# CAPSTONE: BUSINESS REQUIREMENTS

Course Coordinator: Marcos Bittencourt

## Problem statement

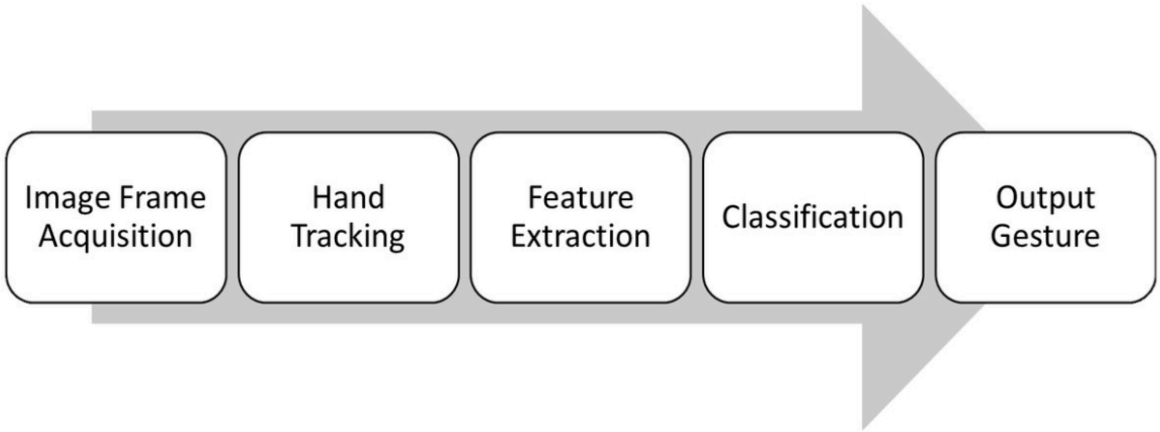
There are about 70 million deaf people who use sign language as their first language or mother tongue. It is also the first language and mother tongue to many hearing people and some deaf blind people (tactile sign languages). Each country has one or sometimes two or more sign languages, although different sign languages can share the same linguistic roots in the same way as spoken languages do. 2008 is also the year when the United Nations' Convention on the Rights of Persons with Disabilities adopted by the U.N. General Assembly in 2006, came into force, having been ratified by the 20th country (India ratified on October 1,2007). This document, which has also been ratified by India, should, in theory, advance the status of India's deaf, and also their language ISL: Article 9, Section 2(e): Provision of professional sign language interpreters; Article 21, Section3(b): Acceptance and facilitation of the use of sign language in official interactions; Article 24, Section 3(b):

Facilitation of the learning of sign language and the promotion of the linguistic identity of the deaf community; and Article 30, Section 4: "Recognition and support of their specific cultural and linguistic identity, including sign languages and deaf culture". Considering the current state of affairs in India, this is a tall order. A sign language (also signed language or simply signing) is a language which uses manual communication and body language to convey meaning, as opposed to acoustically conveyed sound patterns. This can involve simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions to fluidly express a speaker's thoughts. They share many similarities with spoken languages (sometimes called "oral languages", which depend primarily on sound), which is why linguists consider both to be natural languages, but there are also some significant differences between signed and spoken languages.

Linguistic work on Indian Sign Language (ISL) began in the 1970’s. In 1977, Vasishta, Woodward, and Wilson visited India with partial support from the National Science Foundation (USA) and collected signs from four major urban centres (Delhi, Calcutta, Bombay, and Bangalore) for linguistic analyses. Vasishta et al. (7), found that ISL is a language in its own right and is indigenous to the Indian subcontinent. Subsequent efforts by Vasishta et al between 1977 and 1982, resulted in four dictionaries of ISL regional varieties and some articles (8, 5, 9). The All India Federation of the Deaf, supposedly distributed these dictionaries to selected programmes serving the Deaf in India. In 2001, another dictionary was published by the Ramakrishna Mission Vidyalaya in Coimbatore (10). Some articles on the existence of a rural sign language in India, were also published (11, 12). It is claimed that a rural Indian Sign Language (RISL) also exists and it is very different from the ISL included in Vasishta et al’s dictionaries. The examples given by the author, however, do not support this contention. Development of any language, including sign languages, requires ongoing interaction between the speakers of that language. Deaf people in rural areas have little, or no opportunity, to meet other deaf people. What the author had observed, appears to be literally the gestures used by hearing people to communicate with the deaf. Such systems of manual communication used by isolated deaf people and their immediate environment are known as “home signing”. Home sign systems lack the linguistic complexity of sign languages that are used in deaf communities (13).

