Real Time Sign Language Recognition Using Deep Learning

Sanket Bankar[1], Tushar Kadam[2], Vedant Korhale[3], Mrs. A. A. Kulkarni[4]

[1,2,3]Studenst, VIII Semester B.Tech, Department of Electronics And Telecommunication Engineering,
College Of Engineering, Pune (COEP) Wellesley Rd, Shivajinagar, Pune, Maharashtra 411005, India
[4]Assistant Professor, Department of Electronics And Telecommunication Engineering,
College Of Engineering, Pune (COEP) Wellesley Rd, Shivajinagar, Pune, Maharashtra 411005, India

---***---

Abstract - *Sign language is an extremely important communication tool for many deaf and mute people. So we proposed a model to recognize sign gestures using YOLOv5 (You only look once version 5). This model can detect sign gestures in complex environment also. For this model we got the accuracy of 88.4% with precision of 76.6% and recall of 81.2%. The proposed model has evaluated on a labeled dataset Roboflow. Additionally we added some images for training and testing to get better accuracy. We compared our model with CNN (convolutional neural network) where we got accuracy of 52.98%. We checked this model for real time detection also and got the accurate results.*

Key Words: YOLO (You Only Look Once), CNN, Deep Learning, Python, OpenCV

1. INTRODUCTION

In our surrounding we can see there are people having various disabilities and some of them are found to be deaf and mute. To communicate with others, those people need to learn sign language and normal people are unable to understand sign language. This problem causes miscommunication between people. Due to this miscommunication mute people can live isolated from society. They can't able to take part in social events or any discussion. This create big gap between normal people and people with disabilities. We can reduce this gap by using technologies like computer vision, deep learning etc. So this is the main reason to choose this project. Our project constructed the model to understand the sign language from the user (can be mute or normal) and translated it into the understandable text. There are many object detection algorithms in deep learning. This paper includes the comparison between general CNN and YOLO and why YOLO is better. There are different versions of YOLO like YOLO v1, v2, v3, v4 and v5. In our model we have used the latest version of YOLO which is YOLOv5. YOLO v5 model runs about 2.5 times faster than other versions while managing better performance in detecting smaller objects. Our model can detect the static images as well as gestures from on camera (video).

2. RELETED WORK

We studied several research papers which proposed CNN algorithm for sign language system but CNN is significantly slower due to an operation such as maxpool and we checked CNN for real time detection but it gave inaccurate results. Hand detection and image processing needs to be done in CNN which increases processing time. In contrast, YOLO is specifically developed for real time system. We used latest version of YOLO i.e. YOLOv5 in our project.

3. METHODOLOGY

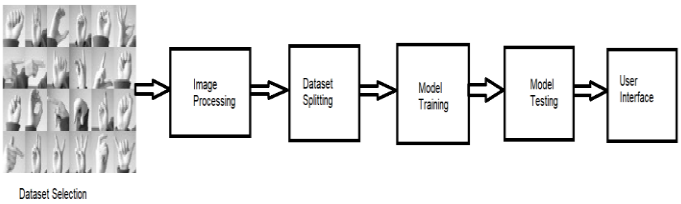


Fig -1: Methodology

First we have selected the sign language dataset and from that dataset we have fetched the images. Using image processing we have converted those images into pixels. We did this image processing for CNN. Then in dataset splitting we have divided this dataset for training and testing purpose. Using this training and testing samples we have trained and tested our model. At last we have created the user interface for real time detection. If Images/features in the training dataset are tilted or rotated then CNN have difficulty in classifying those images.

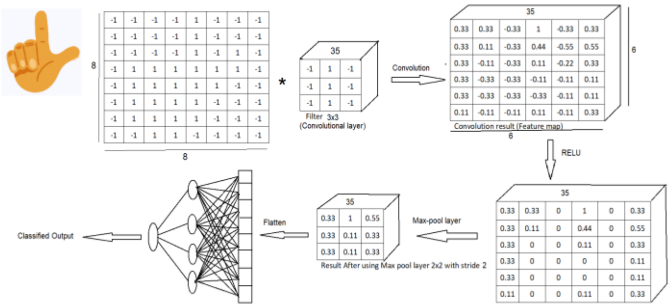
4. DATASET SELECTION

Fig -2: Sample images from Dataset

In our project we used Roboflow dataset. This dataset contains 1728 images and additionally we have added our own 422 images to get better results. There are 1752 images for training, 248 are validation and 150 for testing. The features/images are in .jpg format and labels are in .txt files. These text files include label/sign gesture, x-coordinate, y-coordinate, height and width of the image. These images are in unstructured format as we can see in figure 2. The significance of this dataset is the images have unclear background which makes it difficult to recognize the alphabets. The images taken in the real time mode do not have a clean background, so if we use a regular dataset chances are the model won't work efficiently in real time. So for real time detection we have used this dataset.

