Sign Language Recognition System Using Customized Convolution Neural Network

Chapter · February 2023

DOI: 10.1007/978-981-19-6634-7_59

Dipmala Salunke
Jayawant Shikshan Prasarak Mandal
24 PUBLICATIONS **32** CITATIONS
SEE PROFILE

Nihar Madhaw Ranjan
Rajarshi Shahu College of Engineering
75 PUBLICATIONS **261** CITATIONS
SEE PROFILE

Pallavi M Tekade
Jayawant Shikshan Prasarak Mandal
5 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Sign Language Recognition System Using Customized Convolution Neural Network

Dipmala Salunke, Ram Joshi, Nihar Ranjan, Pallavi Tekade, and Gaurav Panchal

Abstract With the rapid development of computer vision and its expansion of applications, the demand for interaction between human and machine is becoming more and more extensive. Sign language recognition, which is a part of language technology, involves evaluating human hand signs and gestures through machine learning algorithms. Vision-oriented hardware like camera is used to convert these sign languages into meaningful form like text or sound. We have implemented a system that captures the hand sign or gesture of the user and predicts the equivalent meaning of the language. A lot of work has been done in the field of sign language recognition; however, a smaller amount of work done where, kids can learn and practice sign language as well as learn basic mathematical operations using sign language. This paper mainly focuses on making a system that can be used by children with hearing or vocal disabilities to learn basic alphabets, numbers and some words in sign language. Deep learning-based customized convolution neural network is trained on the dataset size 2400 * 44, where seven convolution layers are defined for sign language prediction with the accuracy of 99.92%.

Keywords Convolution neural network · Sign language recognition · Gesture recognition · Machine learning

1 Introduction

According to data from leading health organizations like WHO, about 4.3 crore people across all the countries in the world have problems related to hearing loss which is about 6% of the world's population, of whom 3.4 crore are children. Studies expect that the number will rise to 900 million by 2050 [1]. People can exchange the feelings and ideas and interact only though voice or actions. People communicate

D. Salunke (✉) · R. Joshi · N. Ranjan · P. Tekade · G. Panchal
Department of Information Technology, JSPM's Rajarshi Shahu College of Engineering Tathawade, Pune, India
e-mail: dipmala.salunke@gmail.com

with each other using sign language which is one of the ways for non-verbal communication. Sign language helps in communication for people to express themselves and understand each other using gestures or signs. To communicate with deaf persons, we can use hand, facial, and body movements. We are building a sign language recognition technology to bridge the gap between normal people and deaf people. It will be an excellent tool for communicating with people who have hearing impairments. Also, for non-sign language users to understand the communication, there is a good interpretation. Gesture recognition is a type of perceptual computing which has a user interface that allows computers to capture the user's signs and interpret human gestures into its equivalent meaning. Simple gestures can be used to interface with devices. There are a variety of methods for recognizing gestures, including cameras, computer vision algorithms, and so on. With this, we can readily understand human body language. According to a review of the literature, several experiments on sign language identification have been conducted, although with small datasets, and some of the articles have failed to recognize all alphabets. For American sign language, we customized a convolution neural network that recognizes 26 alphabets, 0-9 digits and 8 words. The authors built a dataset of 2400 gestures for each letter, totaling 105600 gestures. The goal of this system is to obtain high recognition rates. The system is developed using Python OpenCV library, convolutional neural network with hidden layer nodes along with activation function rectified linear unit.

