Recognition of sign language using image processing

Sandhya Arora* and Ananya Roy

Department of Computer Engineering,
Cummins College of Engineering for Women,
Pune, India
Email: sandhya.arora@cumminscollege.in
Email: ananya.roy@cumminscollege.in
*Corresponding author

Abstract: According to World Health Organization, over 5% of the world’s population have hearing and speaking disabilities. The primary language of communication for people who are deaf and mute is the sign language. The proposed system aims to recognise the American Sign Language and converts it to text. Input given to the system is an image of the hand depicting the necessary alphabet. The histogram of the input image is then computed and checked for similarity with the histograms of pre-saved images by using the Bhattacharyya Distance Metric. Implementation of the system will be a small step in overcoming the social barrier of communication between the deaf-mute people and the people who do not understand sign language. OpenCV is used as a tool for implementing proposed system.

Keywords: American Sign Language; histogram; Bhattacharya Distance Metric; OpenCV.


Biographical notes: Sandhya Arora has completed her PhD from Jadavpur University, Kolkata, Indi., She is currently working as a Professor at the Department of Computer Engineering in Cummins College of Engineering for Women, Pune, India. She has a teaching experience of 20 years. She has more than 25 research publications in various national/international conferences/journals. Her areas of interest and research includes machine learning and pattern recognition, artificial intelligence and high performance computing, data mining and information retrieval.

This paper is a revised and expanded version of a paper entitled ‘Recognition of Sign Language using Image Processing’, presented at IRO World Science and Technology Congress, 23-24 September 2016, Hong Kong.

Copyright © 2018 Inderscience Enterprises Ltd.
1 Introduction

As known to many, sign languages are the only medium through which the educated deaf and mutes communicate all over the world. World Health Organization states that over 5% of the world’s population are a part of this category (Lewis et al., 2013). However, most of the people blessed with the ability to hear and speak are not well versed with this sign language, thus leading to communication gap. This system aims to bridge this communication gap and aid the deaf and the mute to use technology to carry out their daily transactions by using a simple approach which is easily implementable.

The last decade witnessed a good number of publications (Budia and Bressan, 2007; Agrawal et al., 2016, 2014; De Silva and Fernando, 2011; Kshirsagar and Doye, 2015) in this field. Our system recognises the sign language alphabet by calculating the Bhattacharya distance metric between the histograms of the captured, processed image and stored image and giving that alphabet as output whose image’s histogram has a lesser Bhattacharyya distance with the captured image’s histogram.

The tool used in this implementation is OpenCV. OpenCV is Open Source Computer Vision and it provides an open source and cross platform library of programming functions aimed at providing real time computer vision. It was designed for computational efficiency and has strong focus on real time applications. Released under the BSD license, it is free for both commercial and academic use.

2 Objectives

1. Help the hearing and speaking impaired to communicate with people who do not understand sign language.
2. Help the deaf and mute to carry out their daily transactions without relying on an interpreter.
3. Broaden the use of technology to solve social problems.

2.1 Sign language

Sign language is a natural language and the primary means of communication adopted by the hearing and speaking impaired all over the world. Sign language typically makes use of certain hand gestures, in order to communicate. The linguistic properties of sign language bear many similarities to the commonly spoken languages.

There is a common misconception among the people that there is just one universal sign language. However, this is not true. According to the 2013 edition of Ethnologue, about 137 sign languages are spoken all over the world, with many countries having more than one native sign language (Kumar and Begam, 2014). However, the American Sign Language is the mostly widely used sign language.

Though primary used for communication between the deaf and the dumb, sign language finds its application in the fields of military communications, underwater communications, etc.
3 Literature survey

Technology has always been the best and fastest medium for communication in recent years. There has already been a great deal of work done in the area of text to sign conversion. Many state of the art applications and devices are available, which achieve this conversion. However, almost all these technologies make extensive use of special hardware components which leads to the increase in the overall price of the device and thus limiting its availability to the masses.

3.1 Sign mobiles: Android application

The sign mobile application makes use of three technologies – video relay service (VRS), Outfit 7 and Mimix, in order to convert the sign language into textual and audible output. It makes use of mobile gesture recognition and also supports remote text-to-sign conversion and vice versa. However, it is an extremely complex application from the developer’s point of view and is also platform dependent (Stepanov, 2012).

