International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 07 | July 2023 www.irjet.net p-ISSN: 2395-0072

Real Time Sign Language Translation Using Tensor Flow Object Detection

John Pius Thayiparampil[1], Kiran k[2], T Binto Binu[3], Sreejith Viju[4]

[1-4]BTECH UG Students ,Department of Computer Science and Engineering ,TOMS college of Engineering APJ Abdul Kalam Technological University, Kerala ,India

---***---

Abstract - *Deaf and mute people frequently use sign language to exchange information. In India, there are 63 million people who have severe hearing loss. At least 76–89 .Due to their limitations, people with speech and hearing impairments frequently have trouble communicating. Translation systems are required to help bridge the gap in under- standing that exists between the hearing and speech-impaired groups and the general population. So introducing a real-time translates sign language is one solution to this issue. It will be able to translate all of the varied gestures by gathering and analyzing motion data. Sign language can be read by humans in every- day culture. Using TensorFlow object identification, a novel real-time sign language translation has been developed that can translate sign language based on hand movements and is understandable to both average individuals and those with major hearing and speech impairments. The Software will detect gesticulation.*

Key Words: Real Time translation, Sign Language, Tensor Flow , Object Detection , Gesture detection, Hearing and speech Impairment.

1.INTRODUCTION

Motion detection tracks shifts in an object's position with relation to its surroundings, and vice versa. We can identify moving objects in front of the camera with the aid of this motion detection program. This software can be used to carry out the following activities, among others: [1] Take assist for education purpose for deaf and muted students and teachers; [2] helps in meeting and conference; [3] helps to get effective treatment in medical sector ; [4] for more effective and more involving communication. A very good way to lessen the communication gap.

This project report's main goal is to give a thorough description of the Sign Language Translator system that was created as part of Real time sign language translator. Through the use of this technique, people who are deaf or hard of hearing and non-sign language users can communicate more effectively. This initiative aims to improve inclusiveness for the deaf population and allow effective communication by utilizing cutting-edge technology and artificial intelligence. Computer vision and machine learning have made considerable strides in recent years, enabling improvements in a number of areas, including the translation and recognition of sign languages. TensorFlow Object identification, a potent framework for object identification tasks, will be used in this project to create a sign language translator. We can close the communication gap between the hearing-impaired community and the rest of society by utilizing the capabilities of deep learning and image processing. For people who have hearing impairments, sign language is an essential form of communication. The majority of people, however, lack the abilities needed to comprehend and interpret sign language. Effective communication and inclusion for the deaf and hard of hearing are severely hampered as a result. To solve this problem, we suggest creating a sign language translator that makes use of TensorFlow Object Detection, a cutting-edge tool for instantly identifying and classifying items.

1.2 GESTURE RECOGNITION

Gesture recognition acts as a translator between computers and people and enables computers to understand human behaviors. This would enable natural human-computer interaction without putting any extra mechanical components in direct contact with users. The sign language used by the deaf and dumb community is gesture-based. When it was difficult to transmit audio or when typing and writing were difficult but there was still the possibility of vision, this group relied on sign language for communication.

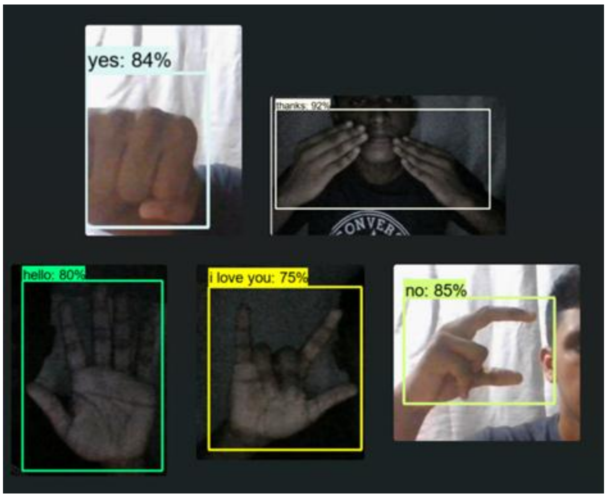


Fig -1: Gesture Recognition

People could only communicate with each other at that time using sign language. Sign language is often used when someone doesn't want to speak; nevertheless, for the community of the deaf and dumb, this is their only form of communication.

