AMERICAN SIGN LANGUAGE RECOGNITION

PULKIT SAGAR, MS. VANDANA CHOUDHARY
Maharaja Agrasen Institute of Technology, New Delhi, India

ABSTRACT

Conceptual— The objective of American Sign Language (ASL) to text is to translate the given input images of ASL letters. That is to design a solution that is intuitive and simple which simplifies the communication for the majority of deaf and dumb people. American Sign Language (ASL) substantially facilitates communication in the deaf community. However, there are only ~250,000-500,000 speakers which significantly limits the number of people that they can easily communicate with. The alternative of written communication is cumbersome, impersonal and even impractical when an emergency occurs. The automatic sign language recognition leads to understand the meaning of different signs without the help from expert persons. Sign language recognition system contains different modules: skin segmentation, feature extraction, and recognition. Skin segmentation is used to extract and locate hands in the Images. The purpose of next modules stands for feature extraction, classification and recognition. Based on segmented hands, extract the hand shape and orientation based features is extracted. Finally, classifiers are trained to recognize the signs.

I. INTRODUCTION

American Sign Language (ASL) is the primary language used by many deaf individuals in North America, and it is also used by hard-of-hearing and hearing individuals. The language is as rich as spoken languages and employs signs made with the hand, along with facial gestures and bodily postures.

American Sign Language (ASL) substantially facilitates communication in the deaf community. However, there are only ~250,000-500,000 speakers which significantly limits the number of people that they can easily communicate with. The alternative of written communication is cumbersome, impersonal and even impractical when an emergency occurs. The automatic sign language recognition leads to understand the meaning of different signs without the help from expert persons.

The goal of this project was to build a neural network able to classify which letter of the American Sign Language(ASL) alphabet is being signed, given an image of a signing hand. This project is a first step towards building a possible sign language translator, which can take the alphabet in sign language and translate them into written and oral language. Such a translator would greatly lower the barrier for many deaf and mute individuals to be able to better communicate with others in day to day interactions.

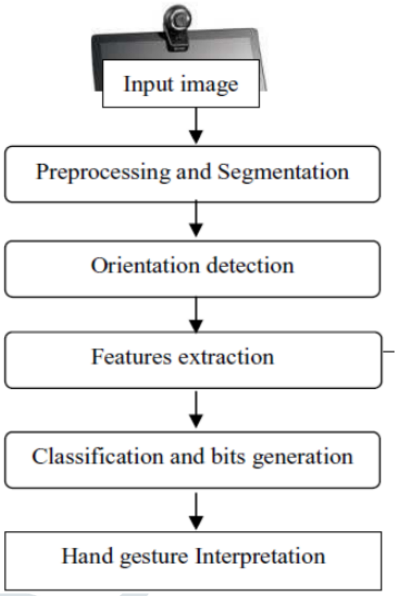
Input image
Preprocessing and Segmentation
Orientation detection
Features extraction
Classification and bits generation
Hand gesture Interpretation

Fig. 1: Data Flow diagram of the Recognition system

II. LITERATURE REVIEW

The importance of sign language may be understood from the fact that early humans used to communicate by using sign language even before the advent of any vocal language. Since then it has been adopted as an integral part of our day to day communication. We make use of hand gestures, knowingly or unknowingly, in our day to day communication. Now, sign languages are being used extensively as international sign use for the deaf and the dumb, in the world of sports by the umpires or referees, for religious practices, on road traffic boards and also at work places. Gestures are one of the first forms of communication that a child learns to express whether it is the need for food, warmth and comfort. It increases the impact of spoken language and helps in expressing thoughts and feelings effectively

The process of gesture recognition can be categorized into a few stages in general, namely data acquisition, pre-processing, segmentation, feature extraction and classification. The input of static gesture recognition is single frames of images, while dynamic sign languages takes video, which is continuous frames of images as input. Vision-based approaches differs from sensor-based approaches mainly by the data-acquisition method. The focus of this section are the methodologies and techniques used by vision-based gesture recognition research.

Data acquisition:

In vision-based gesture recognition, the data acquired is frame of images. The input of such system is collected using image capturing devices such as standard video camera, webcam, stereo camera, thermal camera or more advanced active techniques such as Kinect and LMC. Stereo cameras, Kinect and LMC are 3D cameras which can collect depth information. In this paper, sensor-based recognition involves all techniques of data acquisition which does not use cameras.

