**Smart Bridge-Remote Internship Program**

**TOPIC : CONVERSATION ENGINE FOR DEAF AND DUMB**

Devolped by:Monika K J,Nanditha K N, Spoorthy M N,Navya Gadina, Nethravathi B

**ABSTRACT**

Sign language is used by deaf and hard hearing people to exchange information between their own community and with other people. Computer recognition of sign language deals from sign gesture acquisition and continues till text/speech generation. Sign gestures can be classified as static and dynamic. However static gesture recognition is simpler than dynamic gesture recognition but both recognition systems are important to the human community.

Inability to speak is considered to be true disability. People with this disability use different modes to communicate with others, there are number of methods available for their communication one such common method of communication is sign language. Developing sign language application for deaf people can be very important, as they’ll be able to communicate easily with even those who don’t understand sign language. Our project aims at taking the basic step in bridging the communication gap between normal people, deaf and dumb people using sign language.

**1. INTRODUCTION**

Deaf people around the world communicate using sign language as distinct from spoken language in their everyday a visual language that uses a system of manual, facial and body movements as the means of communication. Sign language is not a universal language, and different sign languages are used in different countries, like the many spoken languages all over the world. Some countries such as Belgium, the UK, the USA or India may have more than one sign language. Hundreds of sign languages are in used around the world, for instance, Japanese Sign Language, British Sign Language (BSL), Spanish Sign Language, Turkish Sign Language. Motion of any body part like face, hand is a form of gesture. Here for gesture recognition we are using image processing and CNN. Gesture recognition enables computer to understand human actions and also acts as an interpreter between computer and human. This could provide potential to human to interact naturally with the computers without any physical contact of the mechanical devices. Gestures are performed by deaf and dumb community to perform sign language. This community used sign language for their communication when broadcasting audio is impossible, or typing and writing is difficult, but there is the vision possibility. At that time sign language is the only way for exchanging information between people. Normally sign language is used by everyone when they do not want to speak, but this is the only way of communication for deaf and dumb community. Sign language is also serving the same meaning as spoken language does. This is used by deaf and dumb community all over the world. Sign language can be performed by using Hand gesture either by one hand or two hands. It is of two type Isolated sign language and continuous sign language. Isolated sign language consists of single gesture having single word while continuous ISL or Continuous Sign language is a sequence of gestures that generate a meaningful sentence.

The algorithmic models are classified into two categories: Generative and Discriminative. Generative models rely heavily on the basis of prior knowledge and use more of hand engineered components like Support Vector Machine (SVM) and other feature extraction techniques. Discriminative models have little prior knowledge and learn from the data given like Neural Networking, specifically Convolutional Neural Network (CNN) which is an efficient approach. Convolutional Neural Network is a cutting-edge method which is used for object and edge detection in images. It is a layered structure which involves kernels or filters that work in a pipe line method to extract multiple complex features. After convolutional layer max pooling is applied, a fully connected layer yields all of the desired features. Softmax is then applied to further detailed results. In order to build an efficient system, training data was augmented.

**1.1 Purpose**

Our aim from the project is to make use of pandas, matplotlib, numpy, keras libraries from python to extract the libraries for deep learning for the hand gesture prediction.

**1.2 Overview**

Deep-learning-based techniques and methods are becoming popular in hand gesture recognition studies, as their performance is superior in image analysis fields, such as object detection, image classification and semantic segmentation. Deep learning techniques have achieved state of-the-art performance for automatic segmentation of hand gestures through multi-model image sensing. The Convolutional Neural Network (CNN) is a powerful method for image recognition and prediction. However, CNN is mostly used for segmentation, classification, and prediction and by using the algorithm of a Flask model has been implemented and tested. Among all the deep learning methods and techniques, CNNs perform better for image segmentation, classification, and prediction.

**1.3 Proposed Solution**

We are making use of convolution neural network to create a model which is trained on different hand gestures. Using opencv the sign languages are predicted. This enables deaf and dumb people to convey their information using signs which gets converted to human understandable language.

