Hand sign detection using deep learning single shot detection technique

Narendra Kumar
School of Computing
DIT University Dehradun
Uttarakhand , India
narendra298@gmail.com

Atul Kumar Singh Bisht
School of Computing
DIT University Dehradun
Uttarakhand , India
5atulbisht@gmail.com

2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN) | 979-8-3503-4798-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/VITECON58111.2023.10157550

***Abstract*— Communication with deaf and dumb people has always been difficult because very few people understand sign language, and the communication gap is growing, resulting in a lack of social interaction and missed opportunities, such as a technical interview. Various works had previously been done to solve this problem, but the previous work in this area had some flaws, such as the inability to interpret the signs in real time. This work was not able to solve this problem properly. To overcome this communication barrier, we have created a realtime hand sign language detection model that can detect many hand signs and Indian alphabet sign language in real time and interpret their meaning for us. We built this model with the TensorFlow object detection API, a Zoo-trained TensorFlow model, OpenCV, and several Python libraries. We have created and used our own data set using OpenCV and the camera. We have created a diverse and large data set to increase the accuracy of the model. The end result of the paper is that our trained model can understand the sign that a person is making through our camera and interpret its meaning.**

Keywords*— *Hand sign, deep learning, single shot detection, deaf people, image detection

I. Introduction

When two individuals who are deaf or hard of hearing communicate using sign language, the information is transmitted mostly via hand and arm motions. This makes sign language the most expressive mode of communication between those with hearing loss. When it comes to the development of gesture-based human–computer interaction systems, sign language recognition plays a significant and prominent role [1]. Throughout the most recent decades of statistics, there has been a growing interest in studying the interface with intelligent computers. Computers have evolved into an essential component of our society for interaction in the modern era. While hand signals are a strong interacting human communication modality, they are also an intuitive and convenient means of communication for both hearing impaired and normal humans. This is because hand signs can be used by both groups. This is due to the fact that hand signals comprise an effective mode of human connection and communication. Static signs and dynamic signs make up the two primary categories that may be applied to the signage. The movement of various parts of the body is a common feature of dynamic signs. Depending on the message that the gesture is trying to express, it might also incorporate feelings.

When we speak about communication, what we're referring to is the act of passing on information from one place, person, or community to another, whether that be physically or virtually. There are three parts to it: the person doing the talking, the message being transmitted, and the person receiving the information.

People who are deaf or dumb may communicate more effectively with one another and with others when they use non-verbal communication. Deafness is a hinder that prevents a person from being able to hear and impacts their capacity to hear sounds and speech. On the other side, a person's capacity to speak is negatively impacted by dumbness, rendering them unable to communicate verbally. When a person is unable to converse or listen to others, it may be quite difficult for them to interact with other people.

Because of this, sign languages play an important role in communication since they make it possible for people to exchange ideas without using spoken language. However, there is still a problem because only a small percentage of people are fluent in sign language. People who are deaf may well be able to interact with one another using languages of sign. However, it is challenging for them to engage in conversation with others. who have normal hearing since most people are not familiar with sign languages, and vice versa. This problem can be fixed by using a technology-based solution. With the use of a solution like this one, it is simple to convert the hand motions used in gestures into the language that is most widely spoken, which is English.

A hand gesture is a non-verbal way of communicating that involves moving the fingers to show what you want to say. People who are deaf or mute may communicate with one another via hand gestures, which are also employed in sign language. Gestures can also be used to operate equipment, such as those that are connected to the Internet of Things (IoT) [11]-[17].Hand gestures have grown in popularity in recent years because they help to bridge the communication gap between people who are deaf and those who can speak [2]-[5] because it is an object that consists of distinct features that must be extracted and recognized before the gestures or signs can be accurately identified. The photos go through several stages of image processing. Image capture, preliminary processing, segmentation, feature extraction, and classifiers are the components that make up the processing phases. Machine learning and image processing approaches and algorithms are used here in order to achieve a higher degree of precision in the results. The purpose of the research paper is to build a real-time system using the TensorFlow object recognition API and then train it using a dataset that will be constructed with the assistance of a camera.

II. LITERATURE REVIEW

In the literature, many algorithms have been proposed for hand gesture recognition.

In [1], the method that researchers suggest for commanding a computer uses eight dynamic and six static hand movements. Hand form detection, tracking of detected hands (if the hand is dynamic), and converting detected data into the appropriate command are the three basic procedures.

