Sign Language Recognition System For Differently Abled Person

Harish.P and Jagadeeswar.M
Department of Information Technology
Meenakshi Sundararajan Engineering College, kodambakkam, Chennai.

Abstract

Nonverbal communication, such as sign language, uses outward body movements to convey key messages either simultaneously and concurrently with spoken words or in place of speech. The hands, face, or other body parts can move when using the performance of different neural network models in this area to obtain balanced neural network models, which not only recognize sign languages with good accuracy but also runs on lower-end embedded systems and smartphones with acceptable speed. This would help develop real-time and portable sign language. In contrast to gestures, which convey a specific language recognition system. Messages, pure expressive displays, proxemics, and displays of shared attention are examples of physical nonverbal communication. Culture-specific gestures can communicate significantly different meanings in various social or cultural contexts. The goal of this research is to develop a Deep Learning algorithm that can categorize photos of various sign languages, including alphabetic and numeric images. The accuracy of hand gesture types classification using CNNs is higher than that of the suggested and existing algorithms, according to a comparison it gives more accuracy than the ANN algorithm where ANN doesn't contain any hidden layers so the accuracy will be lesser. In CNN AlexNet consists of eight layers five convolutional layers, two fully-connected hidden layers, and one fully-connected output layer. Whereas the AlexNet in ANN contains only eight convolution layers. It increases the accuracy.

I.INTRODUCTION

Sign language is a way of communicating by using the hands and other parts of the body. According to the World Health Organization, around 466 million people worldwide have disabling hearing loss and often use sign language for communication[1]. Sign languages differ from country to country, and we mainly examine American Sign Language (ASL) in this paper. ASL is expressed by movements of the hands and face and is the primary language of North Americans who have disabling hearing loss [2]. Still, there will be problems when these people try to communicate with ordinary people who are unfamiliar with ASL. As a result, an automatic and real-time sign language interpreter is not only necessary but also in high demand for people with and without hearing loss alike [3]. When Deaf people constitute a relatively small proportion of the general population, Deaf communities often develop that are distinct from the surrounding hearing community. These Deaf communities are very widespread in the world, associated especially with sign languages used in urban areas and throughout a nation, and the cultures they have developed are very rich. One example of sign language variation in the Deaf community is Black ASL. This sign language was developed in the Black Deaf community as a variant during the American era of segregation and racism, where young Black Deaf students were forced to attend separate schools from their white Deaf peers. The expressions in ASL are difficult to recognize with conventional algorithms. However, with the help of convolutional neural networks, we can recognize and translate sign language more efficiently [4]. In this paper, we analyzed

II.RELATED WORK

Gesture recognition is a competitive area of research. It may include different approaches to recognizing the gesture such as the usage of sensory hardware. But using hardware is costly and less convenient in real life. Thus, using computer vision techniques, researchers are trying to get the best recognition accuracy. The performance of different glove-based and vision-based sign language detection was compared by Sruthi C. J. and A. Lijiya [5].

Deep Learning techniques are easier to use and advantageous as they can extract the features from the raw database. The depths of the Convolutional Neural Networks and the combination of CNN layers alter the performance of the system. Gesture recognition modules have been designed to recognize the sign from a selfie video taken on mobile phones [6]. A new dataset is created for Indian Sign Language with five different subjects performing 200 signs in 5 different viewing angles. The CNN consists of 4 convolution layers. Each convolution layer has a different filter size which improves speed and accuracy. A stochastic pooling layer is introduced for better feature extraction as it sums up the advantages of max and means pooling. Surely this approach provides a platform for visual interaction to deaf and mute users but may put limitations on performing the signs as the user has to hold the phone in one hand.

Skin detection algorithms help remove unnecessary background and this increases the model's accuracy [7]. Though the skin detection process eliminates the noise in the image, it also elongates the recognition time. The use of fast GPUs can help overcome this issue but will result in a costly system.

Skin models are designed to eliminate the effects of non-uniform illuminance [8]. Also, this skin model followed by the rectification of the hand orientation and obtaining a binary image with the Gaussian Mixture Model, further increases the validation accuracy of the CNN.

As the research in sign language recognition started encouraging scientists and students to implement their own CNN architecture or to use the Transfer Learning approach, there is a great demand for the American Sign Language database. Some Computer Vision Scientists and laboratories

III.PROPOSED METHOD FOR GESTURE RECOGNITION

As researchers prefer to use Convolutional Neural Networks for feature extraction from gesture datasets, after investigating the existing related designs and studying the available alternatives given several constraints, we can implement a design that is based on a combination of image pre-processing and deep learning to achieve our sought-after objectives. It is easier for users to simply capture a photo of a hand sign and feed it to the system. The task of feature extraction is performed by the CNNs only. Real-time prediction accuracies can be improved by trying transfer learning, and CNN fine-tuning methods.

