

A Deep Learning Based Model To Recognize Sign Language

by Josphineleela R

Submission date: 12-Apr-2023 04:58PM (UTC+0530)

Submission ID: 2062446037

File name: A_Deep_Learning_Based_Model_To_Recognize_Sign_Language.pdf (1.21M)

Word count: 4749

Character count: 25701

SIGN I-NET : A Deep Learning Based Model To Recognize Sign Language

Dr.R.Josphineleela¹, P.Annamalai², A.Ashwin Kumar³, D.Abishek Benjamin Raj⁴

¹Professor, Dept. of Computer Science and Engineering, Panimalar college, Tamil Nadu, India

^{2,3,4}Student, Dept. of Computer Science and Engineering, Panimalar college, Tamil Nadu, India

Abstract - Among the people living in the world, of whom 97.6 million are both deaf and blind. People in the age group of 15-49 years, which happens to be age group that contributes the most to science and technology, makes about 9% of this population. To communicate with other humans, deaf-blinded people use communication techniques such as sign language, and written text. Recent advancement in AI, Computer vision has made a lot of progress in helping the hearing and speech impaired people to lead a more comfortable independent life. Several solutions have been proposed to help them in navigation and in many other day-to-day tasks. Solutions to help them access the world of knowledge are comparatively less explored. The information conveyed to general masses from person of hearing-disabilities are primarily rely upon through lip reading. As there is a need for a solution for hearing impaired individuals, we integrated a AI based sign language interpreter to overcome communication hurdles among peers for online based video and real-time conversations. Through our efforts, we hope to have moved an inch closer to bridging the gap between people with hearing ,speech impairment and the world's knowledge pool.

1.INTRODUCTION

Sign language is a type of communication used by persons who have hearing and speech impairments. Sign language movements are used by people to communicate their thoughts and emotions through nonverbal communication. Non-signers, on the other hand, find it exceedingly difficult to comprehend, which is why skilled sign language interpreters are required for medical and legal visits, as well as educational and training sessions. Throughout the last five years, there has been an increase in demand for interpreting services. Alternative methods, such as video remote human interpretation over high-speed Internet connections, have been established. They will therefore give an easy-to-use sign language interpreting service, which can be used but has significant restrictions.

According to the 2011 Census, there are about 1027835 persons with hearing and speech impairments among India's

handicapped population of 2.68 crores (2.21% of the overall population), comprising 545179 males and 482656 females.

These group of people evolved their own language to communicate with one another, which we call Sign language. To convey one's thoughts, sign languages employ visible human body motions and gestures. Sign languages differ from place to the next. There is American Sign Language in the United States of America, for example, and Indian Sign Language in India. Several sign languages also include a section where native language alphabets are conveyed by fingerspelling hand signals.

Over the past years, we can find many trials to develop sign language recognition (SLR) systems. Although extensive effort has been done in American sign languages across the world, it appears that considerably less study has been done in Indian dialects. On the basis of input data, there are two types of SLR architecture: glove-based [1], [2], [3] and vision-based [4], [5]. Glove-based SLR architecture use smart gloves to monitor hand position, orientation, velocity, and so on utilizing microcontrollers and sensors. Vision-based SLR techniques use cameras to detect hand gestures. Computer vision-based SLR systems often extract characteristics including edge detection, skin color segmentation, gesture identification, hand form detection, and so on. Nevertheless, most of these solutions need too much processing power to operate in real-time on low-end computation devices such as mobile phones and are thus confined to platforms with high computational capacity. Moreover, the hit and miss of hand-tracking mechanisms is almost evident in all of these approaches.

This study adds to a deep learning-powered Indian Sign Language Recognition system that focuses on identifying basic letters of the American Alphabet. The method comprises the implementation of a hand tracking solution given by Google's open-source project, MediaPipe [6]. A deep learning method is also developed on this solution to provide a quick, economical, lightweight, and simple-to-deploy system that may be utilized as the core of a comprehensive sign language recognition system. The hand landmarks detection technique is used at the heart of this work. Using MediaPipe's hand tracking model, 21 hand landmarks in each image containing Sign Language hand signals were recognised. The landmarks are then gathered as coordinate points, normalized, and recorded as dataset in a.csv file. A feedforward neural network model is then trained on these data points, and real time hand sign

recognition using OpenCV is done using the trained model. Figure 1 depicts an overview of the entire project. The finished model is very efficient, precise, portable, and light. This paper goes into the full procedure for building our model, including the results and evaluation of alternatives.