1.3 LITERATURE REVIEW

A defined collection of hand gestures with specific meanings used by hearing-impaired people to communicate in daily life is referred to as a sign language [3]. They communicate through body, face, and hand movements since they are visual languages. Around the world, there are more than 300 different sign languages [5]. Even though there are numerous distinct sign languages, only a small portion of the community is conversant in any of them, making it challenging for people with special needs to freely interact with the general public. SLR offers a way to communicate with sign language even if you don't know it. It can recognize gestures and translate them into languages like English that are widely spoken.

SLR is a fairly broad area of study, and while much work has been done in this area, there are still many issues that need to be resolved. The electronic systems are now able to make decisions based on data and experience thanks to machine learning algorithms. The training dataset and testing dataset are required by the classification algorithms. The classifier learns from the training set's experiences, and the model is evaluated using the testing set [6]. Numerous authors have created effective strategies for data collection and classification [3][7]. Previous research can be divided into two categories based on the data collecting method: direct measurement methods and vision-based approaches [3]. Particle filters and Kalman [10][12].

2. METHODOLOGY

With the use of a TensorFlow object identification API, the proposed system is designed to create a real-time sign language detector and train it using transfer learning for the created dataset [37]. Following the methods indicated in Section 3, images from a camera are gathered for data acquisition using Python and OpenCV.

A labelled map that represents all the model's objects and includes their id and the label for each letter of the alphabet is created after the data collection. The alphabet's first 26 letters are represented by one of the 26 labels on the label map. Each label has a unique ID with a value between 1 and 26. The open-source framework TensorFlow's object detection API makes it straightforward to build, train, and use an object identification model. They offer a variety of detection models that have previously been pre-trained on the COCO 2017 dataset through their technology, the TensorFlow detection model zoo. The pre-trained TensorFlow model utilized is SSD MobileNet v2 320x320. The SSD MobileNet v2 Object identification model is combined with the FPN-lite feature extractor, shared box predictor, and focus loss on training images scaled to 320x320. Pipeline configuration, or setting up and altering the settings of the pre-trained model to train it using the recently created dataset, is finished. Dependencies such as TensorFlow, con- fig_util, pipeline_pb2, and text_format have been imported for configuration. The main adjustment was to reduce the number of classes originally 90 to 26, which corresponds to the number of signs (alphabets) the model will be trained on. The model was trained in 10000 steps after configuration setup and upgrading. The hyper-parameter that was employed during training was to establish the model's training step count, which was set at 10,000 steps. The model experiences classification loss, regularization loss, and localization loss during training. Between the true values and the anticipated bounding box correction, the localization loss is times out of whack. Eq. (1) to (5) contains the localization loss's [38] formula.

On training images scaled to 320x320, the FPN-lite feature extractor, shared box predictor, and focus loss are integrated with the SSD MobileNet v2 Object identification model. The pre-trained model's settings have been set up and modified to allow it to be trained using the just-created dataset. This process is known as pipeline configuration.

$$L_{loc}(\mathrm{x}, \mathrm{l}, \mathrm{g}) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} smooth_{L1}(l_i^m - \hat{g}_j^m) \quad (1)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \quad (2)$$

$$\hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h \quad (3)$$

$$\hat{g}_j^{w} = \log(g_j^{w}/d_i^{w}) \quad (4)$$

$$\hat{g}_j^{h} = \log(g_j^{h}/d_i^{h}) \quad (5)$$

l is the anticipated bounding box, g is the ground truth bounding box, and N is the number of matching default boxes.

$$L_{con\,f}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \quad (6)$$

where, $\hat{c}_i^p = \exp(c_i^p) / \sum_p \exp(c_i^p)$ is the softmax activated class score for default box i with category p, x_{ij}^p is the matching indicator between default box i and the ground truth box j of category p.

In the section that follows, the various losses that were sustained during the experiment are discussed. Following training, the model is loaded from the most recent checkpoint, enabling real-time detection. The model will be prepared for training after configuration setup and updates.

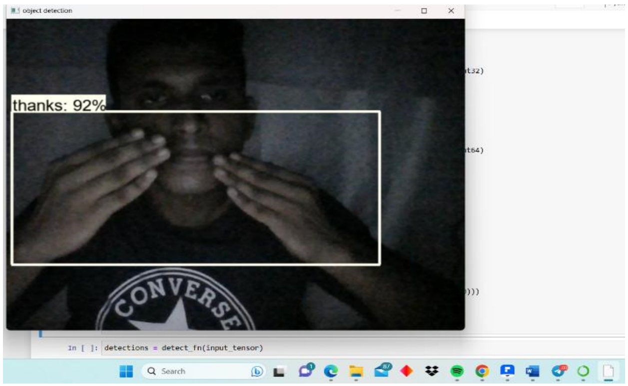


Fig-2: Real Time Sign Language Detection using TensorFlow

The most recent checkpoint generated during model training is used to load the trained model. The model is now finished and prepared for real-time sign language-Detection. The webcam and OpenCV are once again used for real-time detection. Real-time detection is accomplished using CV2 and NumPy dependencies. As depicted in Fig. 2, the system recognizes signs in real-time and translates each gesture's meaning into English. The system is continuously tested by producing and displaying various indicators