In [6], the authors describe a method for recognizing hand gestures that can be used to figure out the alphabets of Indian Sign Language. The genetic algorithm is used for the purpose of gesture recognition. Accurate recognition of single-handed and two-handed movements is possible using the method that researchers suggest using, which is user-friendly and does not need a lot of money.

In[7], work describes an efficient method for recognizing hand gestures, along with the ability to choose hand features for use with low-cost video capture devices. In the model that has been developed, wavelet transforms and decomposition of singular values are the techniques that are used to extract hand information from video frames. To improve the performance of the hand sign identification system, a genetic algorithm that has an efficient fitness function is employed to pick the ideal hand characteristics, which involves removing redundant and unnecessary information.

In [8], researchers build an AdaBoost classifier and skin color model based on hand gestures to account for the uniqueness of hand gestures in terms of skin tone. The authors study a single frame from a movie to learn more about hand movements. Here, the CamShift algorithm not only separates the human hand from the complex backdrop, but also does real-time hand motion tracking.

In [9], developed a quick and easy approach based on motion history images for categorizing dynamic hand movements. The classifiers for up, down, left, and right hand motions were taught to recognize four distinct hear-like directional patterns. Six distinct hand motions were specified, including the familiar "fist" and "waving" movements.

In [10], with the objective of enhancing hand gesture identification, researchers offer a novel method for handwritten text recognition. This method makes use of the hand's natural motions by analyzing its edges and the directions in which the hands are moving. A further improvement is the use of deterministic finite automata and fuzzy logic to increase accuracy in handwriting recognition.

In [11], the method for identifying hand gestures by looking at them is broken down into three steps: feature extraction, preprocessing, and classification. The purpose of this phase of processing is to identify the hand area in the picture. To identify the hand's boundary, image data of hand gestures is processed using a Laplacian of Gaussian filters and a zero crossing detector. In this work, the authors offer a new method for extracting features, one that makes use of the LHFD (local histogram feature descriptor). The suggested feature is found by getting the local histogram of the grayscale gesture picture.

In [18], researchers offer a neural network convolution based hand gesture identification system that uses multi-view gestures as training data. This is done to get around the problem of self-occlusion. For training and testing, each gesture is put through a series of thorough tests for every possible combination of multi view sets.

In [19], researchers describe a way to use Microsoft's Kinect sensor to understand hand gestures. Kinect enables real-time, high-resolution, three-dimensional scanning of an object. Researchers advocate a strategy that combines modeling Kinect's depth function is used to remove the backdrop from photos of hand gestures. The segmented hand pictures' borders are located using techniques for image processing [2].

In [20], this study implements a recognition system for hand gestures with a facial recognition system to help with a wide range of activities. Algorithms are applied to a dynamic video in order to extract dynamic visuals. In the Gesture System, color of skin detection has been carried out in the YCbCr color space, and the convex defect characteristic point of the hand has been utilized to uncover various elements such as the fingertips and the angle between the fingers.

In [21], it has been suggested that an unique strategy be used for the purposes of recognizing and classifying, and many other generally popular models have been contrasted with it. Principal Component Analysis, Histogram of Gradients, were used in the construction of the innovative model.

There are various other work already done in this field but the problem is that it can recognize limited amount of signs and the data is set is not very big and diverse so that it can detect the sign in different condition like low light ,low camera quality etc. Our model can recognise Indian sign language and various different signs with good accuracy.

III. METHODOLOGY AND PROPOSED WORK

The primary purpose of that research is to create a model that recognizes the hand signs of a person who cannot speak and a deaf person who cannot read our wordings.

With the help of the TensorFlow object detection API and Python, the model will be able to understand certain hand signs and the alphabet signs in the real time . we have created our own data set with the help of a Python library labelimg and used it for training and testing our model which is built with the tensorflow 2 model zoo and deep learning ssd ,the end result will we can understand the hand sign and we will get the meaning of those sign as words in our screen .the whole process is shown in the figure 1 and the work flow .

This whole process begins with the collection of a picture, which is then divided into a series of smaller rectangles. In order to determine whether or not a certain segmented rectangular region includes a legitimate item, feature extraction is performed on the area in question. Combining overlapping boxes into a single enclosing rectangle results in the desired effect. Suppression that is not at its maximum. These all process is carried out by providing our own dataset in the pre trained zoo model and how our model is different is the accuracy of the model and the vast number of hand sign it can recognise ,it can recognise more than 30 signs.