2 Literature Survey

Narayana et al. [2] implemented a system that was able to improve the rate of hand recognition from 67.71 to 82.07% on ISO GD dataset. They used ISO GD dataset and NVIDIA datasets as benchmarks and got an accuracy of 91.28% which was increased by 8.9% for the previous best result not using IR dataset. Hossen et al. [3] worked on regional language and developed a system based on Bengali sign language, and with the help of CNN, they were able to implement a system where they could identify 37 static sciences out of the 51 letters in Bengali sign language. They used 1147 images for the system and got an accuracy of 96.33%. They used different layers like convolutional layer, max pooling layer and activation function rectified linear unit for better accuracy. Dieleman et al. [4] proposed a system using two convolutional neural networks for extracting hand features and one for extracting the upper body features in this system; they mainly focused on implementation of 3D Evolutions although a 2D convolution has better validation accuracy than 3D convolution, but they were able to achieve accuracy of 95.68% and also observed false positive rate of 4.13%. Cheng et al. [5] proposed a system where two networks like CNN and RBM network have combined together to give a better accuracy for gesture recognition. They used Kinect sensor to obtain the color and depth of samples and implemented a system which had a better recognition rate with an error of just 3.9%. By incorporating tropic layers and Optimizer, Narang et al. [6] were able to predict sign languages of 26 characters with an accuracy of 97.3%, eliminating background noise and preventing overfitting. Two stochastic gradient algorithms, root mean square propagation and adaptive gradient algorithm, were merged. Rajendran et al. [7] proposed a system in which they trained a three-layered deep convolutional neural network to recognize sign languages with a 99.96% on training data and a % accuracy on testing data. Pradeep Kumar et al. [8] implemented a system using independent Bayesian classification combination where the recognition rates were 96.05 and 94.27% for single- and double-hand gestures. They collected a dataset of 51 dynamic word sign gestures for the system. Beena et al. [9] proposed a system that has trained using 1000 images of each numeral signs. The algorithm extracts features from the block processed images and trained using the artificial neural network (ANN) and obtained an accuracy of 99.46% for the depth image. CNN model was used to recognize 50 different signs in Flemish sign language to recognize a set of 50 different signs in the Flemish sign language with an error of 2.5%, using the Microsoft Kinect.

3 System Architecture

The flow diagram of sign language recognition system is shown in Fig. 1. Web camera is used to capture the gestures for 26 alphabets, 0–9 digits and 8 words including best of luck, like, remember, you, I/me, love, luck and I love you. Customized convolution neural network is used to classify the image, corresponding class of image is recognized, and text is displayed on the screen that gets converted into voice.

The system consists of two modules:

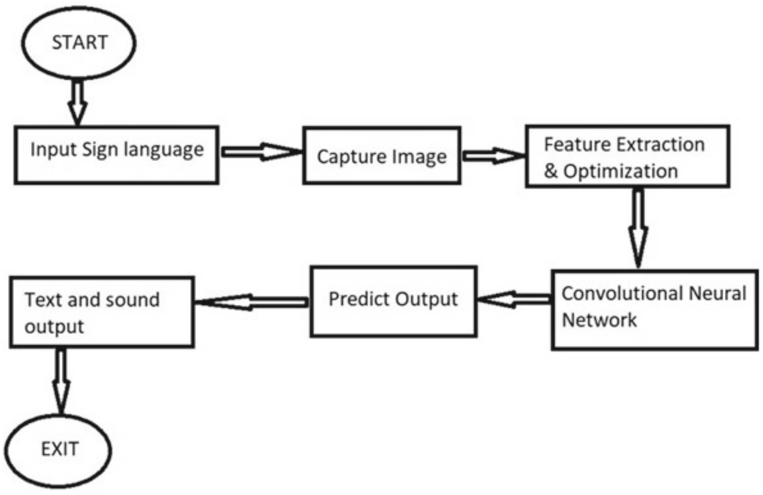


Fig. 1 Sign recognition system flow diagram

3.1 OpenCV

OpenCV is a library of programming functions that aimed at real-time computer vision.

(a) Capturing frame and displaying them:

- We will capture the frame from the camera one by one
- display them on to the display
- Importing libraries
- Creating camera object
- Reading the frame
- Displaying the frames

(b) Extracting region of interest:

Threshholding the image to binary format. Converting it into grayscale (Fig. 2) thus giving out the object or action into a single format.

(c) Finding contours:

Contours are a curve of a points, with no gaps in the curve. Contours are extremely useful for approximation of shape and analysis (Fig. 3).

(d) Finding convexity defects:

Convexity defect is a cavity in an object segmented out from an image. There is some unwanted part inside the main boundary that needs to be removed (Fig. 4).

