Deep Learning Approach to convert American Sign Language to Text for Deaf and Mute

Avtar Chandra[1], Shiv Prakash[2]

[1] *Department of Electronics and Communication, University of Allahabad, Prayagraj, India.*
[2] *Prof. Department of Electronics and Communication, University of Allahabad, Prayagraj, India.*

K e y w o r d s

Deep Learning
Neural Network
CNN
Feature Extraction
ReLU
ADAM optimizer
Gaussian blur filter

A B S T R A C T

Sign Language is a language within which we tend to create use of hand movements and gestures to communicate with other people who are Deaf and Mute. Using convolution neural network, we attempted to develop a real-time finger spelling technique based on "American Sign Language" (ASL). In this paper shows the sign language recognition of 26 alphabets hand gestures of American sign linguistic communication. This organized system contains various modules like pre-processing, training and testing as well, our method is providing **95.8%** accuracy for the 26 alphabets extraction, conditioning, training and testing of model and American sign to text conversion. In this project we have used Deep Learning, OpenCV and Tensor Flow to recognize face masks, we found our dataset performed better accuracy in respect to recognition.

Introduction

Gesture may be called to motion in any body part like hand and face. For recognition of gesture we are using computer vision and image processing. Gesture recognition allows system to grasp human's actions associated additionally act's as an interface in-between human and computer. This could enable genuine human interaction with computers without requiring direct physical touch with machinery. The Deaf and Mute communities sign with gestures.

When it was hard to transmit audio or when writing was challenging, this group employed sign language when vision was still an option. The only mode of inter-person communication at the time was sign language. The deaf and dumb community across the world regularly use this, although in regional variants like ISL and ASL. Hand gestures, made with either one hand or both, can be used to communicate in sign language.

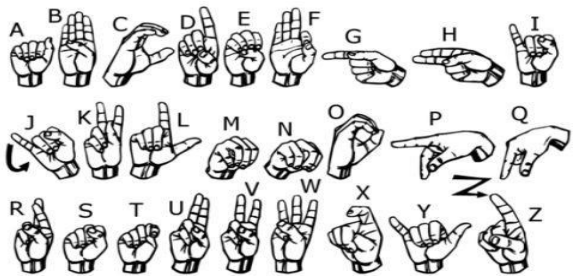
Since the single communication-related handicap Deaf and Mute persons have prevents them from using spoken languages, the only means of communication available to them is through ASL (American Sign Language), very extensively used sign language.

Objective

American Sign Language is a perfectly structured language. American Sign Language is the visual language that deaf and dumb individuals use as their first language. Unlike audio samples that pass through sound, sign language uses body language and hand-to-hand communication to dynamically convey one's thoughts. Gestures might be clearly understood by everyone if a universal link that translated sign language into text existed. In order to enable Deaf and Mute people to communicate without really understanding one another's languages, study has been done on interface system based on vision.

The aim in the project is to project a practical system that's useful for help to people with hearing impairments and uses a very simple and effective method.

Gestures that we have used to train are as specified in the image below:



In an application that is portable and real-time, we are attempting to communicate words, characters, and phrases in American Sign Language with an understanding of symbols.

The major objectives for this project are:

- To create a model that, in comparison to currently existing models, would predict symbols with the greatest accuracy and in the shortest amount of time.
- To lower the cost and create a graphical user interface (GUI) program that is simple to use and requires little upkeep in order to convert a sign to its matching text.
- To offer ideas depending on the present term in order to avoid having to translate the entire word, enhancing accuracy and speeding up sign to text conversion
- To lower the likelihood of spelling errors by proposing appropriate spellings for terms nearby in the English lexicon.

Corresponding Author:
Email Address: avi4each@gmail.com

Literature Review

There has been a lot of research and experiments carried over gesture recognition of hand in past decade.

With the aid of a literature research, we accomplished the fundamental steps in the identification of hand gestures as :

- Acquisition of Data.
- Preprocessing of Data.
- Extraction of Features.
- Gesture classification.

Acquisition of Data:

The different procedures to gain information about hand gesture may be executed in following ways:

1. Using sensor devices

Electromechanically gadgets are taken into consideration to deliver exact hand function (position) and configuration. Various glove based strategies may be used to abstract records .But it isn't user friendly and steeply-priced.

2. Vision based approach

The vision based approaches know a regular contact between computer and humans with no usage of any additional gadgets because they just need a digital camera. Computer cameras are used as input devices in vision-based tactics to examine the statistics of hands and fingers. We will talk about simulated vision system that are applied in hardware and/or software and that frequently serve to supplement natural creativity and foresight. Managing the considerable variation in the look of human hands caused by a large number of hand motions, a diversity of skin tones, as well as changing camera angles, scales, and shutter speeds, is the core challenge in vision-based hand detection.

