2017 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)

Proposed System for Sign Language Recognition

Shashank Salian
Department of Computer Engineering
V.E.S. Institute of Technology,
Mumbai, India.
shashank.salian@ves.ac.in

Indu Dokare
Assistant professor:
Department of Computer Engineering
V.E.S. Institute of Technology,
Mumbai, India.
indu.dokare@ves.ac.in

Dhiren Serai
Department of Computer Engineering
V.E.S. Institute of Technology,
Mumbai, India.
dhiren.serai@ves.ac.in

Aditya Suresh
Department of Computer Engineering
V.E.S. Institute of Technology,
Mumbai, India.
suresh.aditya@ves.ac.in

Pranav Ganorkar
Department of Computer Engineering
V.E.S. Institute of Technology,
Mumbai, India.
pranav.ganorkar@ves.ac.in

Abstract*— The learning aids for hearing and speech disabled people exist but the usage of these aids are limited. The proposed system would be a real time system wherein live sign gestures would be processed using image processing.Then classifiers would be used to differentiate various signs and the translated output would be displaying text . Machine Learning algorithms will be used to train on the data set. The purpose of the system is to improve the existing system in this area in terms of response time and accuracy with the use of efficient algorithms, high quality data sets and better sensors. The existing systems have been able to recognize gestures with high latency as it uses only image processing. In our project we aim to develop a cognitive system which would be responsive and robust so as to be used in day to day applications by hearing and speech disabled people.

Keywords-
ML-Machine Learning
SL-Supervised learning,
NN-Neural Networks,
ASL- American Sign Language,
AUSLAN-Australian Sign Language,
CNN-Convolutional Neural Networks,
SVM- Support Vector Machine.

I. INTRODUCTION

A sign language is a language which mainly uses actions or gestures to convey meaning, as opposed to acoustically conveyed sound patterns. There are significant differences between signed and spoken languages, because of the constraints offered by visual gestures. Yet the two are fundamentally similar as both have their own syntax and semantics. Groups of hearing and speech impaired people have used signs to communicate since many years and so sign language is developed among them.

American Sign Language substantially facilitates communication in the hearing impaired community. However, there are only ~250,000-500,000 speakers which limits the number of people that they can communicate with [7].In order to diminish this obstacle and to enable better communication, we would like to propose an ASL recognition system that uses Convolutional Neural Networks to translate a user's ASL signs into text in real time.

Figure 1. Sign chart for the American Sign Language (ASL)[7]

II. RELATED WORK

ASL recognition is not a new computer vision problem. Over the past 20 years, researchers have used classifiers from a variety of categories that we can classify roughly into linear classifiers, neural networks and Bayesian networks.

A real-time sign language translator is an important milestone in facilitating communication between the deaf community and the general public.Brandon Garcia and Sigberto Alarcon Viesca[1] were able to implement a robust model for the letters a-e and a modest one for letters a-k.The Surrey and Massey datasets on ASL along with the GoogleNet architecture are used to train the system with.The system takes

978-1-5090-4324-8/17/$31.00 ©2017 IEEE

the input of the user's sign language video,classifies the actions for each letter and then tries to come up with the most accurate words. The factors such as lighting conditions and sign language border detection are considered while designing the project.They attained a validation accuracy of nearly 98% with five letters and 74% with ten letters.

Hardie Cate,Fahim Dalvi and Zeshan Hussain[2] use machine learning techniques for temporal classification, specifically the multivariate case.case. Although the results obtained are not very high, we believe that a more efficient implementation of the algorithms can yield bigger and more complex models that will perform well.This system uses the University of South wales dataset which has 95 unique signs. It suggests the use of baseline SVM for high quality data and Temporal classification techniques for lower quality data. It also considers implementing a custom LSTM model that removes the assumptions of the technique that do not apply to the data. A device called as Myo armband is also used to collect new data .They achieved an accuracy of 78.6% on a high quality dataset.

III. Approach and Methods

Convolutional Neural Networks (CNNs) are machine learning algorithms that have seen a great success as it handles a variety of tasks related to processing videos and images. Like other machine learning algorithms, CNNs seek to optimize some objective function, specifically the loss function. CNNs have seen a rapid improvement in image classification with many proposed models like GoogleNet, AlexNet giving an accuracy almost near to human perception. The main cause of the recent improvement in CNNs has been due to the ImageNet Large Scale Visual Recognition Competition(ILSVRC). For image processing we propose to use OpenCV library [9] along with TensorFlow [11] and Keras [10] which will be used for training the classifier. For other mathematical calculations we may use the NumPy Array [13] in Python [12]. The various approaches we considered are explained in the subsequent paragraphs.