5. CNN (CONVOLUTIONAL NEURAL NETWORK)

CNN or Convolutional neural network is a deep learning neural network i.e., we can think CNN as a machine learning algorithm that can take an input image, assign an importance to an object and then to be able to differentiate between one object and others. CNN works by extracting features from the images. Any CNN consist of three things which are – an input layer which is a grey scale image, then output layer which is the binary or multiclass labels and third hidden layers which contains convolution layer, RELU and then pooling layers and finally there is artificial neural network to perform the classification. Now let's see CNN architecture.



-1	-1	-1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	1	-1
-1	1	1	1	1	1	1	-1
-1	1	1	1	1	1	-1	-1
-1	1	1	1	1	-1	-1	-1
-1	1	1	1	1	-1	-1	-1
-1	-1	1	1	-1	-1	-1	-1

-1	1	-1
-1	1	-1
-1	1	-1

0.33	0.33	-0.33	1	-0.33	0.33
0.33	0.11	-0.33	0.44	-0.55	0.55
0.33	-0.11	-0.33	0.11	-0.22	0.33
0.33	-0.33	-0.33	-0.11	-0.11	0.11
0.33	-0.33	-0.33	-0.33	-0.11	0.11
0.11	-0.11	-0.11	0.11	-0.11	0.33

0.33	0.33	0	1	0	0.33
0.33	0.11	0	0.44	0	0.55
0.33	0	0	0.11	0	0.33
0.33	0	0	0	0	0.11
0.33	0	0	0	0	0.11
0.11	0	0	0.11	0	0.33

0.33	1	0.55
0.33	0.11	0.33
0.33	0.11	0.33

Fig -3: CNN Architecture

In CNN architecture, first we have an input image. Then we need to convert it into pixels. For simplicity let's consider an image size of 8x8. Then we can perform the convolution by passing the image through convolutional filter which would be of size 3x3. This convolution filter will go through each and every part of the image; we called them as strides and extract all the important features. These convolutional layers are in multiple numbers and here we have considered them 35. So it will extract different 35 features from the image. From this convolution we will get a new image of size 6x6. We can also call this convolution result as feature map. Then we have RELU activation function to bring non-linearity in our model. So what it will do is, it will take our feature map and whatever negative values are there, it will just map them to 0. And if values more than 0 then it will keep them as it is. Now here we can see huge amounts of dimensions, so it will lead to curse of dimensionality and computations are more. In order to overcome this, we will pass this image through max-pool layer. Here the size of max-pool layer we have considered as 2x2. This max-pool layer will go through each and every layer of the 6x6 image on each 4 pixels and pick the value having more probability. So due to this max-pool layer, the size of the image will get reduced to half i.e., 3x3. All these steps repeated many times and once we are done with that, we will flatten the entire layer and this is going to be a simple artificial neural network. Every time what happens over here is, we are going to pass these features through multiple layers or deep layers and then we can perform classification here. In this artificial neural network, in order to perform the classification in our last output layer, we will pass an activation function as SoftMax. Because softmax gives us the probability which would range from 0 to 1. So, this is the architecture of convolutional neural network.

6. LIMITATIONS OF CNN

Convolutional neural network has max pooling layers which lead to slow processing. CNN has many layers for training, so the computer takes a lot of time for training the model. CNN requires a lot of data points for training the model. In contrast to CNN, coordinate frames can't be used. These coordinate frames are the part of computer vision. These frames are used to keep track of the orientation and different features of an object. In real time detection we need to define the frame for detection of objects. It will detect images only in constrained area. So this is the main disadvantage of CNN. YOLO can detect images at any position with fast processing. So this is the main reason why are we choosing YOLO.