Data-preprocessing and extracting Features for approach based on feature:

- In [1] the hand detection method combines colour detection by threshold with background removal. Because both the face and the hand have the same skin colour, we may utilise" Adaboost face finder " to tell them apart.
- We could also apply the Gaussian Blur filter in order to abstract the images needed for training. We can easily apply filters by usage of Open Computer Vision, too to be called as OpenCV and is described in [3].
- We can employ instrumented gloves, as suggested in [4], to extract the essential picture for training. In comparison to adding the filters to data extracted from the videos, this will help us minimise computation time taken for preprocessing and provide us with more succinct and precise data.
- Using a colour segmentation method, we attempted to manually segment the image, however as stated in the study paper, the segmentation results we obtained did not match since color of skin and tone are highly dependent on lightening circumstances. The resultant we achieved that are similar to each other such as the "V" sign and the gesture of "2" needs to be trained for the project, so we decided to improve it in terms of better accuracy. Instead of segmenting the hand from any background for a huge number of symbols, we retain the background for a huge number of symbols, we retain the background of hand a single colour. So we don't have to separate it based on skin tone. This would allow us to get better outcomes.

Classification based on Gesture:

- In [1] "Hidden Markov Models (HMMs)" are to be used to classify gestures. This model focuses on the dynamical aspect of gestures. Tracing skin-colored patches corresponding to hands in body and facial space centered on the face of used is used to excerpt gestures from video picture sequences. The purpose is to identify two types of motions. Images are filtered using a quick lookup index table. After filtering, skin-colored pixels are collected as droplets. A blob is a statistical object based on position [x, y] and colorimetric [Y, U, V] of a skin-colored pixel to define a uniform area.
- In [2], an efficient and fast method to recognize static hand gestures is the Naive Bayes classifier. It is based upon the classification of various movements based on geometric extracted from segmented visual data. As a result, unlike some of the other approaches, our method is not affected by skin tone. Gestures are extracted from each frame of a video with a static background. Before attempting to extract geometric invariants from an object of interest, the object must first be classified and given a name. The next stage is to categories gestures using the distance weighting and K-nearest neighbor methods (KNNDW), which will produce data suitable for a locally weighted Nave Bayes Classifier.
- According to the paper "Human Hand Gesture Recognition Using a Convolution Neural Network by Hsien I Lin, Ming-Hsiang Hsu, and Wei-Kai Chen the graduates of the Taiwan National Taipei Institute of Automation Technology" creates a skin model, a picture's hands are extracted, and then a binary threshold is applied to the entire image. The threshold picture is calibrated around the primary axis to centre the image. Using this picture, they train and forecast a convolutional neural network model. They trained the model on more than seven hand gestures and achieved an accuracy of about 95% for those seven gestures..

Hardware and Software Requirements

- Operating System: Windows
- Language used: python
- Platform used: Anaconda, Visual Studio
- Application used: Jupiter
- Library used: NumPy, MatPlotlib, OpenCV, Keras, etc.

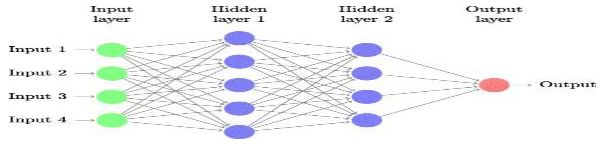
Key Words and Definitions

Extraction Feature and Representation

The 3D matrix representation of an image, where the dimensions of images are height, width, and depth, respectively (1 for Grayscale and 3 for RGB). Additionally, these values of image pixels are employed along with CNN to extract major updates.

Artificial Neural Networks

An artificial neural network is a collection of neurons connected in a way that resembles the way the human brain is organized. Information is sent from one neuron to another through every connection. Before transmitting the inputs to the buried layer of neurons, the first layer of neurons receives and processes them. The information is transmitted to the final output layer after being processed through multiple hidden levels



They can learn and must be trained. There are various strategies of learning as:

- Unsupervised Learning.
- Supervised Learning.
- Reinforcement Learning.

Methodology and Materials:

The American Sign Language to text system has been built on the concept of computer vision. Every alphabet signs or gesture are formed using their own hands that eliminates the requirement for any artificial device or equipment for engagement.