Image pre-processing:

Image pre-processing stage are performed to modify the image or video inputs to improve the overall performance of the system. Median filter and Gaussian filter are some of the commonly used techniques to reduce noises in images or video acquired. In research [1], only median filtering is applied in this stage. Next, morphological operation is also widely used to remove unwanted information.

For instance, Pansare etal. [2] first threshold the input image into binary image, then median and Gaussian filters is used to remove noises followed by using morphological operations as the pre-processing stage. In some researches, the images captured are downsized into a smaller resolution prior to subsequent stages. This technique is used in researches [3][4][5][6] has shown that reducing the resolution of the input image is able to improve the computational efficiency.

Research in [7] tabulated the processing time associated with different downsizing factor of image resolution. In this research, division by 64 is the optimum scale as it reduced processing time by 43.8% without affecting the overall accuracy. Histogram equalization is used in [8] to enhance the contrast of the input images taken under different environment to uniform brightness and illumination of the images.

Segmentation :

Segmentation is the process of partitioning images into multiple distinct parts. It is a stage whereby the Region of Interest (ROI), is segmented from the remaining of the image. Segmentation method can be contextual or non-contextual. Contextual segmentation takes the spatial relationship between features into account, such as edge detection techniques. Whereas a non-contextual segmentation does not consider spatial relationship but group pixels based on global attributes.

Skin color segmentation :

Skin color segmentation are mostly performed in RGB, YCbCr, HSV and HSI color spaces. Several challenges toward achieving a robust skin color segmentation is sensitivity to illumination, camera characteristic and skin color. HSV color space is popular as the Hue of palm and arm differs greatly, hence palm can be segmented from the arm easily. Research [9] segments the face and hand in HSV color space.

Chen et al. [10] performed skin color segmentation in RGB color space, using the rule of R > G > B and matching with pre-stored sample skin color to find the skin color. Research [11] found that YCbCr is more robust for skin color segmentation compared to HSV in different illumination condition. Research also found that CIE Lab color space is more robust as compared to YCbCr under different light variation.

A normalized RG space was introduced in [12] to overcome the weakness of RGB which suffers from non-uniformity. Research in [13] proposed using K-means clustering method on the chrominance channels in YCbCr color space to separate the foreground which is the skin pixel from the rest of the background. Skin color distribution and skin-color model classification can overcome the shortfall of applying constant skin- color threshold.

Elmezain etal. [14] performed skin color segmentation in YCbCr color space. In [15], a single Gaussian Model based on YCbCr are used, and the classifier detects skin pixels from the background effectively [16]. Yang et al. [17] implemented the methodology, however, Gaussian model is built instead of histogram model.

Authors in [7] proposed a dynamic skin color modeling method by introducing weighting factors to locally trained skin model and globally trained skin model to obtain an adaptive skin color model.