7. YOLO ALGORITHM

YOLO stands for You Only Look Once. YOLO algorithm is specifically designed for real time object detection. YOLO employs regression methods for detection and provides probabilities for the output classes. Along with the probabilities of the classes, bounding boxes are also generated for the detected object. To summarize, YOLO predicts the result in the single execution of the algorithm.

There are five versions of YOLO. Yolov1 was the first version ever to be introduced. This was the milestone in the field of computer vision and object detection. Yolov2 was the faster and more precise version of previous one. This version provided the processing at 40-50 fps. Yolov3 provided tradeoff between accuracy and precision. Further v4 and v5 were developed. V4 had more accuracy than previous versions while v5 had pytorch implementation. How YOLO works?

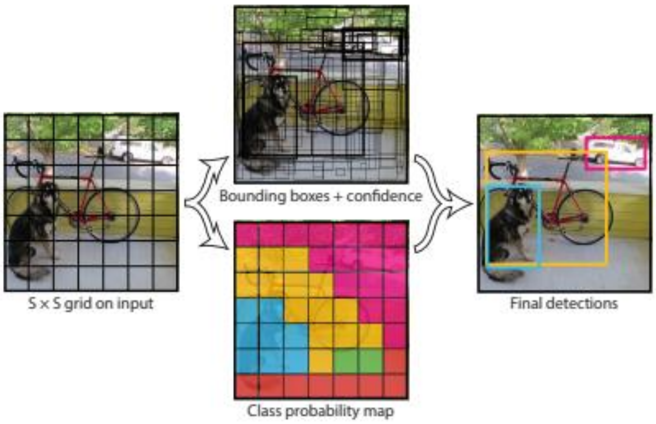


Fig -4: YOLO Algorithm

Firstly, the image is divided into number of grids. Each grid has a dimension of m x m. Hand detection is done for every grid cells. When an object is detected in the grid, bounding boxes are generated. Every bounding box has 4 parameters: height, width, center of the box and class of the object detected. This leads to formation of multiple bounding boxes. So, finally IOU (Intersection over Union) is calculated for all the boxes and the boxes with highest IOU are selected. We have given 26 classes i.e., class for each alphabet for training. So, the algorithm is trained to detect hand in the given image and predict the alphabet denoted by that hand sign. The primary advantage of YOLO is the small processing time, which counts a lot when developing a computer vision model.

8. IMPLEMENTATION

For the implementation, we used a dataset consisting of hand images depicting hand signs having different angles and backgrounds. For training the model, yolov5 model is cloned from the repository. The training module of yolo is programmed for detecting the hand and training the model for prediction. After train-test-validation split, the data is provided to the yolov5 model. The trained model has weight files which will be used for detection of the alphabets. The final model has mAP i.e., mean average precision of 0.88, with precision of 0.76 and recall of 0.81.

9. COMPARISON TABLE OF CNN VS YOLO

Table -1: CNN VS YOLO

Parameters	CNN	YOLO
Accuracy	Less	More
Accuracy score	0.53	0.88
Processing time	Slow	Fast
Real time detection	Slow	Fast

