**Abstract**

Communication is the key for any human being and for the small percentage of people who have a hearing problem, has the communication barrier which is yet to be solved by the help of technology, our Sign-Language Recognition model uses machine learning to solve this problem in hand as our system takes the Image/ Video input and uses the trained machine learning model to accurately predict the signs and gestures and then convert them into english words/ sentences, the training model is chosen by considering the accuracy of the machine learning model. Then the model is chosen for predicting the indian sign language gestures. These recognised gestures are then converted into user readable text as the system provides a Graphical User Interface (GUI) to make it user friendly. We have classifications which include 1 to 9 numbers and alphabets of the indian sign language, making it a total of 35 classifications of image dataset and 10 classifications of video datasets which include basic everyday use case scenario indian sign language gestures. All the classifications help the supervised machine learning model to learn from the thousands of image and video datasets, helping the model train for better prediction of the gestures given.

**Introduction**

**Machine Learning:**

Everyone in this world communicate with each other by voice, there is a small percentage of people who are unable to communicate with everyone because of hearing disability, millions of people are suffering this disability and not all school curriculums have sign language which the mute people use to communicate, so most of the crowd is unable to understand what a mute person wants to say. Sign language is expressed in the form of hand gestures along with some non-verbal expressions. It is not the same as non-verbal gestures and hence is difficult to understand for normal people unless trained. With the technology we have and in the modern era we are able to use artificial intelligence to solve the problems of our mankind, with the help of machine learning we are able to construct a bridge which breaks the communication barrier between a normal person and a mute person. Artificial Intelligence has made break-through solutions to a lot of our problems in modern life. With advanced technology in our age we are able to cross human intelligence at one point. Our sign language translator majorly focuses on indian sign language which is been used currently by mute people in india, our sign language translator has machine learning models which has a good accuracy rate, making it an essential tool for communication, our system takes in image input and video input, recognises the gestures and signs in the footage, predicts the output with the help of the trained machine learning algorithm and converts it into english language, with the user-friendly graphical user interface (GUI) we can provide the output on the screen, making it translate the images and video footage of signs and gestures into readable text, breaking the communication barrier and helping people communicate with each other, our supervised machine learning model has been trained with thousands of image datasets which include various lighting conditions, skin tone and other various scenarios where we probably use the system, this include video datasets of signs, making it possible to recognise some of the signs we use in our daily life, All the diverse dataset helps the model train better and making the model able to detect the signs and gestures in most the cases. The trained model is supervised and has satisfactory accuracy results. Our goal is to predict alphabets, numbers and basic signs in the Indian sign language, making it a little easier to communicate with any person even if he/she has no idea about sign language.

A Sign Language is a language in which communication between people is made by visually transmitting the sign patterns to express the meaning. It is a replacement of speech for hearing and speech impaired people. Thus, because of which has attracted many researchers in this field from long. Many researchers have been working in different sign languages like American Sign Language, British Sign Language, Taiwanese Sign Language, etc. but few works have made progress on Indian Sign Language.

The hearing impaired people become neglected from the society because the normal people never try to learn ISL nor try to interact with the hearing impaired people. This becomes a curse for them and so they mostly remain uneducated and isolated. Thus recognition of sign language was introduced which has not only been important from engineering point of view but also for the impact on society.

Studies prove that Children feel more lonely when they are silent and the problem of this communication gap will add to their depression. There are also a numerous number of problems prevailing due to this communication gap, our project aims to act as a bridge to close this distance and make the flow of thoughts more easy.

**Literature Review**

**Challenges For Solving the problem**

1. No dataset available

There is no standard dataset available in video format to train the system.

1. No feasible model till date

There is no standard Algorithm to train using video data and identify the signs and gestures from video data.

**Algorithms Explored**

**Convolution Neural Networks**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

**Optical Flow**

Optical flow provides a concise description of both the regions of the image undergoing motion and the velocity of motion. In practice, computation of optical flow is susceptible to noise and illumination changes.

**Key frame Extraction**

Key frames are the frames that alone can define the content of the video. We follow a map-like structure and identify the gesture based on the extracted key frames.

