SIGN LANGUAGE TRANSLATOR USING CNN

Akhil A, Akhin A, Ashik SB and Harigovind K

Abstract— **In recent years, human–computer interaction behaviour has appeared more and more in daily life. Especially with the rapid development of computer vision technology, the human- centred. human–computer interaction technology is bound to replace computer-centred human–computer interaction technology. The study of gesture recognition is in line with this trend, and gesture recognition provides a way for many devices to interact with humans. The traditional gesture recognition method requires manual extraction of feature values, which is a time-consuming and laborious method. In order to break through the bottleneck, the implementation of a gesture recognition algorithm based on the convolutional neural network is applied. We apply this method to expression recognition, calculation, and text output, and achieve good results. Through this experiment, we aim to show that the proposed method can train the model to identify gestures with fewer samples and achieve better gesture classification and detection effects. Moreover, this gesture recognition method is less susceptible to illumination and background interference. It also can achieve an efficient real- time recognition effect through which gesture translation for the intended mute populace aid without third party intervention for their ease of living.**

Keywords—**Computer Vision, human–computer interaction, feature values**

I.INTRODUCTION

With the rapid development of science and technology, the way of human-computer interaction has also been greatly changed. Gestures are the most common in all kinds of body language, so it can be used as a simple and free means of human-computer interaction [6]. It has a very broad application prospect. Each individual utilizes language to communicate with others. The listening to disabled individuals likewise utilizes language to communicate among themselves.

Sign language is basically utilized by hearing impaired populace to communicate with others. Communication through signing is a very organized nonverbal language using both non-manual and manual correspondence. Non manual signals are essentially outward appearance, head movement, stance and orientation of the body. While manual signals incorporate movement and orientation of hand that passes on typical significance.

On the other hand, communication with typical individuals is a major impediment for them since not every ordinary person comprehends their gesture-based communication. To beat this issue, a sign language recognition system is expected to assist the deaf and mute people to communicate with normal people. An important process for gesture based human computer interaction is to recognize gestures.

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body. When performing gesture recognition [4] the features of the gesture are first extracted, and then gesture recognition is performed according to the extracted features. There are many common gesture recognition methods. For example, the neural network-based recognition method has a strong ability to classify and identify.[8] However, if the number of neural network layers is generally shallow, it is easy to overfitting; the recognition method based on geometric features performs gesture recognition by extracting gesture structure, edge, contour and other features, it has good stability, but can't improve the recognition rate by increasing the sample size. The recognition method based on the Hidden Markov Model has the ability to describe the time and space changes of gestures, but the recognition speed of the method is not satisfactory.

Some of the major problems faced by a person who are unable to speak is they cannot express their emotion as freely in this world. Utilizing voice recognition and voice search systems in smartphones we can aim to resolve this issue. They are not able to utilize (Artificial Intelligence/personal Butler) like google assistance, or Apple's SIRI etc., because all those apps are based on voice controlling. There is a need for such platforms for such people. American Sign Language (ASL) is a complete, complex language that employs signs made by moving the hands combined with facial expressions and postures of the body. It is the go-to language of many North Americans who are not able to talk and is one of various communication alternatives used by people who are deaf or hard-of-hearing. While sign language is very essential for deaf-mute people, to communicate both with normal people and with themselves, is still getting less attention from the normal people. The importance of sign language has been tending to be ignored, unless there are areas of concern with individuals who are deaf-mute. One of the solutions to talk with the deaf-mute people is by using the mechanisms of sign language. Hand gesture is one of the methods used in sign language for non-verbal communication. It is most commonly used by deaf & dumb people who have hearing or talking disorders to communicate among themselves or with normal people. Various sign language systems have been developed by many manufacturers around the world but they are neither flexible nor cost- effective for the end users.