Work flow of proposed model are given below:

*1) Data set collection :*First step is to collect all the images for training and testing with the help of camera and opencv ,which is a library use for real time computer vision .30 images of each sign is collected that is further splited in testing purpose and training . 80 % of the images(data) is for the training purpose and rest 20% of images(data) is used for testing purposes .after the data is collected .

2) Create Label map : Now after we have collected all the images ,and we have to label them with their respective signs so that we can train the model with that .To do this work we have a python library called labelImg as show in the figure 2 which label each image with its sign and create a label map and a xml file respective for each image .

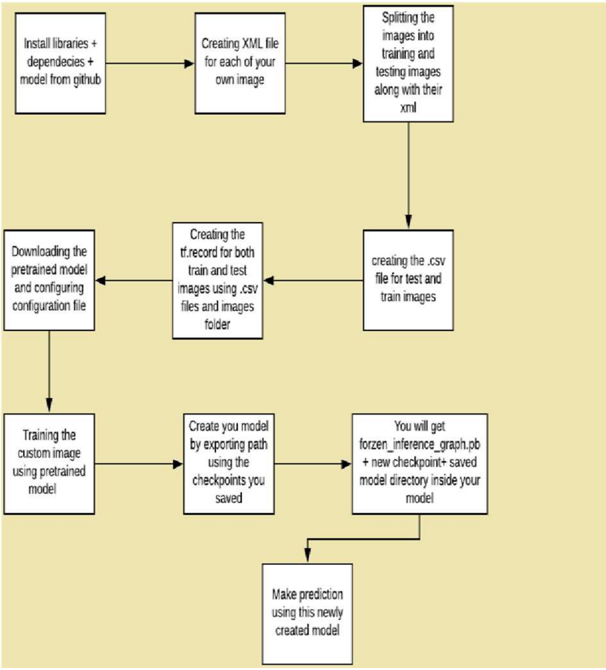
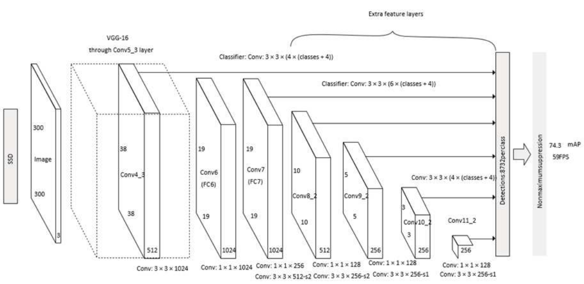


Fig. 1. Flow chart of proposed method

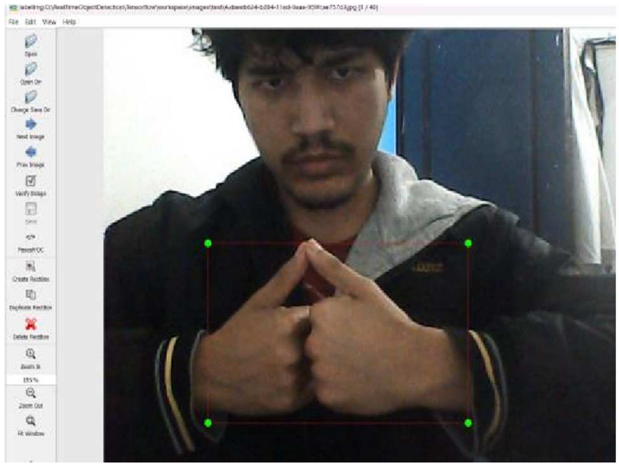
*3)TF RECORD :*Now in order to store all the training and testing data in the form of binary format we have created TF record.

*4)IMPORT Tensorflow2 zoo model :*We have used a predefined model of tensorflow 2 zoo which is used for the object detection. SSD MobileNet with version two 320x320 is the pre-trained TensorFlow model that is currently being used. Using training pictures scaled to 320x320, the SSD MobileNet version two Object identification model is integrated with the , FPN-lite feature extractor ,shared box predictor, and focal loss. Setting up the pipeline setup, also known as the pre-trained model configuration , is followed by updating it for transfer learning in order to train it using the newly constructed dataset.