Data Set Generation

We searched for any previously built datasets for the project, but none of them had the raw photos we need. The only datasets that we could discover were in form of RGB values. We made the decision to build our own Dataset as a result. The steps we applied to construct our dataset are as follows:

- The Open Computer Vision (OpenCV) library was utilised to construct our dataset. The first step was to take roughly 200 images for testing and about 800 pictures of each ASL symbol for training purpose.
- We start by taking a snapshot of each frame that the webcam on our computer displays. A blue enclosing square designates the region of interest (ROI) for each frame.

- We extracted our RGB, Region of Interest (ROI) from this entire image and transformed it to a grayscale-image, as depicted in below figure.

- To extract several elements from the final image, we lastly employed a gaussian blur filter. This is how the image appears following the use of Gaussian blur.

Gesture Classification:

The approach implemented in project:-

Our method uses 2 layer of the algorithm in order to forecast the last gesture of the user.

Algorithm Layer "1":

1. To create processed images following feature extraction, apply the Gaussian blur filter and threshold to OpenCV captured frames.
2. When generating a term, a character is printed and considered. if it is found in more than 60 frames of this processed image, which has been given to the CNN model for prediction.
3. Spaces in-between words are to be considered by using the blank symbol.

Algorithm Layer "2":

1. We detect a variety of symbol sets that provide similar consequences when detected.
2. Then we apply specific classifier to sets to classify between them.

Layer 1:

1. First Convolutional Layer:

The resolution of the original image is 128 x 128 pixels. It was previously processed with 32 filter weights in the first convolutional layer(3x3 pixels each).The result is a 126 x 126 pixel image, one for each filter weight.

2. First Pooling Layer:

Images have been down sampled by the use of 2 x 2 max pooling. That is, itstores the largest value in the square of a 2 x 2 array. So the image is abridged to 63 x 63 pixels.

3. 2nd Convolution Layer:

The first full layer's output (63 x 63) is now fed into the second convolutional layer. It should be processed using 32 filter weights in the second convolutional layer. (3 x 3 each pixel). The result is a 60x60 pixel image.

4. Second Pooling Layer:

The resulting image is down sampled again by the usage of (2 by 2) pool and down sampled to a 30x30 image resolution.

5. First Densely Connected Layer:

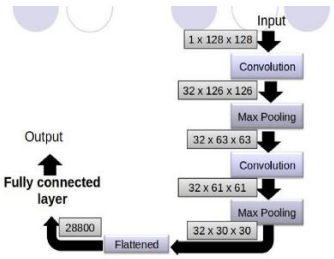
These images are fed into a layer with 128 fully connected neurons, and the output of this layer is transformed into an array. at every step of 30 x 30 x 32 = 28800 values. This layer's output is sent into a second tightly connected layer. We have used a dropout layer of 0.5 to avoid over fitting.

6. Second Densely Connected Layer:

96 neurons and result from very first densely linked layer have to be used as the intake for the completely connected layer.

7. Final layer:

The last layer, which has the same number of neurons as the number of classes categorised, uses the output of the second layer's dense connections as input. (Blank and alphabets symbols).



Activation Function:
Rectified Linear Unit, or ReLu has been engaged in each layer (fully connected neurons along with convolution).
For every pixel input, ReLu determines max[x, 0]. This increases the nonlinearity of the formula and forms it simpler to understand its excessive complicated characteristics. By cutting down on computation time, It aids in the resolution of the vanishing gradient problem and the acceleration of training.

Pooling Layer:
With the reLU activation function, we use Max pooling with a pool size of on the input image (2, 2). The number of parameters is reduced, which decreases computing costs and prevents overfitting.

Dropout Layers:
The issue of over fitting, in which the weight's of the network are so tuned to the exercise instances that the network performs poorly when given fresh examples after training. This layer "drops out" a random set of activation in the same layer by initialising them as 0. Even if certain activations are missing, the network must be capable of providing the correct classification or result for a single sample [5].

Optimizer:
As a result of the loss function's output, Adam optimizer has been used in order to update the model.Adam combines the advantages of "root mean square propagation," improvements to two stochastic gradient descent techniques (RMSProp), and the adaptive gradient algorithm (ADA GRAD).

Layer 2:
In order to get as close as we can to accurately identifying the symbol displayed, to identify and prediction symbols that are increasingly alike each other, we use two layers of algorithms. The following symbols were unclear or showed incorrectly during our testing, we are providing various symbols also:

a. For U: D & R.
b. For S: M & N.
c. For D: R & U.
d. For I: T, D, K & I.

We are making 3 different classifiers to classify this sets, in order to handgrip above occurring case's:

1. (D, U, R)
2. (I, K, T D,)
3. (M, N, S)

Implementation of Finger-spelling sentence formation:
We print the character and add it to the current line once the number of detected characters exceeds a particular value and no additional characters are near to the threshold (We recorded the value as 50 and the difference threshold as 20 in our code.).