## Rational Statement

The Sign language is a visual language used by the people with the speech and hearing disabilities for communication in their daily conversation activities. It is completely an optical communication language through its native grammar, be unlike fundamentally from that of oral languages. In this research paper, presented an optimal approach, whose major objective is to accomplish the transliteration of 24 static sign language alphabets and numbers of American Sign Language into humanoid or machine decipherable English manuscript. Pre-processing operations of the signed input gesture are done in the first phase. In the next phase, the various region properties of pre-processed gesture image are computed. In the final phase, based on the properties calculated of earlier phase, the transliteration of signed gesture into text has been carried out. This paper also presents the statistical result evaluation with the comparative graphical depiction of existing techniques and proposed technique.




## Process (Key Metrics)

This method will use web camera which takes images of the signs by user. The Image are processed to extract characteristics necessary for recognition, which are then use as input for a neural network for recognition. Some basic steps are mention below.

1. **Pre-processing** - The purpose of this steps is the obtain, after the initial image processing, a set of numerical features that uniquely features that uniquely define describe a sign. Moreover, in order to remove overcome error caused by the noise of dataset Data-Pre-processing plays an important role.
2. **Image Preparation** – This include plethora of steps which includes image scaling, Skin detection and noise reduction, to decrease the computation cost of the model. The extracted data of hand images were fed into an autoencoder in order to perform the actual recognition training part of the project.

3. **The Neural network model** – We are using YOLO (You only see once) Model to achieve the goal. This model generalizes object representation and is based on FCNN (fully convolutional neural network). It trains on full images and directly optimizes detection performance. The dataset will be used to train this model and a loss function is setup to optimize the loss. Further testing data is used to check the model performance.

4. **Results**: Finally, the model is tested against unseen data for further evaluation.

