**SIGN LANGUAGE INTERPRETER USING IMAGE CLASSIFICATION**

**Project report for the award of the degree of**

**Bachelor of Technology**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted By**

**NABANITA DASGUPTA** Department of Computer Science & Engineering

University Roll No. **12016009001078** UEM, Kolkata

**TRISHA DHAR** Department of Computer Science & Engineering

University Roll No. **12016009001068** UEM, Kolkata

**KRISHAN KANT SINGH** Department of Computer Science & Engineering

University Roll No. **12016009001610** UEM, Kolkata

**PRITI GUPTA**  Department of Computer Science & Engineering

University Roll No. **12016009001085** UEM, Kolkata

**KRITTIKA BEPARI** Department of Computer Science & Engineering

University Roll No. **12016009001088** UEM, Kolkata

**DEBAROTI CHOWDHURY** Department of Computer Science & Engineering

University Roll No. **12016009001619** UEM, Kolkata

**Under the guidance of**

PROF. **SANKHADEEP CHATTERJEE** and PROF. **SOMARPITA DUTTA**

Department of Computer Science & Engineering



UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.

**CERTIFICATE**

This is to certify that the project titled **Sign Language Interpreter Using Image Classification** submitted by **Nabanita Dasgupta (University Roll No. 12016009001078), Trisha Dhar (University Roll No. 12016009001068)**, **Krishan Kant Singh (University Roll No. 12016009001610), Priti Gupta (University Roll No. 12016009001085), Krittika Bepari (University Roll No. 12016009001088)** and **Debaroti Chowdhury (University Roll No. 12016009001619)**, Students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in partial fulfilment of requirement for the degree of Bachelor of Computer Science & Engineering is a bona fide work carried out by them under the supervision and guidance of Prof. **Sankhadeep Chatterjee** and Prof. **Somarpita Dutta** during 8th Semester of academic session of 2019-2020. The content of this report has not been submitted to any other university or institute for the award of any other degree.

I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Sankhadeep Chatterjee Prof. Somarpita Dutta

Assistant Professor Assistant Professor

Department of CSE Department of CSE

UEM, Kolkata UEM, Kolkata

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Sukalyan Goswami

Head of the Department

Department of CSE

UEM, Kolkata

**ACKNOWLEDGEMENT**

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout this project remains invaluable to us.

We are sincerely grateful to our guide Prof. **Sankhadeep Chatterjee** and Prof. **Somarpita Dutta** of the Department of Computer Science & Engineering, UEM, Kolkata, for their wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. Sukalyan Goswami, HOD, Computer Science & engineering, UEM, Kolkata and all other departmental faculties for their ever-present assistant and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

**Nabanita Dasgupta**  [Enrolment No: **12016009001078]**

**Trisha Dhar** [Enrolment No: **12016009001068]**

**Krishan Kant Singh** [Enrolment No: **12016009001610]**

**Priti Gupta** [Enrolment No: **12016009001085]**

**Krittika Bepari** [Enrolment No: **12016009001088]**

**Debaroti Chowdhury** [Enrolment No: **12016009001619]**

**TABLE OF CONTENTS:**

**Topic Page No**

1. Abstract …………………………………………………….5

2. Introduction…………………………………………………….6

3. Literature Survey……………………………………………...7-8

4. Problem Statement & Discussion …………………………...9-10

5. Proposed solution …………………………………………..11-14

6. Result Analysis …….……………….....................................15-16

7. Conclusion………………………………………………..…….17

8. Future Work …………………………………………..……….18

9. Bibliography…………………………………………………19-20

**1. ABSTRACT:**

Sign language interpreting is the use of a sign language to convey the information contained in the programmed audio(speech and other important sounds) to viewers who are deaf and for whom sign language is their first language. In line with the recent advances in this field of deep learning, there are far reaching implications and applications that neural networks can have for sign language interpretation. So hereby we propose the main aim of this project work -to develop an application which will translate sign language to English in the form of text , thus aiding the process of communication via sign languages. This application acquires the hand gestures as image data and interprets these gestures into readable text. It is evident from the experimental results that the gestures are recognized measuring the positions and the orientations of the fingers with the application of CNN algorithm. Then the proposed model is validated on the test dataset to achieve excellent results in interpreting the signs into letters by classifying them.

