Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-021-00617-x

ORIGINAL RESEARCH

American Sign Language recognition using Support Vector Machine and Convolutional Neural Network

Vanita Jain[1] · Achin Jain[1] · Abhinav Chauhan[1] · Srinivasu Soma Kotla[1] · Ashish Gautam[1]

Received: 29 January 2020 / Accepted: 3 February 2021

Abstract A sign language recognition system is an attempt to help the speech and the hearing-impaired community. The biggest challenge is to recognize a sign accurately. This can be achieved by training the computers to identify the signs. The accuracy depends on the methods used for classification and prediction which is achieved through machine learning. This research proposes the recognition of American Sign Language by using Support Vector Machine (SVM) and Convolutional Neural Network (CNN). In this work we have also calculated optimal filter size for single and double layer Convolutional Neural Network. In the first phase features from the dataset are extracted. After applying various preprocessing techniques, Support Vector Machine with four different kernels i.e., 'poly', 'linear', 'rbf' and 'sigmoid' and Convolutional Neural Networks with single and double layer are applied on training dataset to train the model. Finally, accuracy is calculated and compared for both the techniques. In CNN filters of different sizes have been used and optimal filter size has been found. The experimental results establish that the double layer Convolutional Neural Network achieve an accuracy of 98.58%. Optimal filter size is found out to be **8 × 8** for both single and double layer Convolutional Neural Network. From the experimental results we conclude that accuracy of CNN model can be improved by altering the filter size. This also helps in CNN to learn optimum values for variable sized parameters and tuning of different hyper parameters.

Keywords Sign language recognition · Support Vector Machine · Convolutional Neural Network

1 Introduction

Hand-gesture recognition is the skill of computer to identify hand gestures from sources like images or video feed [1]. A lot of work is being done on gesture recognition, majorly being in the field of computer vision using edge detection and haar-cascade-classifier [2]. Popular sign language involves study of upper body part i.e. from waist level upwards [3]. However, the same sign can show major changes in shape when the location is different [4]. There are many categories of hand gesture such as controlling, communicating, manipulating and conversational gestures [5]. Sign language which is highly structured falls into the category of communicative gestures and most suitable for computer vision [6]. To aid the hand segmentation process, signer used to wear colored gloves or wrist band [7, 8]. There are lots of challenges in gesture recognition and researchers have presented several evaluation criteria to measure the accuracy of hand gesture algorithm in overcoming those challenges. Some of those criteria are real time performance, independence, scalability and robustness of the algorithm [9].

Machine learning is a field that provides the ability to learn with data, without being explicitly coded for the task [10]. In machine learning, classification is the problem of identifying the observations into set of different categories on the basis of existing data used for training the module and later predicting the categorical association of a new observation. Classification is performed using different algorithms i.e. Naive Bayes, Support Vector Machine (SVM), Decision Tree, Neural Networks etc. Support

✉ Vanita Jain
vanita.jain@bharatividyapeeth.edu

[1] Bharati Vidyapeeth's College of Engineering, New Delhi, India

Published online: 25 February 2021

Vector Machines is broadly categorize as supervised learning model which is used for analysis of data using classification and regression techniques. Non-linear classification can also be carried out with SVM by using Kernel trick which is done by mapping low-dimensional input to high dimensional feature space [11]. Artificial Neural Networks aim to replicate the working of biological neural networks by performing "learn" and "test". A Convolutional Neural Network (CNN), is a set of deep, feed-forwarding networks, composed of multiple or singular layer. CNNs are highly efficient at processing visual and two-dimensional data [12].

In this paper, we have used Support Vector Machine [29] with four different kernels for the detection and recognition of American Sign Language gestures. Further Convolutional Neural Network with single and double layer are used to solve the same problem. The paper is organized in six sections. In Sect. 2, we discuss the related work. Section 3 discusses about the methodology used for hand-gesture recognition for Support Vector Machine and Neural Network classification. This section deals with the implementation of the algorithms. Results have been presented in Sect. 4. The paper is concluded in Sect. 5.