II. OBJECTIVES

Some of the major problems faced by a person who are unable to speak is they cannot express their emotion as freely in this world. Utilizing voice recognition and voice search systems in smartphones we can aim to resolve this issue. They are not able to utilize (Artificial Intelligence/personal Butler) like google assistance, or Apple's SIRI etc., because all those apps are based on voice controlling. There is a need for such platforms for such people. American Sign Language (ASL) is a complete, complex language that employs signs made by moving the hands combined with facial expressions and postures of the body.. While sign language is very essential for deaf-mute people, to communicate both with normal people and with themselves, is still getting less attention from the normal people. The importance of sign language has been tending to be ignored, unless there are areas of concern with individuals who are deaf-mute. One of the solutions to talk with the deaf-mute people is by using the mechanisms of sign language. Hand gesture is one of the methods used in sign language for non-verbal communication. It is most commonly used by deaf & dumb people who have hearing or talking disorders to communicate among themselves or with normal people. Various sign language systems have been developed by many manufacturers around the world but they are neither flexible nor cost- effective for the end users.

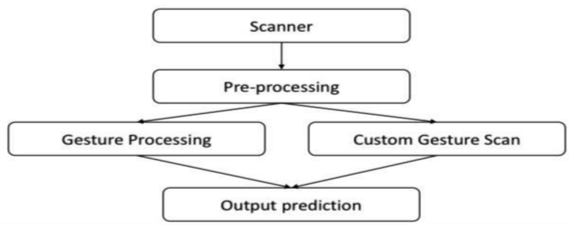


fig 1.Dataflow Diagram

III.APPROACHES AND METHODOLOGY

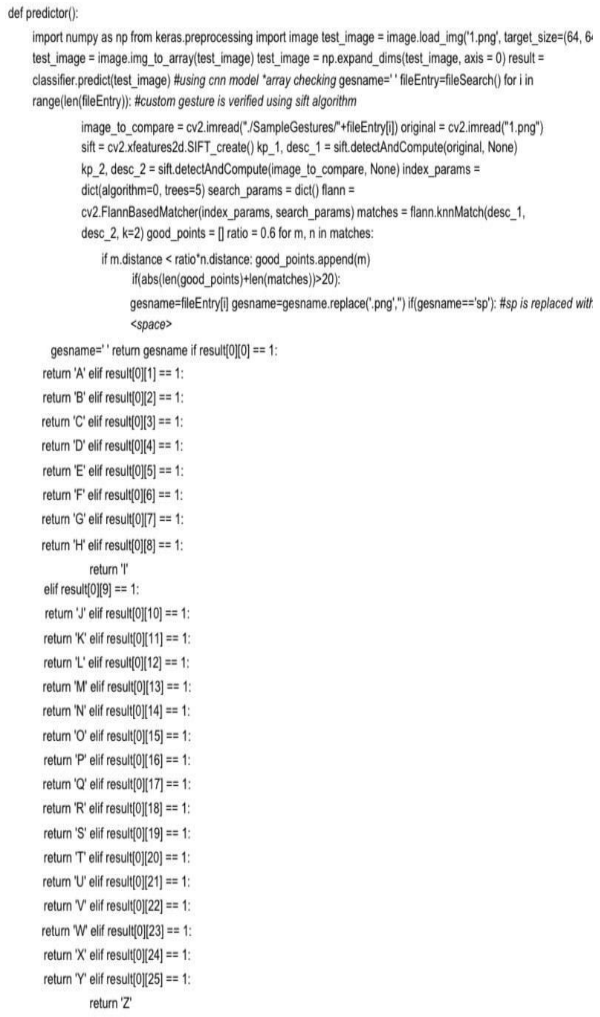
In the project a predesigned CNN model is used in the main program to process the gestures realtime from the user and convert the image captured to specific size mentioned in the code and convert the picture into HSV format. In the basic modules this is covered by scanner and Data Pre-Processing modules, here in the actual program implementation this is also done through them. Sliders are provided in the GUI for adjusting the values between high and low for HSVparameters the image is then captured and stored as "1.png".This is done to gain an optimally clear image from the camera input. This image is then passed to the predictor function which then forwards it to the classifier function which converts the image into a 2D array, in this array there are spaces for storing the value of the letter.For example, suppose a gesture for "A" is given, it will enter value of 1 in the space for 'A' andenter value of 0 in all others. Hence there will be value spaces for all letters in ASL.Predictor uses 2 different algorithms for gesture processing:1. The traditional 2D array for recognizing the gestures with which the mode was trained.2. SIFT algorithm for recognizing the custom gestures added to the system.To create new gestures in this