

**INTERNSHIP REPORT APPROVAL FORM**

July 04, 2019

With immense pleasure, this is to approve that the students of Sona College of Technology i.e

**Sai Bharathwaj S,**

**Mohammed Adhil M,**

**Aravind S.**

successfully completed their Project and Project Report on **“Communication Through Gestures”** under our guidance.

We are highly impressed with the work that they have done and commend them on their quick grasping skills. They have shown good intent to learn and have put the knowledge gained into application in the from of this project. We appreciate the hard work and commitment shown by them.

We hereby approve that this document is completely checked and accepted by Smart Bridge Technical Team. It’s been an absolute pleasure to educate and mentor these students. We hope that this document will also serve as a Letter of Recommendation, to whomsoever applied.

We wish them success in all future endeavors and a great career ahead.

**GD Abhishek**

AI Developer

A Project Report on

**COMMUNICATION THROUGH GESTURES**

*Submitted by*

SAI BHARATWAJ S

MOHAMMED AADHIL M

ARAVIND S

*Submitted to*

Smart Bridge Solutions

*in partial fulfillment for the completion of one-month internship*

*in*

Python in Machine Learning

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**SONA COLLEGE OF TECHNOLOGY**

SALEM-636005.

(BATCH 2016-20)

**ABSTRACT**

Gesture recognition are the growing fields of research. Being a significant part in non verbal communication hand gestures are playing vital role in our daily life. Hand Gesture recognition system provides us an innovative, natural, user friendly way of interaction with the computer which is more familiar to the human beings. Gesture Recognition has a wide area of application including human machine interaction, sign language, immersive game technology etc. By keeping in mind the similarities of human hand shape with four fingers and one thumb, this paper aims to present a real time system for hand gesture recognition on the basis of detection of some meaningful shape based features like orientation, center of mass (centroid), status of fingers, thumb in terms of raised or folded fingers of hand and their respective location in image.

On having the input sequence of images through web cam it uses some pre-processing steps for removal of background noise and employs K-means clustering for segmenting the hand object from rest of the background, so that only segmented significant cluster or hand object is to be processed in order to calculate shape based features. This simple shape based approach to hand gesture recognition can identify around 24 different gestures. This proposed implemented algorithm has been tested more than 450 images and it gives approximate recognition rate of 85%

**INTRODUCTION**

**PYTHON**

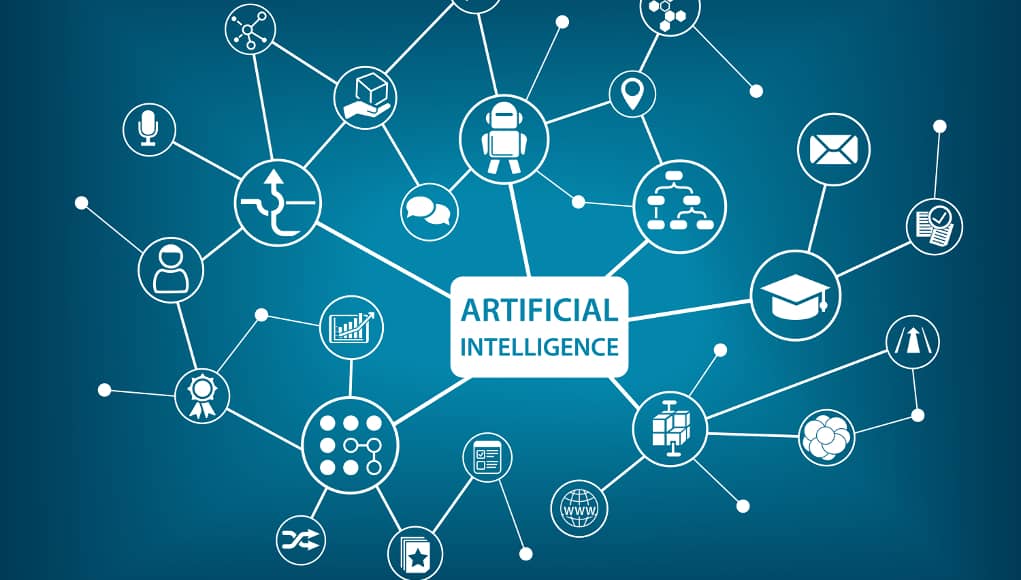
**Python**  is an interpreted, high-level, general-purpose programming language. Created by **Guido van Rossum** and first released in 1991.Its language constructs and object-oriented approach aim to help programmers write clear, logical

code for small and large-scale projects. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. **Python 2.0** was released on 2000 and introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. **Python 3.0** was released on 2008 and was a major revision of the language that is not completely backward compatible. Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development. Python is a multi-paradigm programming language; structured programming is fully supported and many of its features support functional programming and aspect-oriented programming (including by meta-programming and metaobjects (magic methods)). Many other paradigms are supported including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution. Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