A.System Description

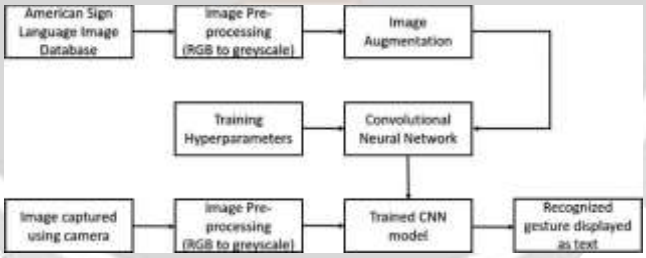


Figure 1. System Block Diagram

The model of the neural networks used in this paper can be described as a multilayer perceptron. A block diagram of the system is shown in Fig. 1. The description for each block is given below.

- Image Database: The database contains images of different hand signs. These images are taken from different users with multiple repetitions. The resolution of the images may be varying. Different datasets are available for American Sign Language.

- Image Pre-processing: Training the raw images as it is might lead to poor performance. Thus, simple image processing algorithms can be implemented to achieve maximum accuracy. Image processing algorithms such as RGB to gray conversion reduce the training time and power consumption. The noise from the images can be eliminated.

- Image Augmentation: Data augmentation helps in the case of a small database. Image augmentation is achieved by doing various operations, including Mirroring – Flip the image horizontally; Cropping – Cutting out a

certain portion of an image; Rotating, shearing, and local warping; Color shifting – For the RGB dataset, the pixel values can be modified.

- CNN Training & Training Options: Deep learning is used for the project. Training options are set accordingly before training the database using any CNN architecture. The training options are maximum batch size, number of epochs, and learning rate.

- Image Acquisition: Any camera, even a laptop webcam can be used to acquire the image to be recognized. Because in the end the image captured will be reduced to the input size of the CNN. Hence the camera need not be high-resolution.

- Display Output: The recognized sign can is displayed in text format or can be also conveyed with an audio description.

B. Hardware Components

As the input to the system is in the form of an image, a camera is required to capture the photograph of the plant. There are no specific requirements regarding the camera. Any good quality and resolution camera can be used to take input. The resolution of the camera can be kept as low as possible.

C. Software Libraries

- OpenCV: to capture and process the images.
- TensorFlow and Keras: to train the plant dataset.
- NumPy: to hold a group of images to test the model.

IV.IMPLEMENTATION

A.Database

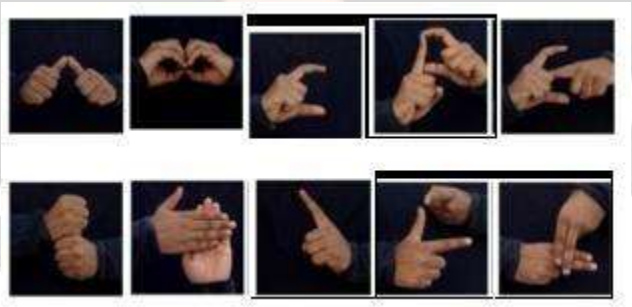


Figure 2. Sample Images from the American Sign Language Dataset [10]

The biggest challenge we faced during the experiments was to find a suitable and sufficient sign language dataset. We used the American Sign Language dataset made available publicly by Turkey Ankara Ayrancı Anadolu High School which comprises approximately 205 images per class (see Fig. 2) [10]. The dataset consists of hand gestures for 0 to 9 digits. The images are in RGB format with a resolution of 100x100.

B. Image Preprocessing

After converting the RGB image dataset to grayscale, three image processing algorithms were implemented and compared for better results. The first method is simple thresholding, for which the source image should be grayscale.

The threshold value is used to classify the pixel values in only two categories- 0 and 1. The pixel values are compared with a pre-defined threshold value by the programmer. The outputs of simple thresholding are binary images.

The second method is Otsu's thresholding, which automatically calculates a threshold value from an image histogram. It is used to perform automatic image thresholding. In the simplest form, the algorithm returns a single intensity threshold that separates pixels into two classes, foreground, and background. The result is an image with a bimodal distribution. The bimodal distribution has two peaks, which are generated by combining two normal distribution curves.

The last method is Canny edge detection. It is a five-stage algorithm [11]: Noise Reduction (use gaussian filter to smooth the image and remove the Noise), Gradient calculation (find t

intensity gradients of the image), Non-maximum Suppression (apply Non-Maximum suppression to get rid of spurious response to edge detection), Double threshold (determine the potential edges) and Hysteresis Thresholding (finalize the detection of the edges by suppressing all the other edges that are weak or not connected to strong edges). As these discussed methods didn't yield correct output images for the entire data, we decided not to make any changes to the obtained grayscale image database.

