Sign Language Detection using Deep Learning

Smit Patel [(1)], Tanushree Pardhi [(2)], Zankhana Shah [(3)]

[(1)] Student, BVM Engineering College
, smitpatel0045@gmail.com

[(2)] Student, BVM Engineering College
, tanushree.pardhi@gmail.com

[(3)] Professor, BVM Engineering College
, zankhana.shah@bvmengineering.ac.in

Abstract: In this paper we put forward a system that bridges the gap between people who use Sign Language to communicate and the computer. Sign Language Recognition systems, though much crucial, suffer the lack of implementation in our common day to day devices. This paper focuses on various techniques and tools to help resolve this gap using Deep Learning. Here we propose a system that recognizes sign language and predicts the right sign using a web camera. The system uses Deep learning techniques, Convolution neural networks, max pooling and ReLU activation function. We aim to create a software which is both affordable, much more accessible to the users and works without compromising with the desired results.

Keywords: Sign Language Recognition; Convolution Neural Networks; Deep Learning; Image Classification.

INTRODUCTION

Growth of any kind means nothing if it is not inclusive, 466 million people across the world (over 5% of the world's population) suffer from significant hearing loss, even with the evolution of technology, the deaf-mute community faces many challenges on a regular basis. Today there is an advent of devices using speech recognition but nearly none of them have the capability of sign language recognition. Which is not the only challenge, the research is limited to only a particular country as just like any other language the sign language also has various dialects. Taking into consideration all the aspects of this problem we put forward here a system which makes use of deep learning to recognize and predict the sign language through the use of web camera.

Further challenge posed was that the complex background and illumination conditions affect the hand tracking and make the Sign language recognition very difficult but there exists a technology called Microsoft Kinect, it enables us to hand and body movements better because in addition to color it also provides depth data. This gives us a distinct 3D motion trajectory of each signing in the sign language vocabulary which leads to far better results[1]. But the main disadvantage of this approach is that it requires a specialized device called Kinect to capture the gestures of the user, which makes it outside the reach of many users.

One of the solutions is using Deep Learning, which we demonstrate in this paper. By using Neural Networks, Max pooling and activation function such as ReLU we will be able to create a solution that proves to be more economical than using a device like Kinect, and will also be more accessible to the users while maintaining the desired results.

LITERATURE STUDY Sign Language Recognition using 3D Convolution Neural Networks

Creating descriptors for hand shapes and hand motion trajectory is a prime difficulty in sign language recognition. Hand motion trajectory also pertains to curve matching and key point tracking. There has been a lot of studies on these topics yet it has been significantly difficult to obtain good results.

These issues are addressed by developing a 3D convolution neural network to naturally integrate the shape of hand, facial expression and the trajectory of the hand. Inputs such as color, depth and body skeleton images taken from Microsoft Kinect are used rather than only using color images as it is done commonly.[5] This leads to the network to consider the changes depth and trajectory in addition to color. Since convolution neural nets have the ability to learn the features from raw data without providing it with any previous knowledge it negates the need for segmenting hands from the background or tracking of hand movements. What convolutional networks bring to the table amidst the already existing models is that it captures the motion information from the raw video data while the other preexisting models need manually set features to do the classification.[5]

Let's look at the stages involved while approaching Sign Language Recognition: deciding the regions of interest, extracting features describing these regions, and with these features training a classifier.[5] Inspired from the biological neural networks, the deep learning model the CNNs fulfill the 3 stages with just a single

network trained on raw pixel to classifier outputs. CNNs have shown some significant progress in object detection and recognition, natural language processing, scene labelling and segmentation tasks.