system run the capture program which will re-initiate the gesture recognition process resulting in a HSV image which is saved in a new folder named in accordance to the user. Feedbacks given c for correct predictions.Sentence formation is formed using multiple letters recognised by the system by temporarily appending the letter to a temporary file.Export function is used to store the temporary for usage by voice assistance for converting the word to speech. i.e, for making the software speak the gestured word.Single scan is used for recognizing the image captured frame by frame to return its value,captured images are passed to the predictor after conversion into HSV.Sentence scan is used when there is the need for continuous prediction of images even in the delay. Usage of Tkinter in GUI is only for gesture creation, for all others PyQt is used. TTS is implemented using Google TTS assistance preinstalled in the system which converts the temporary file content into mp3 audio file which is played with assistance of OS in the browser.The actual functional modules of the system are:1. Predictor: processes the gestures and passes it through one of the two compatible algorithms for classification.2. Single Scan: Image captured is recognized frame by frame to return its value, captured images are passed to the predictor after conversion into HSV.3. Sentence Scan: used when there is the need for continuous prediction of images even inthe delay4. Export File: used to store the temporary for usage by voice assistance for converting the word to speech.5. Create Gesture: used for creating and storing custom gestures from the user, initialized by running the capture program which will re-initiate the gesture recognition process resulting in a HSV image which is saved in a new folder named in accordance to the user.The reason for branching the basic modules into differing function modules were for the ease of access and programming in a more efficient and effective manner. Predictor acts as a classification point for determining the control flow to the next stage of operations which can be from arranging differing scenarios like single gesturing, continuous gesturing or custom gesture inclusion. In the implementation to better facilitate needs of the user the GUI was prepared with an option menu to various other function screens from the current one. That is to say one can change between the function screens a from other screens after entering via Main GUI Screen. The flow of operations from Predictor to single gestures will result in two possible branches being either single gesture recognition or custom gesture recognition. The user can set custom gestures to convey emotions which is not possible through traditional ASL by a single gesture or simply gesture a letter in their daily life.