2. LITERATURE REVIEW

Ansari and Hari [7] investigated the classification of static movements in Indian sign language using photos with 3D depth data.

The photographs were obtained using Microsoft Kinect, which captures 3D depth information in addition to 2D images. The dataset included 5041 photos of static hand movements that were categorized with 140 different classifications. The model was trained using K-means clustering. In the study, they were able to achieve 90% accuracy for 13 signs and 100% accuracy for the signs, bringing the total number of alphabets recognised to 16 (A, B, D, E, F, G, H, K, P, R, T, U, W, X, Y, Z) with an average accuracy rate of 90.68%.

Pugeault and Bowden [8] used color and depth pictures to create a recognition system based on 24 static American Sign Language (ASL) alphabet signs. Gabor filters are utilised to extract features at different scales, and a multiclass random forest classifier is employed. In the leave-one-out trial, they achieved a recognition rate of 49%. They also generated a dataset dubbed American Fingerspelling Dataset and it is the most often used benchmarked dataset in this sort of sign language study.

Lin et al. [9] created a CNN for human gesture recognition and gathered skin color, orientation, and hand position for CNN training. To train the skin color model, the Gaussian Mixture Model is utilized. This model's average accuracy is 95.96%.

Sruthi C. J and Lijiya A [10] developed a static alphabet recognition CNN model for an ISL system. They used the binary silhouette of the signer's hand area to recognise static ISL alphabets. The created model has a 98.64% accuracy.

Rekha et al. [11] worked using the Indian Sign Language dataset's 23 static and three dynamic signals. To find hands, they employed skin color segmentation. Edge orientation and texture were employed as features to train a multiclass SVM, with an 86.3% success rate. Their method, however, was too sluggish to be used as a practical gesture recognition system.

Bhuyan et al. [12] employed a dataset of 400 photos including 8 motions from Indian Sign Language. They employed a skin color-based segmentation approach for hand identification, followed by closest neighbor classification, to reach a recognition rate of more than 90%.

Pugeault and Bowden [13] developed a real-time recognition method for alphabet recognition in American Sign Language. A dataset of 24 classes was employed, comprising 48,000 3D depth photos collected using a Kinect sensor. Gabor filters and multi-class random forests were utilized, resulting in exceptionally accurate classification rates.

Keskin et al. [14] employed object identification by components to recognise signals denoting numerals in American Sign Language. Their dataset included ten classes and 30,000 3D depth photos collected with a Kinect sensor, and they attained an accuracy rate of nearly 99%.

Halder and Tayade [15] used the MediaPipe framework to get multi-hand landmarks and used Support Vector

Machine (SVM) for Real-time detection of hand signs. The average accuracy achieved was about 99%.

Das et al. [16] investigated a deep learning-based sign language recognition system that uses processed static pictures to recognise American Sign Language motions. The collection contained 24 labels of static motions from A through Z, except J. The collection had around 100 photos per class taken on an RGB sensor. To train the model, the Inception V3 convolutional neural network model was used. After training and testing the model, the average validation accuracy rate was above 90%, with the highest validation accuracy being 98%. They determined that when given a dataset of suitably cropped photos, the relatively recent Inception V3 model might be an effective model for static sign language identification.

Sahoo [17] worked on Indian sign language identification using machine learning. This study concentrated on static hand movements in Indian sign language for numerical numbers (0-9). To create the dataset, photos of the signs were captured using a digital RGB sensor. The collection had 5000 photos in total, with 500 images for each digit from 0 to 9. To train the model, two supervised learning classifiers were used: Naive Bayes and kNearest Neighbor. In this study, the K-Nearest Neighbor approach performed marginally better than the Naive Bayes classifier, with average accuracy rates of 98.36% and 97.79%, respectively.

Ren et al. [18] segmented hands from Kinect pictures using a threshold-based method. Their dataset had ten classes and 1000 photos in total. The achieved accuracy percentage was around 93%.