The model proposed in this paper[5] performs 3D convolutions to learn about the spatial features and the temporal features. The deep architecture created pulls information of different kind from the adjacent layers and then separately runs the convolution and subsampling. Next the for the final feature representation the information from every channel is gathered and combined. Finally by using the multilayer perceptron classifier the feature representations are classified. The effectiveness of the method is shown by the results obtained when 3D CNN and GMM-HMM are trained on the same dataset.[5]

A Review of Hand Gesture and Sign Language Recognition Techniques

Sign language recognition consists of methods like recognizing hand motion trajectory for different signs and segmentation of hands from the background to predict and string them into sentences which are both semantically right and meaningful. Furthermore, the challenges faced in gesture recognition involve motion modelling, motion analysis, pattern recognition and machine learning. SLR has models using handcrafted parameters or parameters which are not manually set. The background and the environment plays a big role such as the lighting in the room and the speed of the gestures affects the model's ability to perform the classification. The gesture appears different in 2D space due to differences in viewpoints.

To recognize the gestures there are a couple of approaches such as sensor-based approach and vision-based approach. In the sensor-based approach capturing of the various parameters such as trajectory, position and velocity of the hand is done by devices equipped with sensors. On the other hand vision-based approaches involve using images of video footages of the hand gestures.[6]

For the representation of gestures model-based or appearance-based methods are used where model-based approach the 2D or 3D space is used for the description of the hand while the latter derives it directly using the template database from the video footages and images. Following are some of the stages while approaching sign language recognition.[6]

Data Acquisition: Sign language recognition based on the visuals is data such as frames of images.

Image pre-processing: It is basically performed to modify the input to improve the overall performance.

Segmentation: This process includes the process of partitioning images into multiple distinct parts. In this stage the Region of interest is separated from the remainder of the image.

Feature Extraction: Feature extraction deals with the transformation of interesting parts of the input data into sets of compact feature vectors.

Classification: It can be categorized into supervised and unsupervised machine learning techniques. Supervised machine learning is a technique used to teach a system to recognize patterns in the input data, which are further used in prediction of future data. Supervised machine learning takes in a set of known training data and it is used to infer a function from labelled training data.[6]

DESIGN AND IMPLEMENTATION

Data Set:

For this project, dataset of American Sign Language created by Akash is used. It is a collection of over 87,000 images, around 3000 for each of its classes. This Dataset contains a total of 29 classes, it has two categories, 26 classes are for alphabets A-Z and 3 classes are for SPACE, NOTHING and DELETE which prove quite useful when dealing with a real time application.

Figure 1. Alphabets A-Z

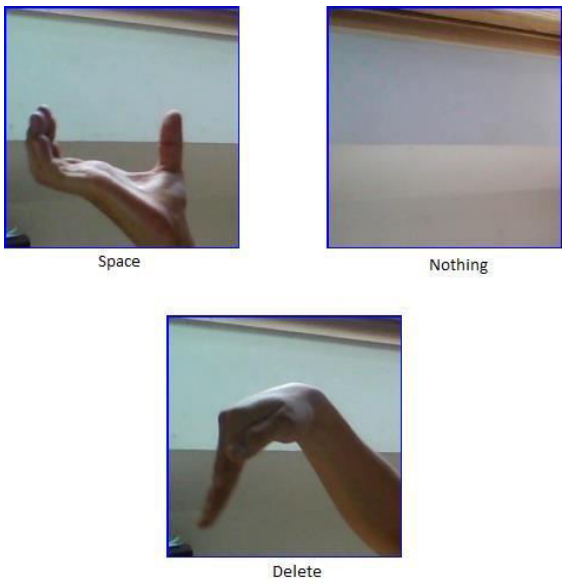

Figure 2. Other Classes

Algorithms Used:

For supervised learning, i.e. learning with a labelled dataset, CNN (Convolution Neural Network) algorithm is used.

- Convolution Neural Network:

  Convolution Neural Networks, is a deep learning approach inspired by the biological neural networks, they take images or videos as input,0 assign some weights and biases to the extracted features of an image and basically learn to classify them. As opposed to primitive methods where we have to manually engineer the filters/characteristics, CNN are capable of learning them with enough training.

