Omilia - The Sign Language Converter

Ajay Bhat[a], ***Avinash Raja S***[a], ***Christina Eunice John***[a], ***Kaladharshini K***[a], ***Karthika Menon***[a], ***Dr.S.Saraswathi***[a], ***Dr.K.Madheswari***[a], ***Dr.Y.V.Lokeswari***[a]

[a]*CSE Department, SSN College Of Engineering, Chennai - 603110, India*
[b]*E-mail addresses: ajay18012@cse.ssn.edu.in, avinashraja18030@cse.ssn.edu.in, christina18038@cse.ssn.edu.in, kaladharshini18067@cse.ssn.edu.in, karthika18070@cse.ssn.edu.in, saraswathis@ssn.edu.in, madheshwarik@ssn.edu.in, lokeswariyv@ssn.edu.in*

Abstract

Omilia - The sign language converter is a tool used for converting sign language into text as well as speech. This proves to be of great use to people who wish to interact using sign language but have no prior knowledge of the same. In India, about 63 million (6.3 percent) of the population suffers from auditory loss. This model aims to mitigate the stigma around deaf people in society and help in easing the process of communication. The model is a real time sign language converter, where the signer stands in front of a camera and signs, the gesture is then translated into a visual and auditory signal. This allows a person with no knowledge of sign language to understand the signer. This model uses a webcam, Logitech C310 HD Webcam, for capturing the signs and with the help of deep learning algorithms, the gesture is recognized. Once the gesture recognition gets completed, it is used to generate text that is displayed on a screen. The text is then converted into speech whose output is provided by a speaker. The whole model is executed on a Raspberry Pi.

Keywords: Computer Vision, Deep Learning, Gesture Recognition, Text to Speech, Image Processing

1 Introduction

A movement, of the hand or face, that expresses an emotion, sentiment or an idea, is defined as a gesture. Sign language is a structured and well- defined form of communication in which each letter or word is associated with a specific sign. It serves as a mode of communication for the specially chal- lenged people of our community. The hand shapes, hand movements, hand and head orientation, hand and head location, and facial recognition are the five essential parameters of a sign language recognition system. This model mainly focuses on recognizing signs by hand shapes. The model works on In- dian Sign Language (ISL). The signs for the ISL Alphabets is given in Fig 1. ISL is widely used by India's deaf community. However, sign language is not used as a medium of communication in schools for the specially-abled. Teachers are not prepared to use ISL-based teaching methods in teacher education programmes. There is no educational material available that includes Indian Sign Language. ISL interpreters are in high demand in institutes and other places where specially-abled people and others communicate, but India only has about 300 certified interpreters [1].

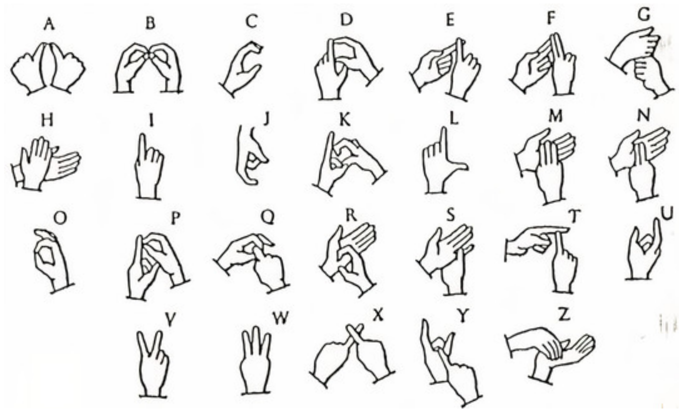


Figure 1: ISL Alphabets

This model uses a webcam, Logical C310 HD webcam, for capturing the signs and with the help of deep learning algorithms, the gesture is recognized. Once the gesture recognition gets completed, it is used to generate text. The text is then converted into speech which is played through speakers. The whole model is executed on a Raspberry pi model 4B. Since the gestures generated are captured as images, CNN can be used for feature extraction. The model uses a 11-layer deep sequential convolutional neural network-based framework that provides optimum accuracy. A rough illustration of the layers in a CNN model is provided by Fig 2. The remainder of the paper is structured as follows. Section II goes over related work. Section III describes the proposed system. Section IV is concerned with the specifics of implementation. Finally, Section V concludes the paper.