A. Softmax Regression

Softmax regression is a generalization of logistic regression to the case where we want to handle multiple classes.Softmax regression allows us to handle y(i) $\epsilon\{1,\ldots,N\}$ where N is the number of classes.

$$Loss = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{e^{f_{i,y_i}}}{\sum_{j=1}^{C} e^{f_{i,j}}}\right) \quad (1)$$

$$f_j(z) = \frac{e^{z_j}}{\sum_{k=1}^{C} e^{z_k}} \quad (2)$$

N = total number of training examples
C = total number of classes

Equation (2) is the softmax function. It takes a feature vector z for a given training example, and simplifies its values to a vector of [0,1]-valued real numbers summing to 1. Equation (1) takes the mean loss for each training example, xi , to produce full softmax loss.

B. Recognizing Hand Gestures

Hand gestures can be recognized using several methods .Two of the most widely used methods are: Haar Cascades and Neural Networks. Haar Cascades were initially used for facial detection and are very easily transferrable to hand gesture recognition.

C. Neural Networks

Neural Networks are inspired by the biological arrangement of processing elements called neurons in the brain.These neurons enable parallel processing of computational tasks.This enables Neural networks to solve complex problems of pattern recognition better than procedural algorithms.CNNs are neural networks in which the response of the neuron can be calculated by a convolution operation.The initial layer of CNN can be used for matching images with respect to a fixed template.The subsequent layer can then be used for detecting variations of the identified image for improved accuracy and for generating patterns of a pattern.

D. Haar Cascade

It is a computer vision learning approach in which we train a cascade function on many positive and negative images.It can therefore be used to detect objects from other images. Initially, the algorithm is trained for facial recognition and is trained with many images.When the model is trained, features could then be extracted.Haar Cascades can be implemented with OpenCV library[9].

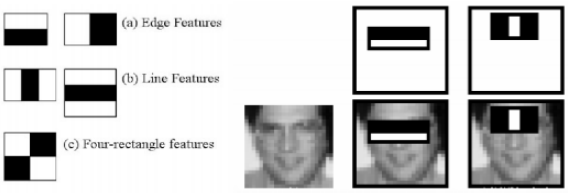

Figure 2. Features of Haar Cascade[9]

E. Deep Learning

Deep learning is used to mask the levels of abstraction in a machine learning algorithm.It contains a set of hidden layers each using the output of the previous layer for better feature extraction and pattern recognition.This is specially used for unsupervised learning of a large unclassified dataset.We have implemented deep learning using the Keras [10] deep learning library in python [12] which contains an important numerical library TensorFlow [11].

F. Keras and TensorFlow python libraries

Keras [10] is a higher level deep learning library which can be used as an interface to TensorFlow [11], which is developed by Google is used as the backend neural network modelling framework.

IV. ARCHITECTURAL DIAGRAM

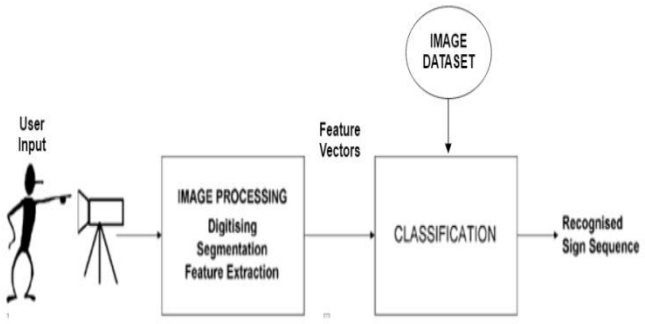

Figure 3. Architecture diagram for our proposed system

The architecture depicts that it is a real time sign language recognition system and also the methods for image processing is mentioned in the paper.The proposed image dataset is the Massey University dataset because it contains an exhaustive set of data.

V. DATASET

The various datasets we considered are as follows:

A. Australian Sign Language signs (High Quality) Data Set

The above mentioned data consists of samples of Auslan signs. In this dataset [4] there 95 Auslan signs with 27 examples each which were captured from a native Auslan signer using very good-quality position trackers.

B. Massey University Dataset

The images are all png in RGB mode, with the hands segmented by colour and with black background (0,0,0).

C. University of Texas Arlington Dataset:

Video dataset [5] containing short sequences of hand gestures , it consists of multiple frames.