**Long Short term Memory**

Long Short-Term Memory(LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more.LSTMs are a complex area of deep learing.

**How to overcome these Challenges**

1. There is no standard dataset available so we decided to make our own dataset
2. We are also including the face gestures so that we can also keep the context  
    Of the sentence.
3. Decomposition of the system is important

**Decomposed system workflow**

| Phase 1 | Implementing object detection for accuracy analysis,Algorithm selection(image dataset) |
| --- | --- |
| Phase 2 | Include Face gestures and training using video data (3 words) |
| Phase 3 | Training complete model(10 words) and including face emotion detection. |

**Phase 1**

We take image dataset and train the model because working with video data is tedious.we can fix on the algorithm and parameters in this phase.

Dataset in phase1

Single hand gestures <https://www.kaggle.com/muhammadkhalid/sign-language-for-alphabets>

Double hand gestures <https://www.kaggle.com/vaishnaviasonawane/indian-sign-language-dataset>

**Phase 2**

We now train the model with video data using 3 gestures initially by using the algorithm fixed in phase 1. After fixing the algorithm we also add the face gestures.

**Phase 3**

Train the model with complete 10 gestures after getting high accuracy we then need to work on face emotion detection.

**Evolution of Proposed System**

Keeping in mind all the advancements in the mind it is not so hard to solve this problem using machine learning and artificial intelligence . so, we propose a solution to this problem using machine learning .

* We train a Machine learning model using supervised learning on the signs and gestures in Indian sign language so that it can identify the signs and gestures in the Indian sign language in future.
* This model takes the input as an area of interest generated by the object identifier and finds their corresponding English words.
* The generated words are then passed to a sentence generator at which the generated words are framed into a proper English sentence.

**Alphabet Recognition :**

With the help of modern technology and computer vision we are able to take images and videos as input and use the supervised models to recognise the alphabets. With the well-known methods of convolutional neural networks (CNN), we train using the thousands of image dataset consisting of various use-case scenarios, the alphabet recognition is one of the phase were we predict the alphabets in indian sign language by recognising the hand signs in the images, this module is modular making it a reusable in the further development of the system. The alphabet recognition takes frames or images as input, the future scope of the system is where frames are extracted from the video footage, all the frames will have essential information in-order to recognise the pattern which further help the model to predict the sign into an English language alphabet. The alphabet recognition module brilliantly detects the hands in the image apart from the background, making it easier for the system to recognize the hand, the hand outline module extracts the hand features from the image and converts the image into a black and white image. This processed image is then sent as input feed to the supervised model which can take advantage of the processed form of the image and predict the alphabet more accurately. Hand outline detection will be discussed further in the paper as it is one of the essential features for the system to accurately predict the sign given by the user.

**Convolutional Neural Networks (CNN) :**

The Convolutional Neural Networks are one of the neural networks models used for computer vision based systems, our model takes advantage of the Convolutional Neural Networks (CNN) to accurately predict the signs in the indian sign language, this model is more advantageous when compared to other neural network models and also follows supervised learning methodology.

**Training Phase:**

In the training phase, we gather thousands of images for our various classifications which include alphabets and numbers with different possible scenarios of the use cases. The training phase takes the training dataset and trains the model by recursively going through the supervised methodology and with each epoch the accuracy of the model alters, this accuracy changes significantly with each epoch, so the accuracy can sometimes be increasing or decreasing. This accuracy is the major and essential part of the training as it is what determines the accuracy of the supervised trained model. All the thousands of images in the dataset are iterated and accuracy is varied throughout, as we have chosen a supervised method, the dataset not only hold the training data but also each data solution in-order for the system to validate its recognition of the sign or gesture on each iteration of the data in the dataset. The training phase is really essential for our system as we are going to do the accuracy analysis of the machine learning models and then choose the best model which has satisfactory and higher accuracy rate when compared to other models, this considered model is the final model chosen for the system to use.