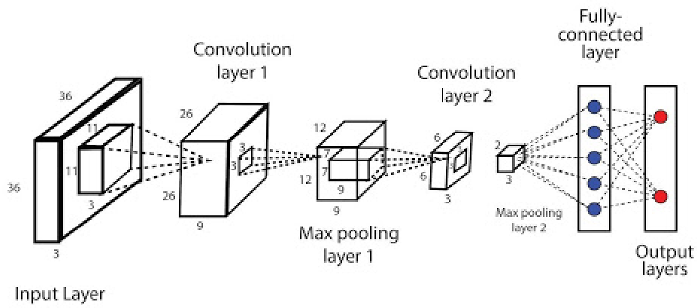


Figure 2: CNN Model (dimensions not to scale)

2 Literature Survey

Technology has been used to make people's lives easier. Helping differently abled people has always been on top of the list. There have been several attempts to bridge this gap by converting sign language into text or speech so that it can be understood by the average person. These approaches range from hand gesture sensing gloves to image processing based sign based sign language conversion.. The following tables show a comparison of sensor-based and camera-based sign language recognition models. Refer Table 1 for summary of sensor-based models and Table 2 for summary of camera-based models.

Table 1 Summary of Sensor-based Models

Paper	Components	Algorithm	Accuracy	Signs Classified
Vutinuntakasame S., Jaijongrak V.R., Thiemjarus S, 2011 [10]	BSN (Body Sensor Networks) Sensor Glove	Segmentation with threshold based classification and a probabilistic model. Multivariate Gaussia3n Model is used.	72.7-73.6% using rules for accuracy enhancement	26 letters of ASL, thumbs up(space) and thumbs down(full stop)
G. Marin, F. Dominio and P. Zanuttigh, 2014 [4]	Leap motion sensor, Kinect Sensor	**SVM** (Support Vector Machine) is applied to the extracted features to recognize the gesture. There are 2 different types of features extracted. For the first one a histogram of the distances of finger tips from the center of the palm is used. In the second one the curvature of the hand contour is used. **RBF** (Non-Linear Gaussian Radial Basis Function) has been used.	91.28% from the combined use of the two sensors, combining all the features.	10 different gestures
Elmahgiubi, Mohammed & Ennajar, Mohamed & Drawil, Nabil & Elbuni, Mohamed, 2015 [2]	Sensory glove, Arduino mega development board, LCD screen/Bluetooth smartphone	The minimum and maximum values of finger flexes are measured, and then mapped in the range of 0 to 200, with 0 being fingers fully flexed and 200 being all fingers completely bent. The flex values of the fingers are then compared to the values stored in the system and the matching letter is displayed.	96% model accuracy	20 out of 26 letters of the English alphabet
N. Tubaiz, T. Shanableh and K. Assaleh, 2015 [11]	Two DG5-VHand data gloves	**MKNN** (Modified K-Nearest Neighbors) classifier is used **HMM** (Hidden Markov Model) is used.	98.9% sentence recognition rate using MKNN	40 sentences formed using 80 word lexicon from Arabic Sign Language