3. DATA COLLECTION PROCEDURE

3.1 Generating the ASL Dataset

The data set contains a collection of photographs of alphabets from American Sign Language, organized into 29 folders that reflect the various classes. The training data set includes 87,000 200x200 pixel pictures. There are 29 classes, 26 for the letters A-Z and three for SPACE, DELETE, and NOTHING.

These three classifications are highly helpful for real-time applications and classification. This test data set is only 29 photos strong in order to encourage the use of real-world test image. we have applied data augmentation techniques to increase data sample sizes for better efficiency and analysis.

3.2 Detection of Hand Landmarks Using MediaPipe

For detecting hands, we used the mediapipe library, which not only detects hands in the frame but creates 22 3D coordinates of the entire hand, giving us rich data to work with. The MediaPipe hand landmarks model gives the coordinates indicating hand landmark points based on the placement of the pixels containing the landmark points in the image. As a result, the coordinates of two photographs of the identical hand gesture with different configuration in the frame might be thousands of miles apart. This makes training the model more challenging. To fix this difficulty, the wrist landmark point (donated by index 0 in the hand landmarks list) was assumed to have x and y coordinates of (0,0) and the coordinates of all the other landmark points were appropriately altered relative to the wrist point.

The coordinate values were then normalized to be in the range [0,1] by dividing all of them by the biggest absolute coordinate value obtained through relative adjustment in the landmarks list. The coordinates were gathered in the.csv file after they were normalized. The coordinate normalizing process is shown in Fig.1.

The coordinates in the.csv file were collected and then sent via a pandas library method to identify null values.

The model occasionally fails to recognise hands in hazy photos, resulting in void entries. Clearing up these vacant entries is required before training an impartial model. As a result, we used their indices to detect null entries and delete them from the file.

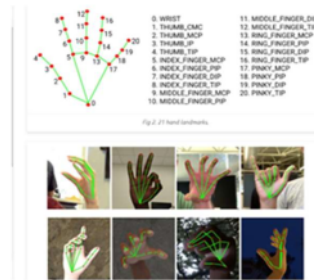


Fig-1

3.3 Min-Max Scaler Based Normalization

Using the minimum and maximum value of each feature, we may scale the data in a dataset using the min-max normalization approach to a particular range. The min-max scaler employs the lowest and maximum value of each column to scale data series instead of the traditional scaling method, which scales data based on the standard normal distribution (with mean = 0 and standard deviation = 1).

3.4 Data Cleaning and Normalization

The MediaPipe hand landmarks model gives the coordinates indicating hand landmark points based on the placement of the pixels containing the landmark points in the image. As a result, the coordinates of two photographs of the same hand sign with different frame configuration might be substantially different. This makes training the model more challenging. To fix this difficulty, the wrist landmark point (donated by index 0 in the hand landmarks list) was assumed to have x and y coordinates of (0,0) and the coordinates of all the other landmark points were appropriately altered relative to the wrist point.

The coordinate values were then normalised to be in the range [0,1] by dividing all of them by the biggest absolute coordinate value obtained through relative adjustment in the landmarks list. The coordinates were gathered in the.csv file after they were normalised. The coordinate normalising process is shown in Fig. 2.

The coordinates in the.csv file were collected and then sent via a pandas library method to identify null values.

The model occasionally fails to recognise hands in hazy photos, resulting in void entries. Clearing up these vacant entries is required before training an impartial model. As a result, we used their indices to detect null entries and delete them from the file.

Jyotishman Bora et al. / Procedia Computer Science 218 (2021) 1384–1393

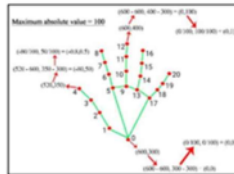


Fig -2

4. PROPOSED SYSTEM

4.1 Convolutional Neural Network(CNN)

CNN is an artificial neural network that processes data using the convolution principle. It is a deep neural network with extra layers between the input and output layers [26]. CNN's main idea is to use a convolutional layer to determine the association between object characteristics and a pooling layer to group comparable features. CNN architecture is made up of at least one convolutional layer (CL), which converts a collection of activation functions with a differential function, each with its own pooling layer (PL), and the end result is a fully connected layer (FCL) [27]. The pooling layer is used to shrink a picture by down sampling and summarizing its characteristics. The most frequent pooling methods for grouping are average pooling, in which the summary is the dominant characteristic, and maximum pooling, in which the strongest feature is summarized [28]. To detect a pattern, CNN omits zero-valued parameters and creates fewer connections between layers than conventional neural networks, which connect all neurons in the next layer. To decrease the number of connections, non-zero parameters can be shared and utilized by more than one connection in the layer. CNN is a powerful approach because structures of multiple dimensions may be built to adapt to input data of different dimensions, which can have a significant impact on final classification/prediction outcomes. Furthermore, because CNN is founded on the notion of locality [29], the representation of raw data is critical.