**Recognition Phase:**

Recognition Phase uses the trained supervised model to recognise the gestures in the images/ frames of the video footage. This phase takes the live video as the input where the length of the video can vary according to the indian sign language, the system makes sures to extract the frames which are needed to recognise the sign language from the footage. These frames are then sent to the a module which detects the hands in the frames and then differentiates the hands from the background, this feature is essential and gives an advantageous gain as the module processed image/ frame makes sure to get a higher recognition rate of the input, the images which are processed by the Hand Outline Detection module are then sent to the supervised model where the trained model takes the processed image as input and uses the training data from the training phase and checks for patterns and other features which are relevant to the recognition from the input and then predicts/ recognises the sign/ gesture. The recognised gestures are then shown in the Graphical User Interface (GUI) which are quality of life features making the system easy user- friendly. The Recognised gestures are not only shown live but also stored under the history panel where the user can revisit the past recognised words, the words recognised are stored in English language and are stored either in sentence or single word format. The recognised words are stored temporarily until the session ends and the user can either give input as image format or video format, where the system can take both types of inputs and can recognise the sign language in both cases.

**Face emotion recognition:**

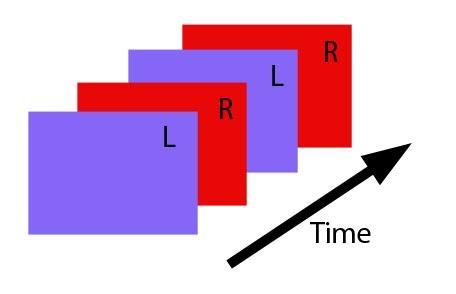
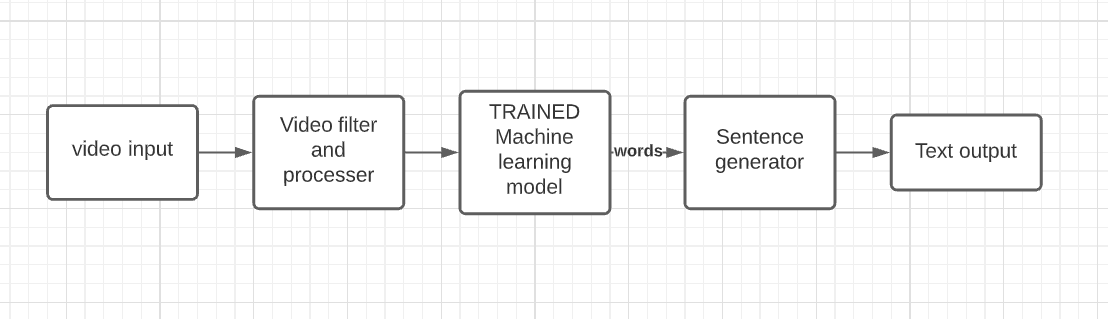
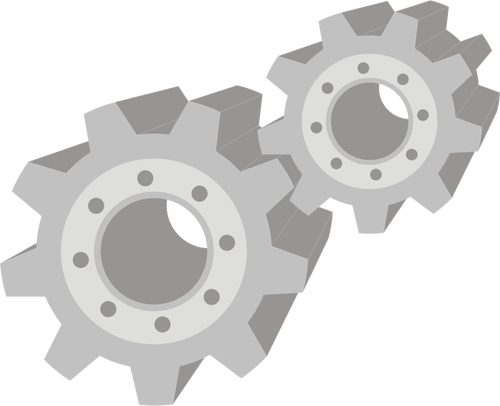
Face emotion recognition is very crucial for exactly identifying the emotion of the user with which we can predict the context of the conversion more accurately. The example shows the same gesture with different meaning based on facial expression.

 + (HAPPY EXPRESSION) = GOOD / ALL IS WELL

 + (DOUBTFUL EXPRESSION - QUESTION) = WHAT

**System Architecture**

The Video Input is taken as input and each frame is taken out of the video. The Video filter and processor labels the hand gestures in the frames. These labeled frames are then sent to the learning model. This model now gets trained by the sent data. After the gestures are recognised, they’re sent to the sentence generator. The Sentence generator takes the text and phrases the sentence, words accordingly for understandable output to the user. This output of the Sentence generator is shown as Output which is an understandable sentence.



**Video input**

Recorded video in real time which is captured from camera from the computer system web cam. It takes the video and generated frames and passes them to the video filter.