III. SIGN LANGUAGE RECOGNITION SYSTEM

The sign language recognition done using cameras may be regarded as vision based analysis system. The idea may be implemented using a simple web camera and a computer system. The web camera captures the gesture image with a resolution of 320x240 pixels. The captured image is then processed for recognition purpose. The idea may be represented by the block diagram as shown in figure. Gesture capture using web camera The first step towards image processing is to acquire the image. The acquired image that is stored in the system windows needs to be connected to the software automatically. This is done by creating an object. With the help of high speed processors available in computers today, it is possible to trigger the camera and capture the images in real time. The image is stored in the buffer of the object.As has been already discussed, the image is acquired using a simple web camera. Image acquisition devices typically support multiple video formats. When we create a video input object, we can specify the video format that you want the device to use. If the video format as an argument is not specified, the video input function uses the default format. Acquired image, gesture 'd' Some image acquisition devices use these files to store device configuration information. The video input function can use this file to determine the video format and other configuration information. The image info function is used to determine if our device supports device configuration files. If the input is an RGB image, it can be of class uint8, uint16, single, or double. The output image, I, is of the same class as the input image. If the input is a colormap, the input and output colormaps are both of class double. The acquired image is RGB image and needs to be processed before its features are extracted and recognition is made.. This image is first converted into grayscale as some of the preprocessing operations can be applied on grayscale image only. The edges are detected in the binary image. A number of edge detection techniques may be used in MATLAB. The Edge Detection block finds the edges in an input image by approximating the gradient magnitude of the image. The block convolves the input matrix with the Sobel, Prewitt, or Roberts kernel. The block outputs two gradient components of the image, which are the result of this convolution operation. Alternatively, the block can perform a thresholding operation on the gradient magnitudes and output a binary image, which is a matrix of Boolean values. If a pixel value is 1, it is an edge. For Canny, the Edge Detection block finds edges by looking for the local maxima of the gradient of the input image. It calculates the gradient using the derivative of the Gaussian filter. The Canny method uses two thresholds to detect strong and weak edges. It includes the weak edges in the output only if they are connected to strong edges. As a result, the method is more robust to noise, and more likely to detect true weak edges. In this project we have used canny edge detection. The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Canny edge detection was developed by John F. Canny (JFC) in 1986. Even though it is quite old, it has become one of the standard edge detection methods and it is still used in research. The Canny edge detector works on gray scale image. In image processing finding edge is fundamental problem because edge defines the boundaries of different objects. Edge can be defined as sudden or strong change in the intercity or we can say sudden jump in intensity from one pixel to other pixel. By finding the edge in any image we are just reducing some amount of data but we are preserving the shape. The Canny edge detection algorithm is known as the optimal edge detector. Canny, improved the edge detection by following a list of criteria. The first is low error rate. Low error rate means edges occurring in images should not be missed and that there are NO responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge . Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero The resulting image contains a number of discrete objects. The discontinuities are joined using kNearest Neighbor search. The k-nearest neighbor (kNN) search helps to find the k closest points in X to a query point or set of points. The kNN search technique and kNN-based algorithms are widely used as benchmark learning rules—the relative simplicity of the kNN search technique makes it easy to compare the results from other classification techniques to kNN results. They have been used in various areas such as bioinformatics, image processing and data compression, document retrieval, computer vision, multimedia database, and marketing data analysis. You can use kNN search for other machine learning algorithms, such as kNN classification, local weighted regression, missing data imputation and interpolation, and density estimation. Sign Language Recognition System. After Canny’s Edge Detection Watershed Transform may be used in place of kNN search. Watershed transform computes a label matrix identifying the watershed regions of the input matrix A, which can have any dimension. The elements of L are integer values greater than or equal to 0. The elements labeled 0 do not belong to a unique watershed region. These are called watershed pixels. Once the edges are detected, our aim is to detect the finger tips. Wavelet family method is one of the techniques that may be used to detect the peaks. Wavelet analysis consists of decomposing a signal or an image into a hierarchical set of approximations and details. The levels in the hierarchy often correspond to those in a dyadic scale. From the signal analyst's point of view, wavelet analysis is a decomposition of the signal on a family of analyzing signals, which is usually an orthogonal function method. From an algorithmic point of view, wavelet analysis offers a harmonious compromise between decomposition and smoothing techniques. Unlike conventional techniques, wavelet decomposition produces a family of hierarchically organized decompositions. The selection of a suitable level for the hierarchy will depend on the signal and experience. Often the level is chosen based on a desired low-pass cutoff frequency. The finger tips are detected by finding the 1s at the minimum rows. The width and height of the finger is predefined. Once the finger tips have been detected, our next aim is to match the gesture with the predefined gesture database. This is done using prediction tables. Fuzzy Rule set may be used to make the classification after detecting the finger tips. The logical image is converted back to RGB. An RGB image, sometimes referred to as a true color image, is stored as an m-by-n-by-3 data array that defines red, green, and blue color components for each individual pixel. RGB images do not use a palette. The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel's location. Graphics file formats store RGB images as 24-bit images, where the red, green, and blue components are 8 bits each. This yields a potential of 16 million colors. The precision with which a real-life image can be replicated has led to the nickname "truecolor image." An RGB array [12] can be of class double, uint8, or uint16. In an RGB array of class double, each color component is a value between 0 and 1. A pixel whose color components are (0,0,0) is displayed as black, and a pixel whose color components are (1,1,1) is displayed as white. The three color components for each pixel are stored along the third dimension of the data array. For example, the red, green, and blue color components of the pixel (10,5) are stored in RGB(10,5,1), RGB(10,5,2), and RGB(10,5,3), respectively. Wavelet family method is one of the techniques that may be used to detect the peaks. Wavelet analysis consists of decomposing a signal or an image into a hierarchical set of approximations and details. The levels in the hierarchy often correspond to those in a dyadic scale. From the signal analyst's point of view, wavelet analysis is a decomposition of the signal on a family of analyzing signals, which is usually an orthogonal function method. From an algorithmic point of view, wavelet analysis offers a harmonious compromise between decomposition and smoothing techniques. Unlike conventional techniques, wavelet decomposition produces a family of hierarchically organized decompositions. The selection of a suitable level for the hierarchy will depend on the signal and experience. Often the level is chosen based on a desired low-pass cutoff frequency. Sign Language Recognition System. Detected Finger Peaks The finger tips are detected by finding the 1s at the minimum rows. The width and height of the finger is predefined. Once the gesture has been recognized, it may be used to generate speech or text.