**ARTIFICIAL INTELLIGENCE**

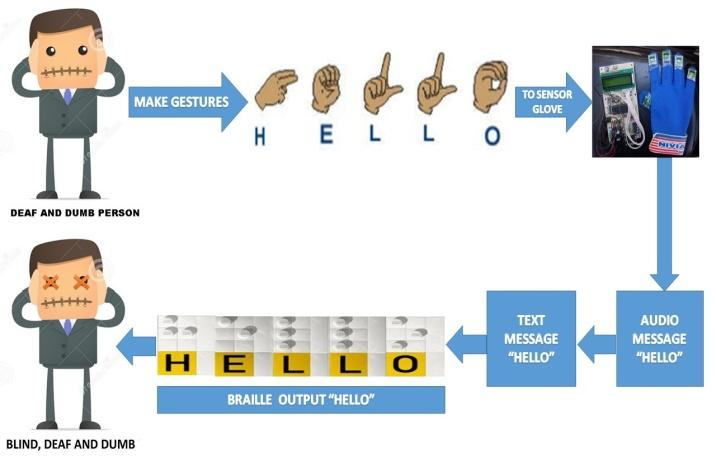
“**Artificial Intelligence**” is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving". Machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the AI effect. Artificial intelligence can be classified into three different types of systems: analytical, human-inspired, and humanized artificial intelligence. **Analytical AI** has only characteristics consistent with cognitive intelligence and using learning based on past experience to inform future decisions. **Human-inspired AI** has elements from cognitive and emotional intelligence understanding human emotions and considering them in their decision making. **Humanized AI** shows characteristics of all types of competencies (i.e., cognitive, emotional, and social intelligence) can be self-conscious and is self-aware in interactions with others. Traditional problems (or goals) of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects.



Approaches include statistical methods, computational intelligence, and traditional symbolic AI. Many tools are used in AI, including versions of search and mathematical optimization, artificial neural networks, and methods based on statistics, probability and economics. High-profile examples of AI include autonomous vehicles (such as drones and self-driving cars), medical diagnosis, proving mathematical theorems, playing games (such as Chess ), search engines (such as Google search), image recognition in photographs, spam filtering, predicting flight delays, prediction of judicial decisions and targeting online advertisements.The overall research goal of artificial intelligence is to create technology that allows computers and machines to function in an intelligent manner. These consist of particular traits or capabilities that researchers expect an intelligent system to display.

**PROBLEM**

Inability to speak is true disability. In our system we intend to overcome this disability by capturing hand gestures performed by the disabled and giving speech as output. This will not only help the mute people but also deaf and blind ones to communicate with anyone around. Our system will contain a manual of our own gestures which will be categorized depending on the situation they are most likely to be used in hence making it easier for the user to convey his message.