C. Convolutional Neural Networks

Convolutional neural networks are specifically designed to make inferences from visual data such as images and videos. The features are extracted and learned to train the model, which gives better recognition accuracy compared to conventional Machine Learning algorithms. CNNs have numerous applications in the field of Signal Processing, robotics, medical imaging, data analysis, Business Intelligence, etc. The learning from the unaltered and smaller dataset with CNNs yields surprisingly better results.

CNN architecture is built with a combination of several layers which can be referred to as functional units of deep learning. The architecture takes input data, with predefined hyperparameters, and learns from the data to decide the values of weights and biases. The accuracy or loss is checked after each iteration of the learning process against a part of the original data set aside before training, called a validation dataset. Descriptions for layers used in the experiments are given below:

Convolution layer- Convolution layers extract features from the fed images. The feature maps are obtained by applying filters on the image termed convolution filters. The number of feature maps is equal to the number of filters. These filters are in the form of 3x3, 5x5, etc. matrices. The feature maps go through the activation function before feeding to the next layer. ReLU is one of the widely used activation functions.

- Pooling Layer: The pooling layer reduces the size of the images by taking the maximum pixel value or the average pixel value from a group of pixels. Here the pooling areas or windows do not overlap on image regions. Pooling layers are useful for reducing computational loads.

- Flatten Layer: Output nodes from the previous layer are taken and separated or flattened and weights are assigned to these individual nodes. Thus, the array or tensor of nodes is reshaped by the flatten layer.

- Dense Layer: The output shape of the dense layer is affected by the number of units specified in the code. It is a regular NN layer that simply applies activation and gives output.

- Dropout Layer: CNN architecture might undergo overfitting in which the model gets trained specifically to given training data and thus fails on any new data. Dropout is the solution to avoid overfitting in which involves the elimination of randomly selected nodes from each iteration learning process.

V.RESULTS

A.Comparison of results obtained for existing CNN architectures

Table I shows the results of the transfer learning approach performed with a 10-class dataset. The raw image dataset was fed to CNNs. The weights taken were learned on the ImageNet database. The last output layer of the MobileNetV2 was modified for our data. The massive difference between training and validation accuracy implies overfitting. Fig. 3 shows the training graph for MobileNetV2.

Overfitting refers to the model trained perfectly on the 'training' subset of the data. It fails to accurately generalize the test data. Thus, though the training accuracy has reached almost unity, the model fails on the validation dataset.

Thus, after seeing the failure with the above approach, we decided to train some CNN architectures on our dataset from scratch. The weights were randomly initialized. Table I also shows the results of training Vgg16 and LeNet-5.

Features of Vgg16 [12]:

- It uses only small 3x3 filters with a stride of 1.
- There are a total of 16 layers with a stack of convolution and max-pooling layers and 3 fully-connected layers and softmax layers at the end.
- There are approximately 138 parameters to be trained and a huge amount of memory is required (96 MB/image).

On the other hand, LeNet-5 has only 5 layers [13]. Thus, as seen in Table I, both Vgg16 and LeNet-5 architecture are not suitable for our data. Thus, a conclusion can be drawn regarding the depth of the required CNN architecture. It should have a moderate depth between Vgg16 and LeNet-5.

TABLE I. RESULTS FOR OTHER CNN MODELS

B.Architecture of designed CNN Model

A model having a moderate number of layers is designed. The final architecture that worked better on the data is shown in Fig. 4. The CNN has 10 layers with 3 convolution layers, 3 max-pooling layers, flatten, dropout, and 2 dense layers. The use of the dropout layer eliminates or skips some randomly selected nodes to avoid overfitting. Generally, a small dropout value of 20%-50% of neurons is used for experimental purposes. Table II shows the training and validation accuracies with different dropout values.

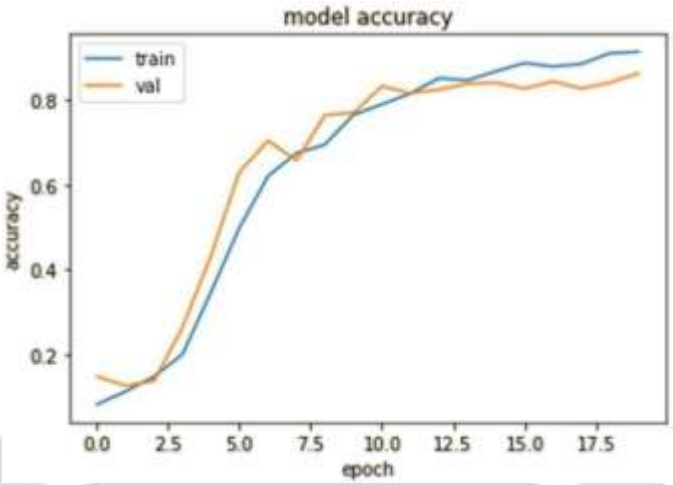
Thus, we took out the test dataset from the whole data. 20 images per class, i.e. a total of 200 images were set out for testing the model. This gives a training accuracy of 91.37%, a validation accuracy of 86.30%, and a testing accuracy of 87.50% as shown in Fig. 5. The other settings are the same as above.