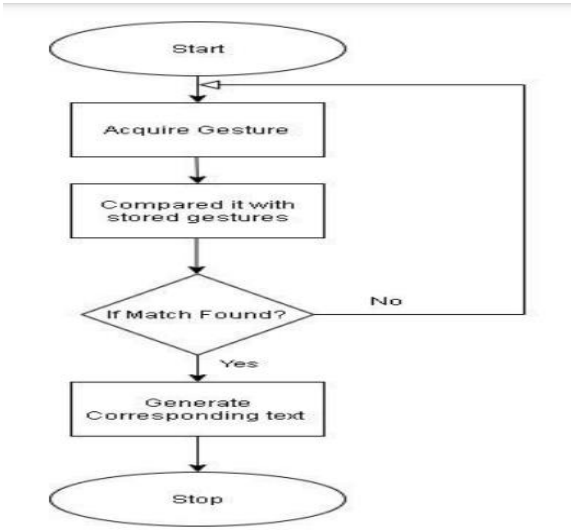
Else, clean the present vocabulary with number of occurrences of the current character to dodge the possibility of predicting an invalid character.

Whenever the total of detected blank's (normal background) surpasses a definite value, if present buffer is empty, no blanks are sensed.

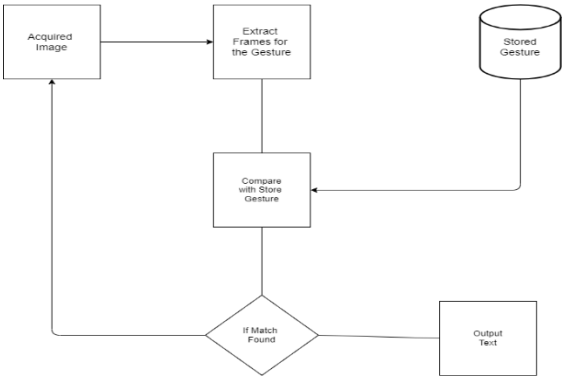
Otherwise, it will print one space to predict the finish of a word, and the current word will be added to the sentence below.

Auto correction Feature:
In Python libraries for each (incorrect) input word, "Hunspell suggests" appropriate replacements, and the user can add phrase to the current sentence by selection. From a list of terms that match the current word. It decreases spelling errors and aids in prediction for difficult words.

Flowchart:



Data Flow Diagram:



Data Flow Diagram

Training and Testing:
To get rid of extra noise, convert the 'RGB' input image to grayscale and add a Gaussian blur Filter. Resize the image to 128 by 128 and use an adaptive threshold to remove the hand from the background.

The pre-processed input photos are then given to the model for training and testing once we have completed all of the aforementioned procedures. The possibility that an image corresponds to one of the classes is estimated at the prediction level. In order for all values in each class to add up to 1, the output is standardised between 0 and 1. The softmax feature helped us do this.

Convert the RGB input image to grayscale, then add a Gaussian blur to remove additional noise. To get the hand out of the background, resize the image to 128 by 128 and apply an adaptive threshold.

Once all of the aforementioned steps have been carried out, the pre-processed input photographs are then supplied to model for the purpose of training as well as testing.

At the guess/prediction level, the likelihood that an image belongs to one of the classes is estimated. The result is standardized between 0 and 1 so that all values in each class total up to 1. This was made possible via the softmax feature.

Challenges Occurred:

We ran across various issues along the process. A lack of data was the first issue we encountered. Because operating with only square pictures was much more practical in Keras, we intended to deal with raw photographs, primarily square images. We chose to develop our own dataset for it because we were unable to find an existing one. The second challenge was deciding which filter to apply to our photographs in order to extract the necessary attributes to input into the CNN model. After testing with a variety of filters, including canny edge detection, binary threshold, and others, we finally opted for the Gaussian-blur-filter. Many other disputes have been faced related to precision of model we have taught in past phases and have been evolved by subsequent increase the size of input image and dataset.

Results:

Using only layer 1 of our algorithm, we were receiving the precision of 91.3 percent in the model, and when we combined both the layers '1' and '2', we could accomplish an accuracy of 95.8 percent. This accuracy surpasses that of the vast majority of American Sign Language research articles that have just been released. The vast majority of research articles focus on the use of kinect-like gadgets for possible hand detection.

A flemish sign language recognition system with a 2.5 percent mistake rate is created in [7] using kinect and convolutional neural networks. [8]Constructs a recognition model and obtains a defect rate of 10.90% applying hidden markov model (HMM) classifier and a dictionary of 30 word's. They accomplished an average accuracy of 86% for 41 static motions in Japanese sign language in [9]. For signers who had already been seen, Map [10] obtained an accurateness of 99.99% utilising depth sensors, and 83.61% and 85.39% for original signers.

