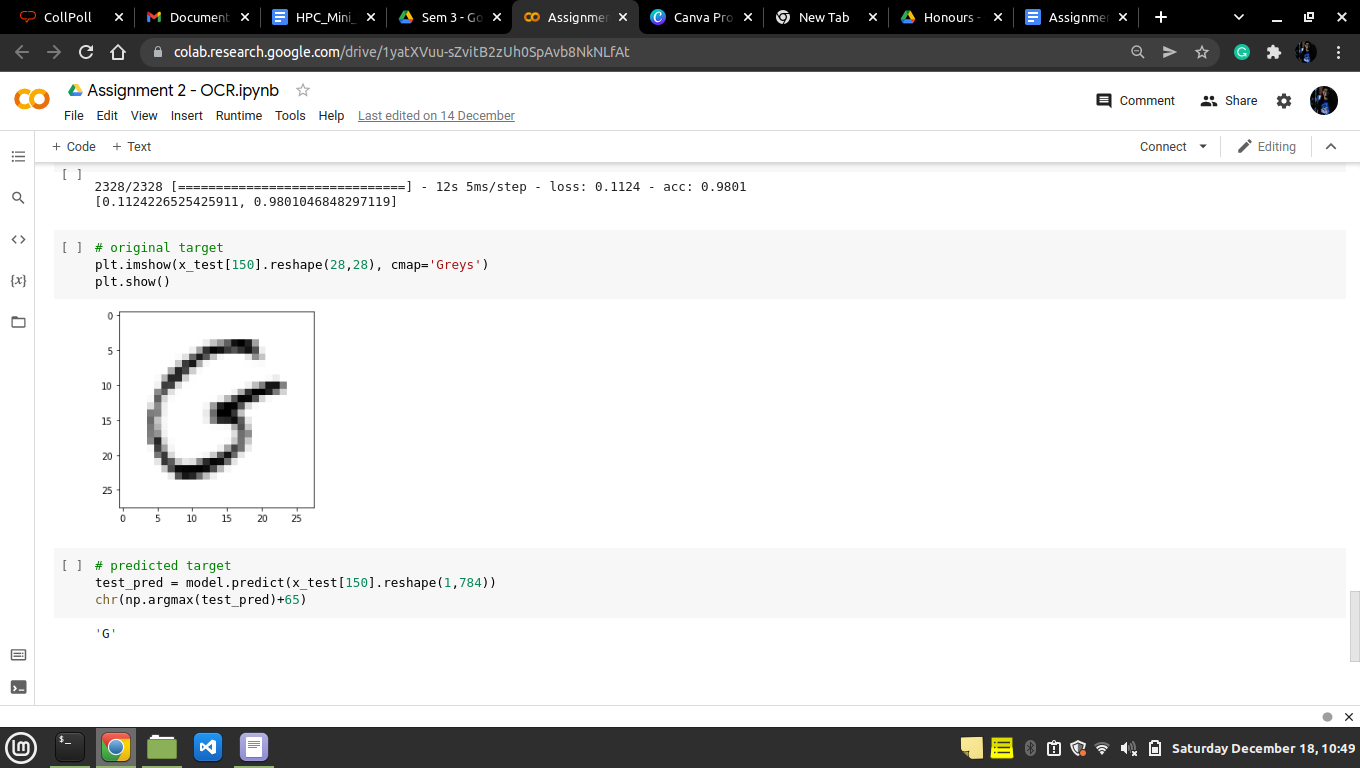
plt.show()

# predicted target

test\_pred = model.predict(x\_test[150].reshape(1,784))

chr(np.argmax(test\_pred)+65)

**OUTPUT:**



**Conclusion:**

Here, we studied how to recognize optical characters using ANN.