2.1 DATA ACQUISITION

For Sign Language, a real-time sign language identification system is being created. Python and OpenCV are used to capture webcam photos for data acquisition. Real-time computer vision is the primary focus of OpenCV's functions. It expedites the incorporation of artificial intelligence into commercial goods and offers a standard infrastructure for computer vision-based applications. More than 2500 effective computer vision and machine learning algorithms are available in the OpenCV library, which may be used for a variety of tasks, including face and object recognition, classification of human behaviors, object identification, tracking camera and object movements, and the extraction of 3D object models [35].The produced dataset is made up of signs that, as illustrated in Fig. 1, correspond to Indian Sign Language alphabets [36].

Fig-3: Selecting a portion of the image to label using labeling

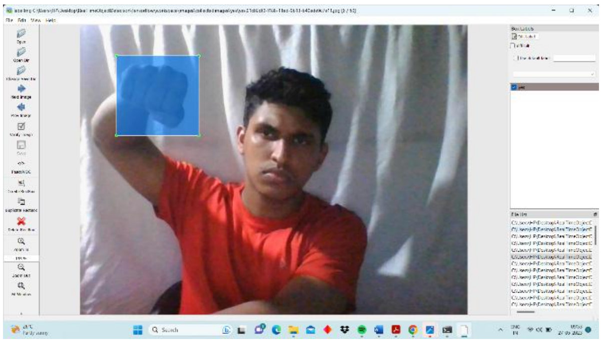
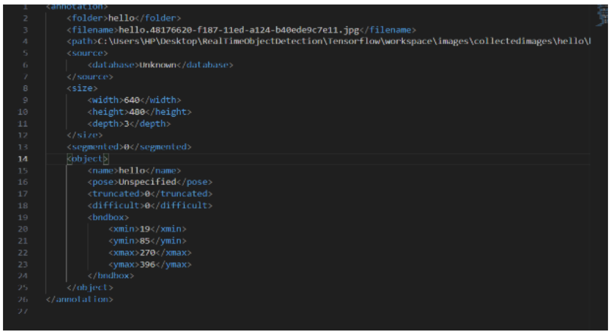


Fig-4: Labelling the selected portion using labeling

To create the dataset, 50 photos for each letter of the alphabet are taken. The photos are taken every two seconds, allowing for time to record gestures with slight variations each time, and a buffer of five seconds is provided between two separate signs, i.e., to go from the sign of one letter to the sign of another alphabet. The photographs that were captured are kept in the appropriate folder.



```
<annotation>
    <folder>hello</folder>
    <filename>hello.48176620-f187-11ed-a124-b40ede9c7e11.jpg</filename>
    <path>C:\Users\HP\Desktop\RealTimeObjectDetection\Tensorflow\workspace\images\collectedimages\hello\
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>640</width>
        <height>480</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>hello</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>19</xmin>
            <ymin>85</ymin>
            <xmax>270</xmax>
            <ymax>396</ymax>
        </bndbox>
    </object>
</annotation>
```

Fig-5: Convert into xml file Labelled portion using labeling

2.2 IMPLEMENTATION

The dataset was made for Sign Language, where the signs are English alphabets. The dataset is produced using the

data collection technique outlined in Section 3 of this document. The experiment was conducted on a computer running Windows 11 that featured an Intel Ryzen 5 4600H 4 GHz processor, 8 GB of memory, and a webcam (HP TrueVision HD camera with 0.31 MP and 640x480 resolution). The development environment consists of TensorFlow Object Detection API, Jupyter Notebook, OpenCV (version 4.2.0), and Python (version 3.7.3).

2.3 RESULT

The developed system is able to detect Indian Sign Language alphabets in real-time. The system has been created using TensorFlow object detection API. The pre-trained model that has been taken from the TensorFlow model zoo is SSD MobileNet v2 320x320. It has been trained using transfer learning on the created dataset which contains 500 images in total, 25 images for each alphabet. According to Fig. 6, the total loss experienced during the final 10,000 steps of training was 0.25; localization loss was 0.18; classification loss was 0.13; and regularization loss was 0.10. The lowest loss of 0.17, as seen in Fig. 6, occurred at Steps 9900.