****

**REVIEW OF LITERATURE**

Recently, there has been a surge of interest on hand detection, tracking, and gesture recognition.. Human hands are the parts that people use the most and with the most ease to interact with the world. Hands are highly articulated structures with some 27 degrees of freedom. Even though hand postures and gestures are frequently considered as being identical, there are differences as explained. While the hand posture is a static motionless pose, such as making a palm posture and holding it in a certain position, the hand gesture is a dynamic process consisting of a sequence of changing hand postures over a short duration, as for instance waving the hand. Because of the inherent complexity of the hand gesture, its recognition can be divided into two stages: the low stage hand posture detection and the high stage hand gesture recognition. For a computer to recognize a hand gesture, first the hand should be detected from the captured image, and then the recognition of the gesture made by that hand should be done in a similar way as humans do. However, this is a more challenging technique to be implemented due to the limitations of such a natural system. Segmentation of the hand image is the first stage of all these systems. In [Aran06, Tokatl05], gloves or finger marks have been used to extract the hand posture information from the image and facilitate the hand segmentation process by removing the varying skin color issue of the problem. This approach permits the system to detect hands easily, and it is invariant against lightning conditions which may change. Another way for solving the problem is by implementing a hand gesture recognition system with a uniform 10 background such as a black curtain [Ionescu05]. Using a uniform background will facilitate the segmentation of the hand region. However, including gloves or uniform background reduce the natural (transparent) interaction feeling of gesture-recognition applications by imposing artificial constraints on the user’s environment. Since the scope of the thesis is detecting hands in normal backgrounds, our system must be able to detect and recognize hand gestures in a cluttered background using face subtraction, skin detection, and contour comparison with hand postures templates. The well-known “come as you are” expression stands for a Human Computer Interaction (HCI) system without constraints for the user to wear markers, gloves, or long sleeves, uniform background, or select a illumination. These constraints hinder the user’s ability to interactively track and recognize gestures. In order to avoid all these problems, the computer vision technique needs to meet the challenge to recognize in real-time hand gestures without requiring the user to wear any additional aids or to be connected to a special hand tracking or haptic device. Many researchers have proposed various techniques for hand gesture recognition systems. Usually, these systems are divided into two techniques, namely the glove-based and respectively the vision-based methods [Garg09]. The Data-Glove based approaches utilize sensor devices for digitizing hand and finger movements into multi-parametric data. The additional sensors facilitate detection of hand configuration and movement. However, the sensor devices are quite expensive and wearing gloves or trackers is uncomfortable and enlarges the “time-to interface,” or setup time. On the other hand, vision-based methods are more natural and useful for real-time applications. Besides, computer-vision-based interfaces provide many outstanding advantages.The need for all these to work in a variety of environments causes several issues because these systems require user and camera independent and invariant against the background and lighting changes to attain real-time performance. 11 A reliable set of characteristic features and relevant information of how they correlate in representing hand gestures are needed in order to successfully recognize hand gestures. A taxonomy for gesture types and a complete gesture modeling framework is presented in [Pavlov97]. It provides a characterization of gestures' spatial properties that divides gesture interpretation into two groups 3D hand model-based and appearance-based. Appearance-based methods were developed for hand posture and gesture recognition directly from images using visual features such as hand contours and fingertips positions. Other methods, called 3D model-based, provide a geometrical representation of the hand configuration using the joint angles of a 3D hand’s articulated structure recovered from the sequence of images

**DATA COLLECTION**

The American Sign Language letter database of hand gestures represent a multi-class problem with 24 classes of letters (excluding J and Z which require motion).The dataset format is patterned to match closely with the classic MNIST. Each training and test case represents a label (0-25) as a one-to-one map for each alphabetic letter A-Z (and no cases for 9=J or 25=Z because of gesture motions). The training data (27,455 cases) and test data (7172 cases) are approximately half the size of the standard MNIST but otherwise similar with a header row of label, pixel1,pixel2....pixel784 which represent a single 28x28 pixel image with grayscale values between 0-255. The original hand gesture image data represented multiple users repeating the gesture against different backgrounds. The Sign Language MNIST data came from greatly extending the small number (1704) of the color images included as not cropped around the hand region of interest. The collection of the data is from large dataset platform KAGGLE. A robust visual recognition algorithm could provide not only new benchmarks that challenge modern machine learning methods such as Convolutional Neural Nets but also could pragmatically help the deaf and hard-of-hearing better communicate using computer vision applications. The National Institute on Deafness and other Communications Disorders (NIDCD) indicates that the 200-year-old American Sign Language is a complete, complex language (of which letter gestures are only part) but is the primary language for many deaf North Americans. ASL is the leading minority language in the U.S. after the "big four": Spanish, Italian, German, and French. One could implement computer vision in an inexpensive board computer like Raspberry Pi with OpenCV, and some Text-to-Speech to enabling improved and automated translation applications

**METHODOLOGY**

TOOLS USED**:**

**Personal Computer** - This is the vital component of the model. Using Jupyter notebook, we run the code that is created to detect animals.

**Jupyter notebook** - The Jupyter Notebook is an open-source web application that allows data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

**Camera** - Secondary camera used to record the field area live 24/7 and this is fed as an input to the model to detect animals.