Table 2 Summary of Camera-based Models

Paper	Components	Algorithm	Accuracy	Signs Classified
A. Kuznetsova, L. Leal-Taixé and B. Rosenhahn, 2013 [12]	Intel Camera	For regression and classification **MLRF** (Multi Layered Random Forest) is used.	84.7% real data accuracy. 97.4% model accuracy.	24 static letters in ASL.
Sahoo, Ashok & Mishra, Gouri & Ravulakollu, Kiran, 2014 [13]	Web camera for identifying regions of hand and face skin color.	**DIA** (Database Indexing Algorithm), **TWA** (Time Wrapping Algorithm).	91% using Generic Cosine Descriptor (GCD) by Using 3D Hopfield NN.	23 Arabic Sign Language gestures.
Tharwat, Alaa & Gaber, Tarek & Hassanien, Aboul Ella & Shahin, Mohamed & Refaat, Basma, 2014.[6]	Camera used for image capturing.	**SIFT** (Scale Invariant Feature Transform) is used to preprocess the gestures. **LDA** (Linear Discriminant Analysis) reduces the number of features and **SVM** (Support Vector Machine) is used to classify the gestures.	99% overall. Varying accuracy based on models used.	30 Arabic characters
Pigou, L., Dieleman, S., Kindermans, P.J., & Schrauwen, B. , 2015 [14]	Kinect sensor for input image	**3D CNN** (Convolutional Neural Network) is used to classify.	94.2% model accuracy when compared to HMM	ASL alphabets
Cooper, Helen & Ong, Eng-Jon & Pugeault, Nicolas & Bowden, Richard, 2012 [15]	Kinect camera for depth perception.	**SPB** (Sequential Pattern Boosting), **HMM** (Hidden Markov Models).	76% overall accuracy by using Markov Chains.	Kinect dataset, 2 D tracking data, 3 D tracking data.
Kishore, P.V.V. & rajesh kumar, Panakala, 2012. [8]	Camera for video input	**DWT** is used for feature extraction. **PCA** (Principal Component Analysis) is used to reduce the number of features. For classification **TSK** (Takagi-sugeno-kang) is used.	92.142% when tested by 10 different users (varying accuracy for varying sign). 96% model accuracy.	8 English Alphabets (A,B,C,D,M,N,X,Y), numbers and Few words in ISL
Mathavan, Suresh & Kumar, Mohan & Angappan, Kumaresan, 2016. [7]	C170 Logitech Camera used for image capturing	**DWT** (Discrete Wave Transform) is used for feature extraction. **KNN** (K-Neural Networks) is the classifier used.	99.23% when using cosine distance.	13 English Alphabets in ISL. (A,B,C,D,G,K,L,P,T,V,W,X,Z)
M. Geetha and Manjusha U C, 2012 [16]	Camera for receiving image input	**B-Spline** curves, **MCP's**(Maximum Curvature Points).	-	ISL Alphabets (A-Z) excluding J and H, and numerals (0-9).
K, Muthukumar; S, Poorani; S, Gobhinath, 2018. [17]	Android application called DroidCam to get images.	Combination of **DWT** and **DCT**. **LBP** (Local Binary Pattern) is used to increase features.	87% recognition rate when tested with 10 users.	ISL alphabets and numbers
N. Intwala, A. Banerjee, N. Gala [9]	Camera for input image capturing	**Transfer Learning, Convolutional Neural Network, GrabCut Algorithm.**	87.69% overall accuracy.	26 letters of the ISL.
Adithya, V., Vinod, P. R., & Gopalakrishnan, U., 2013 [18]	Web camera for image extraction.	Feature extraction used is **Distance transform** and **ANN** is used for classification.	91% model accuracy	English alphabets in ISL.

3 Proposed Model

Omilia - the Sign Language Converter is a CNN model which converts the input sign language to text and then, to speech. The model takes real-time input of sign language gestures via the webcam. The input image is obtained which then undergoes various standard pre-processing techniques like flipping the image around the vertical axis, converting to a wide range of hsv levels, identifying the regions of interest and removing any background noise in the captured image. The processed image is then passed through the Convolutional Network, where the features are extracted via a series of convolutional layers and max-pooling layers. The output from the CNN is then matched to any one of the gestures present in the database which was created. The predicted text displayed on the black-board. It is also converted to speech with the help of pyttsx3 python module and projected out aloud with the help of a speaker. The CNN model is executed with the help of Raspberry Pi. (Ref. Fig 3)

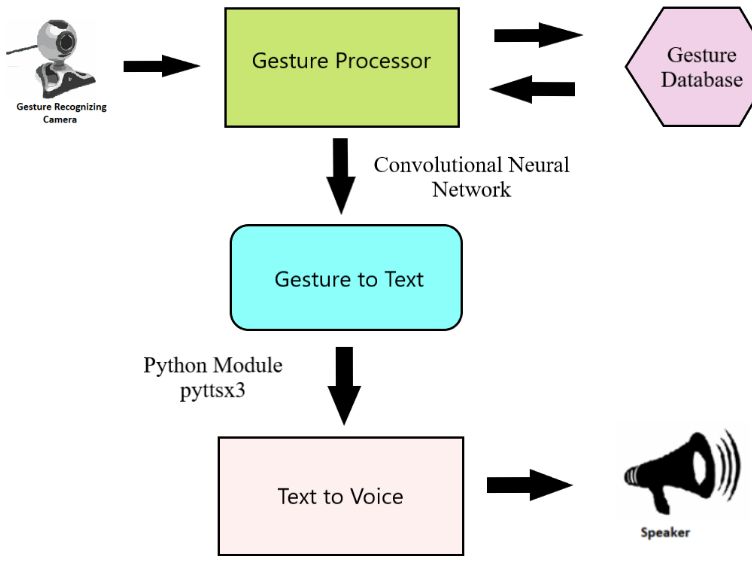
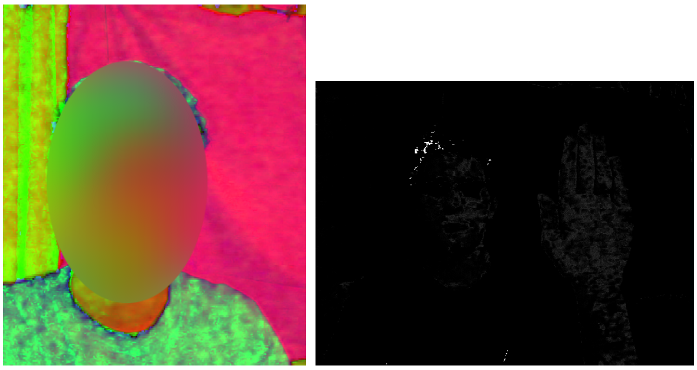