**2. INTRODUCTION:**

Human beings interact with each other either using a natural language channel such as words, writing, or by body language (gestures) e.g. hand gestures, head gestures, facial expression, lip motion and so on. As understanding natural language is important, understanding sign language is also very important. The sign language is the basic communication method within hearing disabled people. People with hearing disabilities face problems in communicating with other hearing people without a translator. For this reason, the implementation of a system that recognize the sign language would have a significant benefit impact on deaf people social live.

Sign Language is a unique type of communication that often goes understudied. While the translation process between signs and a spoken or written language is formally called ‘interpretation,’ the function that interpreting plays is the same as that of translation for a spoken language. In our research, we look at American Sign Language (ASL), which is used in the USA and in English-speaking Canada and has many different dialects. There are 22 hand shapes that correspond to the 26 letters of the alphabet, and you can sign the 10 digits on one hand.

One of the nuances in sign language is how often fingerspelling is used. Fingerspelling is a method of spelling words using only hand gestures. One of the reasons the fingerspelling alphabet plays such a vital role in sign language is that signers used it to spell out names of anything for which there is not a sign. People's names, places, titles, brands, new foods, and uncommon animals or plants all fall broadly under this category, and this list is by no means exhaustive. Due to this reason, the recognition process for each individual letter plays quite a crucial role in its interpretation.

**3. LITERATURE SURVEY:**

American Sign Language (ASL) recognition is not a new computer vision problem. Over the past two decades, researchers have used classifiers from a variety of categories that we can group roughly into linear classifiers, neural networks and Bayesian networks.

In 2009, Eleni Efthimiou, Stavroula-Evita Fotinea, Christian Vogler, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet , Petros Maragos, Jérémie Segouat [5] proposed Dicta-Sign, a project aimed at developing the technologies required for making sign language-based Web contributions possible, by providing an integrated framework for sign language recognition, animation, and language modelling. In 2011, Helen Cooper, Brian Holt and Richard Bowden [6] propounded the task of continuous sign recognition, the work towards true signer independence, how to effectively combine the different modalities of sign. In 2011, Lorena P. Vargas, Leiner Barba, C O Torres and L Mattos [7] presented an image pattern recognition system using neural network for the identification of sign language to deaf people. In 2013, Joyeeta Singha and Karen Das [8] provided a system using Eigen value, weighted Euclidean distance as a classification technique for recognition of various sign languags of India. In 2014, Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, Benjamin Schrauwen [9] contributed a recognition system using Microsoft Kinect, convolutional neural networks (CNN) and GPU acceleration.

In 2014, Maher Jebali, Patrice Dalle and Mohamed Jemni [10] brought the Human Computation Interaction (HCI) performance nearby the human-human interaction, by modelling a sign language recognition system on prediction in the context of dialogue between the system and the interlocutor, to make a ludic application. They also included an empirical tracking method which dynamically changed according to each stage of dialogue. In 2015, Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen [11] propounded an automatic fingerspelling recognition system using CNNs from death maps. In 2015, Jie Huang, Wengang Zhou, Houqiang Li and Weiping Li [12] proposed a novel 3D convolutional neural network (CNN) which extracts discriminative spatial temporal features from raw video stream automatically without any prior knowledge, avoiding designing features. In 2016, Brandon Garcia and Sigberto Alarcon Viesca [13] presented the development and implementation of an American Sign Language (ASL) fingerspelling translator based on a CNN. They also produced a robust model that consistently classifies letters a-e correctly with first time users and another that classifies letters a-k in a majority of classes. In 2016, Ashish S. Nikam, Aarti G. Ambekar [14] introduced a prototype that is able to automatically recognize sign language. They discussed recognition of hand gestures on the basis of orientation, centre of mass, centroid, finger status and thumb position.