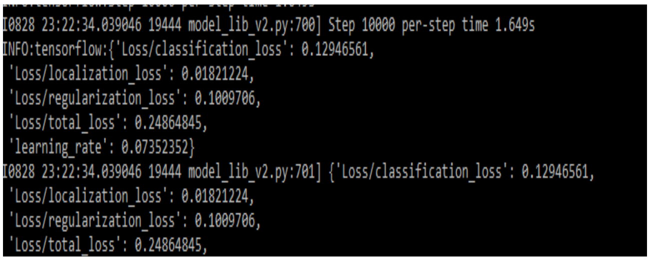


Fig-6: Losses sustained at various steps

The Indian Sign Language Recognition system's cutting-edge technique achieved 93–96% accuracy [5]. It is not a real-time SLR system, despite being quite accurate. This study addresses this problem. Despite the limited dataset, our system has managed to reach an average confidence rate of 87.4 percent.

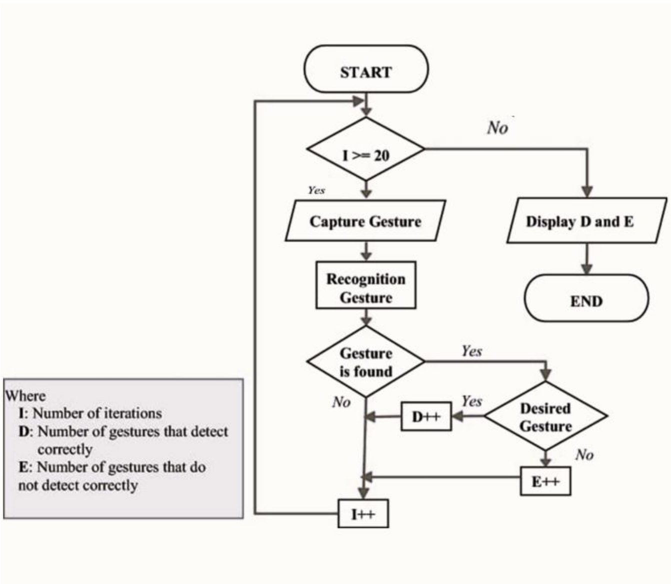


Fig 7: Flowchart to check the accuracy of the translator

3. CONCLUSIONS

The development of a real-time sign language translator using TensorFlow Object Detection offers significant potential for bridging communication gaps between sign language users and non-signers. By leveraging the power of object detection models and the latest version of TensorFlow Object Detection, the translator can accurately detect and localize sign gestures in real-time, enabling efficient and effective translation into spoken or written language. The performance evaluation of such a system from an end user's perspective highlights the importance of accuracy in sign detection, speed and responsiveness, robustness to variations, translation accuracy, user-friendly interface and interaction, adaptability, usability, accessibility, and effective error handling.

These factors contribute to a seamless user experience and facilitate effective communication between sign language users and non-signers. While there is still room for further improvements and advancements, the real-time sign language translator demonstrates promising results in enhancing inclusively and accessibility. It has the potential to empower sign language users by enabling them to communicate more easily and effortlessly in various contexts, such as educational, professional, or social settings. Continued research and development in the field of sign language translation, combined with the advancements in TensorFlow Object Detection and other related technologies, hold great promise for the future. As technology progresses, we can expect further improvements in accuracy, speed, robustness, and user experience, ultimately contributing to a more inclusive and connected society where communication barriers are reduced or eliminated for sign language users.

3.1 FUTURE WORK

In sign language translator may have a very good scope in sign language translating video calling app. The Python programming language can be used to translate sign language systems in a number of ways, some of which are:

Future sign language translators can strive for higher accuracy in recognizing and interpreting sign language gestures. Machine learning algorithms can be further refined and trained on larger datasets, enabling better understanding of subtle hand movements and facial expressions. Currently, many sign language translators require a slight delay in processing and translating signs. Future advancements may focus on reducing this latency, allowing for real-time translation, which would greatly enhance communication between sign language users and non-signers. Recognition of facial expressions: Recognizing facial expressions could be another addition made to these systems.

ACKNOWLEDGEMENT

The task has been successfully completed, and I am very appreciative to God Almighty for his grace and blessings, without which this study would not have been possible.