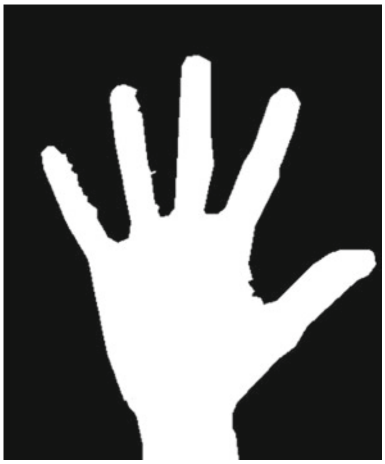


Fig. 2 Gray scaling

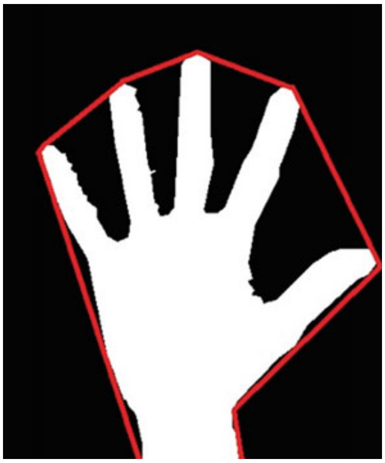


Fig. 3 Region of interest

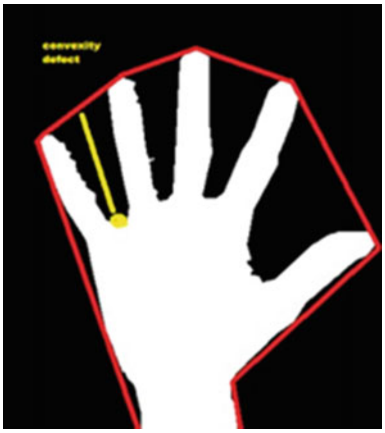


Fig. 4 Convexity defects

3.2 *Convolutional Neural Network*

Convolutional neural network has found out to be the most efficient layer technology for classifying images. In this proposed system, we will train the model with the dataset created and then we will feed the camera with the user gestures which will then be fed to the CNN input layer as shown in Fig. 5. There can be many layers to a CNN, but for this model we will be implementing a customized CNN with one input layer, seven hidden layers and one output layer. Various matrices operations and load balancing in the nodes of the hidden layer will correspond to the prediction of the gestures given by the user on the screen.

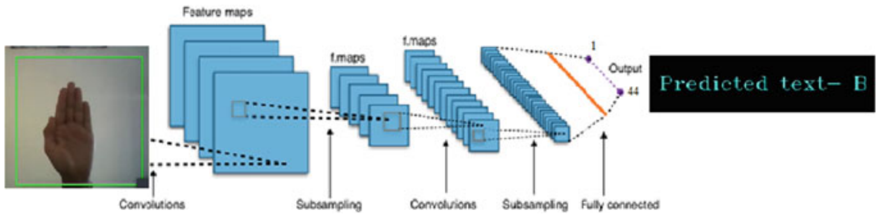


Fig. 5 Convolutional neural network

4 Methodology

The first step of the proposed system is to create a dataset. There are various ways to collect or create gestures, but for the proposed system, we are going to use webcam. The data goes under various processing where the background is detected and eliminated using Hue Saturation Value (HSV) algorithm. Segmentation is done to detect hand from the background. The image is then resized to 50 $\times$ 50 pixels. Our dataset contains 2400 images per gesture; also the training and testing data are split into 80:20 ratio. Binary pixels are extracted from the image and then forwarded to rotate the image where we rotate the data since the images generated by the webcam are mirrored so in this step, we flip the image into right side up. The dataset is trained using customized convolution neural network. The model is then evaluated, and the system then is able to predict the gestures.

4.1 Set Histogram

We first create a histogram data in order to calculate the values of histogram to find the difference between hand and the background which helps to distinguish them. This process creates a histogram object and saves it into the directory. It is created using Pickle library which converts a Python object into byte stream.

This object is then used later on for future operations.