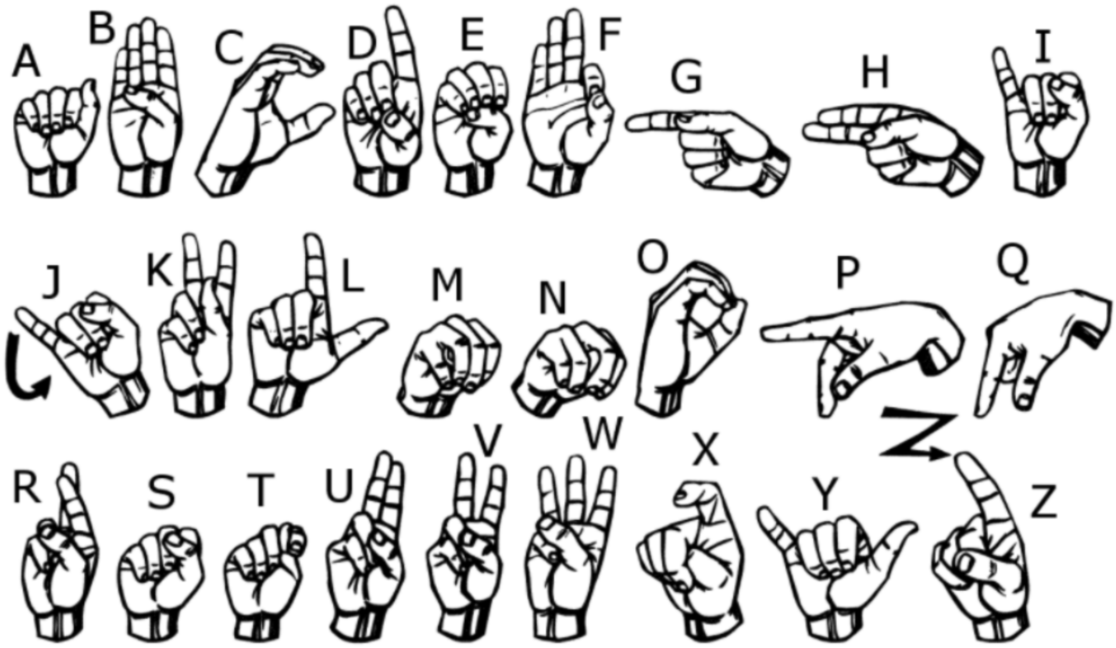
# DATA ANALYSIS

## About the data

The original handwritten MNIST image dataset is a common standard for image-based machine learning methods, but researchers have stepped up their efforts to update it and create drop-in substitutes that are more difficult for computer vision and original for real-world applications. As noted in one recent replacement called the Fashion-MNIST dataset, the Zalando researchers quoted the startling claim that "Most pairs of MNIST digits (784 total pixels per sample) can be distinguished pretty well by just one pixel". The Sign Language MNIST is introduced here to inspire the group to create further drop-in replacements and follows the same CSV format with single-row labels and pixel values. The hand gestures database of American Sign Language letters represents a multi-class problem with 24 letter classes (excluding J and Z which require movement).

## Meaning of signs

Out[6]:



## Importing the Required Packages:

```
111]: import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      import seaborn as sns
      %matplotlib inline
```

Here, we are using some python packages like pandas, numpy, matplotlib, seaborn for data processing and visualization. In addition, we need to analyse the data before moving towards model so that the model is error-free and gives the best performance.

## Load the Dataset

```
In [112]: df_train = pd.read_csv('sign_mnist_train.csv')
```

```
In [113]: df_test = pd.read_csv('sign_mnist_test.csv')
```

Pandas DataFrame is a double-dimensional, probably heterogeneous tabular data structure with defined axes (rows and columns). df_train and df_test are the dataframe which hold the from  the csv file.

## Statistical characteristics

```
In [114]: df_train.describe()
```

Out[114]:

| | label | pixel1 | pixel2 | pixel3 | pixel4 | pixel5 | pixel6 | pixel7 | |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 27455.000000 | 27455.000000 | 27455.000000 | 27455.000000 | 27455.000000 | 27455.000000 | 27455.000000 | 27455.000000 | 2745 |
| **mean** | 12.318813 | 145.419377 | 148.500273 | 151.247714 | 153.546531 | 156.210891 | 158.411255 | 160.472154 | 16 |
| **std** | 7.287552 | 41.358555 | 39.942152 | 39.056286 | 38.595247 | 37.111165 | 36.125579 | 35.016392 | 3 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **25%** | 6.000000 | 121.000000 | 126.000000 | 130.000000 | 133.000000 | 137.000000 | 140.000000 | 142.000000 | 14 |
| **50%** | 13.000000 | 150.000000 | 153.000000 | 156.000000 | 158.000000 | 160.000000 | 162.000000 | 164.000000 | 16 |
| **75%** | 19.000000 | 174.000000 | 176.000000 | 178.000000 | 179.000000 | 181.000000 | 182.000000 | 183.000000 | 18 |
| **max** | 24.000000 | 255.000000 | 255.000000 | 255.000000 | 255.000000 | 255.000000 | 255.000000 | 255.000000 | 25 |

8 rows × 785 columns

Pandas describe() is used to view certain basic statistical details such as percentile, mean, std etc. of a data frame or a series of numerical values. Here, we can observer the total count of features, their mean and standard deviation values and the maximum values.