2 Related work

In [13] authors have proposed a new approach using Hidden Markov Model (HMMs) to classify trajectory, orientation and resultant shape of sign language. The authors in [14] collected 262 signs from two various signers and average accuracy is calculated using HMMs. The authors achieve accuracy of 94%. In [15] authors mentioned the limitations of HMMs in training models which are based on context dependent. In [16] authors have collected 3D translation and rotation of data using Ascension Technologies Flock of Birds devices. Average accuracy achieved by using a bigram and epenthesis modelling is 95.83%. In [17], authors used pre-trained VGG16 model for American Sign Language (ASL) character recognition. VGG16 is developed by Vision Geometry Group from Oxford and it a vision model that uses Convolution Neural Network and the accuracy achieved is about 96%. In [18] authors have worked on Portuguese Sign Language and have used Kinect Camera to obtain hand gestures features. After feature extraction, the authors have done training and testing of the gestures using multiclass SVM. In [19] the authors have extracted histograms of oriented gradients (HOG) features using novel algorithm. The extracted features are then used to train an ANN model to recognize the hand gestures and actions. Authors in [20] proposed a technique based on shape analysis to recognize hand gestures. They used neural networks to classify among six static hand actions and the accuracy achieved by them is 86.38%. In [21] the authors have proposed a system for Italian Sign Language gesture recognition. The proposed system uses Microsoft Kinect alongside graphic processing unit (GPU) accelerated convolutional neural networks that helped the authors to achieve accuracy of 92% with very high accuracy for 20 actions. In [22] authors have used HOG and SIFT techniques to extract image features. The extracted features and further converted into a single matrix which is used to calculate the input (correlation) for KNN classifier. The authors in [23] used edge-oriented histogram to extract the features. Edge count histogram has been used to represent each input image of hand gestures. Accuracy achieved using multiclass SVM classifier was 93.75%.

In [24] authors have used OpenCV and CNN to build an automated system which converts hand gestures into meaningful words and sentences. In [25] authors have used KETI sign language dataset and proposed a sign language translation system based on human keypoint estimation. Their work achieved an accuracy of 93.28% using neural network model. The authors in [26] applied weakly supervised learning methods on video sign language data. The authors have applied multi-stream HMMs with CNN-LSTM models in each HMM stream. This approach in discovering of attributes that lack the power to be identified on their own. In [27] the authors have proposed Hierarchical Grassmann Covariance Matrix (HGCM) model for sign description in Video Sign Language recognition. The model proposed is tested on real continuous sign datasets as well as on HDM05 (Motion Database) and results shows the effectiveness of the work. The authors in [28] proposed modified long short-term memory (LSTM) model to recognize a sequence of connected gestures. The model is tested on Indian Sign Language and accuracy achieved was 89.5% for isolated sign words.

3 Design and implementation

The paper primarily focuses on different machine learning models which can be used to detect and recognize American Sign Language. To achieve this, two machine learning algorithms Support Vector Machine and Convolutional Neural Network have been used. Comparison analysis of both the methods is done using predicted accuracy as metric.

A. Dataset

For the experimental work, we have used American Sign Language Image Dataset from MNIST Kaggle [31] comprising of 25 classes with approximately 1350 instances in each class. In the dataset, each class refers to a different letter in ASL. Each image in the dataset has dimension of 28 × 28 px. Dataset contains the grayscale pixel values of the images containing the signs. The row in the dataset contains 24 labels from 0 to 24 which represent the letter of American Sign Language and corresponding to that 784-pixel values representing the image containing the sign. Sample of the dataset is shown in Table 1 and snapshot of dataset is shown in Fig. 1.

The proposed model is trained on the dataset by using four parameters as shown in Table 2. The flowchart of SVM classification is depicted in Fig. 2.

B. Convolutional Neural Network

Convolution layers are used to accomplish convolution on input pictures or feature maps from the earlier layer with filters. Generally, the first convolution layer is used to excerpt low-level image features such as edges; while the upper layers can extract composite and task-related features [30]. The proposed CNN model is shown in Fig. 3.

Table 3 briefly describes various parameters used.

4 Results

A. Accuracy using SVM

The accuracy of the model trained by using Support Vector Machine Learning Algorithm using different kernels is presented in Table 4.

A maximum of 81.49% accuracy using ‘Poly’ Kernel for SVM has been achieved and is shown in Fig. 4.

B. Accuracy using single layer CNN for various filter size

Single layer CNN is applied on 28 × 28 images with 32 filters followed by dense fully connected neural network. The single layer CNN model was trained with different filter sizes from 5 × 5 to 14 × 14 to find out the optimal filter size. The optimal value of filter size for 32 filters is found to be 8 × 8 where minimum error is 2.656%. Table 5 shows the accuracy and error obtained for various filter sizes obtained by exhaustive search.