**Speake**r - To amplify the sound that is received from the model to the field area to drive away animals.

**LIBRARIES USED:**

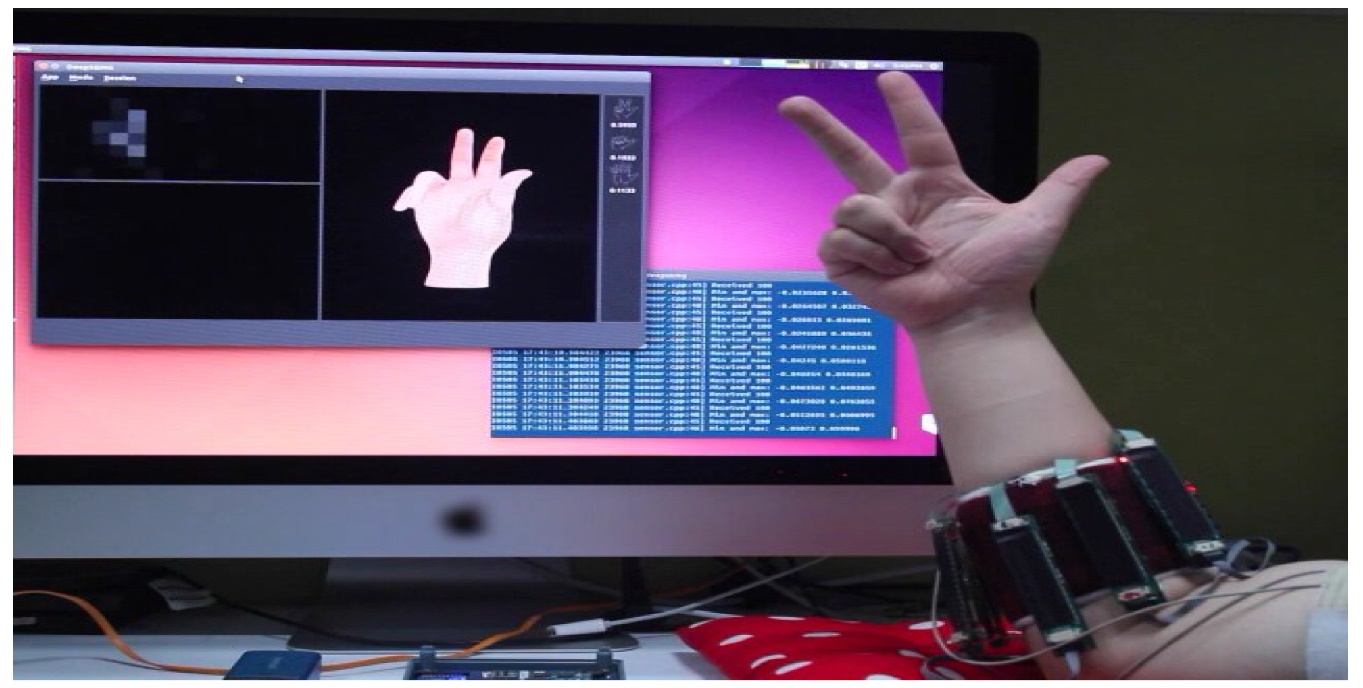
The libraries imported from the above packages that we used in the program are:

**1.Sequential**: Imported from keras.models package. The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs. The Sequential model API is great for developing deep learning models. Models are defined by creating instances of layers and connecting them directly to each other in pairs, and then defining a model that specifies the layers to act as the **input** and **output** to the model, via the parameters inputs and output respectively.

**2.Dense**: Imported keras.layers package implements the operation: output = activation(dot(input, kernel) + bias) where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer (only applicable if use\_bias is True).