A.Convolutional Neural Network

When programming a CNN, the input is a tensor with shape number of images x image width x image height x image depth. Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with shape number of images x feature map width x feature map height x feature map channels. A convolutional layer within a neural network should have the following attributes: Convolutional kernels defined by a width and height. The Number of input channels and output channels. The depth of the Convolution filter must be equal to the number channels of the input feature map. Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neural processes data only for its receptive field.[7] Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for an image of size 100 x 100 has 10,000 weights for each neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using backpropagation.

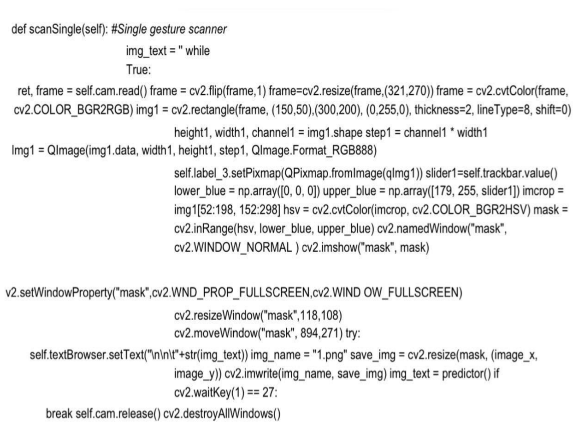
B. Implemented Modules

1)Predictor: processes the gestures and passes it through one of the two compatible algorithms for classification



```
def predictor():
    import numpy as np from keras.preprocessing import image test_image = image.load_img('1.png', target_size=(64, 6
    test_image = image.img_to_array(test_image) test_image = np.expand_dims(test_image, axis = 0) result =
    classifier.predict(test_image) #using cnn model *array checking gesname=' ' fileEntry=fileSearch() for i in
    range(len(fileEntry)): #custom gesture is verified using sift algorithm
            image_to_compare = cv2.imread("./SampleGestures/"+fileEntry[i]) original = cv2.imread("1.png")
            sift = cv2.xfeatures2d.SIFT_create() kp_1, desc_1 = sift.detectAndCompute(original, None)
            kp_2, desc_2 = sift.detectAndCompute(image_to_compare, None) index_params =
            dict(algorithm=0, trees=5) search_params = dict() flann =
            cv2.FlannBasedMatcher(index_params, search_params) matches = flann.knnMatch(desc_1,
            desc_2, k=2) good_points = [] ratio = 0.6 for m, n in matches:
              if m.distance < ratio*n.distance: good_points.append(m)
                  if(abs(len(good_points)+len(matches))>20):
                  gesname=fileEntry[i] gesname=gesname.replace('.png','') if(gesname=='sp'): #sp is replaced with
                  <space>
      gesname=' ' return gesname if result[0][0] == 1:
    return 'A' elif result[0][1] == 1:
    return 'B' elif result[0][2] == 1:
    return 'C' elif result[0][3] == 1:
    return 'D' elif result[0][4] == 1:
    return 'E' elif result[0][5] == 1:
    return 'F' elif result[0][6] == 1:
    return 'G' elif result[0][7] == 1:
    return 'H' elif result[0][8] == 1:
            return 'I'
    elif result[0][9] == 1:
    return 'J' elif result[0][10] == 1:
    return 'K' elif result[0][11] == 1:
    return 'L' elif result[0][12] == 1:
    return 'M' elif result[0][13] == 1:
    return 'N' elif result[0][14] == 1:
    return 'O' elif result[0][15] == 1:
    return 'P' elif result[0][16] == 1:
    return 'Q' elif result[0][17] == 1:
    return 'R' elif result[0][18] == 1:
    return 'S' elif result[0][19] == 1:
    return 'T' elif result[0][20] == 1:
    return 'U' elif result[0][21] == 1:
    return 'V' elif result[0][22] == 1:
    return 'W' elif result[0][23] == 1:
    return 'X' elif result[0][24] == 1:
    return 'Y' elif result[0][25] == 1:
            return 'Z'
```