5)Training: Now we have trained our model. number of training step are 10000.our model give the best result when it is trained 10000 steps, and the accuracy loss is calculated with various formulas as show in figure 3. Now after training ,our model is ready for real time sign detection . Response time of our model is around 1 second.



(a)



(b)

Fig. 2 (a) structure of the VGG16 molecule (b)Labelling images with labelimg

A. Working procedure

Building hand gesture models is an essential part of the process of gesture recognition, and it's the first stage of the process of processing the initial input gesture. The picture is what is brought into this step as the input. When we examine a photograph from the point of view of a human being, we are able to perceive the situation that is being portrayed in the image. Unfortunately, the computer is unable to isolate this particular scene from the source picture. The computer interprets the picture as a matrix, with distinct values present in the various channels and spaces of the matrix. To put it another way, the computer is only capable of capturing the information at the pixel level of a picture. Using low-level information like as pixel values, it is obviously challenging to differentiate between the many things that are being shown. Therefore, if we want to detect hand motions, one of the most successful techniques would be to extract and summarize high-level details such as features and structure from the original photo. This information may be found by looking at the image in its entirety. In our approach, the gesture model functions in the same way. In order to gather high-level information on hand movements, we make use of the VGG16 convolutional neural network. This network has 13 layers that are both convolutional and deep. If the initial picture is provided as an input, VGG16-Net will produce maps of varying resolutions that are representative of the image and include high-level information about it. The number 19 was chosen because it is adequate to extract high-level semantic information for classification and regression, which was the motivation for the decision. Because of the size of our database, using high-level layers might very quickly result in redundant information being stored.

NN is a collection of convolutional neural networks of variable depth, each of which uses extremely tiny convolutional filters, and NN itself is a convolutional neural network. One of the 13 convolutional layers and three fully linked layers is referred to as the VGG16-Net (16 weight layers). Fig. 2(a) illustrates the structure of the VGG16 molecule. In this figure, the parameters of the convolution layer are shown as "conv <receptive field size> - <number of channels>". The function that activates the temporary ReLU state is not shown. A convolution layer is applied to the initial image, and a tiny receptive field filter is used to do the processing. (left, right, top, (minimum size to get bottom and center perception). threshold set to 1 pixel; convolution layer padding is such that the spatial resolution is preserved after convolution, i.e. 1 for filter convolution padding. After multiple rounds of convolution, the pool width is increased to a maximum of five layers. (Maxpooling layers are not present in all convolution layers.) The pixel window is the location for step 2's maxpooling operation. All convolutional layers have nonlinear correction (ReLU) built in. Following a sequence of convolution, maxpooling, and ReLU layers, we are able to produce feature maps that have a small number of dimensions while yet retaining a high level of semantic information. The picture classification method known as the original VGG16-Net uses completely concatenated layers in addition to smooth max layers. This layer is replaced with an SSD layer so that hand segmentation and hand gesture classification may be performed.

The second stage, which consists of using the SSD network to conduct hand segmentation and hand gesture categorization, is the most crucial component of our overall architecture. We went with the SSD variant since it offers both precision and speed. The fundamental concept behind SSD is the application of extremely tiny convolutional filters to individual maps in order to make predictions about category scores and bound boxes for a given collection of bound boxes. In addition, SSD creates distinct scale predictions based on different scale feature maps and splits the predictions based on aspect ratio. The trade-off between speed and accuracy is further exacerbated by this design, which results in straightforward finite element training and high precision.

The SSD feeder is built on a feedforward convolutional neural network known as VGG16. This network gathers bounded boxes and points of a specified size in order to contain the class of items that are in this box. Taking this strategy will result in a significant number of connection boxes, many of which will be redundant with one another. As a result, a step that does not achieve maximum compression is carried out in order to get rid of duplicate bound boxes and get at the final specification. The construction of the SSD is shown in the figure that can be seen above. The picture that is being read in is one that has pixels and RGB channels. The area containing the point box is cut from VGG16 mesh. The SSD model extends the capabilities of the truncated VGG16 network by including many feature layers that vary in size. This layer becomes thinner as one goes deeper and provides the capability of predicting detection over a range of scales. After that, a miniature convolution filter is applied to each point in the feature map that was picked. To be more specific, this filter is referring to a group of bins that all have distinct aspect ratios everywhere else in various feature maps that were chosen for the prediction of picture offsets and confidence scores for each and every item category. The work that we have done contains four different hand movements as well as backdrops in the form of items.