4.2 Creating Dataset

The data collection is an important in order to move forward for the gesture creation. We have created our own dataset containing 105,600 images of 44 gestures using webcam. We have 44 classes each for the gestures. In order to get higher consistency, we have created the dataset at same background with webcam each time commanded. The images captured are in RGB format and are then converted into hsv format as well; other operations are done before saving them. We have used SQLite in order

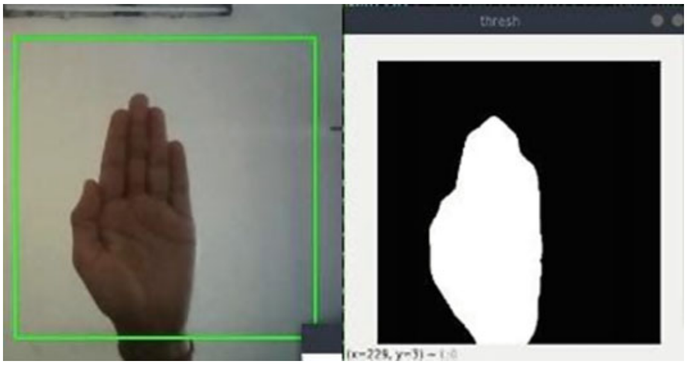


Fig. 6 Image processing

to label the images as they are later referred using 'id' assigned to them. This helps to prepare a proper key–value pair for this image.

4.3 Image Processing

In this, we convert the image from RGB to HSV format as shown in Fig. 6. Then we calculate the back project from the histogram of the background and the image to segment out hand gesture from it. We then threshold the image to get the proper contours of the image. The convexity defects are then removed as well; a fine line of Gaussian and Radian blur is used to remove any unnecessary noise created during creation of the image. The image is then converted to binary in which the image contains two colors, i.e., black and white. The image is then saved into a folder named 'gestures' which contains a folder of all the images named with their 'ids'.

4.4 Model Creation

A customized convolution neural network model is used to extract features from frame to predict hand gestures. It is a multilayered neural network used for image recognition. The architecture containing some seven convolution layers, each containing a pooling layer, activation function, flatten and batch normalization. The first layer contains 16 filters and kernel size of 2 × 2. 'RELU' activation function is used in the layer and the shape of 50 × 50. Then a pooling layer is used with the padding of 'relu' and a pool size of 2 × 2. The sequence of the layers is then 32, 64. Each containing a pooling layer of the specified specification. Then a flatten layer is used to convert the image into a single array. The dense layer is added onto the model of 128 filters with 'relu' as its activation function. The data sometimes

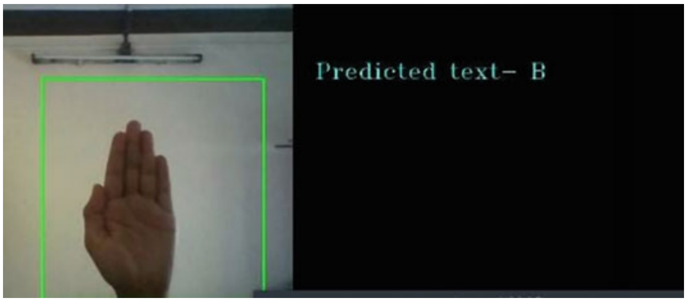


Fig. 7 Display output window

may cause overfitting so a dropout of 0.2 is done to the generated output from the layer. Then another dense layer containing the number of gestures as its filter is used with ‘softmax’ as its activation function. Then model is compiled by using category cross-entropy as the loss function and Adam as the Optimizer.

4.5 Displaying Gestures

All the gestures are arranged into a single picture into gallery format and then saved onto the directory. This file is saved in ‘.jpeg’ format. The file is later used during classification. Figure 7 shows the output window to display predicted text by the model.