Figure 3: Proposed Model

4 Implementation Details

4.1 Dataset

Computer-based detection systems require large datasets for training, validation, testing, and comparison of performance. Due to the unavailability of pre-existing datasets for Indian Sign Language (ISL), a gesture database for ISL was created from scratch. The dataset was very carefully created. The dataset was created by the authors. The need to create our own dataset was necessitated by the fact that a well maintained dataset of Indian Sign Language was not available. Even whatever little was made publicly available consisted of only trivial alphabets. More sophisticated words were created by referring to the official website of Indian Sign Language. Currently the dataset has been incorporated with 36 gestures. The advantage of creating our own dataset especially in the case of Indian Sign Languages is that it can take into account the regional variations of the Sign Language. So the end users in general can very easily add additional gestures as and when needed. The images so obtained were resized to 50X50 pixels to maintain uniformity and



(a) HSV Image **(b)** Normalized + CalcBack

(c) DST (Gaussian Blur) **(d)** DST (Median Blur)

Figure 4: Image Preprocessing methods

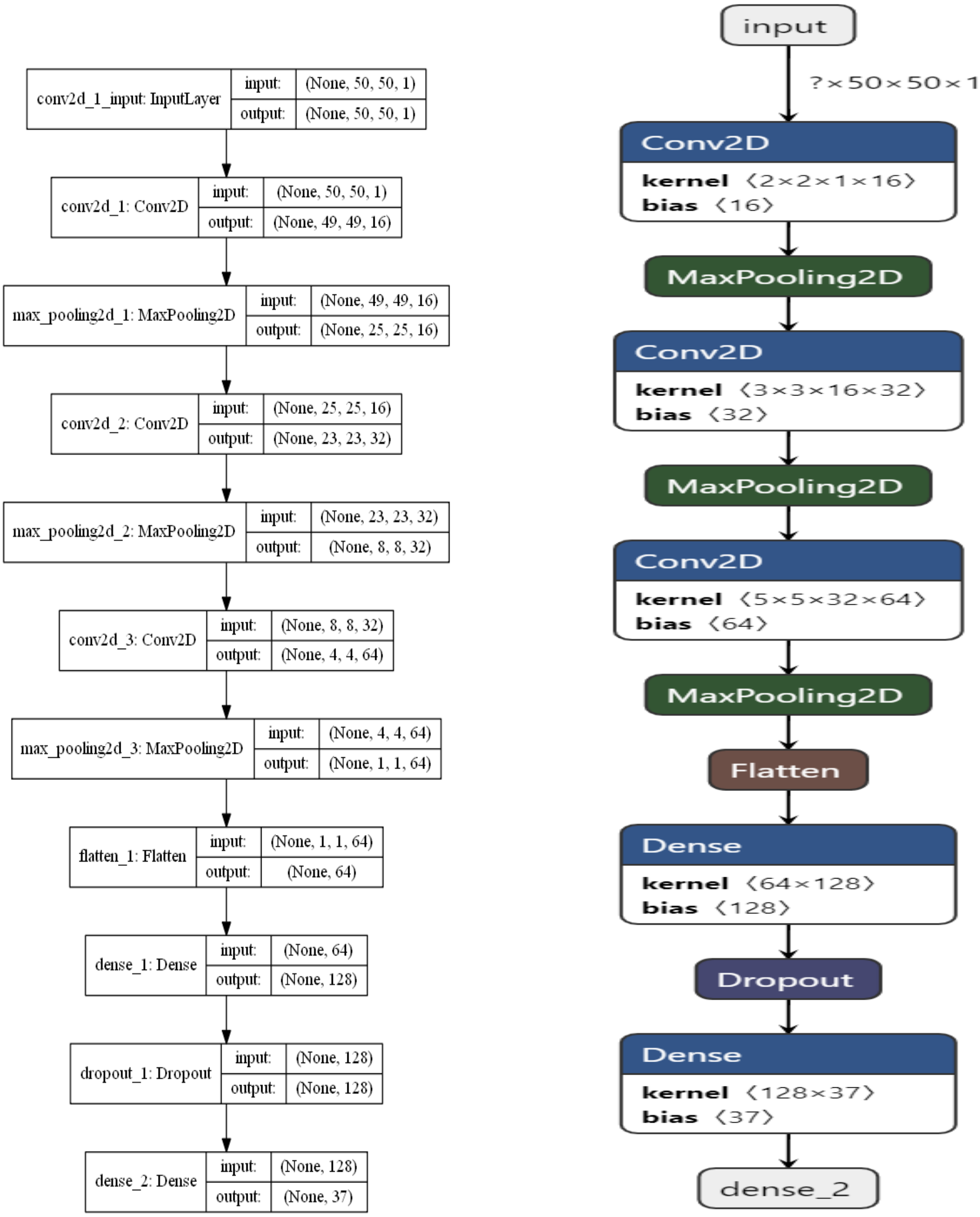
also fit into the computer's memory. The dataset contains a total of 108000 images. All of the images are in.jpg format. The dataset is grouped into three categories: 'train,' 'test,' and 'val.' The train set contains 90000 images, while the test and validation sets contain 9000 and 9000 images, respectively.