**Video filter and processor**

The video filter takes the generated frames and identifies the objects and extracts the areas of interest from each frame . We use object detection algorithms to identify hands and faces.

**Trained machine learning model**

A trained machine learning model that can recognise the signs from the areas of interest and also predict the context based on the facial expressions.The model generates english words corresponding to the signs in the video.

**Sentence generator**

The sentence generator is a trained natural language processing algorithm (NLP) that is used to make properly framed English sentences. It takes generated words from the ML model as input and uses this as key words to generate English sentences.

**Text Output**

The text output generated from the sentence generator is shown as the output and read outloud from the device speaker.

**Overview of technologies**

**Python**

Python is a powerful high-level, object-oriented programming language created by Guido van Rossum. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. Latest version of python is 3.6.5. TRI7

The syntax of the language is clean and the length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

Some features of python are

* Multi-purpose (Web, GUI, Scripting, etc.)
* Object Oriented
* Interpreted
* Strongly typed and Dynamically typed

**Keras**

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Keras méans horn in Greek. It is a reference to a literary image from ancient Greek

and Latin literature, first found in the Odyssey, where dream were divided between those who deceive men with false visions, who arrive to Earth through a gate of ivory, and those who announce a future that will come to pass, who arrive through a gate of horn.

Use Keras if you need a deep learning library that:

* Allows for easy and fast prototyping Supports both convolutional networks and recurrent networks, as well as combinations of the two.
* Runs seamlessly on CPU and GPU.

**Features of Keras**

**User friendliness**

Keras is an API designed for human beings, not machines, It puts user experience

front and center. eras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.

**Modularity**

A model is understood as a sequence or a graph of standalone, fully configurable

modules that can be plugged together with as few restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions and regularization schemes are all standalone modules that you can combine to create new models.

**Easy extensibility**

New modules are simple to add (as new faces and functions), and existing modules

provide ample examples, To be able to easily create new modules allows for total

expressiveness, making Keras suitable for advanced research.

**Work with Python**

No separate models configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility.

**Multiple Backends**

Your Keras models can be developed with a range of different deep learning backends. Importantly, any Keras model that only leverages built-in layers will be portable across all these backends: you can train a model with one hackend, and load it with another.

Available backends include:

* The Tensorflow backend
* The CNTK backend

**Numpy**

NumPy is a general-purpose array-processing package designed to efficiently

manipulate large multi-dimensional arrays of arbitrary records without sacrificing too much speed for small multi-dimensional arrays.

Numpy is built on the Numeric code base and adds features introduced by numarray as well as an extended C-API and the ability to create arrays of arbitrary type which also manipulate large multi-dimensional arrays of arbitrary records without sacrificing too much speed for small multi-dimensional arrays.

Numpy is built on the Numeric code base and adds features introduced by numarray as well as an extended C-API and the ability to create arrays of arbitrary type which also makes numpy suitable for interfacing with general-purpose data-base applications.

There are also basic facilities for discrete fourier transform, basic linear algebra and random number generation.

**Features of NUMPY**

**The ND-array**

The core functionality of NumPy is its "ndarray", for n-dimensional array, data structure. These arrays are strided views on memory. In contrast to Python's built-in list data structure these arrays are homogeneously typed: all elements of a single array must be of the same type.structure. These arrays are strided views on memory. In contrast to Python's built-in list data structure these arrays are homogeneously typed: all elements of a single array must be of the same type. Such arrays can also be viewed into memory buffers allocated by C/C++, Cython, and Fortran extensions to the Python interpreter without the need to copy data around,giving a degree of compatibility with existing numerical libraries. This functionality is exploited by the SciPy package, which wraps a number of such libraries. NumPy has built-in support for memory-mapped ndarrays.

**Limitations**

Inserting or appending entries to an array is not as trivially possible as it is with Python's lists. The np pad routine to extend arrays actually creates new arrays of the desired shape and padding values, copy the given array into the new one and returns it NumPy's p.concatenate operation does not actually link the two arrays but returns a new one, filled with the entries from both given arrays in sequence. Reshaping the dimensionality of an array with np reshape is only possible as long as the number of elements in the array does not change. These circumstances originate from the fact that NumPy's arrays must be views on contiguous memory bullers.A replacement package called Blaze attempts to overcome this limitation.