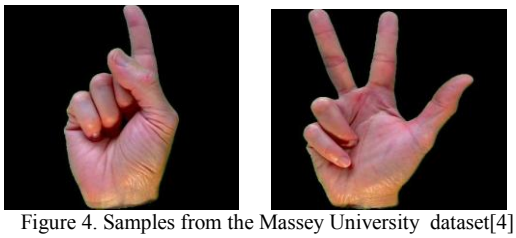

Figure 4. Samples from the Massey University dataset[4]

The Massey University Gesture Dataset [3] 2012 contains 2,524 close-up, color images that are cropped such that the hands touch all four edges of the frame. The hands have all been tightly cropped with little to approximately no negative space and it is placed over a uniform black background.
We recommend using Massey University Dataset[4] because it contains an exhaustive set of data in the form of captured images of different hand gestures. All hand gestures are a part of the American sign language.

VI. IMPLEMENTATION

A. Dataset Preprocessing

The starting point for implementation will be first preprocessing the dataset in the format accepted by the CNN models. Our current dataset does not have a aspect ratio of 1:1 and hence have random width and height. We will be first resizing the images in the dataset to a size of 256x256 so as to attain an aspect ratio of 1:1 which is the accepted image format for the input layer of the CNN. To resize the images in the dataset, we will be using the imagemagick tool, an open source image manipulation library. Currently, Our dataset also contains the images just for the right hand. To make our application compatible with the left hand too, we will be flipping the existing images horizontally.

B. Model of the CNN

The proposed system will be using a CNN model which will consist of an input layer, followed by a set of two convolutional layers each backed up by a max pooling layer and further followed by two fully connected layers. For varying the learning rate of the CNN during the training of the model, we will be using the Stochastic Gradient Descent (SGD) optimizer. SGD reduces cost of training the dataset and it leads to fast convergence. For calculating the final probabilities of the recognized classes, we will be using the softmax regression in the last Fully Connected layer of our CNN. We will use the Sequential API provided by Keras [10] to build the above foresaid model. In the Sequential API, we will take a bottom up approach and start from the Input Layer. The further layers in our CNN model will be stacked one upon the other until we reach the last layer in our model which is the Fully Connected layer. A Diagrammatic Representation of our CNN Model is given in Figure 5 and Figure 6.

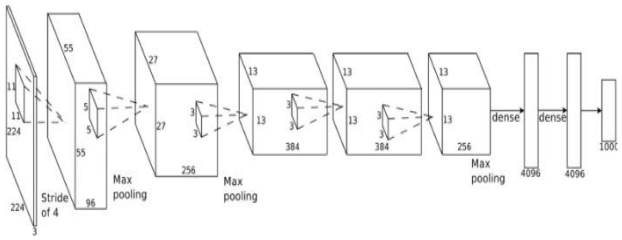

Figure 5. Perspective view of the CNN model[8]

C. Training the CNN

For training the CNN, we will start with a learning rate of 0.01 and a decay rate of 1e-6 using the SGD Optimizer. Initially, we aim to train the CNN for 10 classes of signs in a

batch size of 15 for a total of 50 iterations. In every iteration of the training, the training data will be shuffled so that there will be sufficient randomization during the training process.
We also plan to use a validation split of 20 percent during the training process so that the last 20 percent training data of each class will be used by the CNN for validation. Eventually, we will train the CNN on the whole ASL dataset which consists of 36 signs with a total of 2515 images. To speed up training in that case, we aim to use a batch size of 50 and train for approximately 200 iterations so that we are able to reach a validation accuracy of nearly 98 percent. To achieve a greater validation accuracy, we will be increasing the validation split by an increment of 5 percent and compare it with previously obtained results. Batch size and number of neurons in the fully connected layer of CNN will be incremented by 5 and 25 respectively in each successive trial.As training the CNN model for 36 signs requires extensive use of GPU processing, we will be deploying the training on the g2.2xlarge or p2.xlarge GPU instances provided by Amazon Web Services EC2 Infrastructure.

D. Image Processing

The final step will be to process the images from the webcam in real time and extract the hand image using the image extraction API provided by the OpenCV [9]. To get the image of hand, we will employ various techniques such as Background Subtraction, Convex Hull detection and counting the defects in the resulting Convex Hull. Before giving the real time image to the trained CNN model for classification, we will be performing the same preprocessing steps on it just as we do it for the training data. We will be displaying three windows to the user which will enable the user to position his hand properly so that a proper image of the hand is taken. One of them will also display the largest recognized contour in the real time image.