Since we already have an SSD framework, the next thing that we'll require is a goal function so that we can train the model all the way through to its completion. The overall objective function is equal to the sum of the losses associated with localization (location) and reliability:

$$L(x, c, l, g) = 1/N(Lconf(x, c) + \alpha Lloc(x, l, g)),1 \qquad (1)$$

where N is the number of standard boxes that are equivalent to the truth box. The L, 1 loss is the difference between the ground box and the predicted box parameter, and it is referred to as the localization loss. These parameters are comparable to those used by the more efficient R-CNN in terms of the center coordinates (cx, cy) as well as the width and height of the conventional bounding box :

$$\begin{aligned}
L_{\text{loc}}(x,l,g) &= \sum_{i\in\text{Pos}}^{N} \sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k}\,\text{smooth}_{L1}\left(l_i^m - \hat{g}_j^m\right) \\
\hat{g}_j^{cx} &= \frac{\left(g_j^{cx} - d_i^{cx}\right)}{d_i^w}, \\
\hat{g}_j^{cy} &= \frac{\left(g_j^{cy} - d_i^{cy}\right)}{d_i^h}, \\
\hat{g}_j^{w} &= \log\left(\frac{g_j^w}{d_i^w}\right), \\
\hat{g}_j^{h} &= \log\left(\frac{g_j^h}{d_i^h}\right).
\end{aligned} \qquad (2)$$

Confidence loss is the softest loss of multiclass confidence, as commonly used in multi-classification problems:

$$L_{\text{conf}}(x,c) = -\sum_{i\in\text{Pos}}^{N} x_{ij}^{p}\log\left(\hat{c}_i^p\right) - \sum_{i\in\text{Neg}} \log\left(\hat{c}_i^0\right), \text{ where } \hat{c}_i^p = \frac{\exp\left(c_i^p\right)}{\sum_p \exp\left(c_i^p\right)}. \qquad (3)$$

The loss of the objective function is calculated and reduced while the training is being done by matching the preset bins to the bins that contain the ground truth. This process is repeated many times until we have successfully optimized the SSD model parameters and obtained the optimal model. When determining the aspect ratio of the bins, we will use the k-means group to generate three possible ratios to use as guides.

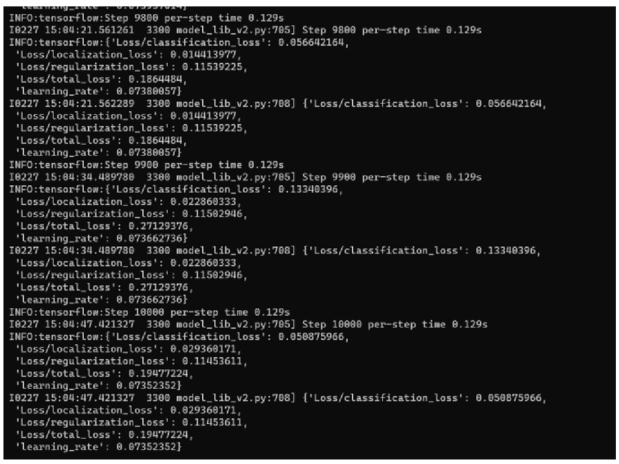


Fig. 3. Training model output.

Then the ratio is 1.9, 1.6 and 1.1 respectively. Additionally, the optimizer used is Human with an initial training level of 0.0001.

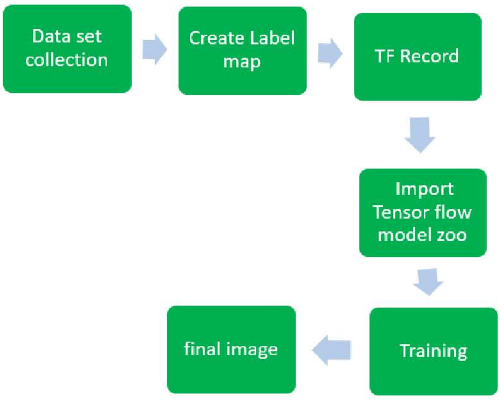


Fig. 4. Process diagram of proposed work

IV. RESULT ANALYSIS

The developed system can detect the Indian sign language alphabet and various other hand signs in real time .