**Scikit-image**

Scikit-image is an open-source image processing library for the Python programming language. It includes algorithms for segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature detection, and more. It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.The scikit-image project started as scikits,.image, by Stéfan van der Walt. Its name stems from the notion that it is a "SciKit", a separately-developed and distributed third-party extension to SciPy. The original codebase was later extensively rewritten by other developers. Of the various scikits, scikit-image as well as scikit-learn were described as well-maintained and popular in November 2012. Scikit-image has also been active in the Google Summer of Code.The scikit-learn project started as scikits,learn, a Google Summer of Code project David Cournapcau, Its name stems from the notion that it is a SciKit, a separately- eloped and distributed third-party extension to SciPy. The original codebase was later written by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre ramfort and Vincent Michel, all from the French Institute for Research in Computer Science and Automation in Rocquencourt, France, took leadership of the project and made the first public release on February the 1st 2010. Of the various scikits, scikit-learn as well as scikit-image were described as well-maintained and popular in November 2012. Scikit-image is largely written in Python, with some core algorithms written in Cython to achieve performance.

**Opencv**

OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. Vadim Paisaresky joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005 OpenCV wns used on a Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later active development continued under the support of Willow Garage, with Gary Bradsky and Vadim Pisarevsky Icading the project. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day. Currently OpenCV supports a wide variety of programming languages like C++, and Python, Java etc and is available on different platforms as well including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on the CUDA and OpenCL are also under

active development for high-speed GU operations. OpenCV-Python is the Python API of OpenCV. It combines the best qualities of the Opencv C++ API and Python language. Python is a general purpose programming language started by Guido van Rossum, which became very popular in a short time mainly because of its simplicity and code readability. It enables me to be a programmer and to express his ideas in fewer lines of code without reducing any readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++.This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules.This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is a very easy to code in Python. This is a code how to implement the OpenCV-Python and it works as a C++ Python wrapper around the original C++ implementation.

**Implementation**

**CNN**

model = tf.keras.models.Sequential([

tf.keras.layers.Conv2D(64, (3, 3), activation='relu', input\_shape=(128, 128, 1)),

tf.keras.layers.MaxPooling2D(2, 2),

tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(512, activation='relu'),

tf.keras.layers.Dense(35)

])

print(model.summary())

**LSTM**

model = Sequential()

model.add(LSTM(64, return\_sequences=True, activation='relu', input\_shape=(30,1662)))

model.add(LSTM(128, return\_sequences=True, activation='relu'))

model.add(LSTM(64, return\_sequences=False, activation='relu'))

model.add(Dense(64, activation='relu'))

model.add(Dense(32, activation='relu'))

model.add(Dense(3, activation='softmax'))

**SVM**

cnn = tf.keras.models.Sequential()

# Step 1 - Convolution

cnn.add(tf.keras.layers.Conv2D(filters=64,padding="same",kernel\_size=3, activation='relu', strides=2, input\_shape=[128, 128, 1]))

# Step 2 - Pooling

cnn.add(tf.keras.layers.MaxPool2D(pool\_size=2, strides=2))

# Adding a second convolutional layer

cnn.add(tf.keras.layers.Conv2D(filters=64,padding='same',kernel\_size=3, activation='relu'))

cnn.add(tf.keras.layers.MaxPool2D(pool\_size=2, strides=2))

# Step 3 - Flattening

cnn.add(tf.keras.layers.Flatten())

# Step 4 - Full Connection

cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))