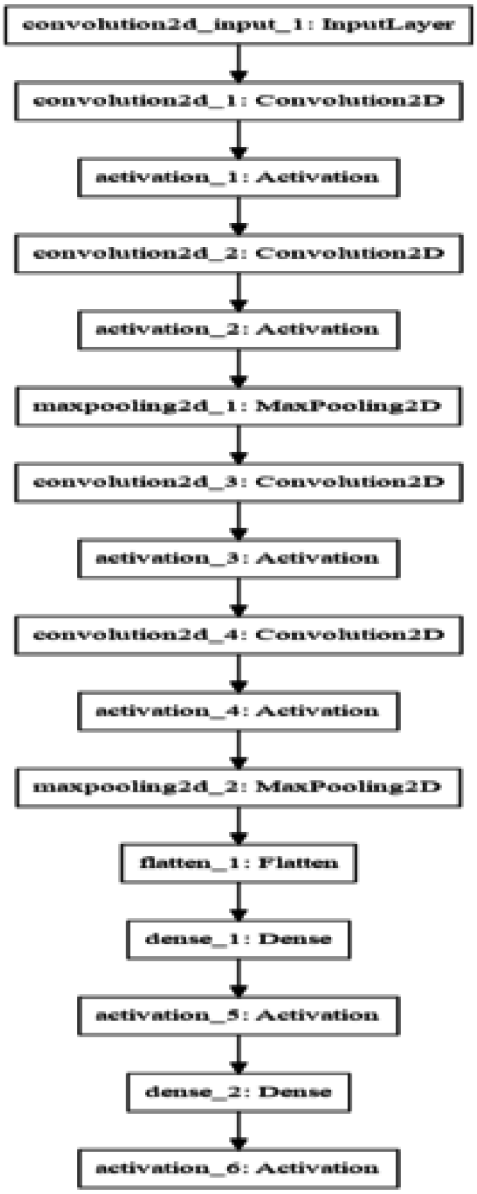

Figure 6. Sequential API model of the CNN

E. Transfer Learning

A further step in our project will be to fine tune already established models like GoogleNet, AlexNet, VGGNet, CaffeNet which have been the previous winners in the ILSVRC challenge. They have been trained on the 2012 ILSVRC dataset. As our dataset is markedly different from ImageNet data, we will vary the model weights and learning rate in the internal layers of these models and modify the output layer to match the input size of our training data.

F. Testing and Evaluation

We will be doing a comparison of the above mentioned models based on the accuracy they give on the real time images. For evaluating the models, we use two metrics which are very popular in previous literature related to CNN. The first metric is Top-1 Val accuracy which gives the percentage of correctly classified labels in which the intended label appears at the 1st position in the predictions done by the CNN model. The second metric is the Top-5 Val accuracy which the

gives the percentage of classifications where the intended label is in the 5 classes with the highest probabilities.

VII. Conclusion

We are going to implement sign language recognition using concepts of machine learning and image processing.After our application is implemented it would benefit the NGOs and various organizations which involve people with special needs as it will be a real-time system for hand gesture identification. The proposed system can also be further carried forward to implement Indian Sign Language recognition which is much more complex as it involves two hand gestures. Also the response time of the system can be reduced with better camera and graphics support.

Acknowledgement

The project team would like to express their heartfelt gratitude to our alma mater V.E.S. Institute of Technology for giving us the opportunity and our project mentor Prof. Indu Dokare for her guidance, encouragement and inputs which led us to undertake this project.

References

[1] Brandon Garcia, Sigberto Alarcon Viesca "Real time American Sign Language Recognition with Convolutional Neural Networks"
[2] Hardie Cate, Fahim Dalvi, Zeshan Hussain December 11, 2015 "Sign Language Recognition using Temporal Classification"
[3] Massey University ASL Gesture Dataset 2012
[4] UCI Machine Learning Repository: Australian Sign Language signs (High Quality) Data Set
[5] University of Texas Austin: ASL_2006_10_10-Lossless compressed videos
[6] A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures"-A.L.C. Barczak, N.H. Reyes, M. Abastillas, A. Piccio and T. Susnja.
[7] Lifeprint.com. American Sign Language (ASL) Manual Alphabet (fingerspelling) 2007.
[8] A Medium Corporation - https://medium.com/@ageitgey
[9] Bradski, G., OpenCV Library, (2000), GitHub repository, https://github.com/opencv/opencv
[10] François Chollet, Keras, (2013), GitHub repository, https://github.com/fchollet/keras
[11] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
[12] Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org
[13] Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37