In 2017, Vivek Bheda and N. Dianna Radpour [15] proposed a method for using deep convolutional network to classify images of both the letters and digits in American Sign Language. In 2017, Beena M.V. and Dr. M.N. Agnisarman Namboodiri [16] focussed on recognition of static gestures of ASL which are collected from Kinect Sensor. They also designed a java GUI application to test the classifier. In 2019, Al Amin Hosain, Panneer Selvam Santhalingam, Parth Pathak, Jana Kosecka, Huzefa Rangwala [17] developed and evaluated a sign recognition system using multiple modalities that can be used by DHH signers to interact with voice controlled devices. They investigated the feasibility of using skeletal and RGB video data for sign language recognition using a combination of different deep learning architectures. In 2019, Shivashankara S and Srinath S [18] focussed on translating video based hand gestures of ASL into human and/or machine readable English text using deep neural networks. They used Gaussian Mixture model for background elimination and foreground detection. In 2019, Kshitij Bantupalli and Ying Xie [19] created a vision based application that offered sign language translation. They used CNN for recognising spatial feature and RNN for training on temporal features.

The above-mentioned extensive literatures claimed that over the last decade there has been growing interest in the interpretation of sign language. Therefore the current work added a contribution by using convolutional neural network to interpret the American Sign Language (ASL).

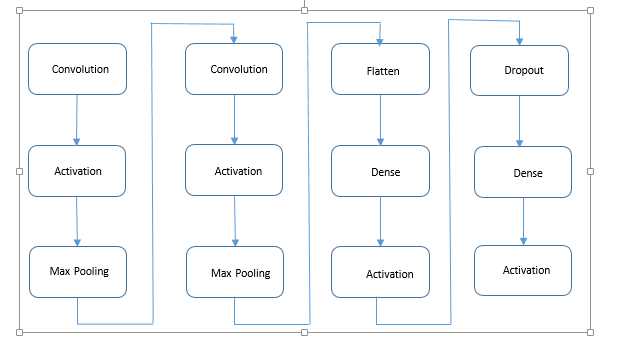
**4. PROBLEM STATEMENT AND DISCUSSION:**

Sign Language Recognition (SLR) targets on interpreting the sign language into text or speech, so as to facilitate the communication between deaf-mute people and ordinary people. This task has broad social impact, but is still very challenging due o the complexity and large variations in hand accuracies.

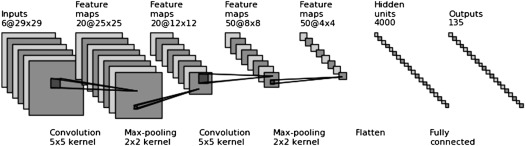
We train a deep convolutional neural network on the MNIST hand sign dataset, we mined from the kaggle repository, and get an average accuracy of 99.134%.The training and testing dataset consists of pixels of images, each row consisting of a particular letter. We train the network and then validate them accordingly to get desired results. The data has been split into features and labels. Once we have trained the model, we will give it sets of new input containing the features, it will return the predicted "label" (alphabet) for those pixels. We use the Sequential API, which allows us to stack recurrent layers of the network from input to output. Next we add a 2Dconvolutional layer to process the 2D MNIST input images. The first argument passed to the Conv2D layer function is the number of output channels, the next input is the kernel size which in our case we have chosen to be a 3x3 moving window. The kernel filter looks for edges and parts. As a result of this layer, we get a convolved feature map. Once the feature maps are extracted, we move on to the RELU layer. The activation function used here is a rectified linear unit. The relu layer performs an element wise operation to set all the negative pixels of the convolved matrix (if any) to zero and introduces non-linearity. As a result, all the neurons are not activated at the same time, means at one time only a few neurons are activated. Input feature map is converted to rectified feature map. Next, we move on to the pooling layer. The rectified feature map is converted to a pooled feature map. It is a down sample operation that reduces the dimension of the feature map.

Next we add another convolutional + activation + pooling layer. Only this time we use 64 output channels instead of 32. The input tensor for this layer is (batch size, 26, 26, 32), 32 being the number of output channels from the previous layer. This allows rapid assembling of network architectures without having to worry too much about the sizes of the tensors flowing around our network.

We flatten the output from these layers to enter our fully connected layers. This determines explicitly the size of our weights and bias variables. We use the sigmoid activation function to get out output in probability.



We compile the model before fitting it. We categorize the class labels using a binary cross entropy as they classify the hot encoded values and optimize the probabilities using RMSprop to get the best accuracy possible.