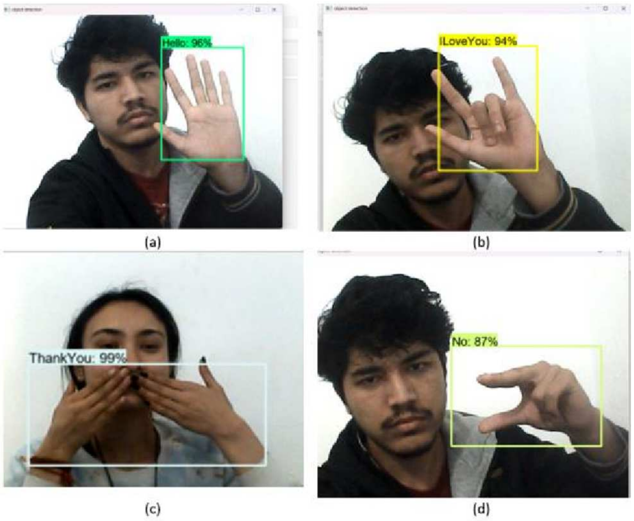


Fig. 5. Hand sign detection (a) hello sign detection with the accuracy. Accuracy graph and accuracy images were formed with the of 96% . (b) I love you sign detection with the accuracy of 94%. help of tensorboard as show in figure 8,9,10. (c) Thank you sign detection with the accuracy of 99% (d) NO sign detection with the accuracy of 87%

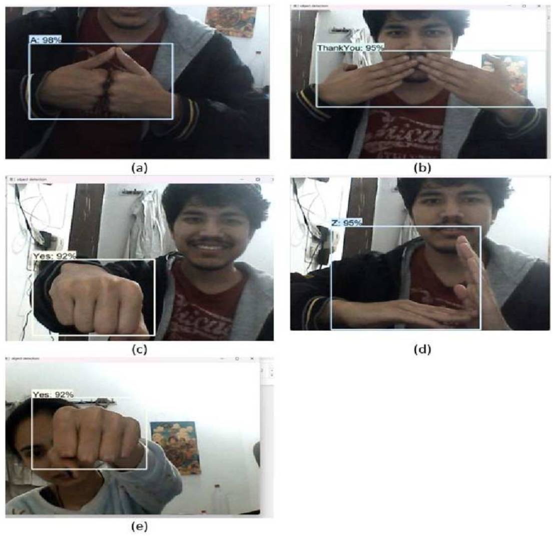


Fig. 6. Hand sign detection . (a)A sign detection with the accuracy of 98%. (B) thank you sign detection with the accuracy of 95% (c) yes sign detection with the accuracy of 92%. (d) z sign detection with the accuracy of 95% (e) yes sign detection with the accuracy of 92%.

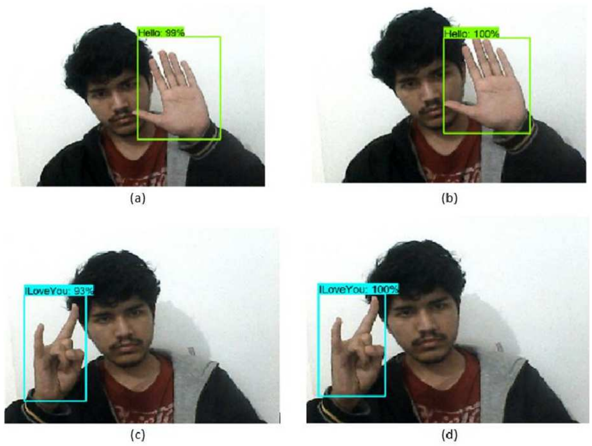


Fig. 7. Hand sign detection (a) hello sign detection with the accuracy of 99% . (b) hello sign detection with the accuracy of 100% . (c) I Love you sign detection with the accuracy of 93%.(d) I love you sign detection with the accuracy of 100%.

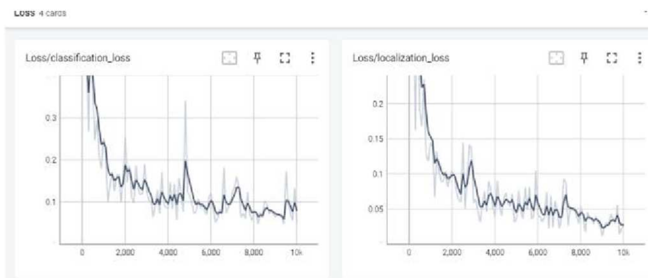


Fig. 8. Accuracy graph

Accuracy graph and accuracy images were formed with help of tensorboard as show in figure 8,9,10.