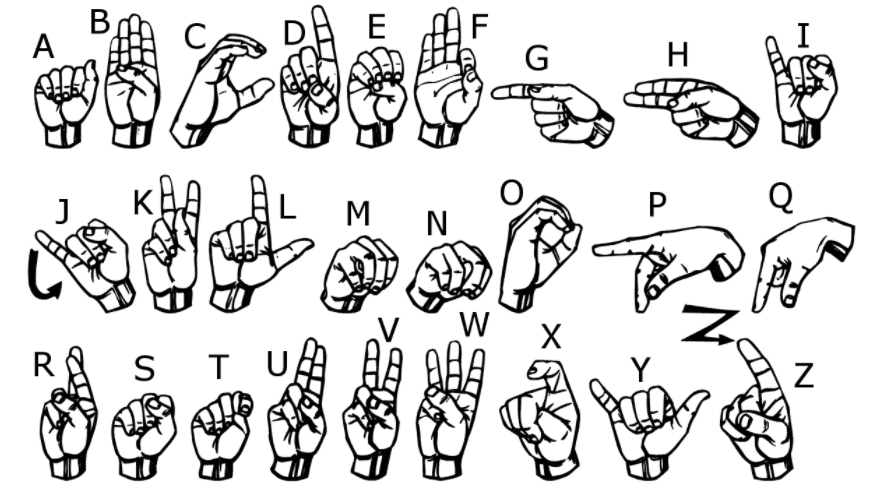
**3.Convo2D**: Imported from keras.layers package. This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. If use\_bias is True, a bias vector is created and added to the outputs. Finally, if activationis not None, it is applied to the outputs as well. When using this layer as the first layer in a model, provide the keyword argument input\_shape (tuple of integers, does not include the batch axis), e.g. input\_shape= (128, 128, 3) for 128x128 RGB pictures in data\_format="channels\_last".

**4.MaxPooling**:Max pooling take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling. **5.Flatten**: As the name of this step implies, we are literally going to flatten our pooled feature map into a column. What happens after the flattening step is that you end up with a long vector of input data that you then pass through the artificial neural network to have it processed further.