4.2 Image Preprocess

Our dataset is made up of images from various sources, considering it was created with the help of contribution of the authors and it is of varying quality, which can make human reading difficult due to blur, contrast, or brightness problems. Furthermore, there is a chance that the classification's accuracy would be harmed by the poor image quality. For this, several OpenCV functions such as convertColor, calcHist, normalize, calcBackProject, MorphEllipse, filter2d and threshold are applied for removal of background noise, adjusting hue and saturation levels and to build the region of interest. This ensures a tolerable range of skin colours is identified. A bounding box is also applied to mask out irrelevant areas and just obtain the region of interest containing the gesture. To make CNNs more generalised for both posterior-to-anterior (PA) and anterior-to-posterior (AP) views, or distortion problem, vertical flip is applied to these ROI images. Then the images are resized and pixel values are normalized to suit the trained CNN. (Ref. Fig 4a,4b,4c,4d)

4.3 CNN Model

The backbone of the model uses a 11 layer Convolutional Neural Network Architecture to perform the gesture recognition. This 11 layer CNN is a sequential model. Sequential model is a stack of layers where each layer has a single input tensor and a single output tensor and the connections between different units do not form a cycle. The stack of layers consists of convolutional layers, pooling layers, dropout layers, dense layers etc. The layers may or may not have parameters. It should be noted that the model presented in this paper is carried out using TensorFlow library and Keras. The code is written and executed on a Spyder notebook. To make our gesture recognition system portable a raspberry pi CPU is used to execute the model. The raspberry pi 4 model B with 4GB internal RAM is equipped with Raspbian OS and has Broadcom BCM2711, Quad core Cortex-A72 64-bit SoC @ 1.5GHz. A 64GB external SD card has been used for storage. The 11 layer CNN Architecture that is used for image recognition and classification is as shown in Fig 5b. Totally, there are 3 convolutional layers, 3 pooling layers, 1 dropout layer, 2 dense layers, an input and output layer. In case of the CNN presented in Fig 5a, the 3 convolutional layers create convolutional kernels of size (2,2), (3,3) and (5,5) respectively that are convolved with the input layer to generate feature maps. The feature maps generated by applying different sizes of kernels will indicate the concentration of the feature required and can hence gauge the region of interest. As indicated by Fig 4d, the output of image preprocessing displays the hand area (region of interest) in white. Hence, the output of the convolutional layer i.e, the feature maps, will find the concentration of white regions and hence map out the gesture to be recognized. In order to introduce non-linearity, Rectified Linear Unit(ReLU) has been used as the activation function. This will dispose of any linearity brought about by transitions in colours, pixels etc. Pooling layers and dropout layers have been inserted after every few convolutional layers (as seen in Fig 5a). Pooling is used to obtain image transition invariance. Pooling thus achieves stability against noise and congestion, and also makes the representation compact. Commonly used pooling operations include average pooling and max pooling. For this model, max pooling has been applied because of its rapid convergence (shown in Fig 5a). It also shows superior functioning in selecting invariant features and hence improves generalization, preventing the model from overfitting. Pool sizes of (2,2), (3,3) and (5,5) and strides of the same length are chosen. Dropout is a convenient way to avoid overfitting for a CNN model. It also helps in improving the performance of neural networks in a number of computer vision tasks, achieving excellent results over a variety of structured datasets. Setting the dropout rate is part of creating a dropout layer. A fraction of input units (the drop out rate) is set to 0 at every update during training. In this model, a dropout rate of 0.2 has been used, because the rate has to be optimum for proper convergence, it should neither be too low nor too high. The dense layers are present at the very end of CNN (shown in Fig 5a). The purpose of dense layers is to perform the final task of classification by doing matrix-vector multiplication to produce a 'm' dimensional vector where the inputs are received from all neurons of the previous layers. The first dense layer has 128 units and makes use of 'ReLU' as the activation function. The activation function for the second dense layer, which has 44 units, is 'Softmax.' The second dense layer is the one that outputs the final gesture class i,e. the prediction for whatever sign has been captured. The number of units in both the dense layers was determined by trial and error.