10. YOLOV5 MODEL RESULTS

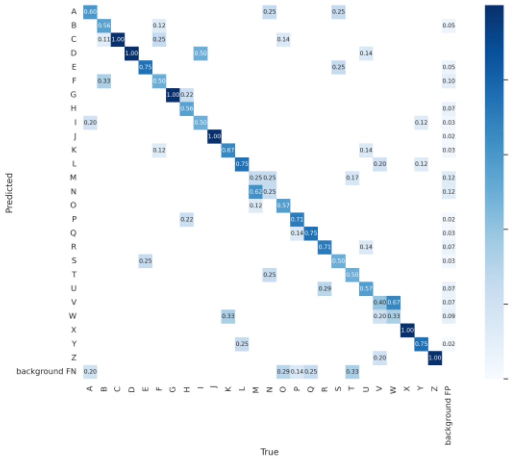


Fig -5: Confusion Matrix

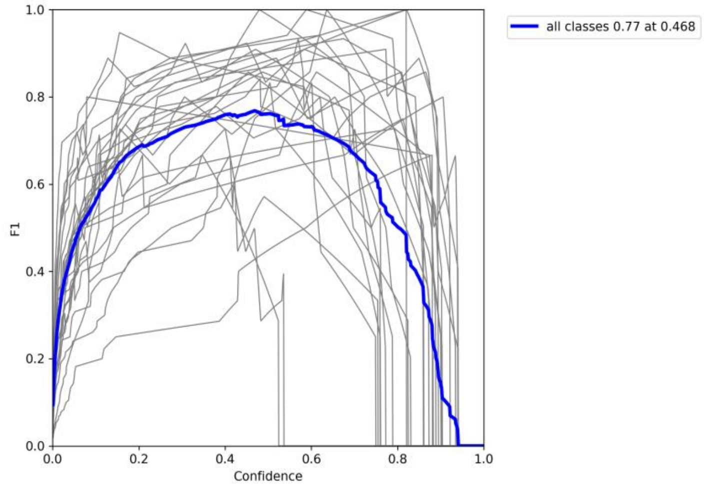


Fig -6: F1- Curve

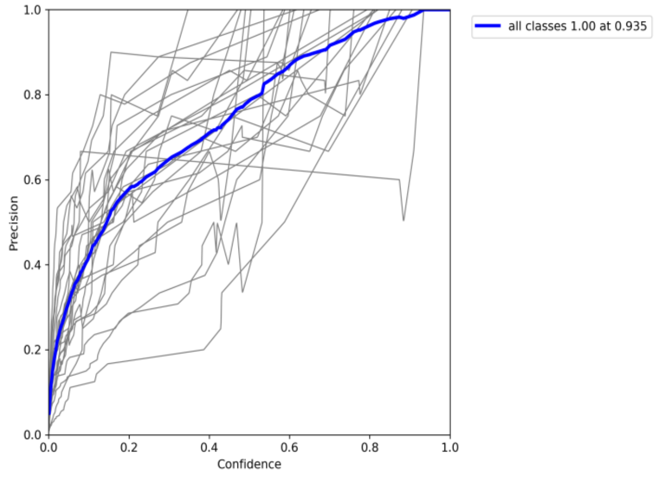


Fig -7: P- Curve

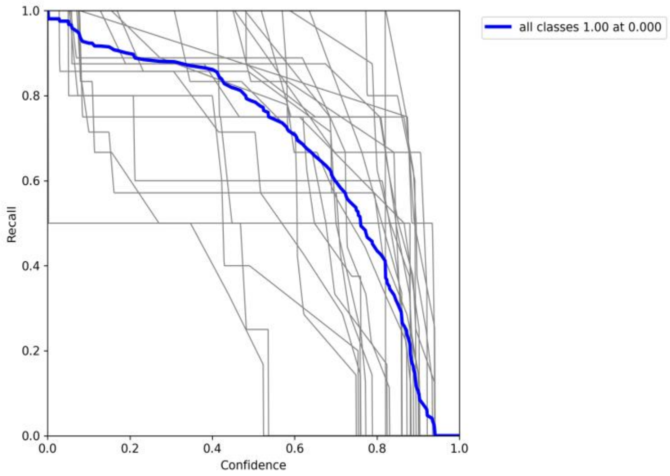


Fig -8: R- Curve

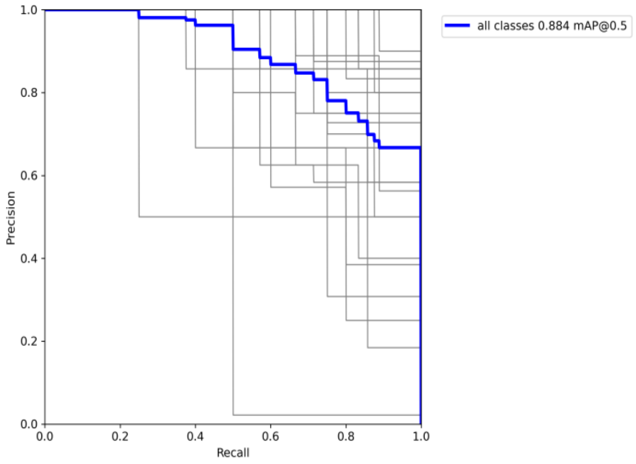


Fig -9: PR- Curve

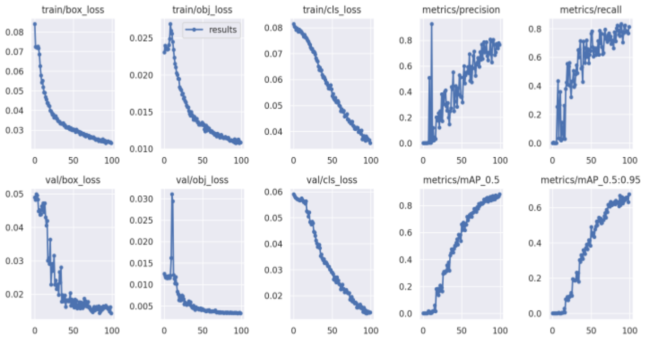
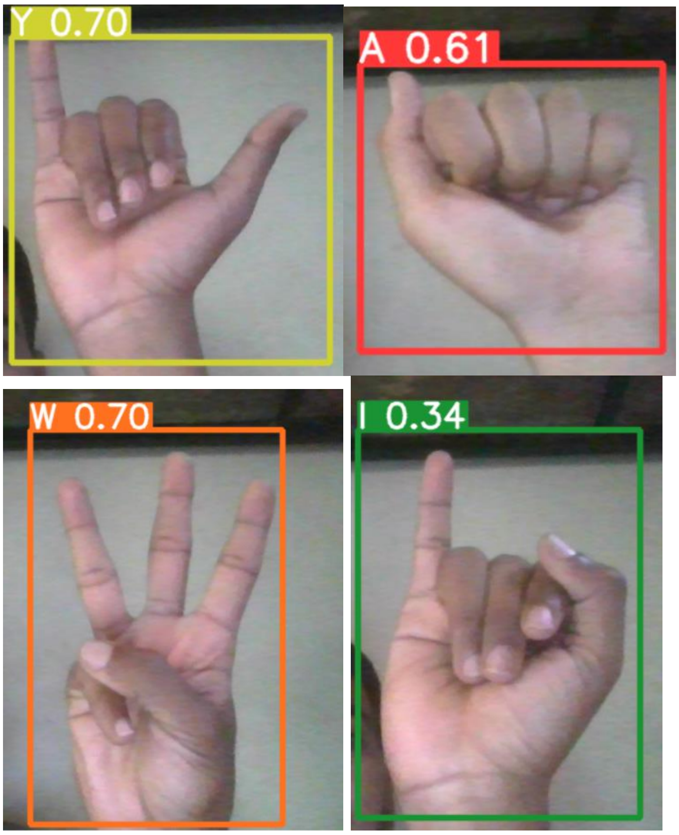


Fig -10: Training and Validation losses

10. DETECTION RESULTS



3. CONCLUSION

In this paper, we have explained a model based on the YOLOv5 for sign language recognition. With an accuracy of 88.4%, the new sign gesture recognition model can detect real time objects and gestures from video in real time. Additionally, we compared the performance and execution times of YOLOV5 with other models, and found that our proposed model was more successful at extracting required features from the hand sign and recognized hand gestures with the accuracy of 88.4% with precision of 76.6% and recall of 81.2%. We predicted all the alphabets successfully.

REFERENCES

[1] V. I. Pavlovic, R. Sharma and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: a review," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 677-695, July 1997, doi: 10.1109/34.598226.

[2] Goyal, Kanika & Singh, Amitoj. (2014). Indian Sign Language Recognition System for Differently-able People. Journal on Today's Ideas - Tomorrow's Technologies. 2. 145-151. 10.15415/jotitt.2014.22011.

[3] Khan, Rafiqul Zaman & Ibraheem, Noor. (2012). Hand Gesture Recognition: A Literature Review. International

Journal of Artificial Intelligence & Applications (IJAIA). 3. 161-174. 10.5121/ijaia.2012.3412.

[4] Wang, Xianghan & Jiang, Jie & Wei, Yingmei & Kang, Lai & Gao, Yingying. (2018). Research on Gesture Recognition Method Based on Computer Vision. MATEC Web of Conferences. 232. 03042. 10.1051/matecconf/201823203042.

[5] 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)*, 2018, pp. 1-4, doi: 10.1109/ISAPE.2018.8634348.