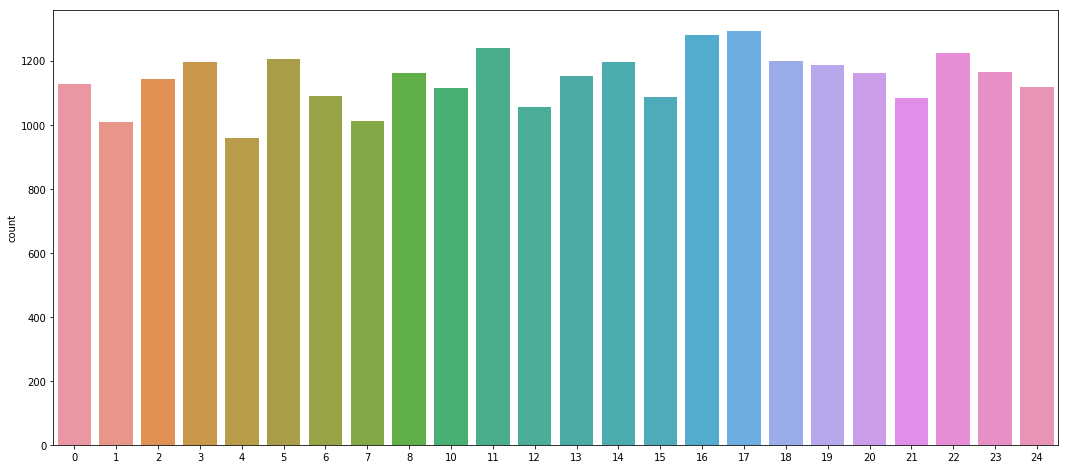


**FIGURES AND TABLE**

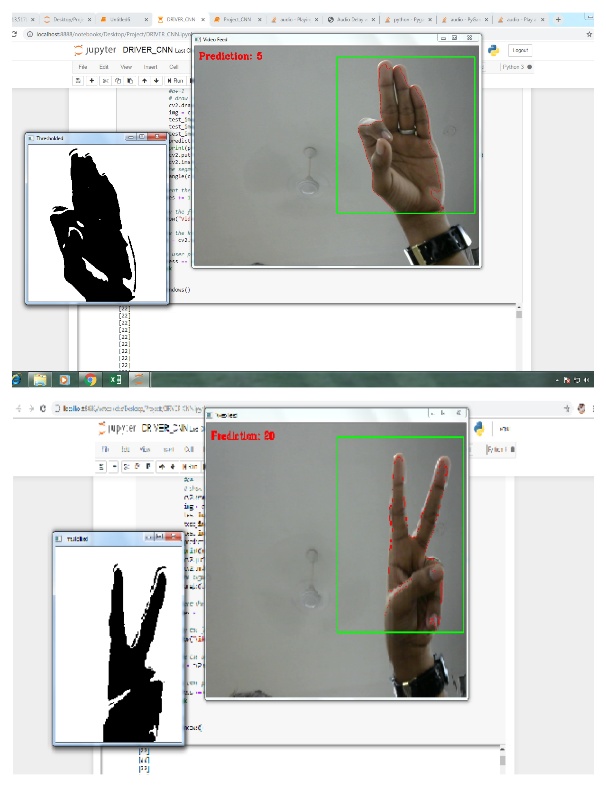
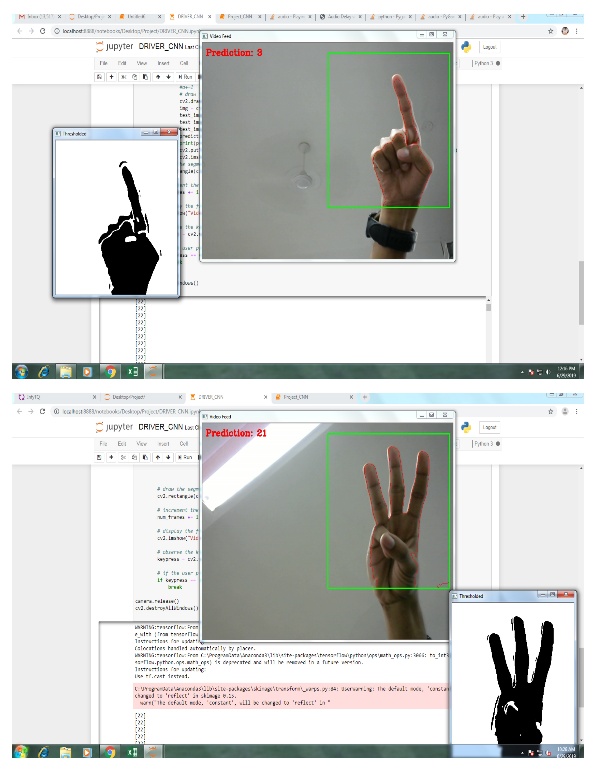
**ACTUALS**



DATASET - COUNT



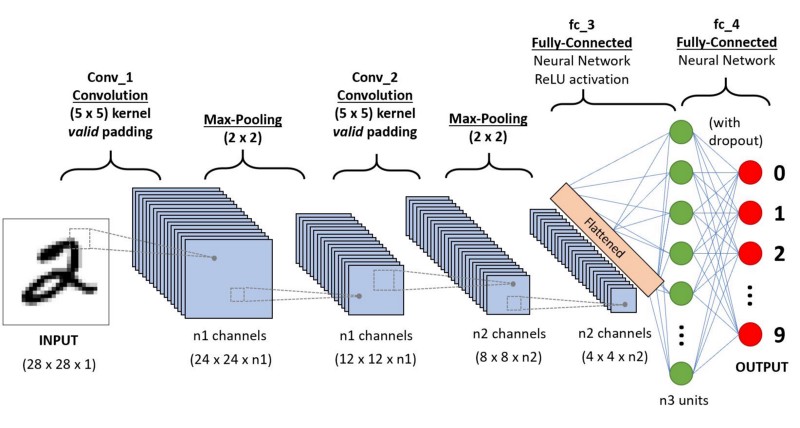
**PREDICTIONS**

**DATA MODELLING**

**CONVOLUTION NEURAL NETWORKS**

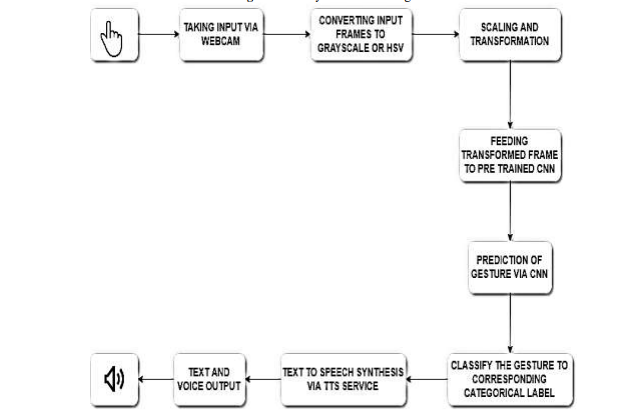
A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets can learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area. The OpenCV library is not enough to start your project. This library provides you the software side, but you also need hardware components.