IV. Conclusion

A good accuracy rate was achieved with promising results from confusion matrix. Although, it can be further improved with larger datasets and better deep learning algorithms. As Real world might be not that sharp and clear.

This result was achieved by following simple steps without the need of any gloves or any specifically colored backgrounds. This work may be extended to recognizing all the characters of the standard keyboard by using two hand gestures. The recognized gesture may be used to generate speech as well as text to make the software more interactive.

Extending it to video or real time detection: We can further extend the project by recognising sign gestures from videos which will help to minimize the gap of communication with deaf and dumb people.We could make the model which would recognise the sign made by people in a video which would simplify the communication with deaf and dumb people.

Image preprocessing: We believe that the classification task could be made much simpler if there is very heavy preprocessing done on the images. This would include contrast adjustment, background subtraction and potentially cropping. A more robust approach would be to use another CNN to localize and crop the hand.

Language Model Enhancement: Building a bigram and trigram language model would allow us to handle sentences instead of individual words. Along with this comes a need for better letter segmentation and a more seamless process for retrieving images from the user at a higher rate.

This is an initiative in making the less fortunate people more independent in their life. Much is needed to be done for their upliftment and the betterment of the society as a whole.

REFERENCES

[1]. Zhang H, Wang Y, Deng C (2011) Application of gesture recognition based on simulated annealing BP neural network. In:Electronic and mechanical engineering and information technology (EMEIT), 2011 international conference, IEEE

[2]. Pansare JR, Gawande SH, Ingle M (2012) Real-time static hand gesture recognition for American sign language (ASL) in complex background. J Signal Inf Process

[3]. Rokade R, Doye D, Kokare M (2009) Hand gesture recognition by thinning method. In: Digital image processing, 2009 international conference, IEEE.

[4]. Lionnie R, Timotius IK, Setyawan I (2012) Performance comparison of several pre-processing methods in a hand gesture recognition system based on nearest neighbor for different background conditions. JICT Res Appl 6:183–194.

[5]. Rekha J, Bhattacharya J, Majumder S (2011) Hand gesture recognition for sign language: a new hybrid approach. In: Proc.conference on image processing computer vision and pattern recognition

[6]. Dardas N, Chen Q, Georganas ND, Petriu EM (2010) Hand gesture recognition using bag-of-features and multi-class support vector machine. In: Haptic audio-visual environments and games , 2010 IEEE international symposium

[7]. Sun HM (2010) Skin detection for single images using dynamic skin color modeling. Pattern Recognit 43(4):1413–1420

[8]. Sethi A, Hemanth S, Kumar K, Bhaskara Rao N, Krishnan R (2012) SignPro—an application suite for deaf and dumb. IJC - SET: 1203–1206

[9]. Dardas NH, Georganas ND (2011) Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Trans Instrum Meas 60:3592–3607

[10]. F-S, Fu C-M, Huang C-L (2003) Hand gesture recognition using a real-time tracking method and hidden Markov models. Image Vis Comput 21:745–758

[11]. Shaik KB, Ganesan P, Kalist V, Sathish BS, Jenitha JM (2015) Comparative study of skin color detection and segmentation in HSV and YCbCr color space. Procedia Comput Sci

[12]. Tsagaris A, Manitsaris S (2013) Colour space comparison for skin detection in finger gesture recognition. Int J Adv Eng Technol 6(4):1431

[13]. Qiu-yu Z, Jun-chi L, Mo-yi Z, Hong-xiang D, Lu L (2015) Hand gesture segmentation method based on YCbCr color space and K-means clustering

[14]. Rekha J, Bhattacharya J, Majumder S (2011) Shape, texture and local movement hand gesture features for indian sign language recognition. In: 3rd international conference on trends in information sciences and computing (TISC2011), IEEE, pp30–35

[15]. Grobel K, Assan M (1997) Isolated sign language recognition using hidden Markov models. In: Systems, Man, and Cybernetics, 1997. Computational cybernetics and simulation. 1997 IEEE international conference, IEEE, pp 162–167

[16]. Yang R, Sarkar S, Loeding B (2010) Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. IEEE Trans Pattern Anal Mach Intell 32:462–477