Figure 5 shows the behaviour of our model on training and testing data. It can be seen from Fig. 5 that the

Table 1 Sample of the dataset

	Label	Pixel1	Pixel2	Pixel3	..	Pixel784
1	3	107	118	127	..	202
2	6	155	157	156	..	149
..	..	..	..	..	..	..

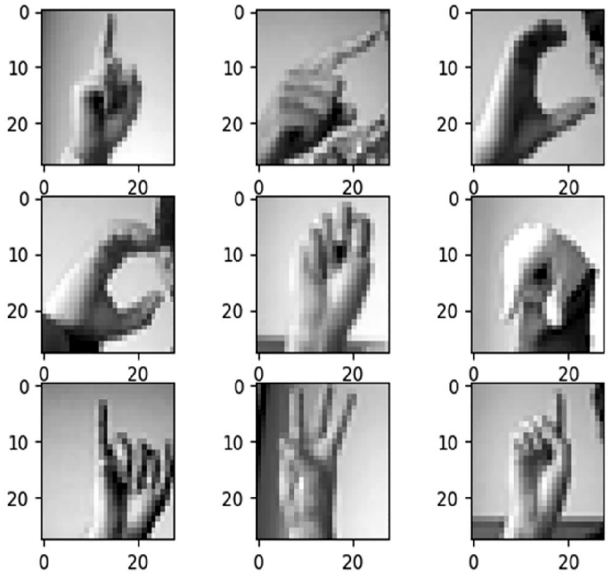


Fig. 1 Snapshot of data

Table 2 Parameters used in model training

Parameter	Value
C	200
Gamma	0.01
Cache Size	8000
Probability	False

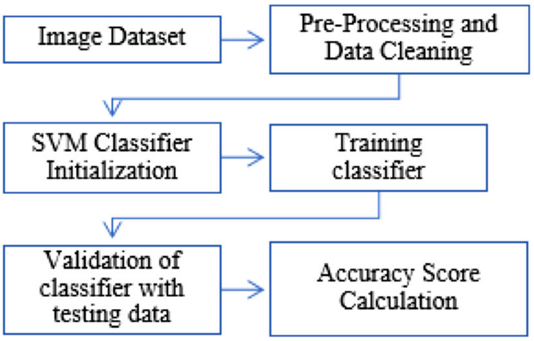


Image Dataset
Pre-Processing and Data Cleaning
SVM Classifier Initialization
Training classifier
Validation of classifier with testing data
Accuracy Score Calculation

Fig. 2 Flowchart of SVM classification

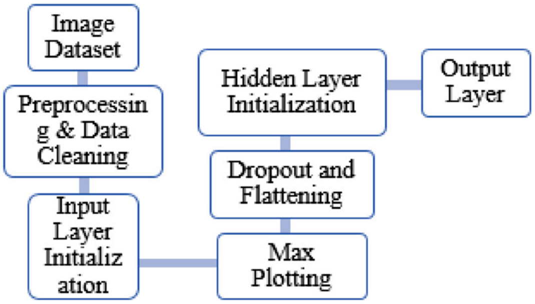


Image Dataset
Preprocessin g & Data Cleaning
Input Layer Initializ ation
Max Plotting
Dropout and Flattening
Hidden Layer Initialization
Output Layer

Fig. 3 Flowchart of CNN classification

Table 3 Parameter used

Parameter	Value
Input layer	Relu activation function with 32 filters
Pooling layer	Max pooling with size equal (2, 2)
Hidden/fully connected layer	Relu activation function with 128 neurons
Output layer	Softmax activation function with 25 neurons
Metrics used	Accuracy, top categorical accuracy, F measure and recall
Loss function	Categorical crossentropy
Optimizer	Adam
Epochs	12
Batch size	32