**1.4 Literature Survey**

In the recent years, there has been tremendous research on the hand sign language gesture recognition. The technology for gesture recognition is given below.

**Visionbased**

In visionbased methods computer camera is the input device for observing the information of hands or fingers. The Vision Based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices. These systems tend to complement biological vision by describing artificial vision systems that are implemented in software and/or hardware. This poses a challenging problem as these systems need to be background invariant, lighting insensitive, person and camera independent to achieve real time performance. Moreover, such systems must be optimized to meet the requirements, including accuracy and robustness.

Vision based analysis, is based on the way human beings perceive information about their surroundings, yet it is probably the most difficult to implement in a satisfactory way. Several different approaches have been tested so far.

1. One is to build a threedimensional model of the human hand. The model is matched to images of the hand by one or more cameras, and parameters corresponding to palm orientation and joint angles are estimated. These parameters are then used to perform gesture classification.

2. Second one to capture the image using a camera then extract some feature and those features are used as input in a classification algorithm for classification.

**2. ALGORITHM (Convolutional Neural Network)**

Neural networks, as its name suggests, is a machine learning technique which is modeled after the brain structure. It comprises of a network of learning units called neurons. These neurons learn how to convert **input signals** (e.g. picture of a cat) into corresponding **output signals,**forming the basis of automated recognition.

A convolutional neural network (CNN, or ConvNet) is a type of feedforward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. CNNs have repetitive blocks of neurons that are applied across space (for images) or time (for audio signals etc). For images, these blocks of neurons can be interpreted as 2D convolutional kernels, repeatedly applied over each patch of the image. For speech, they can be seen as the 1D convolutional kernels applied across time windows. At training time, the weights for these repeated blocks are 'shared', i.e. the weight gradients learned over various image patches are averaged.

**2.1 Steps in CNN**

There are four main steps in CNN: convolution, subsampling, activation and full connectedness.

**1. Convolution**

The first layers that receive an input signal are called convolution filters. 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 cat images it has seen before, the cat reference signal will be mixed into, or convolved with, the input signal. The resulting output signal is then passed on to the next layer.

**2. Subsampling**

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.

**3. Pooling**

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 favoured for its faster training speed.

**4. Fully Connected**

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.

**2.2 Implementation**

Algorithms used in training CNN are analogous to studying for exams using flash cards. First, you draw several flashcards and check if you have mastered the concepts on each card. For cards with concepts that you already know, discard them. For those cards with concepts that you are unsure of, put them back into the pile. Repeat this process until you are fairly certain that you know enough concepts to do well in the exam. This method allows you to focus on less familiar concepts by revisiting them often. Formally, these algorithms are called gradient descent algorithms for forward pass learning. Modern deep learning algorithm uses a variation called stochastic gradient descent, where instead of drawing the flashcards sequentially, you draw them at random. If similar topics are drawn in sequence, the learners might overestimate how well they know the topic. The random approach helps to minimize any form of bias in the learning of topics. Learning algorithms require feedback. This is done using a **validation set** where the CNN would make predictions and compare them with the true labels or ground truth. The predictions which errors are made are then fed backwards to the CNN to refine the weights learned, in a so called backwards pass. Formally, this algorithm is called **backpropagation of errors**, and it requires functions in the CNN to be differentiable (almost).