The simple CNN seen in Figure employs a 1-dimensional array of data as its input layer, with each pixel vector from various pictures representing the input data. The CL then down samples the data, which is subsequently collected by the PL and resulting in the FCL.

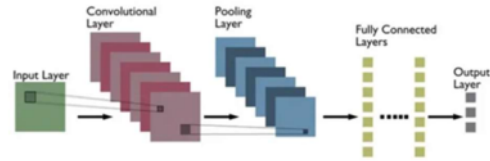


Fig -3: Illustration of CNN Architecture

4.2 Random Forest(RF)

Random forest is an ensemble technique that is formed from a large number of uncorrelated trees that have been refreshed via bootstrap aggregation or bagging. By randomly picking replacement cases from the original training set of specified feature combinations, the bagging strategy generated a training dataset [25]. Because the bagging strategy considers all predictor variables on each splitting in a tree, trees from different samples may have a similar structure, which is known as tree correlation. Random forest improves the bagging strategy by decreasing correlation and variance.

The decision tree predictors in the forest classify samples based on the most popular (high vote) class. Because the decision tree architecture necessitates attribute selection and pruning in order to maximize the options, a tree was grown to its maximum depth each time by utilizing new training data derived from a combination of characteristics. The generalization error converges with pruning as the number of trees rises. The amount of characteristics needed to construct a tree at each node, as well as the number of trees to develop, are user-defined parameters for the RF classifier [25].

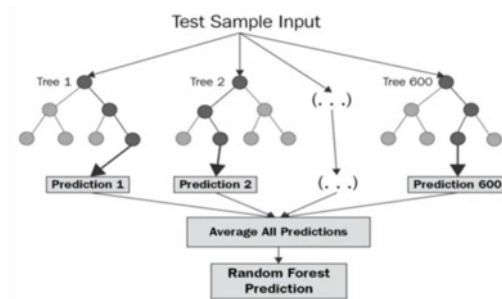


Fig -4: Illustration of Random Forest Classifier Architecture

4.3 Proposed Method

Figure-5 depicts the combined CNN and RF (CNN-RF) architecture which comprises of an image band-derived input layer, a convolutional layer, a pooling layer, and a fully connected layer. The output of the first completely connected layer is the result of extracted features, which are then utilized as input for the RF technique of categorization of the burnt area.

In the previously generated dataset, 6 training algorithms (Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Gaussian NB, Linear Discriminant Analysis) were tried in order to obtain the highest accuracy from the testing set.

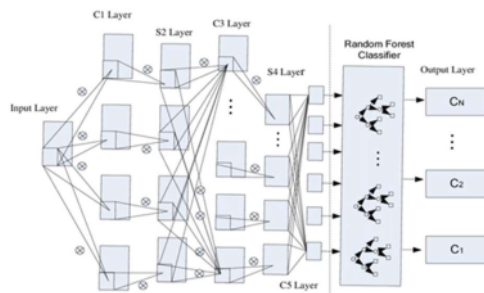


Fig -5: Illustration of CNN-RF Based Hybrid Architecture

5. RESULTS AND ANALYSIS

5.1 Quantitative Analysis of the Model

The classification report and confusion matrix modules from Python's scikit-learn toolkit were used to perform quantitative analysis on the test dataset. The classification

report package generates an assessment report for our model that includes matrices for accuracy, precision, recall, and F1 score. Together with them, the 'support' matrix indicates the model's performance in real-time recognition.

The accuracy matrix computes the number of labels successfully predicted by the model from the entire dataset. The mathematical definition of the accuracy matrix is shown in Equation (1).