3.2 EnableTalk: sensor enabled gloves

The EnableTalk Gloves, developed by a group of students in Ukraine, are fitted with a variety of sensors to detect the position of the fingers and palm in space. This data is transmitted through the controller fitted on the back of the glove, via Bluetooth to a mobile device. The mobile device contains the Microsoft Speech API and Bing API in order to convert the sign language to text and speech. A major disadvantage of this system is its cost. It also requires a variety of hardware components to bring about the translation (Truong, 2013).

3.3 Sign language rings

Sign language ring is designed by a group of students from the Asian University and is the recipient of the Red Dot Design Award. The Buddhist prayer beads are an inspiration for this device. Though conceptual, it shows a great future potential.

This device comprises of three rings on each hand and a bracelet. The rings have motion sensors which are responsible to track the movement of the fingers of the person wearing them and for translating the sign language depicted to spoken language. The word in the spoken language is then displayed on the LED Screen present on the bracelet and is also played by the speaker and amplified by the microphone. Both, the speaker and the microphone are a part of the bracelet (Glenn et al., 2005).

4 Proposed method

4.1 Overview

The input given to the system is an image of the hand depicting the sign alphabet. The image is captured by the camera at the given instance of time. The image is then saved onto the disk. The pre-saved images of sign letters and the test image are then loaded.
They are changed to the HSV colour space from the BGR colour space and the histograms of all the images are computed. Bhattacharyya distance, which is a measure to calculate similarity between the histograms, is then calculated and compared. The image whose histogram is closer to the histogram of the test image, i.e., has a lower Bhattacharyya value, is then searched for the corresponding English alphabet. The alphabet is then displayed as output.

4.2 Working

The logical view of the working of the system is given in Figure 1. The input is taken from the camera at real life where the user signs the required alphabet. Then histogram of the captured image is generated and compared with the histogram of the pre-saved images using Bhattacharyya distance measure and the letter corresponding to the minimum Bhattacharyya distance measure is given as output.

**Figure 1  Work flow (see online version for colours)**

4.3 Constraints

There are certain constraints to be abided by, in order to ensure that the system gives efficient results.

1. The input image should have a black background.
2. If there is any other image present in the background, the system might give incorrect results.
3. Accessories should not be present on the hand depicting the letter.
4. The finger spelling should not contain any kind of movement. Hence, the letters J and Z cannot be detected by this system (Wikipedia, n.d.b).

4.4 Processing the input image and calculating the histogram

The input image is captured by the web camera. The camera will remain active for an indefinite period until the ‘Enter’ key is pressed. The captured image is then saved on the disk. The saved image is then converted from the Red-Green-Blue (RGB) colour space to the Hue-Saturation-Value (HSV) colour space.
HSV colour space represents the RGB or BGR colour space in a cylindrical coordinate form. The geometrical space of RGB is rearranged in an attempt to be more intuitive in manipulating with colour as compared to the Cartesian or cube representation used by RGB. It was designed to estimate the way humans understand, interpret and perceive colour. The HSV colour model describes colours in terms of their tints and luminance. Hue is a number between 0 to 360 degrees; Saturation describes the amount of grey in the colour form 0% to 100% and Value works along with saturation and gives a measure of intensity of the colour between 0% and 100%. (Wikipedia, n.d.a).

The comparison between the images is done by comparing the similarity using Bhattacharyya distance measure between the histograms. Thus, there is the need to generate the histogram of each input image. In order to get an accurate histogram of a particular image, it is better if only the intensity component is considered and the colour components are ignored. HSV separates luma, or the image intensity, from chroma or the colour information. This is not achieved in the RGB colour space. Moreover HSV information is less noisy as compared to RGB colour space. Due to these reasons the captured input image is converted to the HSV colour space.

Histogram can be considered as a plot or a graph that gives us certain information about distribution of numerical data. It gives us an estimation of the probability distribution of a continuous quantitative variable. It was introduced by Karl Pearson. In this system, our underlying data is an image. Thus histogram can also be described as a method to gain better perception of an image. By the analysis of the histogram of a particular image, we can get an idea about brightness, contrast, intensity distribution and various other parameters of the image (Docs.opencv.org, 2016).