4.6 Real-Time Classification

The model is now used to predict the gestures in real time. The process for real time classification uses the same image segmentation methods. The image is then sent to the model for prediction, and the model outputs the ‘id’ assigned to the gesture which in turn does a SQL query in the background to get the label associated with the gesture. The predicted text is displayed to the user in the predicted text block. If the prediction stays for more than 2 seconds and the predicted gesture has accuracy of more than 70%, the label associated with the gesture is added to the string which helps create a meaningful sentence out of many gestures in this step, we also have included a ‘pyttsx3’ which converts a text into speech which makes the text audible to the user. The audio is only played after the gestures are removed from the assigned box, and the threshold box doesn’t contain hand in it. The steps for real time sign language recognition are shown in Fig. 8.

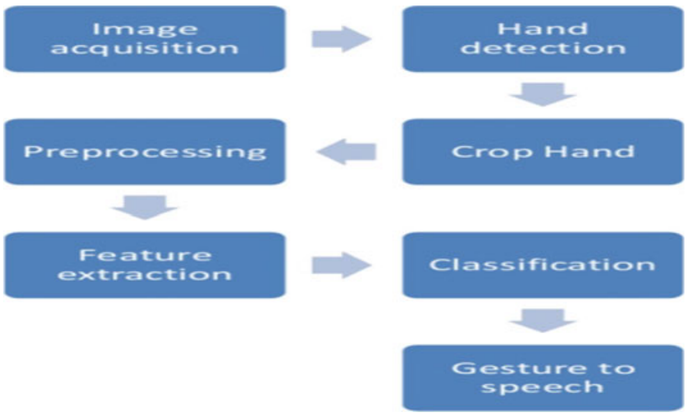


Fig. 8 Algorithm

5 Results and Discussions

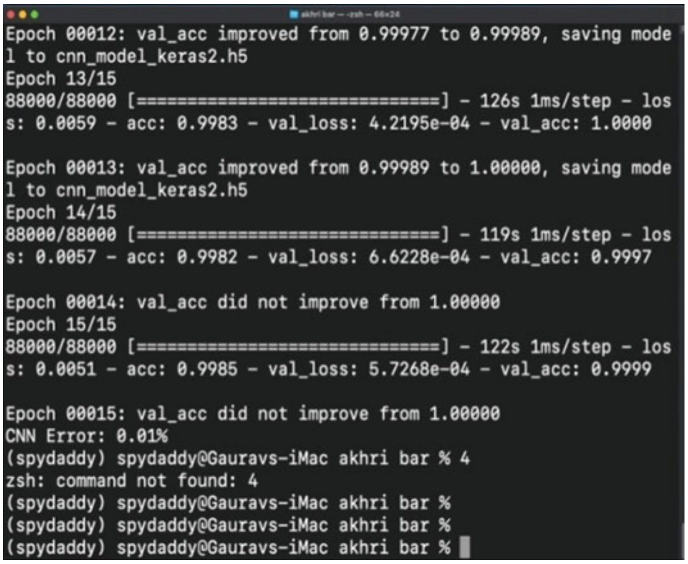
5.1 Dataset

The database used for this system is manually created using the webcam and contains more than 2400 images of the different gestures. Currently, it can detect all the alphabets as well as numbers; also the database contains few words for practice purposes.

5.2 Software Used

To implement this deep learning model, we have used various Python libraries mainly keras and TensorFlow. The language used is Python 3.6. The libraries used in this are:

- h5py = 1.5
- NumPy
- scikit-learn
- keras = 2.5.0
- OpenCV-Python
- pyttsx3
- TensorFlow = 1.5.0.



```
Epoch 00012: val_acc improved from 0.99977 to 0.99989, saving mode
l to cnn_model_keras2.h5
Epoch 13/15
88000/88000 [==============================] - 126s 1ms/step - los
s: 0.0059 - acc: 0.9983 - val_loss: 4.2195e-04 - val_acc: 1.0000

Epoch 00013: val_acc improved from 0.99989 to 1.00000, saving mode
l to cnn_model_keras2.h5
Epoch 14/15
88000/88000 [==============================] - 119s 1ms/step - los
s: 0.0057 - acc: 0.9982 - val_loss: 6.6228e-04 - val_acc: 0.9997

Epoch 00014: val_acc did not improve from 1.00000
Epoch 15/15
88000/88000 [==============================] - 122s 1ms/step - los
s: 0.0051 - acc: 0.9985 - val_loss: 5.7268e-04 - val_acc: 0.9999

Epoch 00015: val_acc did not improve from 1.00000
CNN Error: 0.01%
(spydaddy) spydaddy@Gauravs-iMac akhri bar % 4
zsh: command not found: 4
(spydaddy) spydaddy@Gauravs-iMac akhri bar %
(spydaddy) spydaddy@Gauravs-iMac akhri bar %
(spydaddy) spydaddy@Gauravs-iMac akhri bar %
```