The precision matrix assesses the model's accuracy in terms of expected positives. It computes the proportion of real positives among anticipated positives. When the False Positive (FP) cost is significant, this is a great metric to consider. Equation (2) displays the precision matrix's mathematical formula. The recall matrix counts how many positive predictions our model properly labels. It is a metric used when the high cost of False Negatives is evaluated (FN). The recall matrix is mathematically expressed in Equation (3). By combining recall and accuracy, the F1 score gives a cumulative assessment. The mathematical formulation of this matrix is given in Equation (4).

$$A (\text{Accuracy}) = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$P (\text{Precision}) = \frac{TP}{TP+FP} \quad (2)$$

Fig -6

$$R (\text{Recall}) = \frac{TP}{TP+FN} \quad (3)$$

$$F (F1 \text{ score}) = \frac{2 \cdot P \cdot R}{P+R} \quad (4)$$

Fig -7

In equations (1), (2), (3), and (4), TP, TN, FP, and FN indicate True Positive, True Negative, False Positive, and False Negative, respectively.

Table -1 details the classification report of the CNN-RF model. The model's average accuracy is 99%, as stated in the report.

	precision	recall	f1-score	support
a	0.97	0.97	0.97	95
b	0.98	1.00	0.99	113
c	0.95	1.00	0.97	97

d	0.97	0.93	0.95	100
del	0.95	0.96	0.96	106
e	1.00	0.99	0.99	99
f	0.99	0.98	0.98	96
g	1.00	1.00	1.00	95
h	1.00	1.00	1.00	103
i	0.98	1.00	0.99	90
j	1.00	0.97	0.98	100
k	0.98	1.00	0.99	93
l	0.98	0.96	0.97	113
m	0.86	0.98	0.92	98
n	0.95	0.91	0.93	104
o	0.96	0.95	0.96	103
p	1.00	0.98	0.99	100
q	0.97	0.99	0.98	94
r	0.99	0.98	0.99	110
s	0.97	0.95	0.96	93
space	0.98	0.97	0.98	111
t	0.99	0.93	0.96	101
u	0.95	0.93	0.94	85
v	0.95	0.98	0.97	106
w	0.99	1.00	1.00	109
x	0.97	0.95	0.96	92
y	0.97	0.98	0.97	87
z	1.00	1.00	1.00	107
accuracy			0.97	2000
macro avg	0.97	0.97	0.97	2000
weighted	0.97	0.97	0.97	2000

avg				
-----	--	--	--	--

Table -1: Table Illustrates the Accuracy of the Algorithm

The confusion matrix package is used to analyse the model's real-time performance. It generates a confusion matrix that counts the number of correctly predicted labels. It also allows for the depiction of prediction deviation when the model makes incorrect predictions.

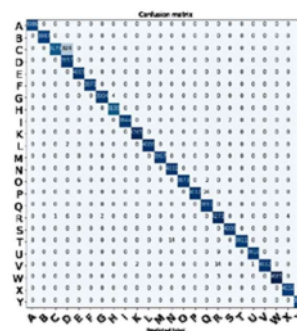


Fig -8: Depicts the model's confusion matrix

5.2 Real-Time Recognition

Figure-9 shows snapshots from a real-time recognition system built with the OpenCV open-source library and the model. The model can distinguish live motions and form sentences through auto correct mechanism.



Fig -9: Real Time Sign Recognition on Webcam Video Stream

5.3 Comparative Analysis

Table-2 shows a comparison of the results obtained in this investigation with those found in the literature. The table below summarizes the work done in relation to the 2D image and vision-based method.

Sign Language	Authors	Approach	Accuracy	Year of Development
American Sign Language	Das et al. [5]	CNN based Inception V3 model	90%	2018
Indian Sign Language	Rekha et al. [8]	Skin Color segmentations with SVM	86.3%	2011
Indian Sign Language	Sahoo [7]	k-NN and Naive Bayes classifier	98%	2021
American, Indian, Italian and	Halder and Tayade	MediaPipe with SVM	99%	2021

Turkey Sign	[12]			
-------------	------	--	--	--

Table -2: Comparative Analysis

6. CONCLUSION

In regional Indian languages with different sets of alphabets, sign language recognition is a challenge that this work seeks to address visually. The solution was implemented utilising powerful technology such as MediaPipe employing a cutting-edge methodology. By training a self-generated 2D picture dataset to a CNN-RF Neural Network, both real-time and static motions were attempted to be identified. According to the classification results, this approach of sign language recognition using the MediaPipe hand tracking solution is more effective and faster for classifying complex hand signs and gestures, including alphabets, than other models in the literature, which require high computation power and longer training time.

