**Image Classification of an American Sign Language**

**Instruction**

***Need following libraries:***

import numpy as np

import pandas as pd

import tensorflow as tf

import tensorflow.keras as keras

from tensorflow.keras import layers

import matplotlib.pyplot as plt

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

from sklearn.metrics import confusion\_matrix

import random

***#Loading csv into data frame***

* ***Link to download database:*** [Sign Language MNIST | Kaggle](https://www.kaggle.com/datasets/datamunge/sign-language-mnist)
* ***pd.read\_csv***
  + sign\_mnist\_train.csv
  + sign\_mnist\_test.csv
* ***Hints:*** “train\_df = pd.read\_csv('../input/mnist-s-l/sign\_mnist\_train.csv')”

***#Exploring the Data***

* train\_df.head()

***#Extracting the Labels***

* y\_train = train\_df['label']
* y\_test = test\_df['label']
* del train\_df['label']
* del test\_df['label']

***#Extracting the Images***

* x\_train = train\_df.values
* x\_test = test\_df.values

***#Summarizing the Training and Validation Data***

* x\_train.shape
* y\_train.shape
* x\_test.shape
* y\_test.shape

***#Visualizing the Data***

* import matplotlib.pyplot as plt
* plt.figure(figsize=(40,40))
* num\_images = 20
* for i in range(num\_images):
* row = x\_train[i]
* label = y\_train[i]
* image = row.reshape(28,28)
* plt.subplot(1, num\_images, i+1)
* plt.title(label, fontdict={'fontsize': 30})
* plt.axis('off')
* plt.imshow(image, cmap='gray')

***#Exercise: Normalize the Image Data***

* x\_train.min()
* x\_train.max()

**Write: your codes to normalize the data**

**Give a summary of the Training Data after normalization**

**Build the NN Model**

***#Exercise: Build the Model***

* from tensorflow.keras.models import Sequential
* from tensorflow.keras.layers import Dense

***# SOLUTION***

* model = Sequential()
* model.add(Dense(units = 512, activation='relu', input\_shape=(784,)))
* model.add(Dense(units = 512, activation='relu'))
* model.add(Dense(units = num\_classes, activation='softmax'))

***#Model Summary***

* model.summary()

**Explain: About NN model charcteristics**

***#Compiling the Model***

* model.compile(loss='categorical\_crossentropy', metrics=['accuracy'])

***# Run the Model***

* model.fit(x\_train, y\_train, epochs=20, verbose=1)

**Write and explain: About the accuracy of the model**