  The ConvNets are a multilayer artificial neural network developed to process 2D or 3D data which is given as input. Every layer in the network is made up of multiple planes which can be 2D or 3D, and each plane consists of multiple independent neurons composition, here the adjacent layer neurons are connected however the same layer neurons aren't.[10]

  A ConvNet has the capability of capturing the Spatial and Temporal features of image by applying relevant filters. Also, reductions of parameters involved and reusability of weights results in the architecture performing better fitting to the image dataset. The main objective of ConvNet is to make processing of an image easier by extracting useful features from an image, without losing the critical features crucial for making accurate predictions. This is of great value when designing an architecture which is not only competent at capturing and learning the features but also can handle huge amounts of data.

- ReLU:

  It is a nonlinear activation function. ReLu really surpasses the earlier methods in decreasing the time consumed by deep learning models for training.

- Pooling:

  The Pooling layer reduces the Spatial size of the convoluted feature, by reducing dimensionality thus the computational power required is reduced. It is also useful for extracting the dominant features in an image which are rotational and positional invariant, which maintains the process of effective training of the model.

  Max Pooling is a process that is sample-based discretization. It mainly downsamples the input representation, making reductions in its dimensionality and leaves space for assumptions about features that are contained in the sub-region binned.

  The convolution layer features are extracted and given as input to the classifier for training which leads to the final classification and then the result can be output. For larger sized high-res images the computational overhead, when the features extracted from the convolution layers is directly provided as

input to the classifier is way too large.

The high-dimensional features given as input to the classifier require a huge computational resource. This also brings up the issue of overfitting. Although due to the factor of the image having a "static attribute", there is a probability that the feature extracted in a local region of the image is applicable equally in some other local area. Which makes it feasible to perform aggregate statistical operations on distinct location characteristics in a local area of an image, and this is known as pooling.[10]

- Fully Connected Layer (FC Layer):

Here every neuron of the former layer is connected to neuron of the next layer, it is simply a feed forward network.

- Soft-max:

The Sigmoid function worked for two values soft-max is basically the extension of this methods which works for more than just 2 values. In various non-linear probabilistic models soft-max function is of significance.[12] This function essentially converts a vector of K real values into a vector of K real values which add to a sum of 1. The numbers provided as input may differ, they can be anything positive, negative, zero, or greater than 1, but the SoftMax function converts these values and makes them fall between 0 and 1, this enables them to be identified as probabilities. The input if small or negative is turned into small probability and if large then it is turned into a large probability. Whatever maybe the case the values will always fall between 0 and 1.

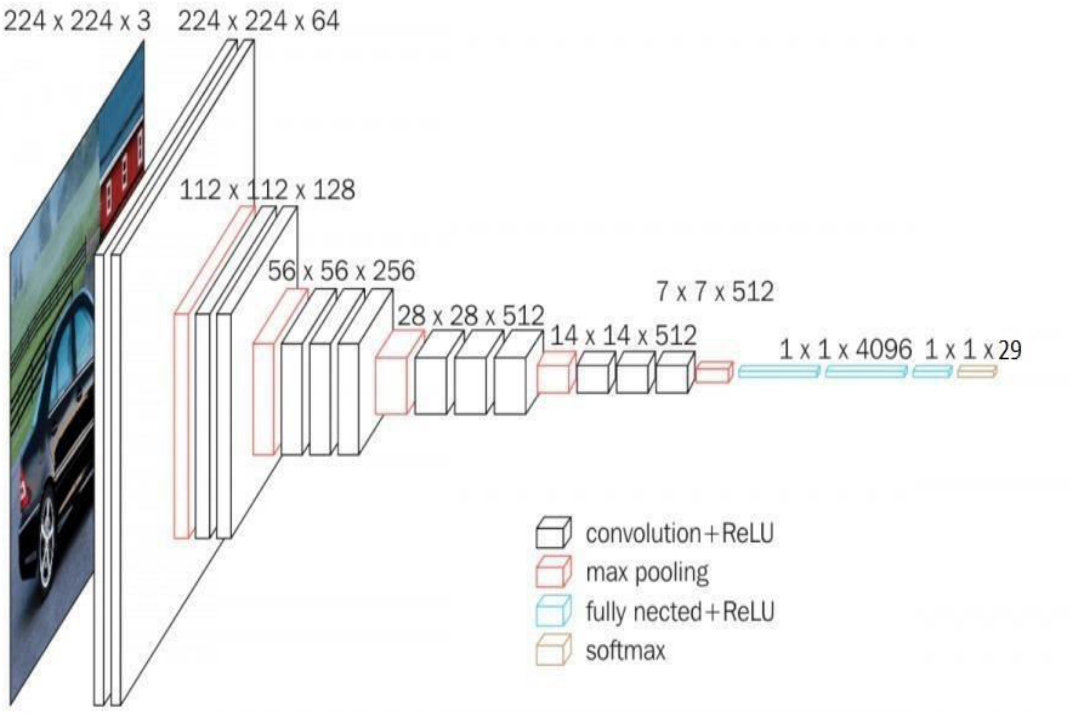

Figure 3. Layers in the model **Modules:**

- Machine Learning Module: Trained neural network architecture to interpret the signs in the given test image. Provide dataset to the model and train it for guessing the right sign.
- User Interface Module: An interface that makes it easier to use the machine learning module.

Tools used:

- TensorFlow:

TensorFlow library is a symbolic math library mainly used for dataflow and differentiable programming across a range of task and also for machine learning applications such as neural networks. The main advantage this library provides is that its open source and free to use.[3]

- Keras:

It is an open-source neural-network, capable of running on top of TensorFlow, Theano, or PlaidML. It makes the implementation of deep learning fast and easy for research and development. It uses four

guiding principles Modularity, Minimalism, Extensibility and everything is in native python.[4]

- OpenCV:

  OpenCV is an open sourced highly optimized library in python aimed at solving computer vision problems, mainly focused on the real time application providing computational efficiency for handling huge amounts of data. [11] It helps process images and videos to help identify objects, faces or even handwriting of a human. When integrated with NumPy it becomes immensely useful, because whatever operations that can be done in NumPy can be combined with OpenCV.

- Tkinter:

  Python provides various options for building Graphical User Interfaces (GUI), we used Tkinter in this particular project. It is a standard library in python used for building GUIs and is fast and easy to use. It provides a powerful object-oriented interface to make for better user experience.

- PIL (Python Imaging Library):

  Python Imaging Library (also known as Pillow in the newer versions) is another free open source library, that supports multiple file formats and supports for opening, manipulating and saving them. It provides powerful image processing and graphics capabilities.

- NumPy:

  NumPy builds on (and is a successor to) the successful Numeric array object. NumPy short for Numerical Python is a free open source python library. It is mainly used for manipulation of arrays and also for working in linear algebra, Fourier transform, matrices etc.[2]

CONCLUSION: MODEL ANALYSIS

The model was trained using the technique of Transferred learning to reduce the training time. The pretrained model used was VGG16.

- Transferred learning:

  Transfer learning is a methodology that uses knowledge obtained while working through one problem to solve the next problem. For example, knowledge gained while learning to recognize one breed of dog could be applied when trying to recognize other breeds. This research study is much inspired from the long history of psychological literature on transfer of learning even though the relation between the two areas of study is very limited. From the practical point of view, reusing or transferring information from previously learned tasks for the learning of new tasks has the potential to provide notable improvements in the sample efficiency of a reinforcement learning agent.[8]

- VGG16: This convolutional neural network model is put forward in their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" by K. Simonyan and A. Zisserman from the University of Oxford.[9] The model accomplishes 92.7% top-5 test accuracy in the ImageNet dataset. This is a dataset which consists of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014.[10] It made developments over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU's.[10]

  The base model we used for transferred learning was VGG16 Model with the ImageNet weights with top included, the last layer was modified to fit the output of 29 characters. The last layer was a dense layer with SoftMax activation function with 29 outputs. Only the last layer was made trainable, and hence from the Total parameters of 134,379,357 only 118,813 parameters were trainable. This significantly cuts down the training time for the model. The model was trained with different quantity of images and their accuracy was measured with 500 new unknown images per character to the model. The following table shows the result of the tests done based on the 14,500 images shown to the model.