## Identifying the Null values

```
In [115]: df_train.sum().isna()

Out[115]: label       False
          pixel1      False
          pixel2      False
          pixel3      False
          pixel4      False
                      ...
          pixel780    False
          pixel781    False
          pixel782    False
          pixel783    False
          pixel784    False
          Length: 785, dtype: bool
```

Checking the null values plays import role in model accuracy. Here in this dataset, we don't have any null values of any feature. We don't need to clean the values for perform any steps related to data cleaning.

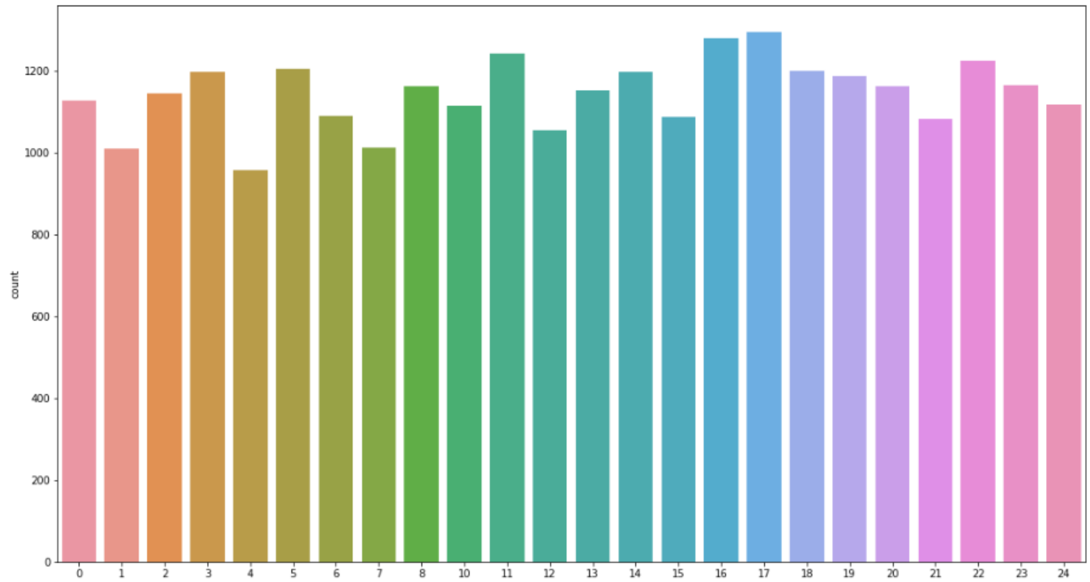
## Shape of the Dataset

```
In [116]: train_label_data = df_train["label"].values
          train_label_class = np.unique(np.array(df_train))
          train_label_data.shape

Out[116]: (27455,)
```

This function tells us the total number of rows in a dataset. We have 27455 rows of target variable.

## Bar Graph

```
In [117]: plt.figure(figsize=(18,10))
          sns.countplot(x=train_label_data)

Out[117]: <matplotlib.axes._subplots.AxesSubplot at 0x1284a8d10>
```




Bar graph is being used to compare the different features. The data is uniform distribution among features. X-axis signifies the total count of a features. On the other hand Y-axis signifies the values of different features. Since our data is already equally distributed, further we don't need to clean or pre-process the data further

## Reshaping the Images to Desired size

```
In [120]: images = df_train.values
          images = np.array([np.reshape(i, (28, 28)) for i in images])
          images = np.array([i.flatten() for i in images])
```

Most of Convolution neural network algorithm takes input images of size (28,28). In this step Numpy is used to reshape the data to appropriate input format. In the last time flatten method is used to convert multi-dimension array to single dimension array.