Fig. 9 Accuracy

5.3 *Specifications*

The system runs on a machine having 8GB RAM. The operating system used is Linux, and the processor is Intel i5 with integrated intel graphics.

5.4 *Confusion Matrix*

A confusion matrix is used to describe the performance of a classification model. The described performance is based on set of test data for which the true values are known. Figure 9 shows the accuracy value for the model. Figure 10 shows the training vs validation loss for the model. Figure 11 shows the training vs validation accuracy of the model and confusion matrix for this system having 44 of classes with accuracy value 99.92 and misclassification 0.0008 is depicted in Fig. 12.

6 Conclusion

For those with speech and hearing disabilities, this paper proposes a customized convolution neural network model for American sign language recognition. Image binarization, image optimization and image segmentation algorithms are explored.

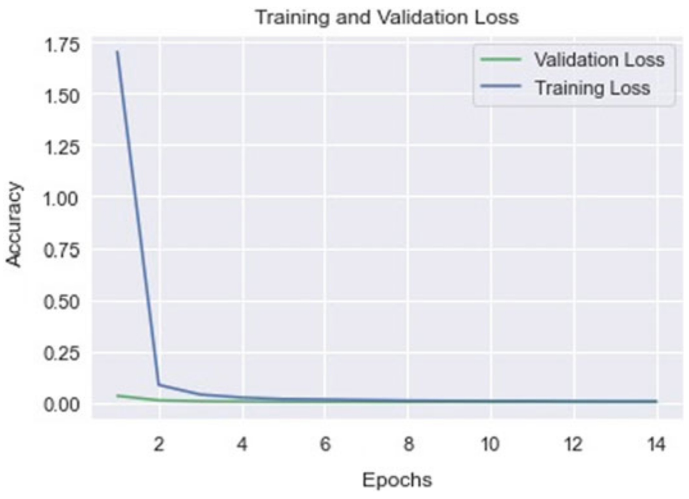


Fig. 10 Training versus validation loss

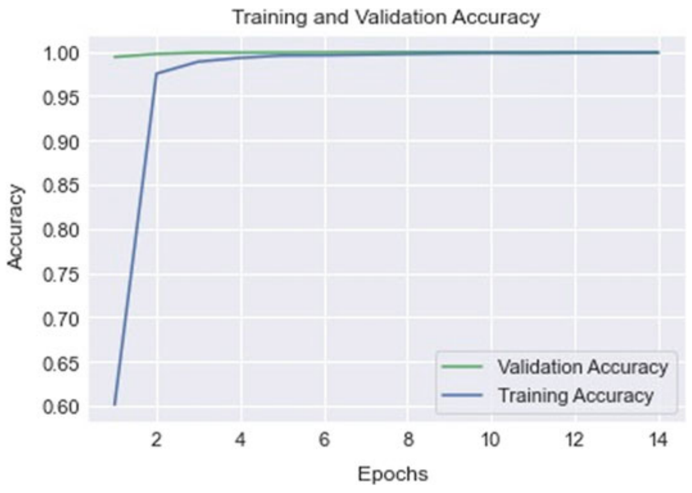


Fig. 11 Training versus validation accuracy

The dataset contains 44 signs with 2400 gestures for each of the 26 alphabets, 0–9 numerals and 08 words. The model was trained on 80% of the data and tested on 20% of it. The research outlines our method for translating sign language into equivalent text, which has a 99.92% accuracy after 15 epochs. Several variables contribute to the excellent accuracy, beginning with pre-processing the dataset images and effectively applying augmentation to the images. By expanding the quantity of the dataset, the system can predict a greater number of gesture signs.