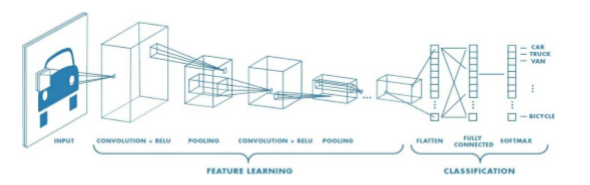


We validate the training data on the MNIST dataset as well as our own dataset.

**5. PROPOSED SOLUTION:**

Here the proposed solution used to build the sign language identification system is using convolution neural network.

Convolutional Neural Network (CNN) is a class of deep, feed-forward artificial neural networks that has been successfully applied to analyzing visual imagery. CNN use variation of multilayer perceptron’s designed to require minimal processing. CNNs has the ability to be able to detect abstract and complex features that makes them so attractive in image recognition problem.



2D Convolutional Layer: The most common type of convolution that is used is the 2D convolutional layer, and is usually abbreviated as conv2D. This layer creates a convolutional kernel that is convolved with the layer input to produce a tensor of outputs. 2D convolutional layer (e.g. spatial convolutional over images). A filter or a kernel in a conv2D layer has a height and a width.

They are generally smaller than the input image and Conv2D filters extend through the three channels in an image (Red, Green and Blue). And each filter in this layer is randomly initialized to some distributions (Normal, Gaussian, etc.). So, by having different initialization criteria, then each filter gets trained slightly differently. These are used in the first few convolutional layers of a CNN to extract simple features. Conv2D filters are used only in the initial layer of a CNN. They are put there to extract the high-level features from an image. The conv2D layer works fairly impressively.

In this project the MNIST sign language dataset has been used. The original MNIST image dataset of handwritten digits is a popular benchmark for image-based machine learning methods but researchers have renewed efforts to update it and develop drop-in replacements that are more challenging for computer vision and original for real-world applications. The American Sign Language letter database of hand gestures represents a multi-class problem with 24 classes of letters (excluding J and Z which require motion).

The dataset format is patterned to match closely with the classic MNIST. Each training and test case represents a label (0-25) as a one-to-one map for each alphabetic letter A-Z (and no cases for 9=J or 25=Z because of gesture motions). The training data (27,455 cases) and test data

(7172 cases) are approximately half the size of the standard MNIST but otherwise similar with a header row of label, pixel1,pixel2….pixel784 which represent a single 28x28 pixel image with gray scale values between 0-255. The original hand gesture image data represented multiple users repeating the gesture against different backgrounds.

A robust visual recognition algorithm could provide not only new benchmarks that challenge modern machine learning methods such as Convolutional Neural Nets but also could pragmatically help the deaf and hard-of-hearing better communicate using computer vision applications. The National Institute on Deafness and other Communications Disorders (NIDCD) indicates that the 200-year-old American Sign Language is a complete, complex language (of which letter gestures are only part) but is the primary language for many deaf North Americans. ASL is the leading minority language in the U.S. after the "big four": Spanish, Italian, German, and French. One could implement computer vision in an inexpensive board computer like Raspberry Pi with OpenCV, and some Text-to-Speech to enabling improved and automated translation applications.

Following are the steps that are followed towards the solution:

1. The MNIST sign language dataset has been used. The sign MNIST has test and train data.

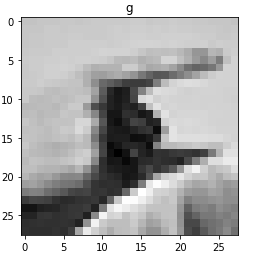
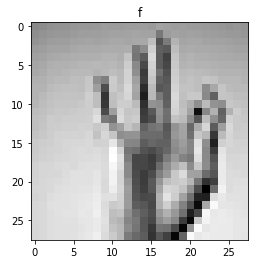
Here is a sample of the pictures used:

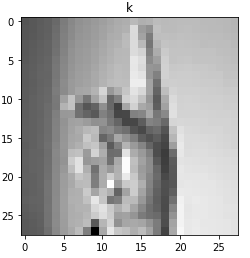
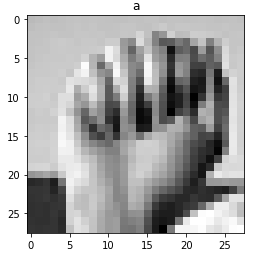


Fig: Standard American Sign Language

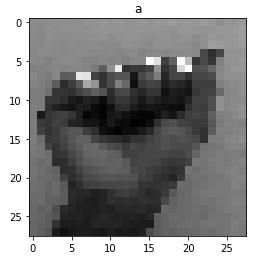
1. The train and test data is split into features and labels.
2. The input data is reshaped into a format suitable for the convolutional layers, using X\_train.reshape() and X\_test.reshape()
3. For class-based classification, one-hot encode the categories using to\_categorical()
4. The model is built using the Sequential.add() function.
5. A convolutional layer is added, for example using Sequential.add(Conv2D(…))
6. Activation functions Relu and softmax have been used.
7. A pooling layer is added, for example using the Sequential.add(MaxPooling2D(…))
8. A “flatten” layer is used ; for example using Sequential.add(Flatten()).
9. A fully connected layer using Sequential.add(Dense)) is used, and a dropout layer.
10. Model is compiled using model.compile()
11. Model is trained using model.fit(), supplying X\_train(), X\_test(), y\_train() and y\_test()
12. Model prediction is generated using model.predict()

Images after converting to grayscale:

Example of an image after model prediction:



**6. RESULT ANALYSIS:**

**Training Loss:** Loss is the result of a bad prediction. Loss value implies how well or poorly a certain model behaves after each iteration of optimization or how bad the model's prediction was on a single example. Ideally, one would expect the reduction of loss after each, or several, iteration(s).

If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples. Higher loss is the worse(bad prediction) for any model.

The loss is calculated on **training** and **validation** and its interpretation is how well the model is doing for these two sets. Unlike accuracy, a loss is not a percentage. It is a sum of the errors made for each example in training or validation sets.

In our project, the trained CNN model had Initial training and validation loss of 0.1602 and 0.063 respectively in epoch-1 and final training and validation loss of 0.0062 and 0.0566 respectively at end of epoch-10.

Our model training loss plot is given below:

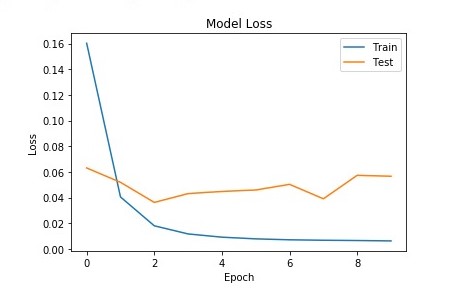


Figure: Training Loss

**Accuracy:** Accuracy is one metric for evaluating classification models or we can also say the accuracy metric is used to measure the algorithm’s performance in an interpretable way. Informally, **accuracy** is the fraction of predictions our model got right. Accuracy of a model is usually determined after the model parameters and is calculated in the form of a percentage. It is the measure of how accurate your model's prediction is compared to the true data.

In our project the CNN model is trained using the MNIST sign language dataset and the accuracy obtained here is 99.12% which is possible as we also tested the model using the test data of the MNIST sign language dataset.

It may not be possible to obtain such a high accuracy in real life scenarios but the accuracy of any model can be improved if we follow certain methods such as:

1. Adding more data

2. Treating missing and Outlier values

3. Feature Engineering- includes **feature transformation, feature creation**

**4.** Feature Selection- based on **Domain Knowledge, Visualization, Statistical Parameters**

**5. Using** Multiple algorithms

6. Algorithm Tuning

7.  Ensemble methods-can be achieved through many ways- Bagging(Bootstrap Aggregating), Boosting

8. **Cross Validation**

**Our model accuracy plot is given below:**

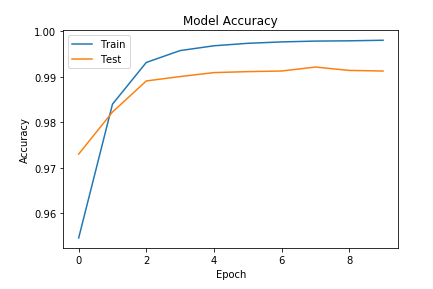


Figure: Model Accuracy

**7. CONCLUSION:**

Hence we make a sign language interpreter and understand how it interprets pictures of hand signs. This will help to facilitate communication between hearing and deaf clients. Human hand gesture provides the most common form of non-verbal interaction among people. At present, artificial neural network is emerging as the technology of choice for many applications such as pattern recognition, gesture recognition, system identification and control. We have used multi scale feature extraction to develop an accurate hand gesture recognition method. This method has a promising performance with various gestures.