REFERENCES

[1] Kapur, R.: The Types of Communication. MIJ. 6, (2020).

[2] Suharjito, Anderson, R., Wiryana, F., Ariesta, M.C., Kusuma, G.P.: Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output. Procedia Comput. Sci. 116, 441–

448 (2017). https://doi.org/10.1016/J.PROCS.2017.10.028

[3] Konstantinidis, D., Dimitropoulos, K., Daras, P.: Sign language recognition based on hand and body skeletal data. 3DTV-Conference. 2018-June, (2018). https://doi.org/10.1109/3DTV.2018.8478467.

[4] Dutta, K.K., Bellary, S.A.S.: Machine Learning Techniques for Indian Sign Language Recognition. Int. Conf. Curr. Trends Comput. Electr. Electron. Commun. CTCEEC 2017. 333–336 (2018). https://doi.org/10.1109/CTCEEC.2017.8454988.

[5] Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., Morris, M.R.: Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. 21st Int. ACM SIGACCESS Conf. Comput. Access. (2019). https://doi.org/10.1145/3308561.

[6] Rosero-Montalvo, P.D., Godoy-Trujillo, P., Flores-Bosmediano, E., Carrascal-Garcia, J., Otero-Potosi, S., Benitez-Pereira, H., Peluffo-Ordonez, D.H.: Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques. 2018 IEEE 3rd Ecuador Tech. Chapters Meet. ETCM 2018. (2018). https://doi.org/10.1109/ETCM.2018.8580268.

[7] Zheng, L., Liang, B., Jiang, A.: Recent Advances of Deep Learning for Sign Language Recognition. DICTA 2017 - 2017 Int. Conf. Digit. Image Comput. Tech. Appl. 2017-Decem, 1–7 (2017). https://doi.org/10.1109/DICTA.2017.8227483.

[8] Rautaray, S.S.: A Real Time Hand Tracking System for Interactive Applications. Int. J. Comput. Appl. 18, 975–8887 (2011).

[9] Zhang, Z., Huang, F.: Hand tracking algorithm based on super-pixels feature. Proc. - 2013 Int. Conf. Inf. Sci. Cloud Comput. Companion, ISCC-C 2013. 629–634 (2014). https://doi.org/10.1109/ISCC-C.2013.77..

[10] Lim, K.M., Tan, A.W.C., Tan, S.C.: A feature covariance matrix with serial particle filter for isolated sign language recognition. Expert Syst. Appl. 54, 208–218 (2016). https://doi.org/10.1016/J.ESWA.2016.01.047.

[11] Lim, K.M., Tan, A.W.C., Tan, S.C.: Block-based histogram of optical flow for isolated sign language recognition. J. Vis. Commun. Image Represent. 40, 538–545 (2016). https://doi.org/10.1016/J.JVCIR.2016.07.020.

[12] Gaus, Y.F.A., Wong, F.: Hidden Markov Model - Based gesture recognition with overlapping hand-head/hand-hand estimated using Kalman Filter. Proc. - 3rd Int. Conf. Intell. Syst. Model. Simulation, ISMS 2012. 262–267 (2012). https://doi.org/10.1109/ISMS.2012.67.

[13] Gaus, Y.F.A., Wong, F.: Hidden Markov Model - Based gesture recognition with overlapping hand-head/hand-hand estimated using Kalman Filter. Proc. - 3rd Int. Conf. Intell. Syst. Model. Simulation, ISMS 2012. 262–267 (2012). https://doi.org/10.1109/ISMS.2012.67.

[14] Mohandes, M., Aliyu, S., Deriche, M.: Arabic sign language recognition using the leap motion controller. IEEE Int. Symp. Ind. Electron. 960–965 (2014). https://doi.org/10.1109/ISIE.2014.6864742.

[15] Enikeev, D.G., Mustafina, S.A.: Sign language recognition through Leap Motion controller and input prediction algorithm. J. Phys. Conf. Ser. 1715, 012008 (2021). https://doi.org/10.1088/1742-6596/1715/1/012008.

[16] Cheok, M.J., Omar, Z., Jaward, M.H.: A review of hand gesture and sign language recognition techniques. Int. J. Mach. Learn. Cybern. 2017 101. 10, 131–153 (2017). https://doi.org/10.1007/S13042-017-0705-

[17] Cheok, M.J., Omar, Z., Jaward, M.H.: A review of hand gesture and sign language recognition techniques. Int. J. Mach. Learn. Cybern. 2017 101. 10, 131–153 (2017). https://doi.org/10.1007/S13042-017-0705-

[18] Camgöz, N.C., Koller, O., Hadfield, S., Bowden, R.: Sign language transformers: Joint end-to-end sign language recognition and translation. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 10020–10030 (2020). https://doi.org/10.1109/CVPR42600.2020.01004.