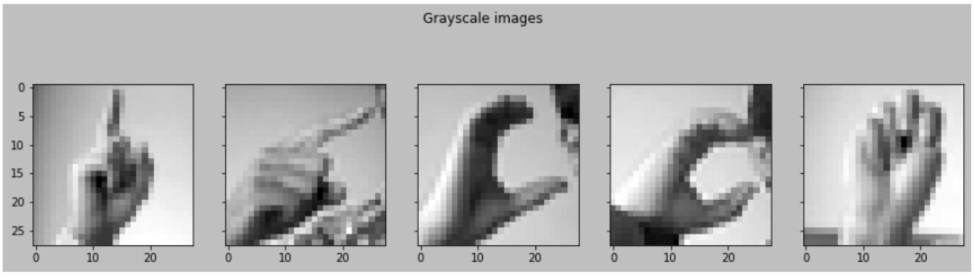
## Convert Target Variable:

```
In [121]: from sklearn.preprocessing import LabelBinarizer
          label_binrizer = LabelBinarizer()
          labels = label_binrizer.fit_transform(labels)

In [122]: labels

Out[122]: array([[0, 0, 0, ..., 0, 0, 0],
                 [0, 0, 0, ..., 0, 0, 0],
                 [0, 0, 1, ..., 0, 0, 0],
                 ...,
                 [0, 0, 0, ..., 0, 0, 0],
                 [0, 0, 0, ..., 0, 0, 0],
                 [0, 0, 0, ..., 0, 1, 0]])
```

Since the target variable is in categorical form, it needs to be convert into numerical form to get a systematic approach. Label Binarizer is used to achieve this task. Here label is our target variable.