Table 4 Accuracy of SVM model in different kernels

Kernel name	Accuracy of SVM (%)
Linear	80.53
Poly	81.49
Rbf	64.09
Sigmoid	13.78

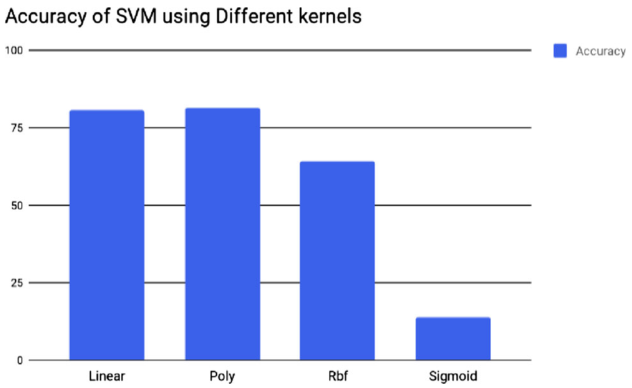


Fig. 4 Accuracy of SVM kernels

Table 5 Accuracy in different filter size

Filter size	Accuracy	Error	Top_k_ca
5 × 5	96.189	3.811	0.99393
6 × 6	97.193	2.807	1.00000
7 × 7	96.390	3.610	0.99682
8 × 8	97.344	2.656	0.99896
9 × 9	96.733	3.267	0.99653
10 × 10	96.648	3.352	0.99552
11 × 11	95.186	4.814	0.99364
12 × 12	96.131	3.869	0.99711
13 × 13	96.001	3.999	0.99697
14 × 14	96.120	3.880	1.00000

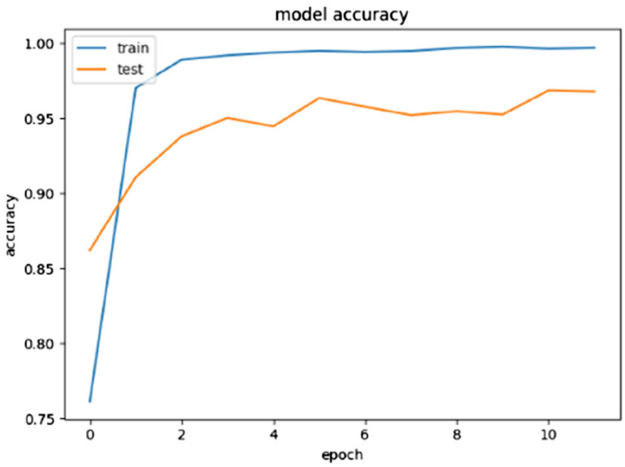


Fig. 5 Model accuracy of CNN

accuracy of the testing data is correctly converging to the accuracy of the training data to avoid underfitting and overfitting. Figure 6 shows the behaviour of our proposed model on training and testing data. It can be seen that the loss of the testing data is correctly converging to the loss of the training data. Figure 7

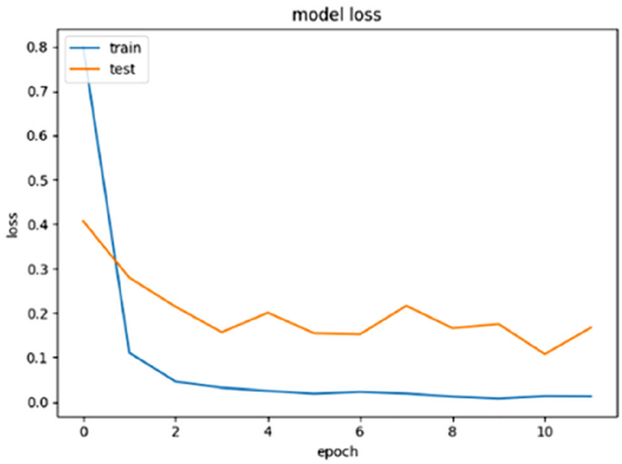


Fig. 6 Model loss of CNN

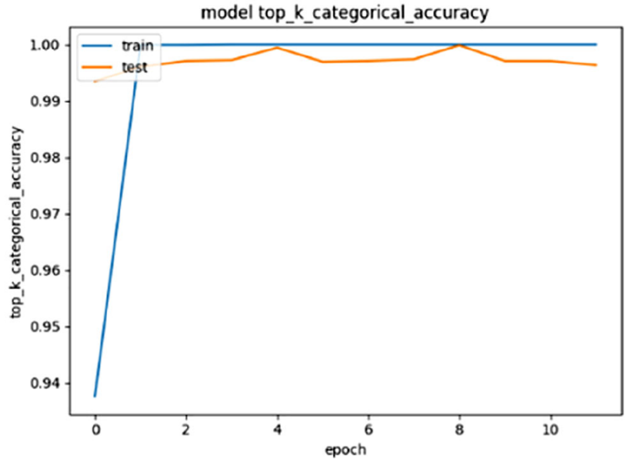


Fig. 7 Top_K categorical accuracy

demonstrate the behaviour of the proposed model on training and testing data for Top_K (= 5) values. The graph in Fig. 7 shows that the accuracy of the testing data is correctly converging to the accuracy of the training data for Top_K. Figures 8, 9 and 10 shows F measure, recall score and confusion matrix respectively.