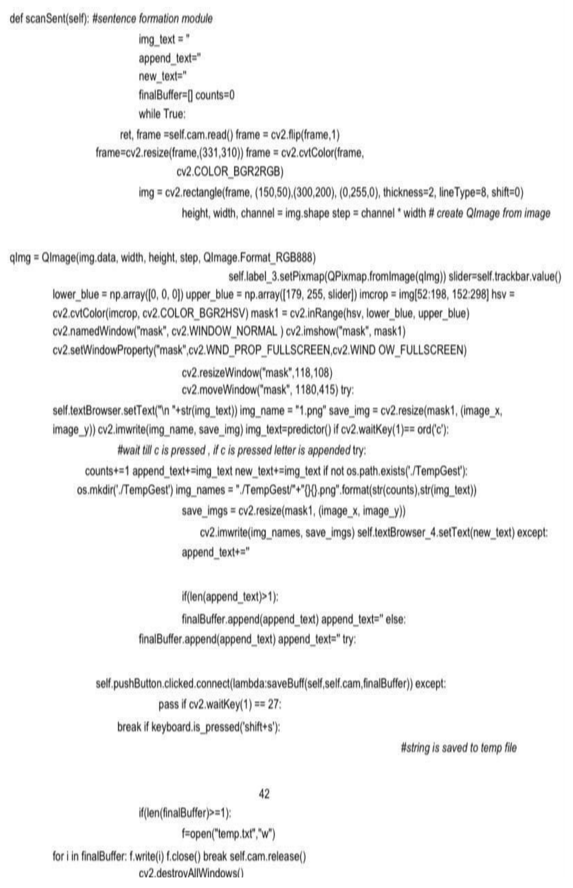
2) Single Scan: Image captured is recognized frame by frame to return its value, captured images are passed to the predictor after conversion into HSV.



```
def scanSingle(self): #Single gesture scanner
                              img_text = '' while
                              True:
 ret, frame = self.cam.read() frame = cv2.flip(frame,1) frame=cv2.resize(frame,(321,270)) frame = cv2.cvtColor(frame,
 cv2.COLOR_BGR2RGB) img1 = cv2.rectangle(frame, (150,50),(300,200), (0,255,0), thickness=2, lineType=8, shift=0)
                                      height1, width1, channel1 = img1.shape step1 = channel1 * width1
Img1 = QImage(img1.data, width1, height1, step1, QImage.Format_RGB888)
                                      self.label_3.setPixmap(QPixmap.fromImage(qImg1)) slider1=self.trackbar.value()
                                      lower_blue = np.array([0, 0, 0]) upper_blue = np.array([179, 255, slider1]) imcrop =
                                      img1[52:198, 152:298] hsv = cv2.cvtColor(imcrop, cv2.COLOR_BGR2HSV) mask =
                                      cv2.inRange(hsv, lower_blue, upper_blue) cv2.namedWindow("mask",
                                      cv2.WINDOW_NORMAL ) cv2.imshow("mask", mask)

v2.setWindowProperty("mask",cv2.WND_PROP_FULLSCREEN,cv2.WIND OW_FULLSCREEN)
                                      cv2.resizeWindow("mask",118,108)
                                      cv2.moveWindow("mask", 894,271) try:
    self.textBrowser.setText("\n\n\t"+str(img_text)) img_name = "1.png" save_img = cv2.resize(mask, (image_x,
                                      image_y)) cv2.imwrite(img_name, save_img) img_text = predictor() if
                                      cv2.waitKey(1) == 27:
       break self.cam.release() cv2.destroyAllWindows()
```

3) Sentence Scan: used when there is the need for continuous prediction of images even in the delay



```
def scanSent(self): #sentence formation module
                              img_text = ''
                              append_text=''
                              new_text=''
                              finalBuffer=[] counts=0
                              while True:
                           ret, frame =self.cam.read() frame = cv2.flip(frame,1)
                       frame=cv2.resize(frame,(331,310)) frame = cv2.cvtColor(frame,
                                     cv2.COLOR_BGR2RGB)
                              img = cv2.rectangle(frame, (150,50),(300,200), (0,255,0), thickness=2, lineType=8, shift=0)
                                      height, width, channel = img.shape step = channel * width # create QImage from image

qImg = QImage(img.data, width, height, step, QImage.Format_RGB888)
                                               self.label_3.setPixmap(QPixmap.fromImage(qImg)) slider=self.trackbar.value()
        lower_blue = np.array([0, 0, 0]) upper_blue = np.array([179, 255, slider]) imcrop = img[52:198, 152:298] hsv =
        cv2.cvtColor(imcrop, cv2.COLOR_BGR2HSV) mask1 = cv2.inRange(hsv, lower_blue, upper_blue)
        cv2.namedWindow("mask", cv2.WINDOW_NORMAL ) cv2.imshow("mask", mask1)
        cv2.setWindowProperty("mask",cv2.WND_PROP_FULLSCREEN,cv2.WIND OW_FULLSCREEN)
                                      cv2.resizeWindow("mask",118,108)
                                      cv2.moveWindow("mask", 1180,415) try:
        self.textBrowser.setText("\n "+str(img_text)) img_name = "1.png" save_img = cv2.resize(mask1, (image_x,
        image_y)) cv2.imwrite(img_name, save_img) img_text=predictor() if cv2.waitKey(1)== ord('c'):
                           #wait till c is pressed , if c is pressed letter is appended try:
              counts+=1 append_text+=img_text new_text+=img_text if not os.path.exists('./TempGest'):
            os.mkdir('./TempGest') img_names = "./TempGest/"+"{}{}.png".format(str(counts),str(img_text))
                                      save_imgs = cv2.resize(mask1, (image_x, image_y))
                                         cv2.imwrite(img_names, save_imgs) self.textBrowser_4.setText(new_text) except:
                                      append_text+=''

                                      if(len(append_text)>1):
                                      finalBuffer.append(append_text) append_text='' else:
                              finalBuffer.append(append_text) append_text='' try:

                self.pushButton.clicked.connect(lambda:saveBuff(self,self.cam,finalBuffer)) except:
                             pass if cv2.waitKey(1) == 27:
                       break if keyboard.is_pressed('shift+s'):
                                                                                #string is saved to temp file

                                                42
                              if(len(finalBuffer)>=1):
                                      f=open("temp.txt","w")
        for i in finalBuffer: f.write(i) f.close() break self.cam.release()
                              cv2.destroyAllWindows()
```

