SYNOPSIS
ON

Sign Language Recognition using Machine Learning

Submitted By

Roll No.	Name of Student	Email ID	Mobile No.
62	Nikhil Dapkekar	nikhildapkekar42031@gmail.com	9405611564
63	Piyush Dabare	piyushdabare2001@gmail.com	7744806955
42	Ayush Raut	aayushr512@gmail.com	9172553507
50	Jay Rathod	jayrathod5588@gmail.com	8407903157

Under The Guidance Of

Prof. Pradnya Moon

Department Of Computer Technology

YESHWANTRAO CHAVAN COLLEGE OF ENGINEERING, NAGPUR

SESSION 2022-2023

1. INTRODUCTION

There is an undeniable communication problem between the deaf community and the hearing majority which can be solved by using innovation ideas and techniques which can be used to automatically recognise sign language and convert it into text. Sign Language is used by the deaf and voiceless community to be able to communicate with others, but the most commonly faced problem here is that everyone around may not be able to understand sign language. Written communication can be also be used but it is slow as compared to face-to-face communication. In case emergency situations its more efficient to use sign language than written communication therefore, many sign languages were created in different countries of world.

American Sign Language was created in 1817 for deaf students. The main goal was to represent the letter and structure of the English language using hands, so that deaf students can use English. Until 1835, ASL was used as a language of instruction and student communication in schools for the deaf. Sooner the use of Sign Language spread throughout the world.

To solve this problem, we can use a custom Convolution Neural Networks (CNN) model to recognize hand gestures. Sign language recognition is a collaborative research field that includes natural language processing, computer vision, pattern matching, and linguistics. Its goal is to compile various methods and algorithms in order to identify already created signs and perceive their meaning.

2. AIM

Sign language recognition (SLR) is a research area in computer vision. Sign language recognition (SLR) required to solve problems like video trimming, sign extraction, sign video background modelling and sign classification. Sign Language recognition (SLR) is very complex topic because of the complexity and diversity of Sign Languages. For example, every country has its own Sign Language and standard. Like Indian and America Sign Language are different from each other. For example, Indian sign language (ISL) uses two hand to symbolize letter "a" while American Sign Language (ASL) required only one hand. So, we are going to make use of Convolution Neural Networks (CNN) to recognise and convert the American hand sign language into text. Convolutional Neural Networks (CNN)is a class of Artificial Neural Network (ANN) which is mostly used in image recognition, image classification, object detection, etc**.**

3. OBJECTIVES

- Our main objective is to capture images from camera, extract hand image from the input image and convert it into text. We are going to use CNN for classification of images. CNN are regularized versions of multilayer perceptron's. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. For that we need to implement a convolutional layer, pooling layer, and fully connected layer.

- Convolution layer performs a dot product between two matrices, where one matrix is a filter and the other matrix is the restricted portion of the image.

- At the end of the convolution process, we have a featured matrix which has lesser parameters(dimensions) than the actual image as well as more clear features than the actual one.

- Pooling layer is applied to decrease the computational power required to process the data. It is done by decreasing the dimensions of the featured matrix even more. After pooling layer, we get a matrix containing main features of the image and this matrix has even lesser dimensions.

- After we have done highlighting some features in an image and reduces the dimensions of the image drastically. we are going to do the classification process and display the letter corresponding to that image.

4. LITERATURE SURVEY

4.1 Machine Learning-based Hand Sign Recognition

Greeshma Pala et. al. [1], has done a comparison study between K-Nearest neighbours (KNN), Multi-class Super Vector Machine (SVM), and Convolutional Neural Networks (CNN) algorithms to determine which algorithm will be able to recognized hand signs with more accuracy among three of them. Algorithms such as K-Nearest neighbours (KNN), Multi-class Super Vector Machine (SVM), and also experiments using hand gloves were used to decode the hand gesture movements before to find the best way with high accuracy. Approximately 29,000 images were used and split into test and train dataset , after that all the images are pre-processed to fit into the KNN, SVM, and CNN models. After implementation and test of KNN, SVM, and CNN algorithms accuracy of 93.83%, 88.89%, and 98.49% was obtained respectively. It was concluded after that CNN is comparatively the best among KNN, SVM, and CNN algorithms with a good accuracy and minimum loss.