Furthermore, the use of MediaPipe provides precise monitoring of hand movements with various motions in the finger phalanges and finger joint aberrations. Furthermore, because of its lightweight nature, the model becomes more durable and can be deployed across a variety of computing devices with varying computational power without sacrificing speed or accuracy.

This study may be expanded to incorporate identification of more American Sign Language signals, including dynamic movements used in everyday conversation. After installing MediaPipe's hand tracking solution, several deep learning approaches may be evaluated to improve the model's accuracy and efficiency.

REFERENCES

- [1] A. Z. Shukor, M. F. Miskon, M. H. Jamaluddin, F. bin Ali, M. F. Asyraf, M. B. bin Bahar and others. (2015) "A new data glove approach for Malaysian sign language detection," Procedia Computer Science, vol. 76, p. 60-67
- [2] M. Mohandes, M. Deriche and J. Liu. (2014) "Image-based and sensor-based approaches to Arabic sign language recognition," IEEE transactions on human-machine systems, vol. 44, p. 551-557.
- [3] N. M. Kakoty and M. D. Sharma. (2018) "Recognition of sign language alphabets and numbers based on hand kinematics using a data glove," Procedia Computer Science, vol. 133, p. 55-62.
- [4] Continuous dynamic Indian Sign Language gesture recognition with invariant backgrounds by Kumud Tripathi, Neha Baranwal, G. C. Nandi at 2015 Conference on Advances in Computing, Communications and Informatics (ICACCI)

- [5] Bernard Boulay, Francois Bremond, Monique Thonat, Human Posture Recognition in Video Sequence. IEEE International Workshop on VSPETS, Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2003, Nice, France.
- [6] Z. A. Ansari and G. Harit. (2016) "Nearest neighbour classification of Indian sign language gestures using kinect camera," *Sadhana*, vol. 41, p. 161-182.
- [7] A. Das, S. Gawde, K. Suratwala and D. Kalbande. (2018) "Sign language recognition using deep learning on custom processed static gesture images," in *International Conference on Smart City and Emerging Technology (ICSCET)*.
- [8] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee and others. (2019) "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*.
- [9] Long Short-Term Memory, Sepp Hochreiter et al., *Neural Computation* 9(8): 1735-1780, 1997
- [10] A. K. Sahoo. (2021) "Indian sign language recognition using machine learning techniques," in *Macromolecular Symposia*.
- [11] Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S. Chen and Thomas S. Huang (2003, February). Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding* 91.
- [12] J. Rekha, J. Bhattacharya and S. Majumder. (2011) "Shape, texture and local movement hand gesture features for indian sign language recognition," in *3rd international conference on trends in information sciences & computing (TISC2011)*.
- [13] M. K. Bhuyan, M. K. Kar and D. R. Neog. (2011) "Hand pose identification from monocular image for sign language recognition," in *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*.
- [14] N. Pugeault and R. Bowden. (2011) "Spelling it out: Real-time ASL fingerspelling recognition," in *2011 IEEE International conference on computer vision workshops (ICCV workshops)*.
- [15] Recognition of Isolated Indian Sign Language Gesture in Real Time, Anup Nandy, Jay Shankar Prasad, Soumik Mondal, Pavan Chakraborty, G. C. Nandi, *Communications in Computer and Information Science book series (CCIS, volume 70)*
- [16] C. Keskin, F. Kırac, Y. E. Kara and L. Akarun. (2013) "Real time hand pose estimation using depth sensors," in *Consumer depth cameras for computer vision*, Springer, p. 119-137.
- [17] Carol Neidle, Ashwin Thangali and Stan Sclaroff [2012] "Challenges in Development of the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus," *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC 2012, Istanbul, Turkey, May 27, 2012*.
- [18] Going Deeper with Convolutions, Szegedy et al. *CVPR 2015, IEEE Explore*.
- [19] A. Halder and A. Tayad. (2021) "Real-time vernacular sign language recognition using mediapipe and machine learning," *Journal homepage: www.ijrpr.com ISSN, vol. 2582, p. 7421*.
- [20] Adam: A Method for Stochastic Optimization, Diederik P. Kingma, Jimmy Ba, Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015
- [21] D. A. Kumar, A. S. C. S. Sastry, P. V. V. Kishore and E. K. Kumar. (2022) "3D sign language recognition using spatio temporal graph kernels," *Journal of King Saud University-Computer and Information Sciences*.
- [22] Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN, Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, Yanbo Gao, *CVPR 201*
- [23] Dynamic Routing Between Capsules, Geoffrey Hindon et al, Nov 2017
- [24] Z. Ren, J. Yuan and Z. Zhang. (2011) "Robust hand gesture recognition"
- [25] Pal, M. Random forest classifier for remote sensing classification. *Int. J. Remote Sens.* 2005, 26, 217-222. [Google Scholar]
- [26] Riyanto, I.; Rizkinia, M.; Arief, R.; Sudiana, D. Three-Dimensional Convolutional Neural Network on Multi-Temporal Synthetic Aperture Radar Images for Urban Flood Potential Mapping in Jakarta. *Appl. Sci.* 2022, 12, 1679. [Google Scholar]
- [27] Belenguer-Plomer, M.A.; Tanase, M.A.; Fernandez-Carrillo, A.; Chuvieco, E. CNN-based burned area mapping using radar and optical data. *Remote Sens. Environ.* 2021, 260, 112468. [Google Scholar]
- [28] Scherer, D.; Müller, A.; Behnke, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN) 2010, Thessaloniki, Greece, 15-18 September 2010*; pp. 92-101. [Google Scholar]
- [29] Wang, Y.; Fang, Z.; Hong, H.; Peng, L. Flood susceptibility mapping using convolutional neural network frameworks. *J. Hydrol.* 2020, 582, 124482. [Google Scholar]