For their recognition system, they have also utilised CNN. It should be noted that unlike some of the models listed above, our model does not use a background subtraction approach. As a result, attempting to apply background subtraction to our project may result in varying degrees of accuracy.

The bulk of the projects mentioned earlier need the use of Kinect devices, but our main objective is to design a project that can be utilised using readily accessible resources. The sensor similar to Kinect now is no longer most effective isn't effortlessly to be had however is also high priced for maximum of target market to shop for and our version makes use of ordinary webcam of the computer as a result it is beyond plus point.

Applications:

- The National Deaf Association (NAD) estimates that there are 18 million deaf people within India. Therefore, benefit to significant portion of community can be benefitted by this project of the disabled by giving them the means to use sign language to interact with the outside world. This will result in the removal of the intermediary, who mostly serves as a translator. The model's simplicity makes it suitable for implementation in mobile applications, which is what we intend to do in the future.
- Deaf human beings do now no longer available with alternatives for speaking and listening to person, and all the options have most important flaws that Interpreters are not normally existing, and additionally should be costly. Our venture as cited earlier than is pretty within your means and calls for minimum quantity of management. Hence is pretty high quality in phrases of fee decrease.
- Using pens and paper is not a good option because it takes a long time and is uncomfortable and messy for both hearing and deaf people.
- The concept of this project can also be applied to the Deaf and Mute community in a variety of settings, such as airport (in security checks or communication while boarding or in aircraft), school & college (for educational purpose), doctor offices (to properly understand the illness) and community facility organisations and jury.

Conclusion Drawn:

In this work, an efficient real time and vision based ASL alphabet recognition system for Deaf & Mute users was developed. On our dataset, we ended up with a final precision of 95.8%. We have successfully increased the accuracy of our prediction after building algorithm with two layers by predicting and confirming symbols which are quite like to each other.

The goal of this project is to create a practical system that is useful for those who struggle with hearing and who generally rely on the highly straightforward and efficient approach of sign language. This technique may be used to translate between text and sign language as well. A gesture recognition system was released for text conversion. It records the indications and displays them as in textual form on the screen.

If symbols are presented properly, and noise is avoided along with the lighting is right, gestures can nearly always recognise them in this way.

This system offers high reliability and fast response.

Scope In Future:

Future development will focus on creating a mobile application for a system that makes it possible for everyone to communicate with deaf individuals.

By utilising new approaches, we intended to expand this concept for words and sentences. The entire concept of this study was intended to be applied to smart phones as well. Implementing the image processing technologies is the difficult part of putting this concept into a mobile phone.

1. Dumb individuals must rely on an interpreter or another form of visual communication since they typically lack the ability to communicate normally with others. This effort can assist reduce reliance on the interpreter because they are no longer constantly available.
2. To fully comprehend the context and tone of the input speech, the system may be expanded to add knowledge of body language and facial expressions as well.
3. In order to increase its reach a mobile or web based approach could help well.
4. Integrating computer vision with a hand gesture detection system to create a two-way communication system.

References:

[1] T. Yang, Y. Xu, and "A. , Hidden Markov Model for Gesture Recognition", CMU-RI-TR-94 10, Robotics Institute, CarnegieMellon Univ.,Pittsburgh,PA, May 1994.

[2]Pujan Ziaie, Thomas M¨uller , Mary Ellen Foster , and Alois Knoll“A Na¨ıve Bayes Munich,Dept. of Informatics VI, Robotics and Embedded Systems,Boltzmannstr. 3, DE-85748 Garching, Germany.

[3]https://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html

[4]Mohammed Waleed Kalous, Machine recognition of Auslan signs using PowerGloves: Towards large- lexicon recognition of sign language.

[5]aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/

[6]Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using ConvolutionalNeural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham

[7]Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision based features. Pattern Recognition Letters 32(4),572–577 (2011)

[8]N. Mukai, N.Harada and Y.Chang, "Japanese Fingerspelling Recognition based on Classification Tree and Machine Learning," *2017 Nicograph International (NicoInt)*, Kyoto, Japan, 2017, pp.19-24. doi:10.1109/NICOInt.2017.9

[9]Byeongkeun Kang , Subarna Tripathi , Truong Q. Nguyen ”Real-time sign language fingerspelling recognition using convolutional neural networks from depth map” 20153rd IAPR Asian Conference on Pattern Recognition (ACPR)

[10]https://www.jetir.org

[11]https://ijsred.com

[12]https://samplius.com