The OpenCV is a free and open-source library focused on real-time image processing. It can detect and recognize a large variety of objects, but our focus now is to apply techniques and methods to detect and recognize the gestures of a human hand.

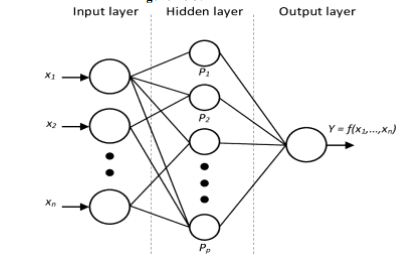
**Keras**: Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error. This makes Keras easy to learn and easy to use. As a Keras user, you are more productive, allowing you to try more ideas than your competition, faster -- which in turn helps you win machine learning competitions.

**Webcam**: A webcam is a video camera that feeds or streams its image in real time to or through a computer to a computer network. When "captured" by the computer, the video stream may be saved, viewed or sent on to other networks travelling through systems such as the internet, and e-mailed as an attachment. When sent to a remote location, the video stream may be saved, viewed or on sent there. Unlike an IP camera (which connects using Ethernet or Wi-Fi), a webcam is generally connected by a USB cable, or similar cable, or built into computer hardware, such as laptops. Convolution is a process where the network tries to label the input signal by referring to what it has learned in the past. If the input signal looks like previous “Hello” gesture images it has seen before, the “Hello” reference signal will be mixed into, or convolved with, the input signal. The resulting output signal is then passed on to the next layer. The output signal strength is not dependent on where the features are located. Inputs from the convolution layer can be “smoothened” to reduce the sensitivity of the filters to noise and variations. This smoothing process is called subsampling , and can be achieved by taking averages or taking the maximum over a sample of the signal. Examples of subsampling methods (for image signals) include reducing the size of the image, or reducing the color contrast across red, green, blue (RGB) channels. The next step is Activation.



The activation layer controls how the signal flows from one layer to the next, emulating how neurons are fired in our brain. Output signals which are strongly associated with past references would activate more neurons, enabling signals to be propagated more efficiently for identification. CNN is compatible with a wide variety of complex activation functions to model signal propagation, the most common function being the **Rectified Linear Unit (ReLU)**, which is favored for its faster training speed.

The last layers in the network are fully connected, meaning that neurons of preceding layers are connected to every neuron in subsequent layers. This mimics high level reasoning where all possible pathways from the input to output are considered. When training the neural network, there is additional layer called the loss layer. This layer provides feedback to the neural network on whether it identified inputs correctly, and if not, how far off its guesses were. This helps to guide the neural network to reinforce the right concepts as it trains.



The image that is captured through the webcam is initially passed and processed through the Original Image filter, due to which it processed into required format and according to the threshold values the image is being detected and the respective word and the speech is displayed and spoken

**CONCLUSION**

In this paper, we presented the concept of gesture-to-speech conversion concept, due to which the communication between the vocally impaired people of the society and the common people will be carried out without any obstruction. The explanation of the design and implementation is presented along with the prototype which captures the gesture of the ARDA sign language. As compared to the other system this concept not only focuses on the gesture to word display but also on the speech synthesis.

Based on the results presented in the previous section, we can conclude that our algorithm successfully classifies different hand gestures images with enough confidence (>90%) based on a Deep Learning model. The accuracy of our model is directly influenced by a few aspects of our problem. The gestures presented are reasonably distinct, the images are clear and without background. Also, there is a reasonable quantity of images, which makes our model more robust. The drawback is that for different problems, we would probably need more data to stir the parameters of our model into a better direction. Moreover, a deep learning model is very hard to interpret, given it’s abstractions. However, by using this approach it becomes much more easier to start working on the actual problem. we don’t need to pre-process the images with edge or blob detectors to extract the important features; the CNN does it for us. Also, it can be adapted to new problems relatively easily, with generally good performance.

REFERENCE

* <https://towardsdatascience.com/tutorial-using-deep-learning-and-cnns-to-make-a-hand-gesture-recognition-model-371770b63a51>
* <https://medium.com/adventures-with-deep-learning/hand-gesture-recognition-with-3d-cnn-part-1-63db0a2f91c9>
* <https://medium.com/@aggirma/keras-convolutional-neural-network-cnn-implementation-for-hand-gesture-recognition-d7dd11958af6>
* <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html>