

DATE	21 October 2022
TEAM ID	PNT2022TMID25357
PROJECT NAME	Real time communication system powered by AI specially abled

## Real-Time Communication System Powered By AI For Specially Abled

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output

### Pre-Requisites

- To complete this project, you must require the following software's, concepts, and packages
  - **Anaconda (IDLE / Spyder / PyCharm)(Python 3.7):**
    - o [Link](#)
  - Computer Vision
    - o [Link](#)
  - Flask Concepts
    - [Link](#)
- Python Packages

3. **Tensorflow**- This package is used as backend support to Keras
4. **Keras**-This package is used for building Neural Network layers
5. **OpenCV**-This package is used for image processing
6. **Flask**- To build a web application

If you are using anaconda navigator, follow the below steps to download the required packages:

- Open anaconda prompt as administrator.
- Type "**pip install tensorflow==1.14.0**" and click enter.
- Type "**pip install keras=2.2.4**" and click enter.
- Type "**pip install opencv-python**" and click enter.
- Type "**pip install imutils**" and click enter.
- Type "**pip install flask**" and click enter.

## Project Structure

This is the project structure which needs to be followed for building Conversation Engine

## Data Collection

In this, we will be collecting data for building our project. We will be creating two folders one for training and the other for testing. Images present in the training folder will be used for building the model and the testing images will be used for validating our model.

## Create Train And Test Folders

**Step1:** Create Train and Test folders with each folder having folders with images of different hand signs. A minimum of 100 images needs to be present in each category folder to get the maximum no of features.

## Create Train And Test Folders

**Step1:** Create Train and Test folders with each folder having folders with images of different hand signs. A minimum of 100 images needs to be present in each category folder to get the maximum no of features.

Dataset can also be downloaded from the reference provided

## **Image Preprocessing**

In this, we will pre-process the images which will be used for building the model. Image pre-processing includes zooming, shearing, flipping to increase the robustness of the model after it is built. We will be using the Keras package for pre-processing images.

## **Import ImageDataGenerator Library And Configure It**

Import ImageDataGenerator and create an instance for which include shearing, rescale, zooming, etc to make the model robust with different types of images.

## **Apply ImageDataGenerator Functionality To Train And Test Set**

Specify the path of both the folders in the `flow_from_directory` method.

Flow from directory loads the images from a given directory and can bring all the images to the target size. We will be loading all the images of the train and test using the flow from directory method.

## **Model Building**

**In this milestone, we start building our model by:**

1. Initializing the model
2. Adding Convolution layers
3. Adding Pooling layers
4. Flatten layer
5. Full connection layers which include hidden layers

At last, we compile the model with layers we added to complete the neural network structure

## **Import The Required Model Building Libraries**

Import the libraries that are required to initialize the neural network layer, create and add different layers to the neural network model.

## **Initialize The Model**

Initialize the neural network layer by creating a reference/object to the Sequential class.

## **Add The Convolution Layer**

The first layer of the neural network model, the convolution layer will be added. To create a convolution layer, Convolution2D class is used. It takes the number of feature detectors, feature detector size, expected input shape of the image, activation function as arguments. This layer applies feature detectors on the input image and returns a feature map (features from the image).

## **Add The Pooling Layer**

After the convolution layer, usually, the pooling layer is added. Max pooling layer can be added using MaxPooling2D class. It takes the pool size as a parameter. The efficient size of the pooling matrix is (2,2). It returns the pooled feature maps. (Note: Any number of convolution layers, pooling and dropout layers can be added)

## **Add The Flatten Layer**

The flatten layer is used to convert the n-dimensional array to a 1-dimensional array. This 1D array will be given as input to ANN layers.

## **Adding The Dense Layers**

Three dense layers are added which usually takes the number of units/neurons. Specifying the activation function, kind of weight initialization is optional.

## **Compile The Model**

After adding all the required layers, the model is to be compiled. For this step, loss function, optimizer, and metrics for evaluation can be passed as arguments.

## **Fit And Save The Model**

Fit the neural network model with the train and test set, number of epochs, and validation steps.

The weights are to be saved for future use. The weights are saved in as .h5 file using save().

## Test The Model

Now we test the model by passing an image to get predictions. While test the model we should make sure that the test image should meet the target size of the model, dimensions need to meet, and should undergo rescaling before giving it to the model.