C. Accuracy using double layer CNN

Double layer CNN is applied on 28 × 28 images with 64 filters (layer 1) and 128 filters (layer 2) followed by dense fully connected neural network. The double layer CNN model was trained with filter size from 5 to 12 to find out the optimal filter size for the CNN. The optimal value of filter size for 32 filters is 8 × 8 where error is 1.419% and cannot be minimized further for same configuration of CNN. Model accuracy is plotted in Fig. 11 and Table 6 shows the accuracy obtained in double CNN for various filter sizes obtained by exhaustive search.

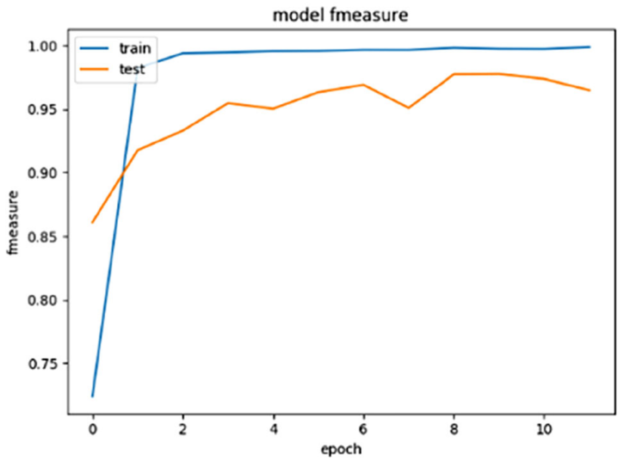


Fig. 8 Model F measure score

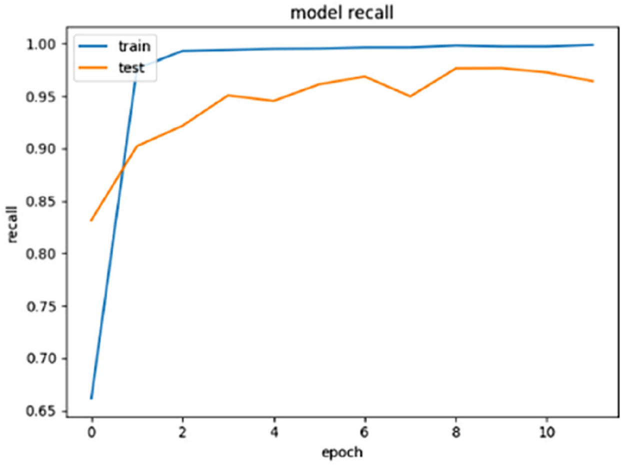


Fig. 9 Model recall score

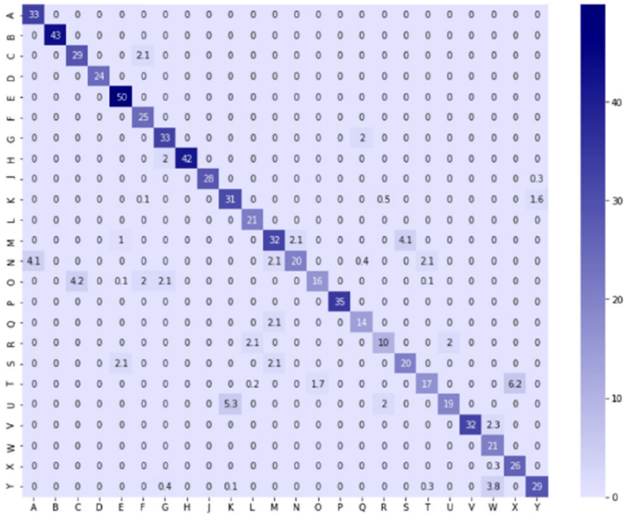


Fig. 10 Confusion matrix

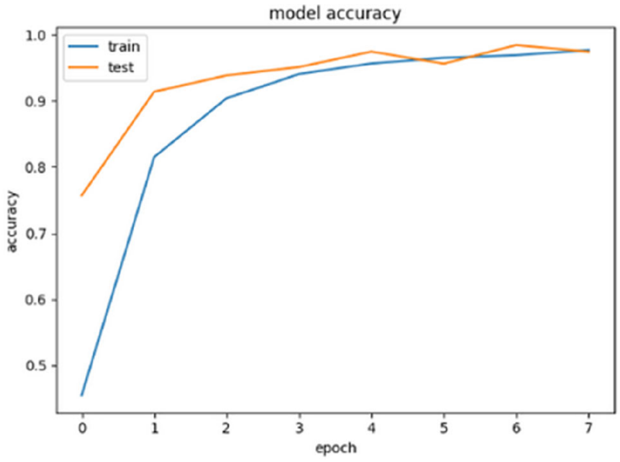


Fig. 11 Model accuracy

Table 6 Double CNN accuracy for various filter Size

Filter size	Accuracy	Error	Top_k_CA
5 × 5	98.094	1.906	1.00000
6 × 6	98.065	1.935	1.00000
7 × 7	97.704	2.296	1.00000
8 × 8	98.581	1.419	0.99985
9 × 9	96.852	3.148	1.00000
10 × 10	97.473	2.527	1.00000
11 × 11	97.212	2.788	0.99956
12 × 12	96.876	3.124	1.00000

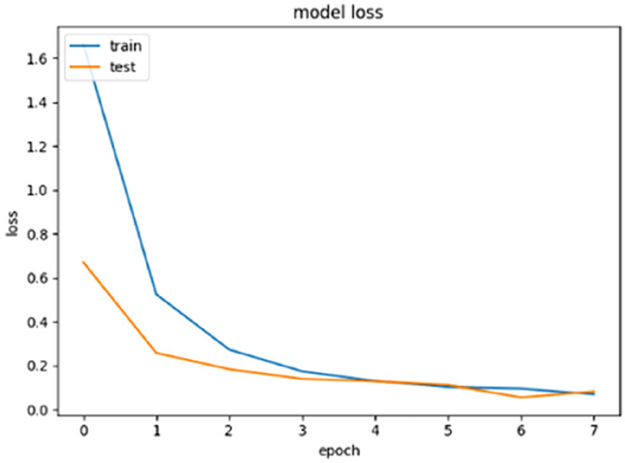


Fig. 12 Model loss

Figure 12 shows the behaviour of our model on training and testing data. The graph in Fig. 12 shows that the loss of the testing data is correctly converging to the loss of the training data.

Figure 13 demonstrate the behaviour of our model on

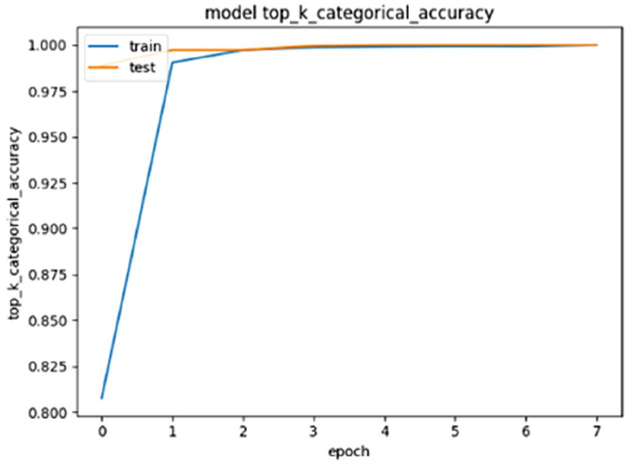


Fig. 13 Model Top_k_categorical accuracy

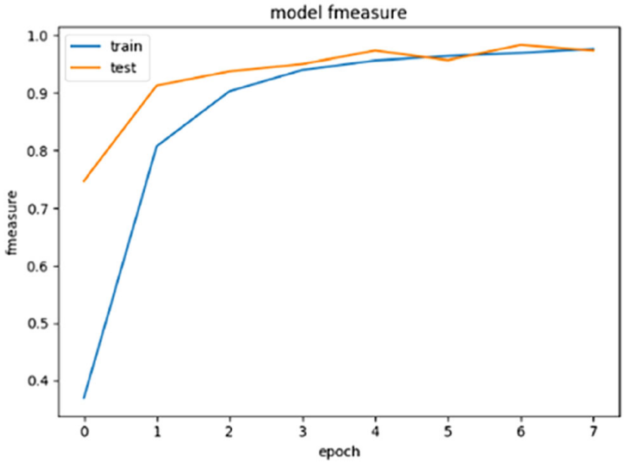


Fig. 14 Model F-measure

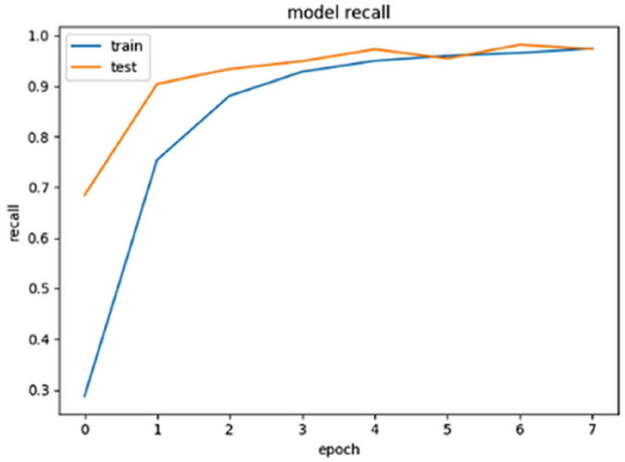
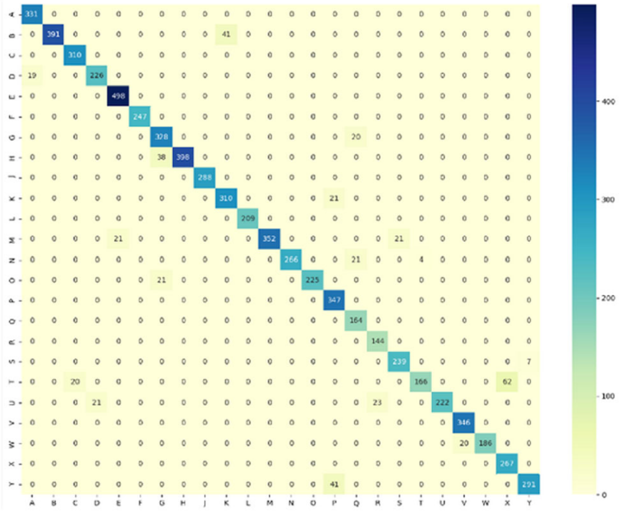


Fig. 15 Model recall



	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
A	331	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	391	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	310	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	19	0	0	226	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	498	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	247	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	328	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	38	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	288	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	310	0	0	0	0	21	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	209	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	21	0	0	0	0	0	0	352	0	0	0	0	0	21	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	266	0	0	21	0	0	4	0	0	0	0	0
O	0	0	0	0	0	0	21	0	0	0	0	0	0	225	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	347	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	164	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	144	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	239	0	0	0	0	0	7
T	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	0	0	0	62	0
U	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	222	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	346	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	186	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	267	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	291

Fig. 16 Confusion matrix

training and testing data for Top_K (5) values. The graph in Fig. 13 shows that the accuracy of the testing data is correctly converging to the accuracy of the training data for Top_K.

Figures 14, 15 and 16 shows F measure, recall score and confusion matrix respectively. From Figs. 14 and 15 it is clear that F-measure and recall values are converging for training and testing data in same pattern.

5 Conclusion and future scope

In this work, we have achieved minimum error for single as well as double layer CNN. The single layer CNN gives an accuracy of 97.344% and the double layer CNN gives an accuracy of 98.581% which is far better than the accuracy of 81.49% achieved using SVM with ‘Poly’ kernel. The optimum filter size for both single and double layer CNN after exhaustive search is found out to be 8 × 8. Traditional CNN’s have a predefined and fixed integral filter sizes for each convolutional layer, which may not give minimum error. In future this can be improved by implementing CNN to learn the filter size for each channel, which would result in learning of various variable sized parameters and the tuning of different hyper parameters.

