Script

**Hello there,**

**My name is Dhyuti**

And My name is Vishal

We are Team Visionaries presenting to you our Sign-Language Translator modeled as a part of OpenCV AI Competition 2022. - 1

**PS:**

**Over 5% of our global population, i.e, 432 million adults and 34 million children have hearing disability. One of the major difficulties faced by deaf people is their *means of communication*. Not everyone around the world is aware of** [**sign-language**](https://en.wikipedia.org/wiki/Sign_language)**. Although many services are deaf-people-friendly, like provision of running subtitles in cinemas, providing written instructions near shops, etc, there are not many services available that put in efforts to understand what the deaf people want to convey.**

**In order to cater to the needs of deaf people and help them communicate better, there is a surging need for sign language translation.**

Approach;

Our main aim is to facilitate the process of sign language translation by use of various ML algorithms and models.

**Our Input** is a*n image of a hand gesture of a person communicating in sign language.*

**Our Output** will be *English Text* corresponding to the translation of what the person said.

There are 4 major steps to facilitate the following:

1. Image Preprocessing
2. Identification of human hand
3. Train a CNN to identify hand gestures

The CNN architecture will be designed based on various logical and experimental iterations - 2

**Dataset:**

**We have chosen the Standard** [**MINST Sign Language Dataset**](https://www.kaggle.com/datasets/datamunge/sign-language-mnist) **for testing and training purposes.**

Model:

We have used a CNN Architecture with an input layer followed by 3 Convolution and Max pooling layers which is a result of trial and error based experimentation. - 3

Codes & Outputs:

**We have included various python modules like CV2, Keras, pandas and matplotlib to proceed with our project** - 1 vid

**So the first step is Loading our data.**

**Our dataset is embedded in the form of an excel sheet**

**Where every row represents a 28x28 image and their pixel data is given -** 1 vid

**To generate the dataset in a format suitable for us, we first read the data in the excel sheets using pandas,**

**We then arrange the data in an 2d-array image format and saved it.**

**We then used opencv’s cv2.imread function to read the thus generated image**

**In this way, we generate both: the testing and training data along with their labels.**

**Next up, we run this data through our defined CNN model to train it and check for accuracy.**