Figure 5. Model Accuracy and Model Loss for Selected CNN Architecture

Thus, a training accuracy of 91.37% and validation accuracy of 86.30% was achieved. Table III shows the obtained confusion matrix.

Figure 4. Selected CNN Architecture with 10 Layers TABLE II. RESULTS FOR OUR CNNMODELa

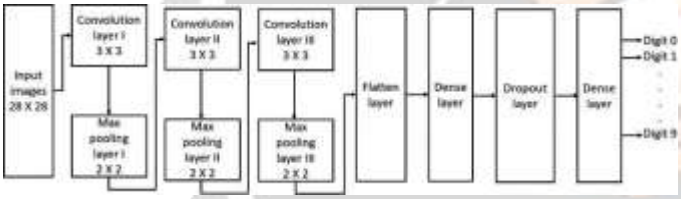
TABLE III. Model Accuracy



a. Settings: Validation split = 0.2, optimizer = SGD(LR=0.01), batch size = 64, the number of epochs = 20

Thus, we see the validation accuracy remains in the range of 90 (±2%).

As of now, we have not kept out the test dataset and have not evaluated the test accuracy for any of the models. The training



CONCLUSION

Data is a group of images sent through the CNN to train the weights and biases of the model. Validation data is another section of the whole dataset used to verify the accuracy of the model. Thus, after each iteration of training, the trained model is evaluated against this validation dataset. Test data is used to evaluate the entire model and it is never touched during the training process. This data gives the final accuracy of the entire CNN architecture.

It focused on how the image from the given dataset (trained dataset) in the field and past data sets were used to predict the pattern of different hand gestures using the CNN model. This brings some of the following different gestures predictions. We applied different types of CNN to compare the accuracy and saw that AlexNet makes the better classification and the .h5 file is taken from their application or desktop application. To optimize the and that is deployed in Django framework for better user interface.. If more feature extraction techniques are added to support the CNN method and correctly categorize different fruit types on images, it is projected that the success of the results would grow. Sign Language prediction to connect with AI model. To automate this process by showing the prediction results in web work to implement in Artificial Intelligence environment applications and embedded systems, increasing recognition speed while keeping the size and complexity of the neural network model at an acceptable level.

REFERENCES

[1] R. Cui, H. Liu, and C. Zhang, “A deep neural framework for continuoussign language recognition by iterative training,” IEEE Transactions on Multimedia (TMM), vol. 21, no. 7, pp. 1880– 1891, 2019.

[2] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[3] Z. Zhang, J. Pu, L. Zhuang, W. Zhou, and H. Li, "Continuous sign language recognition via reinforcement learning," in International Con- ference on Image Processing (ICIP), 2019.

[4] H. Wang, X. Chai, and X. Chen, "A novel sign language recogni-tion framework using hierarchical grassmann covariance matrix," IEEE Transactions on Multimedia (TMM), vol. 21, no. 11, pp. 2806–2814, 2019.

[5] J. Pu, W. Zhou, and H. Li, "Dilated convolutional network with iterativeoptimization for continuous sign language recognition." in International Joint Conference on Artificial Intelligence (IJCAI), 2018.

[6] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: End- to-end hand shape and continuous sign language recognition," in IEEE International Conference on Computer Vision (ICCV), 2017.

[7] N. Cihan Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen, and M. Zhou, "Sign language recognition and translation with kinect," in International Conference on Automatic Face and Gesture Recognition (FG), 2013.

[9] A. Orbay and L. Akarun, "Neural sign language translation by learning tokenization," arXiv preprint arXiv:2002.00479, 2020.

[10] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and transla- tion," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10 023–10 033.

[11] K. Yin, "Sign language translation with transformers," arXiv preprint arXiv:2004.00588, 2020.

[12] A. Schembri, J. Fenlon, R. Rentelis, S. Reynolds, and K. Cormier, "Building the british sign language corpus," Language Documentation& Conservation, vol. 7, pp. 136–154, 2013.

[13] https://www.awhamburg.de/en/research/long-term- scientific-projects/dictionary-german-sign-language.html.