4.2 Deep Convolutional Neural Networks for Sign Language Recognition

G. Anantha Rao et. al. [2], has created a mobile Indian sign language gestures recognition app using Convolutional Neural Networks (CNN) in which a video is captured as input so that the other can capture the video and text is displayed on the screen. They create the dataset with five different subjects performing 200 signs in 5 different viewing angles under various background environments where each sign occupied 60 frames in video. Different CNN architectures were designed and tested to obtain better accuracy in hand sign recognition. At the end authors achieved accuracy of 92.88% compared with other classifier models on the same dataset.

4.3 A Comprehensive Study on Deep Learning-Based Methods for Sign Language Recognition

N. Adaloglou et. al. [3], has done a comparative experimental assessment of computer vision-based methods for SLR by implementing and evaluating multiple datasets which available to public. They have done an in-depth analysis of the most characteristic Deep Neural Network based SLR model architectures. Continuous Sign Language Recognition (CSLR) is a task very similar to the one of continuous human action recognition. There are sequence of glosses (instead of actions) needs to be identified in a continuous stream of video data. But glosses are involved in very small number of frames. The authors also find out that 3D CNN based architectures are more efficient in isolated SLR than 2D CNN based model.

4.4 Sign Language Recognition Using Modified Convolutional Neural Network Model

Herman Gunawan et. al. [4], try to implement one of model which is i3d inception which is also a new Action Recognition model with very high accuracy. They try to find out if it is possible to adopt Action Recognition behaviour into Sign Language Recognition. The goal of this paper is to implement the i3d inception model to Sign Language Recognition model.

4.5 The Efficiency of Sign Language Recognition using 3D Convolutional Neural Networks

N. Soodtoetong and E. Gedkhaw [5], has studied process and method which related with the recognition of sign language using deep learning. 3D-CNN algorithm was used for recognized process and input was taken by capturing images from the Kinect Sensor. The Kinect Sensor was used to collect RGB image, which were then classified using 3D CNN. The result show that the 3D-CNN algorithm could recognize the gesture motion and have the accuracy of 91.23%.

4.6 Sign Language Recognition Systems: A Decade Systematic Literature Review

Ankita Wadhawan and Parteek Kumar [6], has done a Systematic Literature Review and a classification scheme in the paper in between 2007 to 2017. One hundred and seventeen research articles were selected by the authors and reviewed. Each of 117 selected papers were compared with each other on the basis of the method and process they used to acquire data, single or double signs language used, classification technique used and accuracy. It was found that most of the major research on sign language recognition has been performed in isolated and using single handed signs language while taking input from camera. The main aim was to provide readers and researchers a roadmap to guide future research and facilitate knowledge accumulation in sign language recognition.

5. EXISTING METHODLOGIES

There are many algorithms and techniques available for implementation of hand sign recognition systems few of which algorithm are Support Vector Machines (SVM), K-Nearest-Neighbor (KNN) and Convolution Neural Networks (CNN).

5.1 K-Nearest-Neighbor (KNN)

It is one of the best in machine learning algorithms used for classification, it uses a vector space model to classify data points. Firstly, each image is converted into an 2-D array for quick pairwise distance calculation followed by dividing the data into training and testing datasets. In classification process, the algorithmic first assigned k nearest neighbors to the unlabelled query points and after that its check which group has more numbers of nearest neighbors. Generally, K is assigned odd value so that it will easy during the voting process. If we choose small value of k means then noise will have a higher effect on the output and if K is large than it makes the system computationally expensive and slows in performance as well [1].