4) Export File: used to store the temporary for usage by voice assistance for converting the word to speech.

5 Create Gesture: used for creating and storing custom gestures from the user, initialized by running the capture program which will re-initiate the gesture recognition process resulting in a HSV image which is saved in a new folder named in accordance to the user.

IV CONCLUSION

Here a method of gesture recognition based on CNN is introduced and evaluation of the model in a real-world environment. The experimental results show that our model can achieve good results. First, for a specific gesture, by using the recognition model, it can effectively recognize the actual meaning of the gesture. The model can also achieve full automation, and its accuracy can reach a high level. Moreover, in the state where the illumination conditions are not particularly good, the recognition accuracy can be effectively improved by our pre-processing. Next, we will further test and improve our model. We have some preliminary thoughts on how to improve the results. The network also supports the addition of more gestures. In the future, we can even use gestures to play games, chat and email with others. Although the accuracy obtained by the experiment has been very high, we feel that it is necessary to further improve for the application to real life. We plan to further optimize our model by adding training data and changing the network structure.

REFERENCE

[1] Shobhit Agarwal, “What are some problems faced by deaf and dumb people while using todays common tech like phones and PCs”, 2017 [Online].

[2] NIDCD, “American sign language”, 2017 [Online]. Available: https://www.nidcd.nih.gov/health/american-sign-language, [Accessed April 06, 2019].

[3] NAD, “American sign language-community and culture frequently asked questions”, 2017 [Online].

[4] J. Lee, Y. Lee, E. Lee, and S. Hong, ``Hand region extraction and gesture recognition from video stream with complex background through entropy analysis,'' in *Proc. Conf. IEEE Eng. Med. Biol. Soc.*, Jan. 2004, vol. 2, no. 2, pp. 1513_1516.

[5] Wei Fang1,2, Yewen Ding 1, Feihong Zhang1, And Jack Sheng3, ``Gesture Recognition Based on CNN and DCGAN for Calculation and Text Output'', February 27, 2019,

[6] T. Takahashi and F. Kishino, ``Hand gesture coding based on experiments using a hand gesture interface device,'' ACM Sigchi Bull., vol. 23, no. 2, 1991.

[7]Wikipedia, “Convolutional neural network”, 2017 [Online]. Available: https://en.wikipedia.org/wiki/Convolutional_neural_network, [Accessed April 08, 2019].

.[8] Ricky Anderson ,FannyWiryana ,Meita Chandra Ariesta, ``Sign Language

Recognition Application Systems for Deaf-Mute People: A Review Based on Input- Process-Output,''2017 [Online].

[9] R. Meng, S. G. Rice, J. Wang, and X. Sun, ‘‘A fusion steganographic algorithm based on faster R-CNN,’’ CMC, Comput., Mater. Continua, vol. 55, no. 1, pp. 1–16, 2018.

[10] C. Szegedy et al., ‘‘Going deeper with convolutions,’’ presented at the CVPR, Boston, MA, USA, Jun. 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image

recognition,’’ in Proc. CVPR Las Vegas, NV, USA, 2016, pp.770–778.