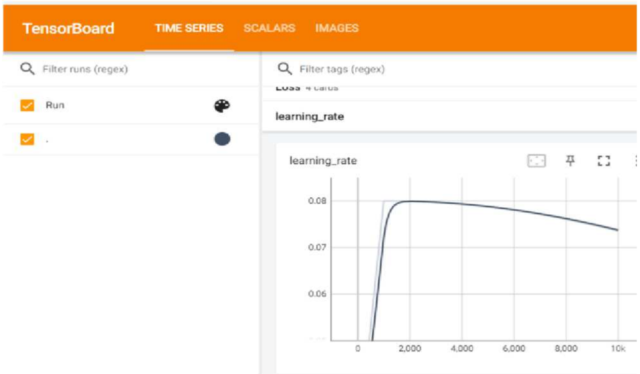


Fig. 9. learning loss of model

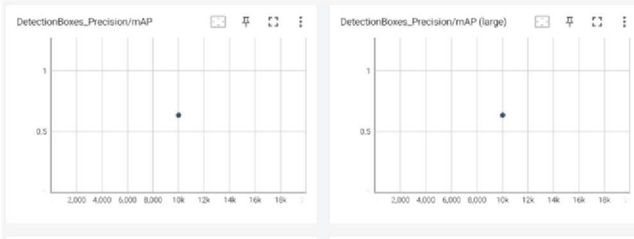


Fig. 10. Average precision of the model

The developed system can detect the Indian sign language alphabet and various other hand signs in real time .TensorFlow's object detection API was used in the development of the system. The MobileNet version two 320x320 SSD is the pre-trained model that can be downloaded from the TensorFlow zoo model. It was taught via transfer learning on a dataset that was automatically created, and the training set consisted of 30 photos for each symbol. The fact that each sign has the same quantity of pictures contributes to the creation of an objective outcome. To create the accurate result the model was trained 10000 steps.

V. CONCLUSION

The motions of the hands, the body, and the face are all used in sign language as a method of communication. Sign language is a kind of the visual language known as "body language." It is essential for persons with particular talents to have a method of communication, and sign language is that method. Because of this, they are able to speak with others, express their sentiments, and share their expressions with others. The fact that nobody is proficient in sign language is a drawback, which means that communication is restricted. This constraint may be overcome by using an automatic sign language recognition system that is able to quickly transform sign language movements into regular spoken English. Using such a system will allow users to communicate more effectively. TensorFlow's Object Detection API was used in order for us to do this. The Indian Sign Language Alphabet dataset was used throughout the training process of the system. It determines sign language in real time. To save costs during the data gathering process, I used a webcam and Python in combination with OpenCV to capture photos. The dependability of the system that was designed is, on average, 85.4 percent. The training dataset was tiny and restricted in size, despite the fact that the system managed to obtain a high average level of confidence. Future work- In different region, different hand sign are used and in order to make our model understand all those signs, more dataset is to be added and vast amount of data is needed to be generate and train our model on that.

REFERENCES

[1] S. Hussain, R. Saxena, X. Han, J. A. Khan, and H. Shin, “Hand gesture recognition using Deep Learning,” 2017 International SoC Design Conference (ISOCC), 2017.

[2] K. Kumar,N. Kumar and A. K. Dahiya, "Image Restoration Using a Fuzzy-Based Median Filter and Modified Firefly Optimization Algorithm", Int. J. Adv. Sci. Technol., vol. 29, no. 4s, pp. 1471-1477, 2020.

[3] N. Kumar, A. K. Dahiya and K. Kumar, "Modified Median Filter for Image Denoising", Int. J. Adv. Sci. Technol., vol. 29, no. 4s, pp. 14951502, 2020.

[4] N. Kumar, H. Shukla, and R. Tripathi, “Image restoration in noisy free images using fuzzy based median filtering and adaptive particle swarm optimization - richardson-lucy algorithm,” International Journal of Intelligent Engineering and Systems, vol. 10, no. 4, pp. 50–59, 2017.

[5] N. Kumar, K. Kumar, and A. Kumar, “Application of internet of things in image processing,” 2022 IEEE Delhi Section Conference (DELCON), 2022.