(a) CNN Model **(b)** CNN Architecture

Figure 5: CNN

The hyper parameters used for the CNN training process are given in Table 3. As this network was created to solve a problem with multiple classifications, categorical cross entropy was chosen as the loss function. The objective that the model tries to minimize during training is the loss function. SGD is the optimizer for the loss function of this model. Stochastic Gradient Descent (SGD) is an effective tool for gradient-based loss function optimization.The learning rate has been set at 0.01 as this is widely used in case of Stochastic Gradient Descent Optimization method.

Table 3 Hyper-parameters used for the CNN training

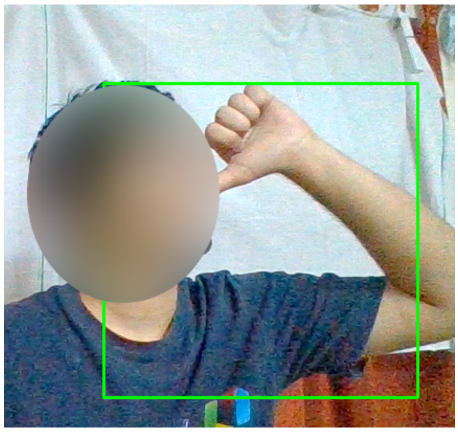
Hyper-parameters	Values
Learning Rate	0.01
Loss Function	Categorical Cross Entropy
Batch Size	500
Optimizer	Stochastic Gradient Descent
Number of Epochs	20
Steps per Epoch	180

In the field of image classification, data augmentation has been proven to be extremely useful. This allows researchers to increase the size of the dataset and also improve robustness of the data which eventually helps in preventing overfitting of the model. This model has employed flipping of images around the vertical axis to achieve the same. The learning rate has been set at 0.01 as this is widely used in case of Stochastic Gradient Descent Optimization method. In the field of image classification, data augmentation has been proven to be extremely useful. This allows researchers to increase the size of the dataset and also improve robustness of the data which eventually helps in preventing overfitting of the model. This model has employed flipping of images around the vertical axis to achieve the same.

4.4 Display

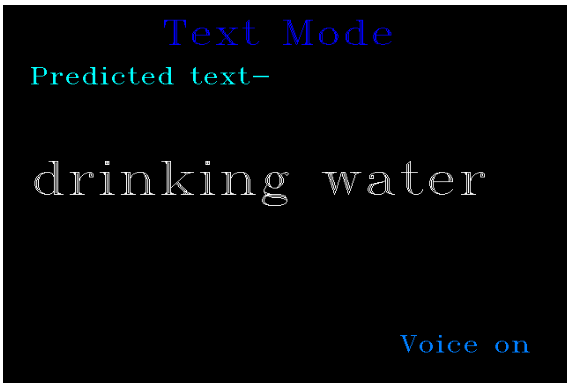
The model, after recognition and identification of the gesture, converts it into text, which is displayed on a screen, Waveshare 7 inch HDMI LCD Screen and it being touchscreen, doesn't need extra equipment like a keyboard or a mouse to operate, which makes it more handy. It is also Raspberry pi compatible. The text is then converted into speech, which is made audible with the help of a speaker. (Ref. Fig 6a,6b,6c,6d)

(a) Thresh for setting hand histogram



(b) Live-action Window

(c) Thresh for setting Live-action Window



(d) Prediction board

Figure 6: Output Window

5 Results

The accuracy and loss curves for the 11 layer CNN model are shown in Figures 14 and 15. Each epoch consists of 180 steps and is repeated for a total of 20 epochs. In each iteration, a set of 500 images is propagated through the neural network. The accuracy and loss on the training set is shown by the blue curve in figures 7 and 8 respectively, while the orange curve represents the same on the testing data. The accuracy increases exponentially over time while the loss decreases. The fit between the prediction and the ground truth label is represented by the loss. As the training progresses, the loss value should decrease. The classification report of the model consisting of precision, recall, f1-score for the 36 gestures and the accuracy of the model is given in Table 4.