A Deep Learning Based Model To Recognize Sign Language

ORIGINALITY REPORT

45%
SIMILARITY INDEX

20%
INTERNET SOURCES

39%
PUBLICATIONS

9%
STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|----------|--|------------|
| 1 | Jyotishman Bora, Saine Dehingia, Abhijit Boruah, Anuraag Anuj Chetia, Dikhit Gogoi. "Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning", Procedia Computer Science, 2023
Publication | 23% |
| 2 | www.mdpi.com
Internet Source | 6% |
| 3 | Submitted to National Institute Of Technical Teachers' Training & Research
Student Paper | 5% |
| 4 | Rajesh George Rajan, M. Judith Leo. "American Sign Language Alphabets Recognition using Hand Crafted and Deep Learning Features", 2020 International Conference on Inventive Computation Technologies (ICICT), 2020
Publication | 2% |
| 5 | Dodi Sudiana, Anugrah Indah Lestari, Indra Riyanto, Mia Rizkinia et al. "A Hybrid Convolutional Neural Network and Random | 2% |

Forest for Burned Area Identification with Optical and Synthetic Aperture Radar (SAR) Data", Remote Sensing, 2023

Publication

6	ijrsrset.com Internet Source	1 %
7	Submitted to University of Northumbria at Newcastle Student Paper	1 %
8	ijircce.com Internet Source	1 %
9	"Mobile Radio Communications and 5G Networks", Springer Science and Business Media LLC, 2022 Publication	1 %
10	users.cecs.anu.edu.au Internet Source	1 %
11	www.researchgate.net Internet Source	<1 %
12	Ankit Kumar, Vijayakumar Varadarajan, Abhishek Kumar, Pankaj Dadheech et al. "Black hole attack detection in vehicular ad-hoc network using secure AODV routing algorithm", Microprocessors and Microsystems, 2021 Publication	<1 %

13	Kshitij Bantupalli, Ying Xie. "American Sign Language Recognition using Deep Learning and Computer Vision", 2018 IEEE International Conference on Big Data (Big Data), 2018 Publication	<1 %
14	medium.com Internet Source	<1 %
15	Submitted to Kennesaw State University Student Paper	<1 %
16	hoge huga.com Internet Source	<1 %
17	link.springer.com Internet Source	<1 %
18	sumc.lt Internet Source	<1 %
19	www.slideshare.net Internet Source	<1 %
20	Eunice Jennifer Robert, Hemanth Jude Duraisamy. "A review on computational methods based automated sign language recognition system for hearing and speech impaired community", Concurrency and Computation: Practice and Experience, 2023 Publication	<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On