The first step in constructing a histogram is to divide the entire set of numerical data into a series of intervals called ‘Bins’. The second step is to count how many values fall into each bin. This will give the information about how many data points are within a particular range. In context of an image, counting the number of values in each bin means calculating the number of pixels in a given intensity range (provided intensity is the chosen parameter).

### 4.5 Histogram comparison and alphabet recognition

There are a number of metrics available to compare two histograms and see how well do they match or are close to identical. Some of the most popular metrics to compute matching are correlation, chi-square, intersection and Bhattacharyya distance. Bhattacharyya distance is a bounded and symmetric measure of similarity between histograms. It is preferred when the task involves comparison between two luminance or intensity histograms in image analysis. This metric is named after the coveted statistician-Anil Kumar Bhattacharyya. He worked at the Indian Statistical Institute in 1930s.

The Bhattacharyya Distance is measured with the help of the following formula,

\[
d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{H_1, H_2}} \sum_{l=1}^{N} \sqrt{H_1(l) \cdot H_2(l)}}
\]

where \(H_1\) and \(H_2\) denote the two histograms and \(N\) is the total number of histogram bins. Lower the Bhattacharyya distance between the two histograms, higher is the accuracy of
the match. In other words, if two histograms are similar to each other, they will have a lower Bhattacharyya Distance metric as compared to two histograms which are different (Rajam and Balakrishnan, 2011).

The Bhattacharyya distance between the input image’s histogram and histograms of all the pre-saved images in the database individually is calculated. The pre-saved image’s histogram, with which the histogram of the input image is the most similar, i.e., has minimum Bhattacharyya distance, is selected. Then the alphabet associated with that image is printed out and in this way the conversion from sign language to text is achieved using histogram comparison.

4.6 Advantages

1. It is extremely easy to use and implement.
2. All the technologies used are open source technologies.
3. It is cost-efficient as compared to other techniques available for the conversion of sign language to text, it does not have any special hardware or software requirements. Even the environment used for the development is open source, hence available for free.
4. The given implementation can be executed on any environment and operating system that provides support for OpenCV.
   
   Since OpenCV is a cross platform application and hence is portable, our implementation imbibes portability as a feature.
5. OpenCV is faster as compared to its counterparts and also uses lesser amount of system resources.
6. Using Bhattacharyya distance as a similarity metric ensures that the similarity between the images is not affected due to rotation, shifting or partial capture of the image.
7. No costly hardware components such as sensor-enabled gloves required.

5 Experimental results

In order to implement the proposed method, OpenCV platform has been used for its image processing capabilities. Python is used as a programming language for implementing the algorithm for conversion from sign language to text. After the implementation, two test cases have been designed and tested in order to identify the scope of the method mentioned. Figure 2 represents the implementation flow.

Table 1 shows a sample of the images which are pre-saved in the system along with their calculated histograms and the associated English alphabets.
Figure 2  Implementation flow
Table 1  Signed alphabets and corresponding histogram and English alphabets (see online version for colours)

<table>
<thead>
<tr>
<th>Base image</th>
<th>Histogram</th>
<th>English alphabet</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Base Image" /></td>
<td><img src="image2" alt="Histogram" /></td>
<td>C</td>
</tr>
<tr>
<td><img src="image3" alt="Base Image" /></td>
<td><img src="image4" alt="Histogram" /></td>
<td>Y</td>
</tr>
<tr>
<td><img src="image5" alt="Base Image" /></td>
<td><img src="image6" alt="Histogram" /></td>
<td>I</td>
</tr>
<tr>
<td><img src="image7" alt="Base Image" /></td>
<td><img src="image8" alt="Histogram" /></td>
<td>L</td>
</tr>
</tbody>
</table>
5.1 Test case 1: Input = Complete image of hand

The Input is captured by the camera and saved onto the disk as test image. For the first test case, we consider the complete image of hand (Umbaugh, 2010).

- Input image:

Figure 3  Input image of test case 1 (see online version for colours)

- Corresponding histogram:
Now, comparing the input image’s histogram with the histogram of the pre-saved images, we get the following values of Bhattacharyya distance metric.