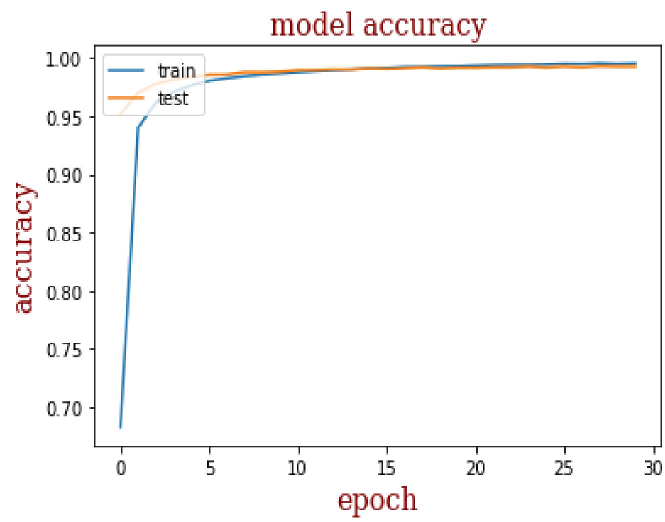


Figure 7: Model Accuracy Curve

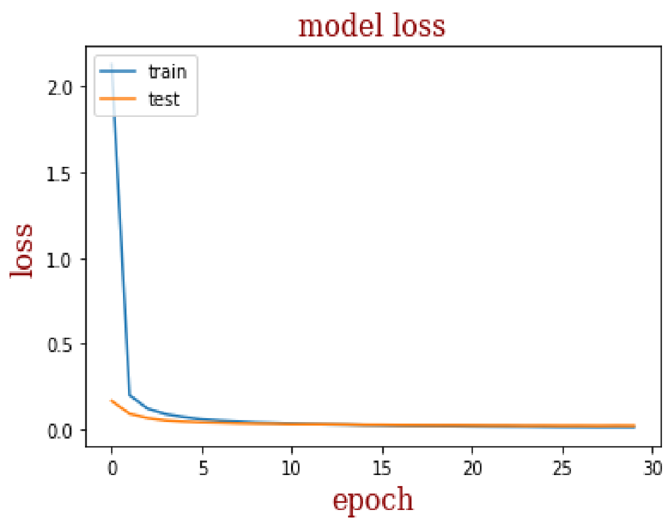


Figure 8: Model Loss Curve

Table 4 Classification Report

	Precision	Recall	F1-Score	Support
0	0.99	0.99	0.99	204
1	1.00	1.00	1.00	204
2	1.00	1.00	1.00	199
3	1.00	1.00	1.00	208
4	1.00	1.00	1.00	202
5	0.99	1.00	1.00	199
6	1.00	1.00	1.00	210
7	0.98	0.99	0.99	184
8	1.00	1.00	1.00	203
9	1.00	1.00	1.00	204
10	1.00	1.00	1.00	209
11	0.99	1.00	1.00	201
12	0.99	1.00	1.00	193
13	1.00	1.00	1.00	204
14	1.00	0.99	0.99	185
15	0.99	1.00	1.00	187
16	1.00	1.00	1.00	216
17	0.99	1.00	1.00	197
18	1.00	1.00	1.00	207
19	0.99	1.00	0.99	201
20	1.00	0.99	0.99	212
21	1.00	1.00	1.00	203
22	1.00	1.00	1.00	222
23	1.00	1.00	1.00	221
24	0.99	0.98	0.99	179
25	1.00	1.00	1.00	212
26	1.00	1.00	1.00	203
27	1.00	1.00	1.00	208
28	0.93	0.83	0.88	205
29	0.80	0.90	0.85	195
30	1.00	0.99	1.00	195
31	0.94	0.97	0.96	198
32	1.00	1.00	1.00	194
33	1.00	1.00	1.00	190
34	1.00	1.00	1.00	189
35	0.95	0.89	0.92	159
36	0.99	0.99	0.99	198
		Accuracy	0.99	