**8.** **FUTURE SCOPE:**

In future, with the use of sensor based technology in ASL some words which uses wrist movement can also be detected so that those signs can be expressed by glove. Similarly elbow movement and facial recognition can be the two areas. Concept of video conferencing with the deaf and dumb people can be introduced for American Sign Language for advanced communication. Communication between the deaf and dumb people in Indian sign language without using sensor based technology can also be done using Smartphone platform which will provide easy to use environment, mobility and high growth rate. American sign language sentence formation with both single handed and double handed gestures without using sensor based approach as vision based is more easy to use and is less cost effective.

Dynamic hand gesture recognition or real time gesture recognition of sentences can also be done by using Deep Learning Approach. A Bot based technology can be the further extension introduced in the Indian sign language which will reply to the question asked in conversation with the hearing impaired people.

**9. BILIOGRAPHY:**

[1] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In Advances in Neural Information Processing Systems (NIPS), 2014.

[2] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In IEEE International Conference on Computer Vision, 2015.

[3] Barczak, A.L.C., Reyes, N.H., Abastillas, M., Piccio, A., Susnjak, T. (2011), A new 2D static hand gesture colour image dataset for ASL gestures, Research Letters in the Information and Mathematical Sciences, 15, 12-20

[4] Kim, Taehwan & Livescu, K & Shakhnarovich, Greg. (2012). American sign language fingerspelling recognition with phonological feature-based tandem models. In IEEE Spoken Language Technology Workshop (SLT), 119-124.

[5] Eleni Efthimiou, Stavroula-Evita Fotinea, Christian Vogler, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet , Petros Maragos, Jérémie Segouat (2009): Sign Language Recognition, Generation, and Modelling: A Research Effort with Applications in Deaf Communication.

[6] Helen Cooper, Brian Holt, Richard Bowden (2011): Sign Language Recognition.

[7] Lorena P. Vargas, Leiner Barba, C O Torres, L Mattos (2011): Sign Language Recognition System using Neural Network for Digital Hardware Implementation.

[8] Joyeeta Singha, Karen Das (2013): Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique.

[9] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, Benjamin Schrauwen (2014): Sign Language Recognition using Convolutional Neural Networks.

[10] Maher Jebali, Patrice Dalle, Mohamed Jemni (2014): Sign Language Recognition System Based on Prediction in Human-Computer Interaction.

[11] Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen (2015): Real-time Sign Language Fingerspelling Recognition using Convolutional Neural Networks from Depth map

[12] Jie Huang, Wengang Zhou, Houqiang Li, Weiping Li (2015): Sign Language Recognition using 3D Convolutional Neural Networks.

[13] Brandon Garcia, Sigberto Alarcon Viesca (2016): Real-time American Sign Language Recognition with Convolutional Neural Networks.

[14] Ashish S. Nikam, Aarti G. Ambekar (2016): Sign Language Recognition Using Image Based Hand Gesture Recognition Techniques.

[15] Vivek Bheda, N. Dianna Radpour (2017): Using Deep Convolutional Networks for Gesture Recognition in American Sign Language.

[16] Beena M.V., Dr. M.N. Agnisarman Namboodiri (2017): Automatic SignLanguage Finger Spelling Using Convolution Neural Network: Analysis.

[17] Al Amin Hosain, Panneer Selvam Santhalingam, Parth Pathak, Jana Kosecka, Huzefa Rangwala (2019): Sign Language Recognition Analysis using Multimodal Data.

[18] Shivashankara S, Srinath S (2019): American Sign Language Video Hand Gestures Recognition using Deep Neural Networks.

[19] Kshitij Bantupalli, Ying Xie (2019): American Sign Language Recognition Using Machine Learning and Computer Vision.

[20] Agarwal, Anant & Thakur, Manish. Sign Language Recognition using Microsoft Kinect. In IEEE International Conference on Contemporary Computing, 2013.