References

1. Cheok MJ, Omar Z, Jaward MH (2019) A review of hand gesture and sign language recognition techniques. Int J Mach Learn Cybern 10(1):131–153
2. Gu Y, Liu T, Jia X, Benediktsson JA, Chanussot J (2016) Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification. IEEE Trans Geosci Remote Sens 54(6):3235–3247
3. Bellugi U, Fischer S (1972) A comparison of sign language and spoken language. Cognition 1:173–200
4. Yang R, Sarkar S, Loeding B (2010) Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. IEEE Trans Pattern Anal Mach Intell 32:462–477
5. Wu Y, Huang TS (1999) Human hand modeling, analysis and animation in the context of HCI. In: Image processing, ICIP 99. Proceedings. 1999 international conference, IEEE, pp 6–10
6. Wu Y, Huang TS (1999) Vision-based gesture recognition: a review. In: International gesture workshop, Springer, pp 103–115
7. Lockton R (2002) Hand gesture recognition using computer vision 4th year project report, pp 1–69
8. Starner TE (1995) Visual recognition of American sign language using hidden Markov models. Dept of Brain and Cognitive Sciences, Massachusetts Inst of Tech, Cambridge
9. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. Artif Intell Rev 43:1–54
10. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. Artif Intell Rev 43(1):1–54
11. Chen Q, Georganas ND, Petriu EM (2007). Real-time vision-based hand gesture recognition using haar-like features. In: 2007 IEEE instrumentation and measurement technology conference IMTC 2007, pp 1–6
12. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R et al (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, pp 675–678
13. Starner T, Weaver J, Pentland A (1998) Real-time American sign language recognition using desk and wearable computer-based video. IEEE Trans Pattern Anal Mach Intell 20:1371–1375
14. Grobel K, Assan M (1997) Isolated sign language recognition using hidden Markov models. In: Systems, man, and cybernetics, 1997. Computational cybernetics and simulation. 1997 IEEE international conference, IEEE, pp 162–167
15. Vogler C, Metaxas D (1998) ASL recognition based on a coupling between HMMs and 3D motion analysis. In: Computer vision, 1998. Sixth international conference, IEEE, pp 363–369
16. Vogler C, Metaxas D (1997) Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods. In: Systems, man, and cybernetics, computational cybernetics and simulation. 1997 IEEE international conference, IEEE, pp 156–161
17. Ahuja R, Jain D, Sachdeva D, Garg A, Rajput C (2019) Convolutional neural network based American sign language static hand gesture recognition. Int J Ambient Comput Intell IJACI 10(3):60–73
18. Ko SK, Kim CJ, Jung H, Cho C (2019) Neural sign language translation based on human keypoint estimation. Appl Sci 9(13):2683
19. Koller O, Camgoz C, Ney H, Bowden R (2019) Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. IEEE Trans Pattern Mach Intell 42:2306–2320
20. Wang H, Chai X, Chen X (2019) A novel sign language recognition framework using hierarchical Grassmann covariance matrix. IEEE Trans Multimed 21:2806–2814
21. Mittal A, Kumar P, Roy PP, Balasubramanian R, Chaudhuri BB (2019) A modified-LSTM model for continuous sign language recognition using leap motion. IEEE Sens J 19:7056–7063
22. Masood S, Thuwal HC, Srivastava A (2018) American sign language character recognition using convolution neural network. In: Satapathy S, Bhateja V, Das S (eds) Smart computing and informatics. Smart innovation, systems and technologie, vol 78. Springer, Singapore
23. Trigueiros P, Ribeiro F, Reis LP (2014) Vision-based Portuguese sign language recognition system. In: New perspectives in information systems and technologies, vol 1, pp 605–617. Springer International Publishing
24. Tavari NV, Deorankar AV (2014) Indian sign language recognition based on histograms of oriented gradient. Int J Comput Sci Inf Technol 5:3657–3660
25. Hasan H, Abdul-Kareem S (2014) Static hand gesture recognition using neural networks. Artif Intell Rev 41(2):147–181
26. Pigou L, Dieleman S, Kindermans PJ, Schrauwen B (2014) Sign language recognition using convolutional neural networks. In: Workshop at the European conference on computer vision 2014, pp 572–578. Springer International Publishing
27. Gupta B, Shukla P, Mittal A (2016) K-nearest correlated neighbor classification for Indian sign language gesture recognition using feature fusion. In: 2016 International conference on computer communication and informatics (ICCCI), pp 1–5

28. Nagarajan S, Subashini TS (2013) Static hand gesture recognition for sign language alphabets using edge-oriented histogram and multi class SVM. Int J Comput Appl 82(4):28–35
29. Singh K, Kumar S, Kaur P (2019) Support vector machine classifier based detection of fungal rust disease in Pea Plant (Pisam sativam). Int J Inf Technol 11:485–492. https://doi.org/10.1007/s41870-018-0134-z
30. Solanki A, Pandey S (2019) Music instrument recognition using deep convolutional neural networks. Int J Inf Technol. https://doi.org/10.1007/s41870-019-00285-y
31. Kaggle (2017) Sign language MNIST: drop-in replacement for MNIST for hand gesture recognition tasks. https://www.kaggle.com/datamunge/sign-language-mnist. Accessed Jan 2018