5.2 Support Vector Machines (SVM)

Support Vector Machines (SVM) is a supervised learning model which is used for regression analysis and classification. It provides a review to the support vector machines Is call-back vector clustering and is use as substitute when data is not labelled. The main purpose of SVM is to create an optimal separating hyperplane so that it can be used for classifying the training vectors. SVM chooses the extreme points that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine [1].

5.3 Convolution Neural Networks (CNN)

CNN are regularized versions of multilayer perceptron's. Multilayer perceptron's usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. For that we need to implement a convolutional layer, pooling layer, and fully connected layer. Convolution layer performs a dot product between two matrices, where one matrix is a filter and the other matrix is the restricted portion of the image. Pooling layer is applied to decrease the computational power required to process the data. It is done by decreasing the dimensions of the featured matrix even more. For example, Max pooling in which we select the maximum value from matrix. After pooling layer, we get a matrix containing main features of the image and this matrix has even lesser dimensions. Next the image is flattened in 1 D array and weight are find use back propagation method and model is train to classify input images[1].

6. PROPOSED METHODOLOGY

HIGH LEVEL DESIGN

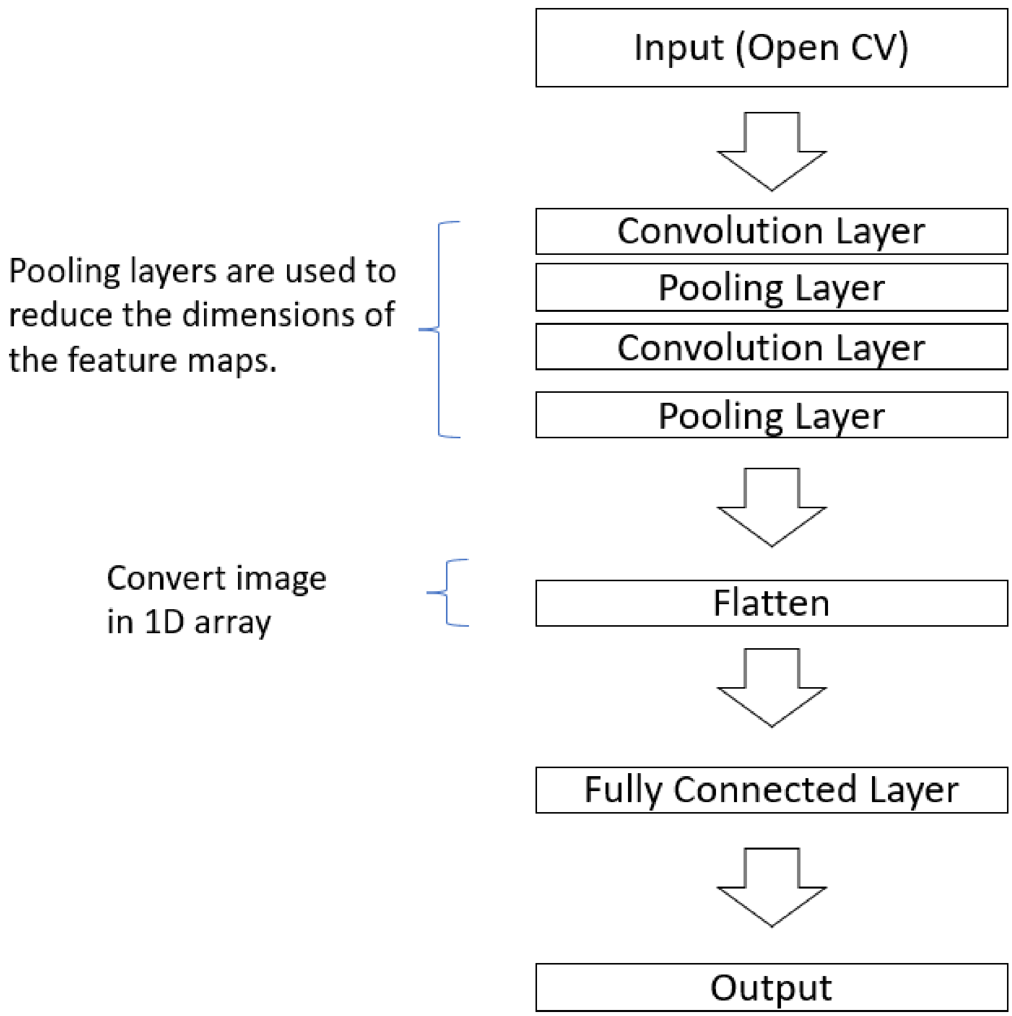

Figure 6.1 Implementation of CNN Layers

Convolutional Neural Network (CNN) is a class of deep neural networks which is mostly used to do image recognition, image classification, object detection, etc. The CNN model consists of many convolution layers like convolutional layer, rectified non-linear layer, pooling layer, and fully connected layer. Each convolutional layer contains a set of filters/kernels/feature maps which helps identify the different patterns in the image. Convolution layer performs a dot product between two matrices, where one matrix is a filter and the other matrix is the restricted portion of the image, while before implementing convolution layer we to do padding. At the end of the convolution process, we have a featured matrix which has lesser parameters(dimensions) than the actual image as well as more clear features than the actual one. Pooling layer is applied to decrease the computational power required to process the data. It is done by decreasing the dimensions of the featured matrix even more. After pooling layer, we get a matrix containing main features of the image and this matrix has even lesser dimensions. The filters of Convolution and Pooling layer are repeated few more time for a better understanding of the image. After we have converted our input image into a suitable form for our multi-Level fully connected architecture, we flatten the image into 1-D vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model can distinguish between dominating and certain low-level features in images and classify them the letter which it symbolizes is displayed.