## Import The Packages And Load The Saved Model

As a first step to start prediction we import packages that are used for loading the model and used to expand the dimension of the image. We use the Keras package to load the model which was saved when we built the model.

## Load The Test Image, Pre-Process It And Predict

Pre-processing the image includes converting the image to the array and resizing according to the model. Give the pre-processed image to the model to know to which class your model belongs to.

## Application Building

Now we will be building a Flask application that is used for building our UI which in backend can be interfaced to the model to get predictions. Flask application requires an HTML page for Frontend and a Python file for the backend which takes care of the interface with the model.

## Build A Flask Application

**Step 1:** Load the required packages

**Step 2:** Initialize graph, load the model, initialize the flask app and load the video  
Graph element is required to work with tensorflow. So, graph element is created explicitly.

**Step 3:** Configure the home page

## **Building Flask Application -Part 3**

Each frame is taken from the camera and processed and sent to the model for prediction. As discussed image undergoes different processing steps to meet model requirements to get predictions.

This below is the snippet for calling video feed from the HTML page.

## **Build The HTML Page**

Build an HTML page to display the processed video on the screen, so that the person can show signs which can be detected.

Run the application by going to the location of your program and run

`python webstreaming.py`

When the python file is executed the localhost is activated on 5000 port and can be accessed through it.

## **Output**

**Step 2:** Open the browser and navigate to localhost:5000 to check your application

## **Train CNN Model On IBM**

You can also train your Image classification Models on IBM Cloud using IBM Watson Studio Service.

This milestone lets you

- Train your model on IBM
- Store your Model on IBM
- Download the Stored model to the Local system

## Register For IBM Cloud

### IBM Account:

- Please click [here](#) to register for IBM
- Please click [here](#) to log in to IBM Account

**Watch the below video to register and login into your IBM account**

## Train Image Classification Model

**Please watch the below video to**

- Train the model on IBM
- Store your Model
- Download model to local system
- Test your model locally

**Note :** Please do skip the flask integration part . Once you download the trained model from IBM continue with the milestone "Video Analysis"

## First Milestone

First milestone test

## Ideation Phase

In this milestone you are expected to get started with the Ideation process.

## Literature Survey On The Selected Project & Information Gathering

In this activity you are expected to gather/collect the relevant information on project use case, refer the existing solutions, technical papers, research publications etc.

## Prepare Empathy Map

In this activity you are expected to prepare the empathy map canvas to capture the user Pains & Gains, Prepare list of problem statements.

## **Ideation**

In this activity you are expected to list the ideas (at least 4 per each team member) by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.

## **Project Design Phase – I**

From this milestone you will be starting the project design phase. You are expected to cover the activities given.

## **Proposed Solution**

In this activity you are expected to prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.

## **Problem Solution Fit**

In this activity you are expected to prepare problem - solution fit document and submit for review.

## **Solution Architecture**

In this activity you are expected to prepare solution architecture document and submit for review.

## **Project Design Phase -II**

From this milestone you will be continue working on the project design phase. You are expected to cover the activities given

## **Customer Journey**

Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).



## **Functional Requirement**

In this activity you are expected to prepare the functional requirement document

## **Data Flow Diagrams**

In this activity you are expected to prepare the data flow diagrams and submit for review.

## **Technology Architecture**

In this activity you are expected to draw the technology architecture diagram.

## **Project Planning Phase**

In this milestone you are expected to prepare milestones & tasks, sprint schedules.

## **Prepare Milestone & Activity List**

In this activity you are expected to prepare the milestones & activity list of the project.

## **Sprint Delivery Plan**

In this activity you are expected to prepare the sprint delivery plan.

## **Project Development Phase**

In this milestone you will start the project development and expected to perform the coding & solutioning, acceptance testing, performance testing based as per the sprint and submit them.

## **Project Development - Delivery Of Sprint-1**

In this activity you are expected to develop & submit the developed code by testing it.

## **Project Development - Delivery Of Sprint-2**

In this activity you are expected to develop & submit the developed code by testing it.

### **Project Development - Delivery Of Sprint-3**

In this activity you are expected to develop & submit the developed code by testing it.

### **Project Development - Delivery Of Sprint-4**

In this activity you are expected to develop & submit the developed code by testing it.