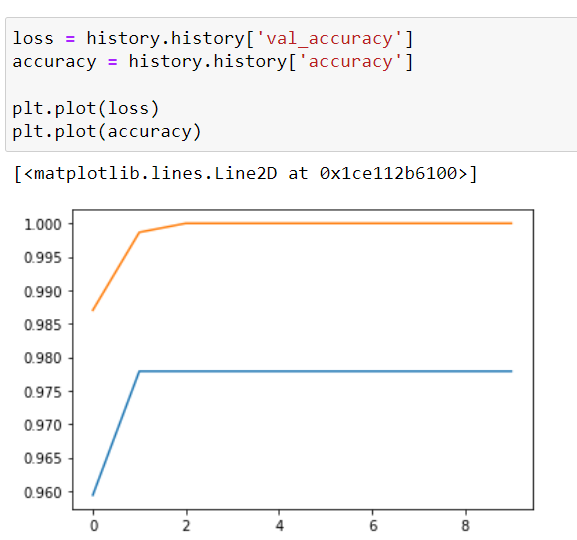
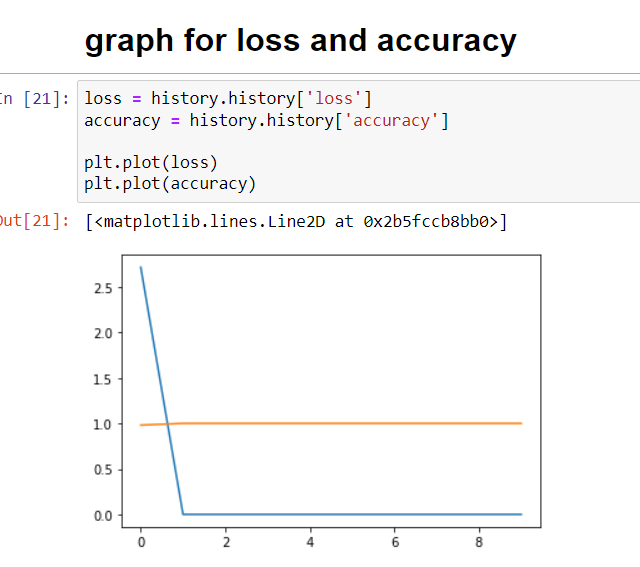
cnn.add(Dense(35, kernel\_regularizer=tf.keras.regularizers.l2(0.01),activation

='softmax'))

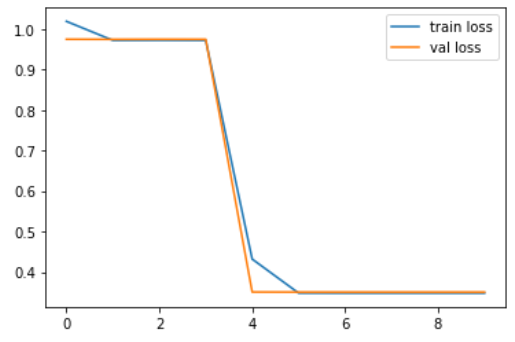
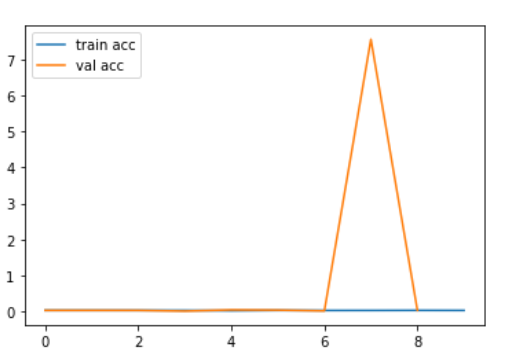
cnn.compile(optimizer = 'adam', loss = 'squared\_hinge', metrics = ['accuracy'])