USE CASE DIAGRAM

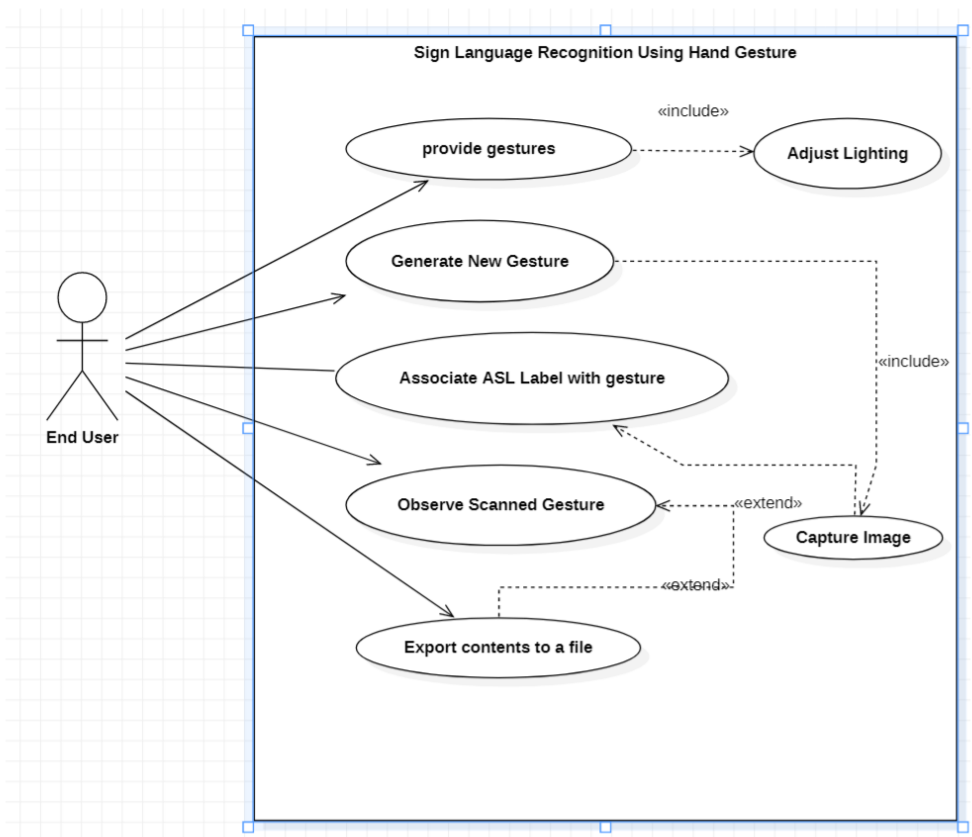

Figure 6.2 Use Case Diagram of Hand Sign Recognition

7. PLAN OF ACTION

7.1 Module 1: Literature Survey

In this module, we will study the various research publications, and implement the knowledge gained from these papers in developing our project.

7.2 Module 2: Collecting dataset

In this module, we will search dataset of images which will be required for training and testing the model. After selecting a proper dataset of images containing images of all letters of American Sign Language, we need to delete all the unnecessary images and folders and files present in dataset. After that we need to split the dataset into two sets - a training set and a testing set. The training set will be need to learn and a testing set is used to check the accuracy of model after the training is completed.

7.3 Module 3: Implementation and Training the CNN model

In this module, we are going to implement the different types of convolution layers and pass the training data to the model so that it can learn from the data and predict the result. After training the model, we will check how it's performing by testing it on the test dataset and find the accuracy of the model.

7.4 Module 4: Thesis writing

In this module, after extensive testing is done a proper thesis will be made.

Activities	**Months**					
	Aug-Sep	Sep-Oct	Dec-Jan	Jan-Feb	Feb-Mar	Mar-Apr
Module 1						
Module 2						
Module 3						
Module 4						

Figure 7.1 Plan of Action

8. EXPECTED OUTCOMES

- The aim of this project is to build a real time SLR computer application that can automatically detect hand gestures in natural lighting conditions and convert that into text.

- We are going to use of Convolution Neural Networks (CNN) to recognise and convert the American hand sign language into text. In CNN, we take an image as an input, assign importance to its various aspects or features in the image and classified one image from another.

- The model can be further trained with a dataset such that its accuracy does not get affected by the background of image gesture which is taken from camera.

- Modules can be optimized and integrated in AI systems which can be used by public. For example, Google assistant.

- Same model can be used to recognized hand sign language which uses two hands for representing a single character such as Indian Sign Language.

Expected chapter scheme of the proposed work

- Chapter 1. Introduction
- Chapter 2. Mathematical Foundation/ Mathematical background
- Chapter 3. Problem definition/Problem formulation
- 3.1 Problem definition (5-6 lines)
- 3.2 Problem formulation (1-2 pages)
- 3.3 Work carried out (Related to objective)
- Chapter 4. Literature Review
- 4.1 General work
- 4.2 Related work
- 4.3 Gaps identified
- 4.4 Scope of improvement and limitations
- Chapter 5. Proposed Methodology/Approach
- Chapter 6. Experimental Setup, Results and Discussion
- Chapter 7. Conclusion and Future Scope