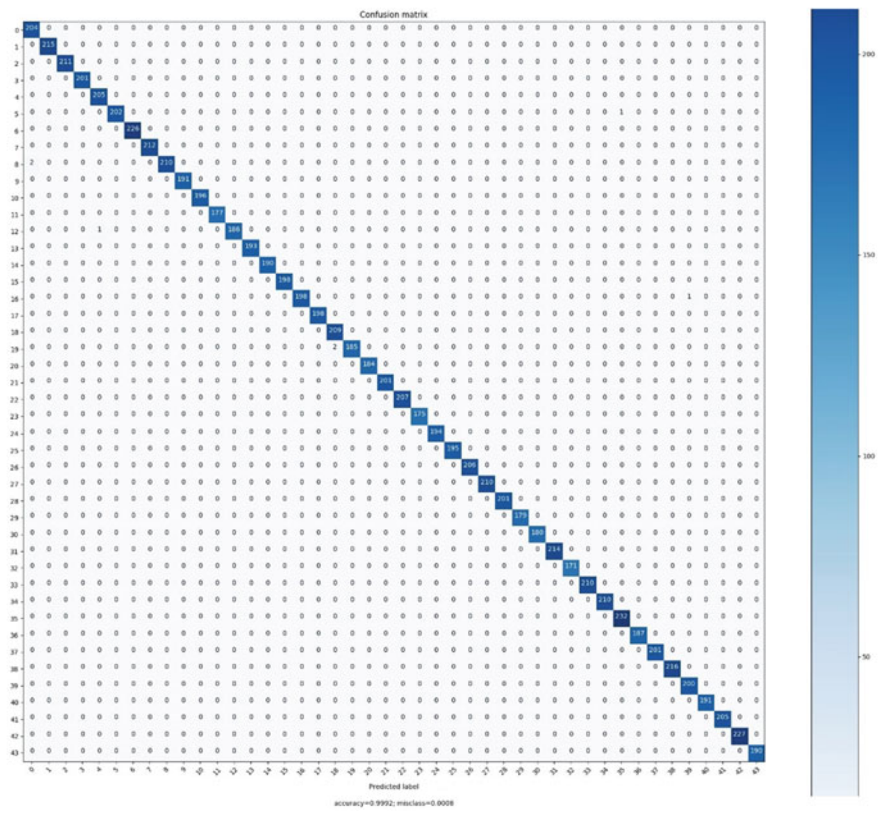


Fig. 12 Confusion matrix

References

1. Tao Y, Huo S, Zhou W (2020) Research on communication APP for deaf and mute people based on face emotion recognition technology. In: 2020 IEEE 2nd international conference on civil aviation safety and information technology (ICCASIT, 2020, pp 547–552. https://doi.org/10.1109/ICCASIT50869.2020.9368771
2. Narayana P et al (2018) Gesture recognition: focus on the hands. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 5235–5244
3. Hossen MA et al (2018) Bengali sign language recognition using deep convolutional neural network. In: International conference on informatics, electronics & vision. International conference on imaging, pp 369–373
4. Dieleman S et al (2014) Sign language recognition using convolutional neural networks. In: ECCV 2014 workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham.https://doi.org/10.1007/978-3-319-16178-5_40
5. Cheng W et al (2014) Jointly network: a network based on CNN and RBM for gesture recognition. Eur Conf Comput Vis ECCV 572–578
6. Yashnarang et al. (2021) Indian sign language to text conversion in real-time using machine learning. Int J Eng Appl Sci Technol (IJEAST) 2455–2143

7. Rajendran R et al (2021) Finger spelled signs in sign language recognition using deep convolutional neural network. Int J Res Eng Sci Manage 4(6):249–253
8. https://Www.Sciencedirect.Com/Science/Article/Abs/Pii/S0020025516307897
9. Beena MV, Agnisarmannamboodiri MN et al (2017) ASL numerals recognition from depth maps using artificial neural networks. Middle-East J Sci Res 25(7):1407–1413. ISSN 1990–9233