**Results and Discussions**

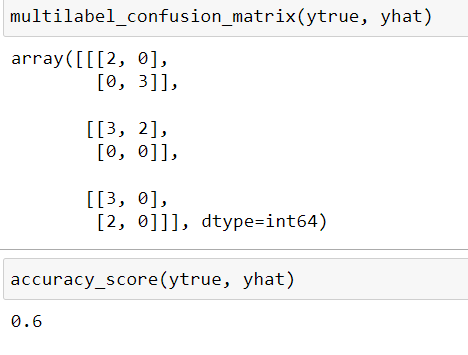
**CNN - DENSE**

****

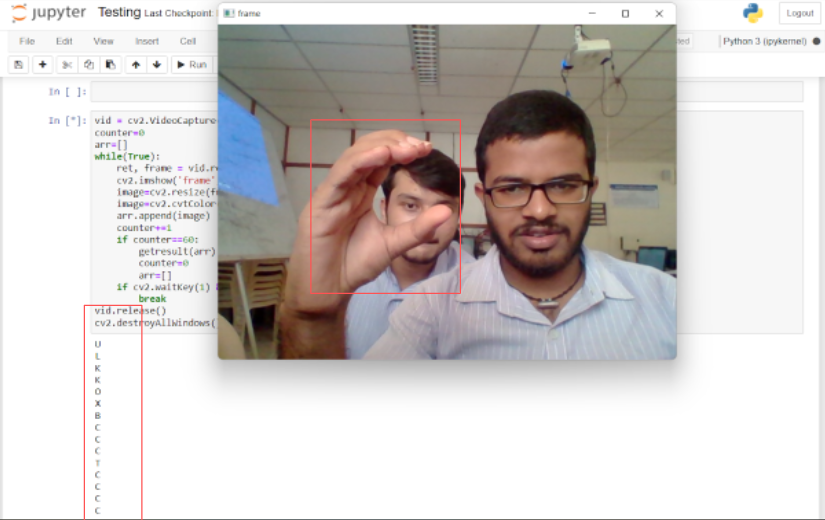
**CNN -SVM**

****

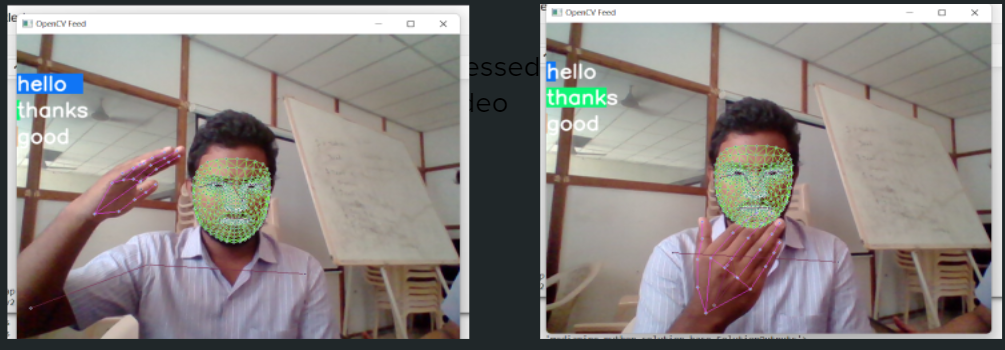
**LSTM**

****

**Screenshots**

****

****

****

**Conclusion and Future Scope**

The Sign Language Translation system can change the lives of many people across the world. It can be used to talk to the mute people and understand them better. This will help to communicate with children at the younger ages well and therefore help them to get out of their loneliness and depression. In Adults this translation system opens up more opportunities to those who are backward until now due to the communication gap.

The development of this system helps the people using Indian sign language and also using this model we can change the training dataset and implement translators for any sign language. Also we can use transfer learning to increase the sign space for recognition. Adding sign generation for English to this system will lead to an even more decrease in communication gap.

with all the advancements in the area of machine learning and artificial intelligence it is high time that we overcame this communication barrier. Our initiative is to make this gap decrease even by a small amount. Moreover we can use the same process to identify signs of any sign language just by changing the dataset used in training and other features to make the system more feasible

**References**

1. Accuracy analysis paper

<https://www.sciencedirect.com/science/article/pii/S1877050917320720>

1. Hardware Implementation paper

<https://www.sciencedirect.com/science/article/pii/S1877050918321331>

1. Software implementation paper

<https://drive.google.com/file/d/1A37c4Ta9CVVFyhwjIFb3Uw_PJCsKs0lb/view?usp=sharing>

1. Selected words

<https://docs.google.com/document/d/1uq2HevDOJ6iAcZ3Sb9WJ8338SDogyoN1sAAxGpcL6tM/edit?usp=sharing>

1. CNN

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

1. LSTM

<https://en.wikipedia.org/wiki/Long_short-term_memory>

1. Official Indian Sign language reference

<https://indiansignlanguage.org/> , <http://www.islrtc.nic.in/>