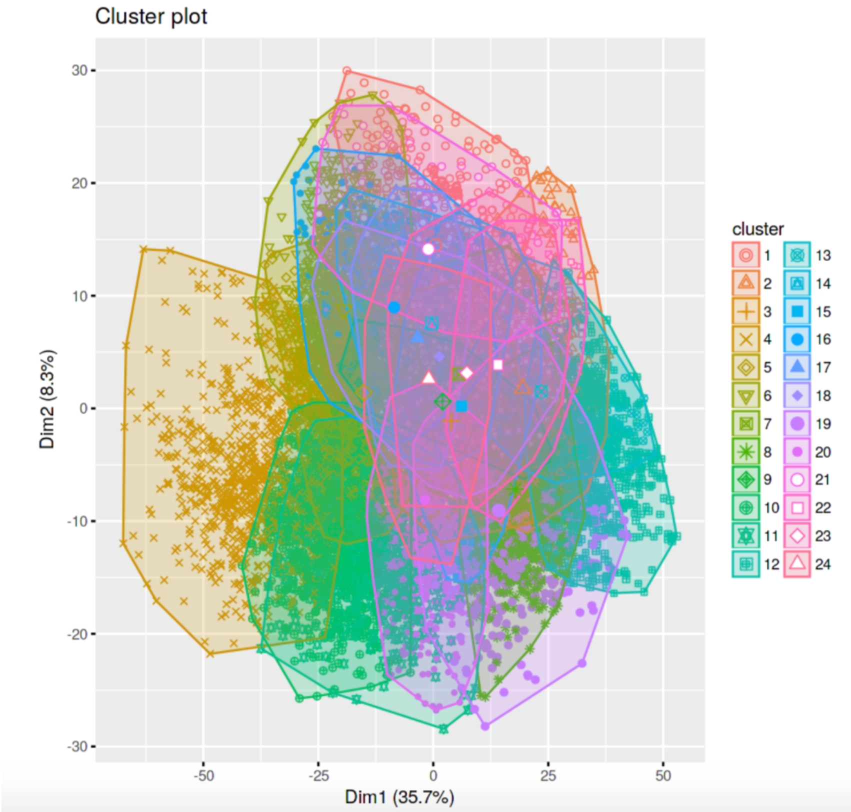
## Subplot of Images

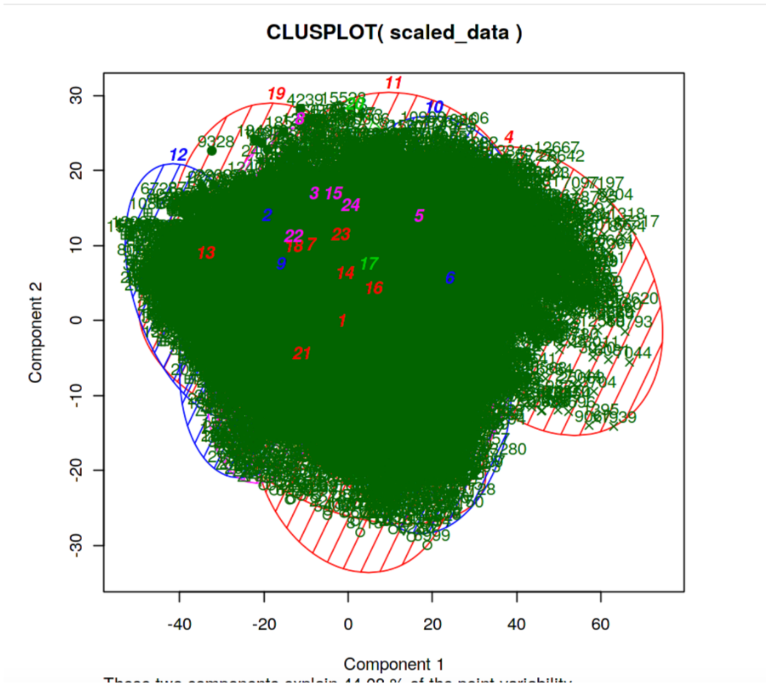
```
In [125]: plt.style.use('grayscale')
          fig, axs = plt.subplots(1, 5, figsize=(15, 4), sharey=True)
          for i in range(5):
                  axs[i].imshow(images[i].reshape(28,28))
          fig.suptitle('Grayscale images')

Out[125]: Text(0.5, 0.98, 'Grayscale images')
```

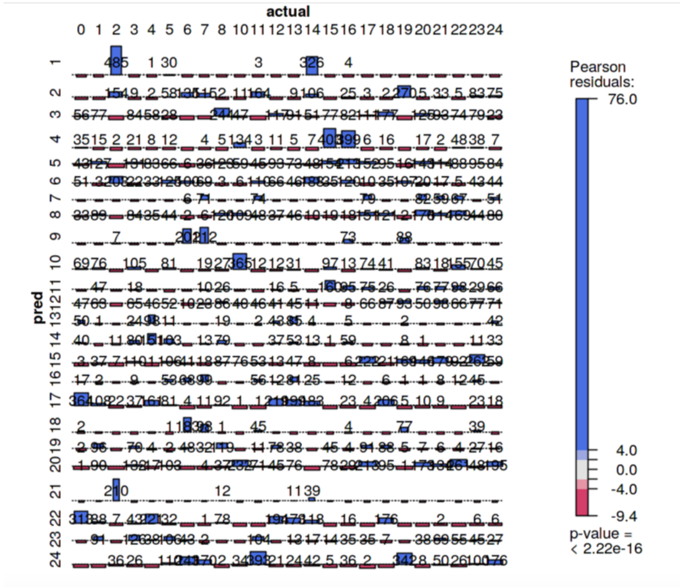


# Clustering using K means

### Association plot to see association between predicted and actual values






After all the steps, we used subplot method to display some of the images to ensure the images are in correct form and are ready to train the CNN model.

# References:

1. https://peerj.com/articles/cs-218/
2. https://pdfs.semanticscholar.org/568c/4f5bdc38059c526c4f8f39094057e71415ca.pdf
3.http://tmu.ac.in/college-of-computing-sciences-and-it/wp-content/uploads/sites/17/2016/10/T203.pdf
4.http://vision.stanford.edu/teaching/cs231a_autumn1213_internal/project/final/writeup/distributable/Chen_Paper.pdf