9. REFERENCES

1. G. Pala, J. B. Jethwani, S. S. Kumbhar and S. D. Patil, "Machine Learning-based Hand Sign Recognition," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 356-363, doi: 10.1109/ICAIS50930.2021.9396030.
2. G. Anantha Rao, K. Syamala, P. V. Kishore, "Deep Convolutional Neural Networks for Sign Language Recognition," Biomechanics and Vision Computing Research Centre, Department of ECE, K.L. University, Green Fields, Guntur (DT), Andhra Pradesh, INDIA.
3. N. Adaloglou, Theocharis Chatzis, Ilias Papastratis, "A Comprehensive Study on Deep Learning-Based Methods for Sign Language Recognition," in IEEE Transactions on Multimedia, vol. 24, pp. 1750-1762, 2022, doi: 10.1109/TMM.2021.3070438.
4. Suharjito, H. Gunawan, N. Thiracitta and A. Nugroho, "Sign Language Recognition Using Modified Convolutional Neural Network Model," 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), 2018, pp. 1-5, doi: 10.1109/INAPR.2018.8627014.
5. N. Soodtoetong and E. Gedkhaw, "The Efficiency of Sign Language Recognition using 3D Convolutional Neural Networks," 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2018, pp. 70-73, doi: 10.1109/ECTICon.2018.8619984.
6. Ankita Wadhawan and Parteek Kumar, "Sign Language Recognition Systems: A Decade Systematic Literature Review," 2019 CIMNE, Barcelona, Spain 2019
7. R. S, Dr. Sabeenian & Bharathwaj, S. & Aadhil, M (2020). "Sign Language Recognition Using Deep Learning and Computer Vision," Journal of Advanced Research in Dynamical and Control Systems. 12. 964-968. 10.5373/JARDCS/V12SP5/20201842.
8. Khalil, Khalid & El-Darymli, Khalid & Khalifa, Othman & Enemosah, Hassan. (2006). "Speech to Sign Language Interpreter System (SSLIS)," IEEE International Conference of Computer and Communication Engineering (ICCCE'06).
9. P. Das, T. Ahmed and M. F. Ali, "Static Hand Gesture Recognition for American Sign Language using Deep Convolutional Neural Network," 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 1762-1765, doi: 10.1109/TENSYMP50017.2020.9230772.
10. T. D. Sajanraj and M. Beena, "Indian Sign Language Numeral Recognition Using Region of Interest Convolutional Neural Network," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 636-640, doi: 10.1109/ICICCT.2018.8473141.
11. I. H. Yemenoglu, A. F. M. S. Shah and H. Ilhan, "Deep Convolutional Neural Networks-Based Sign Language Recognition System," 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2021, pp. 0573-0576, doi: 10.1109/IEMCON53756.2021.9623068.
12. "Sign Language Recognition using Deep Learning", AUG 2022, https://towardsdatascience.com/sign-language-to-text-using-deep-learning-7f9c8018c593
13. Triyono, Liliek & Pratisto, Eko & Bawono, (2018). "Sign Language Translator Application Using OpenCV," IOP Conference Series: Materials Science and Engineering. 333. 012109. 10.1088/1757-899X/333/1/012109.

14. Pathak, Aman, Avinash & Chugh, Editor. (2022), “Real Time Sign Language Detection,” International Journal for Modern Trends in Science and Technology. 8. 32-37. 10.46501/IJMTST0801006
15. Manan Parekh, “A Brief Guide to Convolutional Neural Network,” 2019 https://medium.com/nybles/a-brief-guide-to-convolutional-neural-network-cnn-642f47e88ed4
16. A. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1-5, doi: 10.1109/GET.2016.7916786.
17. Reddygari Sandhya Rani , R Rumana , R. Prema, 2021, A Review Paper on Sign Language Recognition for The Deaf and Dumb, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 10 (October 2021).
18. Amrita Thakur, Pujan Budhathoki, Sarmila Upreti, Shirish Shrestha, Subarna Shakya, “Real Time Sign Language Recognition and Speech Generation,” Journal of Innovative Image Processing (JIIP) (2020) Vol.02/ No. 02 Pages: 65-76 https://www.irojournals.com/iroiip/ DOI: https://doi.org/10.36548/jiip.2020.2.001
19. I. V. Lopez, “HAND GESTURE RECOGNITION FOR SIGN LANGUAGE TRANSCRIPTION,” May 2017, Boise State University.
20. Y. Ji, S. Kim and K. -B. Lee, "Sign Language Learning System with Image Sampling and Convolutional Neural Network," 2017 First IEEE International Conference on Robotic Computing (IRC), 2017, pp. 371-375, doi: 10.1109/IRC.2017.40.