6 Conclusion and Future Works

About 6.3% of the population in India suffers from auditory loss. Sign Language detection using gesture images can greatly improve their livelihood. A personalised, sequential 11-layer CNN model shows excellent classification accuracy of 0.99 in this paper. In addition, the model uses a customized dataset which was created by referring to the official website of Indian Sign Language, and it presently has around 36 gestures. The advantage of creating our own dataset especially in the case of Indian Sign Languages is that it can take into account the regional variations of the Sign Language. However, a limitation of the model is in its inability to make accurate predictions when the real-time images are captured under less than ideal lighting conditions. There is scope for improvement in this regard. This paper also explains why certain design decisions were made. By carefully exploring the application of deep learning and fine-tuning methods, it may be possible to increase accuracy even more. Another area of future study is to incorporate facial expressions and dynamic hand gestures found commonly in the Indian Sign Language. The few possible future works of the model include expansion of the database to include different regional sign language gestures of the ISL, improving the efficiency of the model to work under varied lighting conditions, and extending the model to convert speech to sign language.

References

[1] Official Website of the Indian Sign Language Research and Training Centre (ISLRTC). http://www.islrtc.nic.in/

[2] Vutinuntakasame S. Jaijongrak V.R., Thiemjarus S. An assistive body sensor network glove for speechand hearingimpaired disabilities; Proceedings of the 2011 International Conference on Body Sensor Networks (BSN); Dallas, TX, USA. 2325 May2011; pp. 712.

[3] G. Marin, F. Dominio and P. Zanuttigh, "Hand gesture recognition with leap motion and Kinect devices", Proc. IEEE Int. Conf. Image Process. (ICIP), pp. 15651569, Oct. 2014.

[4] Elmahgiubi, Mohammed & Ennajar, Mohamed & Drawil, Nabil & Elbuni, Mohamed. (2015). Sign Language Translator and Gesture Recognition. 10.1109/GSCIT.2015.7353332.

[5] N. Tubaiz, T. Shanableh and K. Assaleh, "GloveBased Continuous Arabic Sign Language Recognition in UserDependent Mode," in IEEE Transactions on HumanMachine Systems, vol. 45, no. 4, pp. 526533, Aug. 2015, doi: 10.1109/THMS.2015.2406692.

[6] A. Kuznetsova, L. LealTaixé and B. Rosenhahn, "RealTime Sign Language Recognition Using a Consumer Depth Camera," 2013 IEEE International Conference on Computer Vision Workshops, Sydney, NSW, Australia, 2013, pp. 8390, doi: 10.1109/ICCVW.2013.18.

[7] Sahoo, Ashok & Mishra, Gouri & Ravulakollu, Kiran. (2014). Sign language recognition: State of the art. ARPN Journal of Engineering and Applied Sciences. 9. 116134.

[8] Tharwat, Alaa & Gaber, Tarek & Hassanien, Aboul Ella & Shahin, Mohamed & Refaat, Basma. (2014). SIFTbased Arabic Sign Language Recognition System. Advances in Intelligent Systems and Computing. 334. 10.1007/9783319-135724_30.

[9] Pigou, L., Dieleman, S., Kindermans, P.J., & Schrauwen, B. (2015). Sign Language Recognition Using Convolutional Neural Networks. Lecture Notes in Computer Science, 572578.

[10] Cooper, Helen & Ong, EngJon & Pugeault, Nicolas & Bowden, Richard. (2012). Sign Language Recognition using SubUnits. Journal of Machine Learning Research. 13. 22052231.

[11] Kishore, P.V.V. & rajesh kumar, Panakala. (2012). A Video Based Indian Sign Language Recognition System (INSLR) Using Wavelet Transform and Fuzzy Logic. International Journal of Engineering and Technology. 4. 537542. 10.7763/IJET.2012.V4.427.

[12] Mathavan, Suresh & Kumar, Mohan & Angappan, Kumaresan. (2016). An Efficient Framework for Indian Sign Language Recognition Using Wavelet Transform. Circuits and Systems. 07. 18741883. 10.4236/cs.2016.78162.

[13] M. Geetha and C, M. U., "A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using BSpline Approximation", International Journal on Computer Science & Engineering (IJCSE), vol. 4, p. 3, 2012.

[14] K, Muthukumar; S, Poorani; S, Gobhinath. Extraction of Hand Gesture Features for Indian Sign languages using Combined DWTDCT and Local Binary Pattern. International Journal of Engineering & Technology, [S.l.], v. 7, n. 2.24, p. 316320, apr. 2018. ISSN 2227524X.

[15] N. Intwala, A. Banerjee, N. Gala et al., "Indian sign language converter using convolutional neural networks", 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), pp. 15, 2019.

[16] Adithya, V., Vinod, P. R., & Gopalakrishnan, U. (2013). Artificial neural network based method for Indian sign language recognition. 2013 IEEE CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES.