Sign Language Recognition System Using Deep Neural Network

Surejya Suresh
Department of Electronics
Cochin University of Science and Technology
Kochi, India
surejya@cusat.ac.in

Mithun Haridas T.P.
Department of Electronics
Cochin University of Science and Technology
Kochi, India
mithuntp@cusat.ac.in

Supriya M.H
Department of Electronics
Cochin University of Science and Technology
Kochi, India
supriya@cusat.ac.in

***Abstract*— In the current fast-moving world, human-computer- interactions (HCI) is one of the main contributors towards the progress of the country. Since the conventional input devices limit the naturalness and speed of human-computer- interactions, Sign Language recognition system has gained a lot of importance. Different sign languages can be used to express intentions and intonations or for controlling devices such as home robots. The main focus of this work is to create a vision based system, a Convolutional Neural Network (CNN) model, to identify six different sign languages from the images captured. The two CNN models developed have different type of optimizers, the Stochastic Gradient Descent (SGD) and Adam.**

Keywords—Sign Language Recognition, Convolutional Neural Network, SGD, Adam

I. INTRODUCTION

The rapid progress being achieved in the field of information technology leads computer systems to play a vital role in the day-to-day life of human beings. In the current fast-moving world, human-computer- interactions (HCI) is one of the main contributors towards the progress of the country. Input devices such as, keyboards, mouse, joysticks, etc. are the typical HCI modules forming the means of interaction. Since they limit the naturalness and speed of human-computer- interactions, Sign Language recognition system has gained a lot of importance. A prototype of the sign recognition model has been implemented using the Convolutional Neural Network and tested successfully. The primary goal of the system envisages identifying a specific Sign Language and using it to convey information or to control a device. The architecture, results and analysis are discussed in detail in this paper.

The recent advances in understanding human visual processing, clearly reveals the hierarchical fashion of the human perception. Deep learning techniques, a subset of machine learning, using the model of human visual cortex is the most acceptable method for Sign Language than any of the traditional image processing techniques. Understanding the biological motivation behind the human visual cortex system revealed that the primary features from any visual data are transformed into significant features in the upcoming layers. The architecture of Convolutional Neural Network, a deep learning algorithm, is very much analogous to the connectivity pattern underlying in the visual cortex system.

A prototype of the sign recognition model has been implemented using the Convolutional Neural Network and tested successfully. The architecture, results and analysis are discussed in detail in this paper.

II. RELATED WORK

G. A. Rao *et al* [1] has developed a sign language recognition approach based on CNN for a dataset consisting of 200 different sign languages viewed from 5 different angles with different background environments and achieved 92.88% recognition accuracy.

Jie Huang *et al* [2] used 3D Convolutional Neural Network (3D CNN) and extracted discriminative spatial and temporal features from a set of video datasets and observed to have an accuracy of 94.21%.

Static hand gesture classification model by extracting hand features through background elimination method was tried for a dataset of 24 classes comprising the sign language alphabets of Peru [3]. The model attempted to exploit the power of CNN together with image processing techniques to extract hand gesture features and found to have 96.20% accuracy under various noisy and illumination conditions.

A classification problem of three types of hand gestures, labelled as open, closed and un-known, was performed with different architectures of CNN by varying the hyperparameters [4]. On comparison, the best model which uses a matching layer was found to give a classification accuracy of around 73.7%.

A hand gesture-based system to support intelligent vehicles was attempted to develop using deep learning models to increase comfort in driving [5]. This recognition system resulted in 91% accuracy for classification and performs well in real time with a latency of 110ms.

III. PROPOSED SIGN LANGUAGE RECOGNITION ALGORITHM

A framework of CNN and description of the proposed architecture is discussed below. The proposed architecture has been analyzed using two models, the first model using the Stochastic Gradient Descent (SGD) optimizer (M_1) and the second model using Adam optimizer (M_2).

A. Convolutional Neural Network

Neural Network performs as a Universal function approximator that learns the input-output relations by tuning the internal free parameters. CNN is modelled, inspired from the mammalian visual cortex and the hierarchical feature extraction from the image is incorporated into the neural network by exploiting the convolution operation [6], [7].

The great success have been achieved in different areas such as Image Classification, Pattern Recognition, etc. by the application of CNN. CNN may contain several stacks of

978-1-5386-9533-3/19/$31.00 ©2019 IEEE

Convolutional layer, Pooling layer, Dropout layer and Fully Connected layer. The CNN layers are designed in such a way that all relevant features are extracted by bottom layers and abstract features are computed by high level layers. The spatial and local correlation of pixels is extracted and thus derived feature-map is mapped to the higher layer in CNN.

The values of the feature-map derived from the convolutional layer (x,y) position can be calculated as equation (1)

$$f_{ij}^{xy} = \Phi \left\{ b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ijm}^{pq} f_{(i-1)m}^{(x+p)(y+q)} \right\} \quad (1)$$

where f_{ij}^{xy} is the value of j^{th} feature-map in the i^{th} layer at (x,y) position, Φ is the activation function, b_{ij} is the bias corresponding to the feature-map, m is the index of the set of feature-maps in the $(i-1)^{th}$ layer, P_i x Q_i is the dimension of the kernel, w_{ijm}^{pq} is the kernel value at (p,q).

To reduce computation time and to reduce the size of the feature, following the convolutional layer, a pooling layer is used where max function computes the high-level feature in a local window.

B. *Proposed Model Architecture*

The architecture of the proposed CNN model with the configurations as given in Table I is depicted in Fig. 1.

TABLE I. MODEL CONFIGURATION

Layers	Configuration
Convolution (C_1)	32filters, 3x3 kernel, ReLU
Pooling (P_1)	2x2 kernel
Dropout (D_1)	0.2
Convolution (C_2)	32filters, 3x3 kernel, ReLU
Pooling (P_2)	2x2 kernel
Dropout ((D_2)	0.2
Flatten (F)	6272 Neurons
Dense (FC_1)	256 Neurons
Dense (FC_2)	6 Neurons

In this architecture, the input layer receives the resized image of dimension 64 x 64 pixels. A series of two convolution operations are then applied to the image to extract the features. The first convolution layer ($\boldsymbol{C_1}$) uses 32 kernels of size 62 x 62 to generate the feature maps and is activated with Rectified Linear Unit (ReLU) [8] function. Max-pooling ($\boldsymbol{P_1}$) layer of window size 2 x 2 is provided to subsample the obtained feature maps for a reduction in size to 31 x 31. Feature Map is then subjected to the application of a *weight vector* to have a 20 percentage dropout ($\boldsymbol{D_1}$) so that the possibility of overfitting of the network is avoided. The second convolution layer ($\boldsymbol{C_2}$) follows the same procedure with 32 kernels of size 29 x 29 and ReLU activation. The Max-pooling ($\boldsymbol{P_2}$) subsampled with window size of 2 x 2 and a 20 percentage dropout ($\boldsymbol{D_2}$) yields 32 feature maps of size 14 x 14. These feature maps are then flattened(F) resulting in a feature vector with a dimension of 1 x 6272. The first fully collected layer or the dense layer (FC_1) with activation function of ReLU works on this feature vector resulting in a 1 x 256 vector. The second dense layer (FC_2) using *softmax* activation, produces the probabilistic output corresponding to the six classes of sign languages.

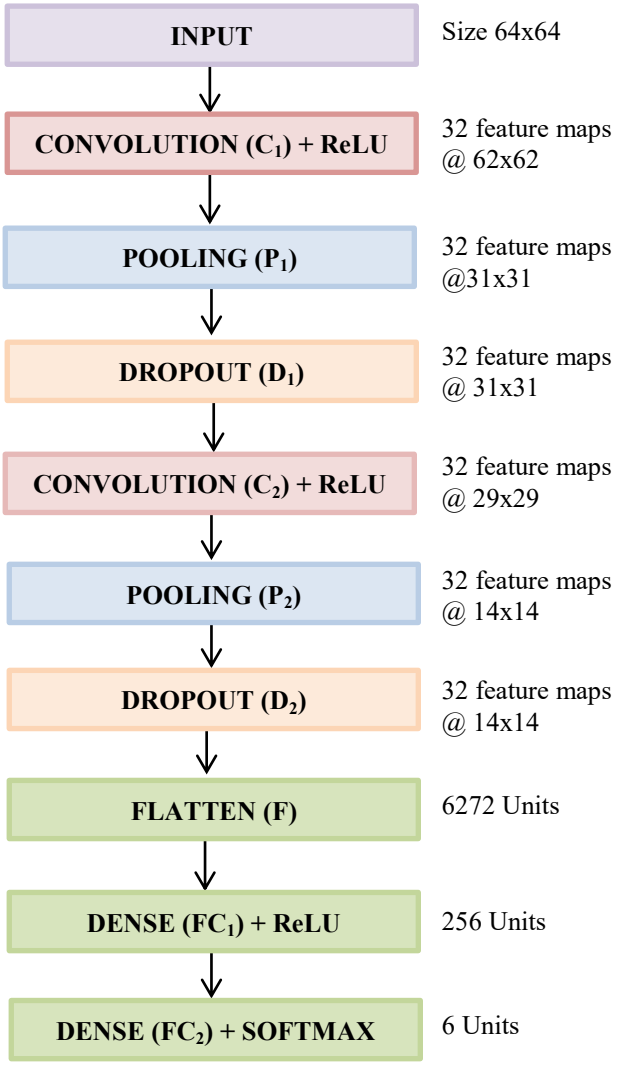


Fig. 1. Architecture of the proposed sign language recognition model using CNN

IV. METHODOLOGY

Hand gestures of different persons have been considered as the dataset for the prototype. The database has been created using images captured on a 5MP camera with a resolution of 2560 x 1920 pixels. Finger count gesture for zero, one, two, three, four and five formed the six different sign language classes. Equal amount of different classes of images were used for training to avoid biasing. Validation of the network performance was also measured along with the training phase to ensure proper learning. Out of the 3060 samples collected, 2448 samples were considered for training (80%), 306 for testing as well as validation as shown in Table II. The images have been resized to 64 x 64 for reduction in computational complexity without loss of relevant information. A sample dataset is as displayed in Fig. 2.

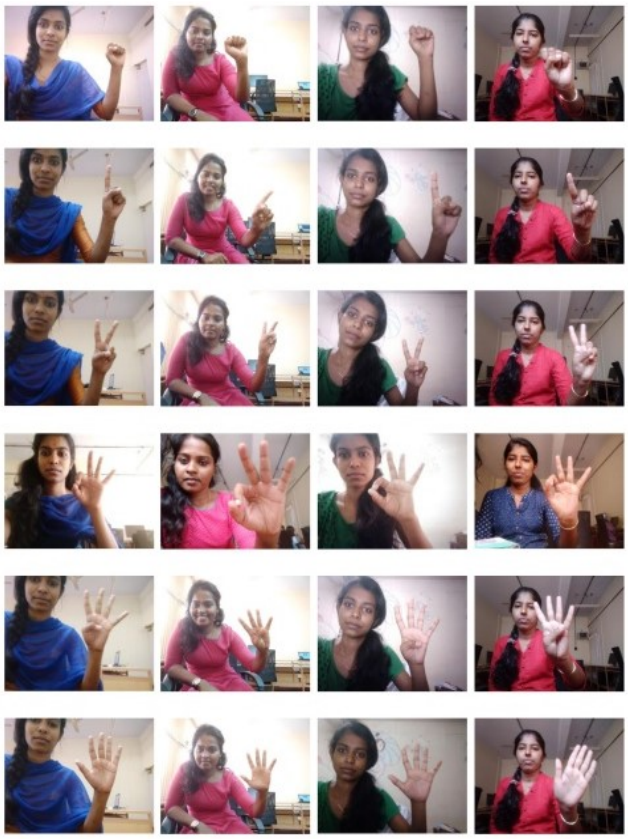


Fig. 2. Dataset showing the 6 sign languages

TABLE II. DATASET SPLIT DETAILS

Dataset	No. of Classes	No. of training images	No. of validation images	No. of testing images
Sign Language Dataset	6	2448	306	306

Python is the programming language used for implementing sign language recognition model. The Jupyter Notebook is the environment used for coding and analysis. The libraries and packages used include keras, scikit-learn, matplotlib, numpy.

Keras is a python deep learning library. Keras library is used to build the sequential network model and to add the convolutional layers in the model. Sckit-learn which is used here for plotting the confusion matrix. The accuracy plot and loss plot are plotted using the matplotlib, which is a Python 2D plotting library. NumPy is the fundamental package for scientific computing with Python is also used for varies array operations.

V. RESULTS AND DISCUSSIONS

The CNN is trained with the optimization based on the gradient descent method. The classification was performed and compared by creating two different models. The SGD optimizer for model 1(M_1) and Adam optimizer for model 2(M_2) are used for optimizing, where the cost function used is Categorical Crossentropy. Table III shows the parameters used in both the models.

Optimization problem is to minimize the cost function for finding the final weights for the network.

$$\min_{w} J(w)$$

where J(.) is the loss function and w is the weight vector.

SGD is an optimization method which uses a fixed learning rate and the weight updation is done in every iteration as described in equation (2)

$$w_k = w_{k-1} - \eta \nabla J(w_{k-1}) \quad (2)$$

Adam, Adaptive Moment Estimation, computes adaptive learning rates for each weight [10]. Adam uses exponentially decaying average of past gradients, m_k (first moment) and past squared gradients, v_k (second moment) as given in equation (4) and (5) respectively. Adam weight update equation can be mathematically represented as equation (5)

$$m_k = \beta_1 m_{k-1} + (1 - \beta_1) \nabla J(w_k) \quad (3)$$

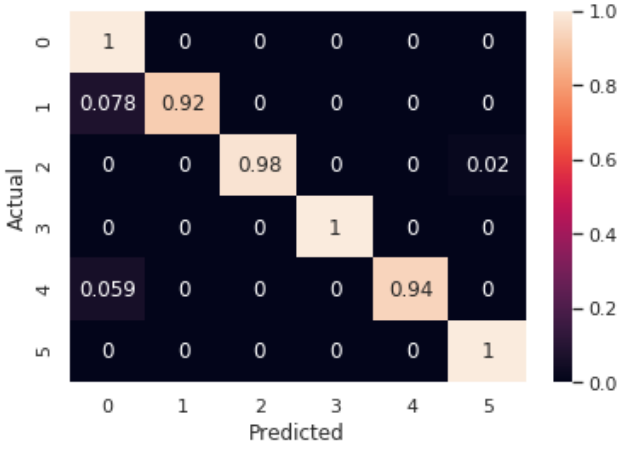
$$v_k = \beta_2 v_{k-1} + (1 - \beta_2) \nabla^2 J(w_k) \quad (4)$$

$$w_{k+1} = w_k - \eta \cdot \frac{\sqrt{1-\beta_1^k}}{1-\beta_2^k} \cdot \frac{m_k}{\sqrt{v_k} + \varepsilon} \quad (5)$$

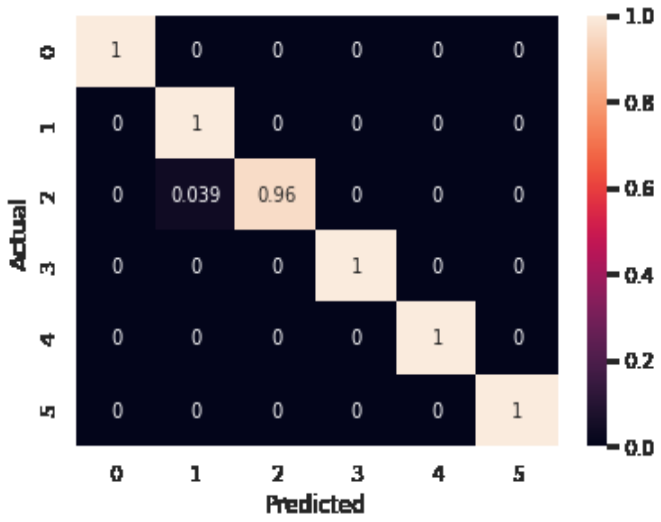
where η is the learning rate, w_k is the weight vector and β_1 $\beta_2 \in [0, 1)$ is a momentum parameter. $(1-\beta_1^k)$ and $(1-\beta_2^k)$ are provided to add the bias correction term for m_k and v_k.

..TABLE III. MODEL PARAMETERS

Network	Model 1	Model 2
Total no. of images	3060	3060
Activation	ReLU and Softmax	ReLU and Softmax
Learning rate	0.01	0.01
Epochs	21	21
Optimization	SGD	Adam
Cost function	Categorical Crossentropy	Categorical Crossentropy



(a)

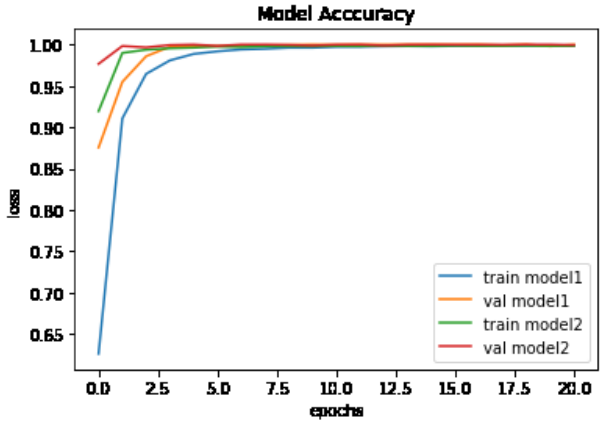


(b)

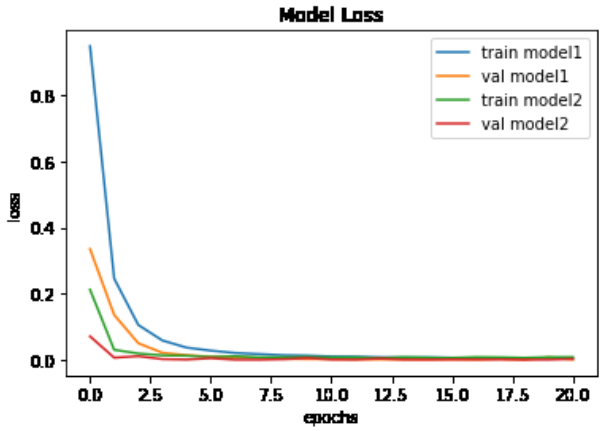
Fig. 3. Normalized Confusion matrix (a) model 1 using SGD optimizer, (b) model 2 using Adam optimizer

Fig. 3. depicts the normalized confusion matrix of model 1 and model 2 when adopting CNN with SGD optimizer and Adam optimizer respectively.

Fig. 4(a) and 4(b) depicts the model accuracy as well as training loss corresponding to both the models respectively. These figuress show that both the models have comparable performance with training and validation datasets which indicates that CNN models are better generalized to the data.



(a)



(b)

Fig. 4. (a) Accuracy of training and testing data both model 1 and model 2, and (b) Loss during the training process of CNN of model 1 and model 2

TABLE IV. CLASSIFICATION RESULTS MODEL1 AND MODEL2

Class	Precision		Recall		Specificity		F1 Score		Accuracy (in %)	
	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2
Zero	0.88	0.93	1.00	1.00	0.97	1.00	0.94	0.96	97.71	98.69
One	1.00	1.00	0.92	0.90	1.00	0.99	0.96	0.95	98.69	98.36
Two	1.00	1.00	0.98	1.00	1.00	1.00	0.99	1.00	99.67	100
Three	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	100	100
Four	1.00	0.98	0.94	1.00	1.00	1.00	0.97	0.99	99.02	100
Five	0.98	1.00	1.00	1.00	0.99	1.00	0.99	1.00	99.67	100

Table IV shows the classification results of Model 1(M_1) and Model 2(M_2) which shows the Accuracy, Precision, Recall Sensitivity, Specificity and F1 Score. There is a good balance between Precision and Recall of the both the models as well. All the actual positive samples of all classes except Two are covered in model 1(SGD Optimizer) whereas only three classes are covered in model 2(Adam Optimizer), which shows the recall sensitivity. Similarly all the actual negative samples of all classes except One are covered in model 1(SGD Optimizer) and four classes are covered in model 2(Adam Optimizer), which shows the specificity of the models

VI. CONCLUSION

This paper proposes a basic 2 layer convolutional neural network (CNN) to classify sign language image datasets. The classifier was found to perform with varying lighting and noisy image datasets. This model has classified 6 different sign languages using two different optimizers, SGD and Adam with an accuracy of 99.12% and 99.51% respectively. More accuracy is obtained when using the Adam optimizer. Future work of this Sign Language Recognition System can be extended to improve the performance by tuning the hyperparameters and implement a sign language recognition system from video sequence using CNN LSTM. This sign language

recognition system can also be made to control certain devices such as home robot.

REFERENCES

[1] G. A. Rao, K. Syamala, P. V. V. Kishore and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), Vijayawada, 2018, pp. 194-197.

[2] Jie Huang, Wengang Zhou, Houqiang Li and Weiping Li, "Sign Language Recognition using 3D convolutional neural networks," 2015 IEEE International Conference on Multimedia and Expo (ICME), Turin, 2015, pp. 1-6.

[3] C. J. L. Flores, A. E. G. Cutipa and R. L. Enciso, "Application of convolutional neural networks for static hand gestures recognition under different invariant features," 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Cusco, 2017, pp. 1-4.

[4] I. Rocco, R. Arandjelovic and J. Sivic, "Convolutional Neural Network Architecture for Geometric Matching," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 39-48.

[5] V. John, A. Boyali, S. Mita, M. Imanishi and N. Sanma, "Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames," 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, QLD, 2016, pp. 1-8..

[6] J. Shijie, W. Ping, J. Peiyi and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," 2017 Chinese Automation Congress (CAC), Jinan, 2017, pp. 4165-4170.

[7] Y. Liu, P. Zhang, "Vision-based Human-Computer System using Hand Gestures", In Proceeding(s) of the International Conference on Computational Intelligence and Security, pp. 529-532, 2009.

[8] A. Krizhevsky, I. Sutskever, and G. E. A Convolutional Neural Network Hand Tracker", in proceedings: Neural Information Processing Systems Foundation. 1995Hinton, "Imagenet classification with deep convolutional neural networks," in Proc. Annual Conference on Neural Information Processing Systems (NIPS), 2012, pp. 1106–1114.

[9] 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), Coimbatore, 2016, pp. 1-5.

[10] Diederik P. Kingma, Jimmy Ba, "Adam: A Method for Stochastic Optimization," 33rd International Conference for Learning Representations, San Diego, 2015