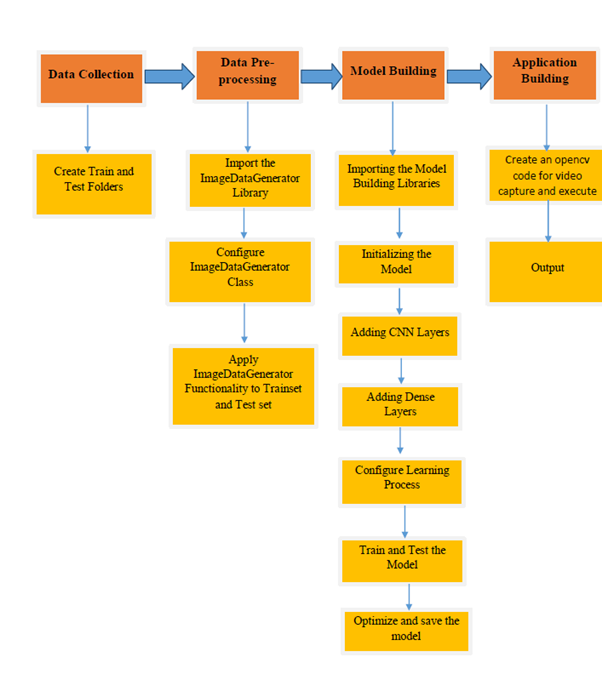
**2.3 Inception**

Inception is a huge image classification model with millions of parameters that can differentiate a large number of kinds of images. By using the pertrained network, we are using that information as input to the final classification layer that distinguishes our dataset.

**3. THEORETICAL ANALYSIS**

The computer-aided mechanisms are applied to detect the hand gestures. The data set used consists of hand gestures of numbers from zero to five, with around 780 images belonging to 6 gestures categories. In **deep learning**, computer-aided mechanisms is generally done by extracting features through a convolutional neural network (**CNN**) and then classifying using a fully connected network.We have trained a convolutional neural network and obtained a prediction accuracy of up to 97%. CNN is a modified variety of deep neural net which depends upon the correlation of neighbouring pixels. It uses randomly defined patches for input at the start, and modifies them in the training process. Once training is done, the network uses these modified patches to predict and validate the result in the testing and validation process. Convolutional neural networks have achieved success in the image classification problem, as the defined nature of CNN matches the data point distribution in the image. As a result, many image processing tasks adapt CNN for automatic feature extraction.

**3.1 Block Diagram**



**3.2 Software Designing**

* Jupyter Notebook Environment
* Spyder
* Python (Sequential,Dense ,Conv2D,MaxPool2D,Flatten)

We developed this hand gesture detection by using the Python language, which is a high level programming language along with Deep Learning Algorithm such as CNN. For coding we used the Jupyter Notebook of Anaconda distributions and Spyder, an integrated scientific programming in python language. opencv is used as a user interface for the prediction.

**Opencv**

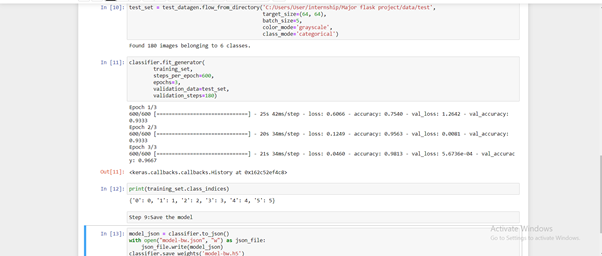
Opencv(Open source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception. The library has more than 2500 optimized algoritms. These algorithms can be used to classify human actions in videos, identify objects.