Table 1. Analysis

Images used to train model	50	100	200	500	1000	2100
Accuracy(%)	45.92	54.83	62.06	72.92	92.87	95.33

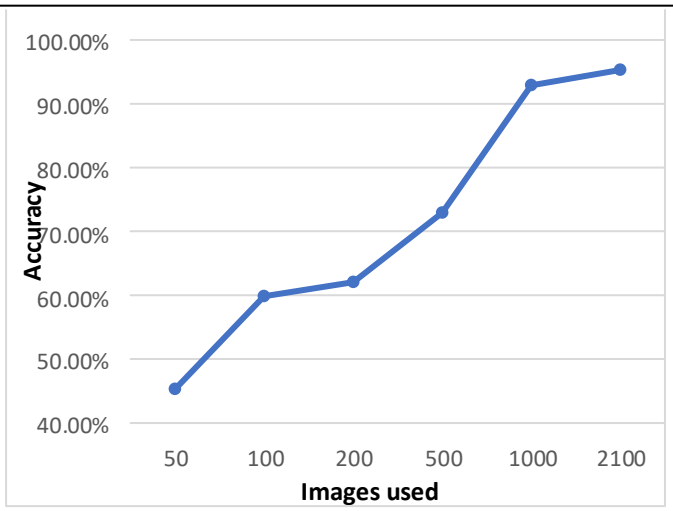

Figure 4. Accuracy Measured on 500 Images per Character

FUTURE WORKS Expanding Dataset

To expand the dataset in such a way that includes images with background noise to improve performance in real life scenarios.

User Trained Model

To make the model User-dependent, the model will be trained on the images provided by the user. In this way the model will become more personalized for the user and provide more accuracy.

Integrating it with a Voice assistance

Voice is the future of all computing interfaces, which poses a great challenge for individuals suffering from speech and hearing impairments. To demonstrate a real-life application of the system we aim to create an interface for communicating with voice assistants (Siri, Google assistant, Alexa, etc) with Sign Language. This interface shall convert the sign language predicted by the system into voice for sending voice commands to the Assistant. Thus, enabling people who use sign language to communicate with a voice assistant.

References

Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., & Zhou, M. (2013, April). Sign language recognition and translation with kinect. In *IEEE Conf. on AFGR* (Vol. 655, p. 4).

Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1, p. 85). USA: Trelgol Publishing.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265-283).

Gulli, A., & Pal, S. (2017). *Deep learning with Keras.* Packt Publishing Ltd.

Huang, J., Zhou, W., Li, H.,& Li, W. (2015, June). Sign language recognition using 3d convolutional neural networks. In *2015 IEEE international conference on multimedia and expo (ICME)* (pp. 1-6). IEEE.

Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics, 10*(1), 131-153.

George Karimpanal, T., & Bouffanais, R. (2019). Self-organizing maps for storage and transfer of knowledge in reinforcement learning. *Adaptive Behavior, 27*(2), 111-126.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*

Zhang, X., Zou, J., He, K., & Sun, J. (2015). Accelerating very deep convolutional networks for classification and detection. *IEEE transactions on pattern analysis and machine intelligence, 38*(10), 1943-1955.

Al-Saffar, A. A. M., Tao, H., & Talab, M. A. (2017, October). Review of deep convolution neural network in image classification. In 2017 *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)* (pp. 26-31). IEEE.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.".

Bouchard, G. (2007). Efficient bounds for the softmax function, applications to inference in hybrid models. In *Presentation at the Workshop for Approximate Bayesian Inference in Continuous/Hybrid Systems at NIPS-07.*