[6] Dong, Cao & Leu, Ming & Yin, Zhaozheng. (2015). American Sign Language alphabet recognition using Microsoft Kinect. 44-52. 10.1109/CVPRW.2015.7301347.

[7] Ruth Campbell, Mairéad MacSweeney, Dafydd Waters, Sign Language and the Brain: A Review, *The Journal of Deaf Studies and Deaf Education*, Volume 13, Issue 1, Winter 2008, Pages 3–20.

[8] Mujahid, Abdullah & Awan, Mazhar & Yasin, Awais & Mohammed, Mazin & Damasevicius, Robertas & Maskeliunas, Rytis & Hameed, Karrar. (2021). Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. Applied Sciences. 11. 4164. 10.3390/app11094164.

[9] Ni, Zihan, Jia Chen, Nong Sang, Changxin Gao and Leyuan Liu. "Light YOLO for High-Speed Gesture Recognition." *2018 25th IEEE International Conference on Image Processing (ICIP)* (2018): 3099-3103.

[10] Lionnie, Regina & Timotius, Ivanna & Setyawan, Iwan. (2012). Performance Comparison of Several Pre-Processing Methods in a Hand Gesture Recognition System based on Nearest Neighbor for Different Background Conditions. ITB Journal of Information and Communication Technology. 6. 183-194. 10.5614/itbj.ict.2012.6.3.1.

[11] J. P. Wachs, H. Stern and Y. Edan, "Cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system," in *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 35, no. 6, pp. 932-944, Nov. 2005, doi: 10.1109/TSMCA.2005.851332.

[12] Barczak, Andre & Reyes, Napoleon & Abastillas, M & Piccio, A & Susnjak, Teo. (2011). A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures. Res Lett Inf Math Sci. 15.

[13] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[14] Daniels, Steve & Suciati, Nanik & Fatichah, Chastine. (2021). Indonesian Sign Language Recognition using YOLO Method. IOP Conference Series: Materials Science and Engineering. 1077. 012029. 10.1088/1757-899X/1077/1/012029.

[15] Heera, S & Murthy, Madhuri & Sravanti, V & Salvi, Sanket. (2017). Talking hands — An Indian sign language to speech translating gloves. 746-751. 10.1109/ICIMIA.2017.7975564.

[16] Bahadure, Nilesh & Juneja, Shubham & Mahapatra, P.D. & Chandra, Chhaya & Verma, Sankalp. (2018). Kinect Sensor based Indian Sign Language Detection with Voice Extraction. International Journal of Computer Science and Information Security,. 16. 135-141.

[17] M. Karthi, V. Muthulakshmi, R. Priscilla, P. Praveen and K. Vanisri, "Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2021, pp. 1-6, doi: 10.1109/ICSES52305.2021.9633834.

[18] Lesha Bhansali and Meera Narvekar. Gesture Recognition to Make Umpire Decisions. *International Journal of Computer Applications* 148(14):26-29, August 2016.

[19] Thangarasu, Rajasekaran & Anandamurugan, S. & Pitchai, Pandiyan & Kaliappan, Vishnu. (2021). Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion. Journal of Plant Diseases and Protection. 10.1007/s41348-021-00500-8.

[20] Khadhraoui, Taher & Faouzi, Benzarti & Alarifi, A. & Amiri, Hamid. (2012). Gesture determination for hand recognition. CEUR Workshop Proceedings. 845. 1-4.

[21] Marcel, Sébastien & Bernier, Olivier & Viallet, Jean & Collobert, Daniel. (2000). Hand gesture recognition using input-output hidden Markov models. Proceedings - 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2000. 456 - 461. 10.1109/AFGR.2000.840674.

[22] Y. Fang, J. Cheng, K. Wang and H. Lu, "Hand Gesture Recognition Using Fast Multi-scale Analysis," Fourth International Conference on Image and Graphics (ICIG 2007), 2007, pp. 694-698, doi: 10.1109/ICIG.2007.52.

[23] S., Manjula & Krishnamurthy, Lakshmi & Ravichandran, Manjula. (2016). A STUDY ON OBJECT DETECTION.

[24] "Online Hand Gesture Recognition Using OpenCV", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.2, Issue 5, page no.1635-1637, May-2015.