**4. EXPERIMENTAL ANALYSIS**

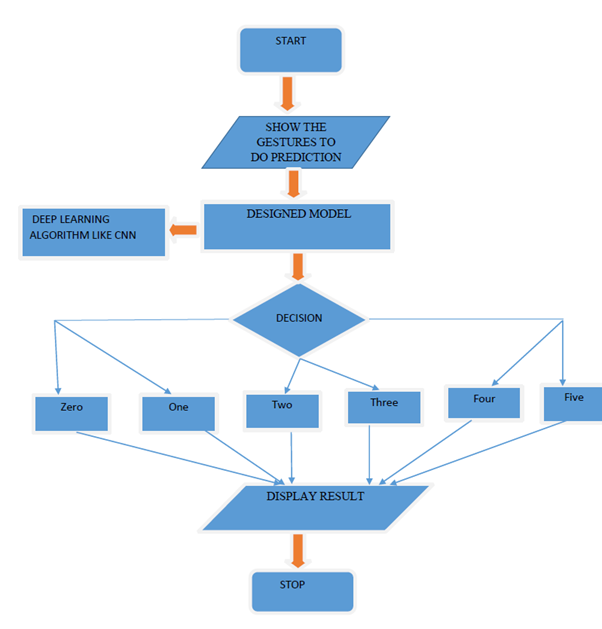
In our project, we have used the hand gestures of numbers dataset. This dataset contains two folders: test set and training set. In test set folder, we have six categories from zero to five. Similarly in the training set folder. Training set consists of 600 images belonging to 6 classes and test set consists of 180 images belonging to 2 classes.







**5. FLOWCHART**

**6.RESULT**

In this paper, the CNN algorithm is used to predict its performance.The results show that, 97% accuracy is achieved.

Screenshots of Output

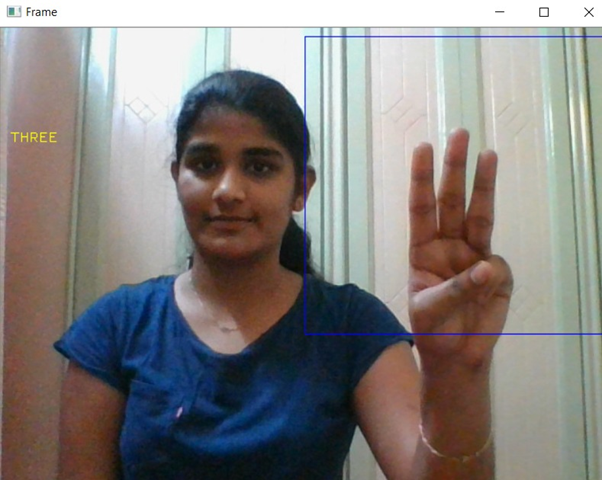


Fig.1 :- Detecting the hand gesture as Three

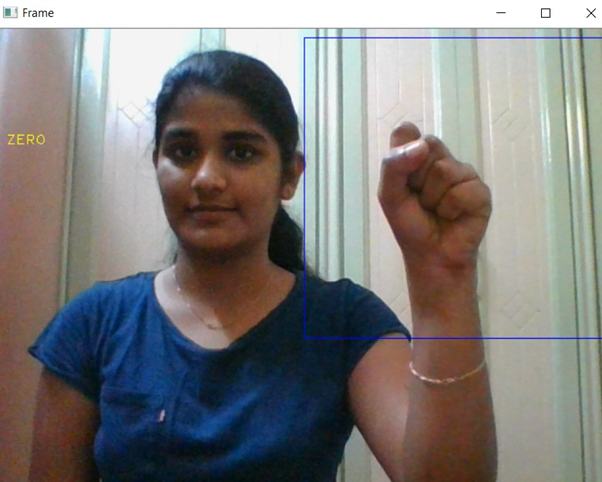


Fig.2 :- Detecting the had gesture as Zero

**7. ADVANTAGES AND DISADVANTAGES**

**Advantages:**

* Easy and simple User Interface for the deaf and dumb people who wants to recognise the signs.
* CNN give the accurate result of the prediction up to 97% which is the algorithm we used for prediction.
* user friendly and low cost
* It can work in real time and predict as soon as the necessary details for prediction are given to the model.

**Disadvantages:**

* datasets challenges can occur.
* it easily generate anatomically incorrect examples.