- On comparison with histogram of letter C: 0.0
- On comparison with histogram of letter O: 0.0
- On comparison with histogram of letter I: 0.278715892992
- On comparison with histogram of letter L: 0.391359245785
- On comparison with histogram of letter V: 0.393755036001
- On comparison with histogram of letter Y: 0.240704936469.

As mentioned earlier, lower the Bhattacharyya Distance, more the similarity. Hence, it can be concluded that the input image depicts letter C.

- Output: C
- Input image 2:

**Figure 5** Input image 2 of test case 1 (see online version for colours)
• Corresponding histogram:

**Figure 6** Histogram of test case 1 (see online version for colours)

Now, comparing the input image’s histogram with the histogram of the pre-saved images, we get the following values of Bhattacharyya distance metric.

- On comparison with histogram of letter C: 0.393755036001
- On comparison with histogram of letter O: 0.393755036001
- On comparison with histogram of letter I: 0.428041469823
- On comparison with histogram of letter L: 0.402408080554
- On comparison with histogram of letter V: 0.0
- On comparison with histogram of letter Y: 0.403131264004.

As mentioned earlier, lower the Bhattacharyya distance, more the similarity. Hence, it can be concluded that the input image depicts letter V.

- Output: V.

### 5.2 Test case 2: Input = Partial image of hand

Similar to the previous test case, the input is captured by the camera and saved onto the disk as test image. For the second test case, we consider the partial image of hand. The partial image means that some part of the image is not clearly visible or is too close to the camera. The image considered in this test case is unclear, in order to show that the implementation will be able to handle unclear images as well.
• Input image:

**Figure 7** Input image of test case 2 (see online version for colours)

![Input image of test case 2](image)

• Corresponding histogram:

**Figure 8** Histogram of test case 2 (see online version for colours)

![Histogram of test case 2](image)

Now, comparing the input image’s histogram with the histogram of the pre-saved images, we get the following values of Bhattacharyya distance metric.

- On comparison with histogram of letter C: 0.694428573944
- On comparison with histogram of letter O: 0.694428573944
- On comparison with histogram of letter I: 0.689730079648
- On comparison with histogram of letter L: 0.757556546074
- On comparison with histogram of letter V: 0.828553158384
- On comparison with histogram of letter Y: 0.649839370717.
As mentioned earlier, lower the Bhattacharyya distance, more the similarity. Hence, it can be concluded that the input image depicts letter Y.

- Output: Y.

6 Future enhancements

As mentioned earlier, the system has a number of constraints, which limits the number of environments in which it can be deployed. In order to overcome these constraints, the following enhancements can be done in the aforementioned system to convert sign language to text.

6.1 Edge detection to compute similarity

Histograms depend on the colour distribution of the image. Thus, to obtain accurate results we need the background to be black and the hand to be free of accessories. This constraint can be removed by using edge detection algorithms in order to determine the similarity of the images. In edge detection methods, the edges corresponding to the boundaries of the image are detected and used for comparison. A major drawback of this technique is that sometimes some other background object which in reality has no importance in the image can be considered to be important, its edges created and used for comparison, which ultimately results in inaccurate output.

However, edge detection methods work best with greyscale images as compared to RGB images. Comparison with partial images might also give inaccurate results (Datta et al., 2005).

6.2 Using content-based image retrieval systems

Instead of calculating histograms of every image and comparing it with the input image, we can use content-based image retrieval systems. The content-based image retrieval systems query the images based on their content. By incorporating such a system, we can have a huge database of images to compare the input image with, thereby not making the system limited to recognising only the American sign language but also other sign languages. This will improve the speed of the system as well as increase its scope.

However, scalability still remains an issue as content-based image retrieval systems will be able to compare with only those images which were originally present during its development stage. Moreover, the complexity of the system highly increases with the implementation of the system (Datta et al., 2005).

7 Conclusions

We have seen a system to convert sign language into text with minimum complexity. On comparison with other methods, this method is extremely cost efficient as it does not require any complex hardware components. The input is given at runtime by capturing the image of the hand which is depicting the sign language and giving us the output. The system gives accurate results, even when the input is captured partially by the camera due
to some reasons. We can, therefore conclude that the system gives sufficiently accurate results with minimum processing overhead. Such a simple system to convert sign language to text can be very useful to improve the life of the deaf and the mute in more than one way. It will make them self reliant and also increase their contribution in the society, leading to overall development.

References