[19] Cui, R., Liu, H., Zhang, C.: A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training. IEEE Trans. Multimed. 21, 1880–1891 (2019). https://doi.org/10.1109/TMM.2018.2889563.

[20] Bantupalli, K., Xie, Y.: American Sign Language Recognition using Deep Learning and Computer Vision. Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018. 4896–4899 (2019). https://doi.org/10.1109/BIGDATA.2018.8622141.

[21] Hore, S., Chatterjee, S., Santhi, V., Dey, N., Ashour, A.S., Balas, V.E., Shi, F.: Indian Sign Language Recognition Using Optimized Neural Networks. Adv. Intell. Syst. Comput. 455, 553–563 (2017). https://doi.org/10.1007/978-3-319-38771-0_54.

[22] Kumar, P., Roy, P.P., Dogra, D.P.: Independent Bayesian classifier combination based sign language

recognition using facial expression. Inf. Sci. (Ny). 428, 30–48 (2018). https://doi.org/10.1016/J.INS.2017.10.046.

[23] Sharma, A., Sharma, N., Saxena, Y., Singh, A., Sadhya, D.: Benchmarking deep neural network approaches for Indian Sign Language recognition. Neural Comput. Appl. 2020 3312. 33, 6685–6696 (2020). https://doi.org/10.1007/S00521-020-05448-8.

[24] Kishore, P.V.V., Prasad, M. V.D., Prasad, C.R., Rahul, R.: 4-Camera model for sign language recognition using elliptical Fourier descriptors and ANN. Int. Conf. Signal Process. Commun. Eng. Syst. - Proc. SPACES 2015, Assoc. with IEEE. 34–38 (2015). https://doi.org/10.1109/SPACES.2015.7058288.

[25] Tewari, D., Srivastava, S.K.: A Visual Recognition of Static Hand Gestures in Indian Sign Language based on Kohonen Self-Organizing Map Algorithm. Int. J. Eng. Adv. Technol. 165 (2012).

[26] Gao, W., Fang, G., Zhao, D., Chen, Y.: A Chinese sign language recognition system based on SOFM/SRN/HMM. Pattern Recognit. 37, 2389–2402 (2004). https://doi.org/10.1016/J.PATCOG.2004.04.008.

[27] Quocthang, P., Dung, N.D., Thuy, N.T.: A comparison of SimpSVM and RVM for sign language recognition. ACM Int. Conf. Proceeding Ser. 98–104 (2017). https://doi.org/10.1145/3036290.3036322.

[28] Pu, J., Zhou, W., Li, H.: Iterative alignment network for continuous sign language recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2019-June, 4160–4169 (2019). https://doi.org/10.1109/CVPR.2019.00429.

[29] Kalsh, E.A., Garewal, N.S.: Sign Language Recognition System. Int. J. Comput. Eng. Res. 6.

[30] Singha, J., Das, K.: Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique. IJACSA) Int. J. Adv. Comput. Sci. Appl. 4, (2013).

[31] Liang, Z., Liao, S., Hu, B.: 3D Convolutional Neural Networks for Dynamic Sign Language Recognition. Comput. J. 61, 1724–1736 (2018). https://doi.org/10.1093/COMJNL/BXY049.

[32] Pigou, L., Van Herreweghe, M., Dambre, J.: Gesture and Sign Language Recognition with Temporal Residual Networks. Proc. - 2017 IEEE Int. Conf. Comput. Vis. Work. ICCVW 2017. 2018-Janua, 3086–3093 (2017). https://doi.org/10.1109/ICCVW.2017.365

[33] Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W.: Video-based Sign Language Recognition without Temporal Segmentation

[34] Cui, R., Liu, H., Zhang, C.: Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. 2017-Janua, 1610–1618 (2017). https://doi.org/10.1109/CVPR.2017.175.

[35] About – OpenCV

[36] Poster of the Manual Alphabet in ISL | Indian Sign Language Research and Training Center (ISLRTC), Government of India.

[37] Transfer learning and fine-tuning | TensorFlow Core.

[38] Wu, S., Yang, J., Wang, X., Li, X.: IoU-balanced Loss Functions for Single-stage Object Detection. (2020).

[39] Kapur, R.: The Types of Communication. MIJ. 6, (2020)..