[6] S. Ghotkar, R. Khatal, S. Khupase, S. Asati and M. Hadap, "Hand gesture recognition for Indian Sign Language," 2012 International Conference on Computer Communication and Informatics, Coimbatore, India, 2012, pp. 1-4, doi: 10.1109/ICCCI.2012.6158807.

[7] R. A. Bhuiyan, A. K. Tushar, A. Ashiquzzaman, J. Shin and M. R. Islam, "Reduction of gesture feature dimension for improving the hand gesture recognition performance of numerical sign language," 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2017, pp. 1-6, doi: 10.1109/ICCITECHN.2017.8281833.

[8] J. -H. Sun, T. -T. Ji, S. -B. Zhang, J. -K. Yang and G. -R. Ji, "Research on the Hand Gesture Recognition Based on Deep Learning," 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China, 2018, pp. 1-4, doi: 10.1109/ISAPE.2018.8634348.

[9] C. -C. Hsieh, D. -H. Liou and D. Lee, "A real time hand gesture recognition system using motion history image," 2010 2nd International Conference on Signal Processing Systems, Dalian, China, 2010, pp. V2-394-V2-398, doi: 10.1109/ICSPS.2010.5555462.

[10] M. Zare, M. Jampour, A. S. Arezoomand and M. Sabouri, "Handwritten Recognition based on Hand Gesture Recognition using Deterministic Finite Automata and Fuzzy Logic," 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), Tehran, Iran, 2019, pp. 93-99,doi:10.1109/PRIA.2019.8786038.

[11] R. P. Tripathi. H.S Shukla N. Kumar, "Image Restoration using modified binary particle Swarm Optimization Richardson-Lucy (MBSO-RL) algorithm", Int. J. Appl. Eng. Res., vol. 10, no. 22, pp. 43077-43081, 2015.

[12] K. Kumar, G. Saini, N. Kumar, M. S. Kaiser, R. Kannan, and R. Shah,“Prediction of energy generation target of hydropower plants using artificial neural networks,” Sustainable Developments by Artificial Intelligence and Machine Learning for Renewable Energies, pp. 309– 320, 2022.

[13] N. Kumar, A. K. Dahiya, K. Kumar, and S. Tanwar, “Application of IoT in Agriculture,” in 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021.

[14] K. Kumar, A. Kumar, N. Kumar, M. A. Mohammed, A. S. Al-Waisy, M. M. Jaber, R. Shah, and M. N. Al-Andoli, “Dimensions of internet of things: Technological taxonomy architecture applications and open challenges—a systematic review,” Wireless Communications and Mobile Computing, vol. 2022, pp. 1–23, 2022.

[15] B. Suneja, A. Negi, N. Kumar, and R. Bhardwaj, “Cloud-based tomato plant growth and health monitoring system using IOT,” 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), 2022.

[16] K. Kumar, N. Kumar, A. Kumar, M. A. Mohammed, A. S. Al-Waisy, M. M. Jaber, N. K. Pandey, R. Shah, G. Saini, F. Eid, and M. N. Al-Andoli, “Identification of cardiac patients based on the medical conditions using machine learning models,” Computational Intelligence and Neuroscience, vol. 2022, pp. 1–15, 2022.

[17] K. Kumar, N. Kumar, and R. Shah, “Role of IoT to avoid spreading of COVID-19,” International Journal of Intelligent Networks, vol. 1, pp. 32–35, 2020.

[18] M. Garg, P. M. Pradhan and D. Ghosh, "Multiview Hand Gesture Recognition using Deep Learning," 2021 IEEE 18th India Council International Conference (INDICON), Guwahati, India, 2021, pp. 1-6, doi: 10.1109/INDICON52576.2021.9691523.

[19] J. Shukla and A. Dwivedi, "A Method for Hand Gesture Recognition," 2014 Fourth International Conference on Communication Systems and Network Technologies, Bhopal, India, 2014, pp. 919-923, doi: 10.1109/CSNT.2014.189.

[20] S. Sharma, S. Jain and Khushboo, "A Static Hand Gesture and Face Recognition System for Blind People," 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2019, pp. 534-539, doi: 10.1109/SPIN.2019.8711706.

[21] A. Sharma, A. Mittal, S. Singh, and V. Awatramani, "Hand gesture recognition using image processing and feature extraction techniques," Procedia Computer Science, vol. 173, pp. 181–190, 2020.