**8. CONCLUSION**

Hand gestures are a powerful way for human communication, with lots of potential applications in the area of human computer interaction. Hand gesture recognition is a difficult problem and the current work is only a small contribution towards achieving the results needed in the field of sign language gesture recognition.This projects consists of the details about the model which was used for the detection of hand getsures using the number sign images. From the resultant graphs, it is proven that the accuracy of the model has reached good level, if it is deployed in the real-time scenario then it

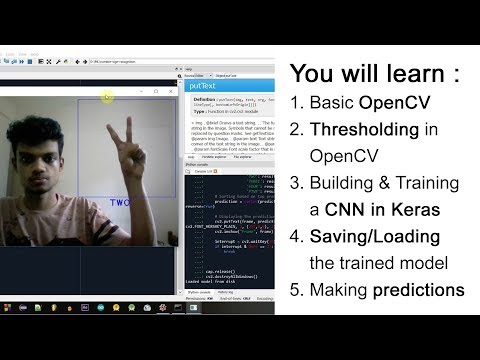
will help as conversation engine.

**9. BIBLIOGRAPHY**

[1]. Cooper, Helen, Brian Holt, and Richard Bowden. "Sign language recognition." In Visual Analysis of Humans, pp. 539562. Springer London, 2011.

[2]. Cooper, Helen, EngJon Ong, Nicolas Pugeault, and Richard Bowden. "Sign language recognition using subunits." Journal of Machine Learning Research 13, no. Jul (2012): 22052231.

[3]. Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning longterm dependencies with gradient descent is difficult." IEEE transactions on neural networks 5, no. 2 (1994): 157166.

[4]. <https://youtu.be/SgKOTgrGnXI>

[5]. https://acadpubl.eu/jsi/2017-116-13-22/articles/21/54.pdf

[6].https://www.goeduhub.com/10560/hand-gesture-recognition-voice-conversion-deaf-dumb-project

**APPENDIX**

**major app.py**

import numpy as np

from keras.models import model\_from\_json

import operator

import cv2

import sys, os

# Loading the model

json\_file = open("model-bw.json", "r")

model\_json = json\_file.read()

json\_file.close()

loaded\_model = model\_from\_json(model\_json)

# load weights into new model

loaded\_model.load\_weights("model-bw.h5")

print("Loaded model from disk")

cap = cv2.VideoCapture(0)

# Category dictionary

categories = {0: 'ZERO', 1: 'ONE', 2: 'TWO', 3: 'THREE', 4: 'FOUR', 5: 'FIVE'}

while True:

\_, frame = cap.read()

# Simulating mirror image

frame = cv2.flip(frame, 1)

# Got this from collect-data.py

# Coordinates of the ROI

x1 = int(0.5\*frame.shape[1])

y1 = 10

x2 = frame.shape[1]-10

y2 = int(0.5\*frame.shape[1])

# Drawing the ROI

# The increment/decrement by 1 is to compensate for the bounding box

cv2.rectangle(frame, (x1-1, y1-1), (x2+1, y2+1), (255,0,0) ,1)

# Extracting the ROI

roi = frame[y1:y2, x1:x2]

# Resizing the ROI so it can be fed to the model for prediction

roi = cv2.resize(roi, (64, 64))

roi = cv2.cvtColor(roi, cv2.COLOR\_BGR2GRAY)

\_, test\_image = cv2.threshold(roi, 120, 255, cv2.THRESH\_BINARY)

cv2.imshow("test", test\_image)

# Batch of 1

result = loaded\_model.predict(test\_image.reshape(1, 64, 64, 1))

prediction = {'ZERO': result[0][0],

'ONE': result[0][1],

'TWO': result[0][2],

'THREE': result[0][3],

'FOUR': result[0][4],

'FIVE': result[0][5]}

# Sorting based on top prediction

prediction = sorted(prediction.items(), key=operator.itemgetter(1), reverse=True)

# Displaying the predictions

cv2.putText(frame, prediction[0][0], (10, 120), cv2.FONT\_HERSHEY\_PLAIN, 1, (0,255,255), 1)

cv2.imshow("Frame", frame)

interrupt = cv2.waitKey(10)

if interrupt & 0xFF == 27: # esc key

break

cap.release()

cv2.destroyAllWindows()