WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real Time Sign Language Detection Using Yolov5 Algorithm

***Mr. D. Chiranjeevulu*[1]*, J. Tejasri*[2]*, I. Sasi Kala*[3]*, K. Hemanth*[4]*, N. Rakesh*[5]**

[1]Assistant Professor, Department of Electronics and Communication Engineering, Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh.

[2,3,4,5]Student, Department of Electronics and Communication Engineering Aditya Institute of Technology And Management, Tekkali, Andhra Pradesh.

ABSTRACT

People with speech or hearing impairments frequently utilize sign language, which is a system of visual motions and signs. It's crucial to comprehend the gestures these people use to communicate in order to integrate them in the community of verbal communicators. Those who don't utilize the gesture in everyday life frequently don't grasp what it means. In this study, we suggest a method for identifying the alphabet and gestures that each motion provides. The goal of this project is to create a real-time sign language identification system for deaf and dumb people utilizing Python programming, OpenCV, and deep learning techniques like YOLOv5 and Convolutional Neural Networks (CNN). To recognize and track hand gestures and other important items in the video stream, the proposed system uses a webcam to gather real-time video input. The video input is enhanced and pre-processed using OpenCV, which is also utilized to present the detection model's results in real-time. This video data is then processed using YOLOv5 and CNN.

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

I. INTRODUCTION

People with various disabilities can be seen around us, and some of them have been identified as being deaf and mute. These people must learn sign language in order to communicate with others, but most of the common people cannot understand sign language. People miscommunicate as a result of this issue. Mute people may live a lonely life because of this misunderstanding with society. They are not able to participate in conversations or social gatherings. This widens the divide between those with impairments and the general population. It is important to have standard data that can be used to compare different algorithms and methods [1].

Using technology like computer vision, deep learning, etc., we can close this gap. This is the primary justification for selecting this project. Our project developed a model to interpret the user's sign language (which can be normal or mute) into text that can be understood. Deep learning uses a variety of object detection algorithms. Convolutional Neural Network (CNN) technology was used to construct YOLO, which can generate quick and accurate object identification [2]. Several research that call for object detection, such as the real-time location of Bhutan number plates [3], the detection of pedestrians [4], and the understanding of traffic signs [5], utilise YOLO. There are several variants of YOLO, including versions 1, 2, 3, and

4. We utilised YOLOv5, the most recent version, in our model. While operating around 2.5 times faster than previous versions, the Yolo v5 model also performs better when recognising tiny objects. Yolo process images in real time 45 frames per second [2]. Our algorithm is capable of recognising both static images and on-camera motions (video).

II. RELATED WORK

Recognizing sign language has been accomplished using a variety of sensor- and vision-based techniques. The visual interpretation of hand gestures for human- computer interaction was studied by Pavlovic et al. [6]. There are two categories in which to separate the current methods. One method detects hands and recognises the gesture they convey by utilising a particular gadget, while the other method uses deep learning.

Referring to various studies that suggested the CNN algorithm for a sign language system, however CNN is slower because of an operation like maxpool, and we tested CNN for real-time detection, but it produced unreliable results. CNN must do hand detection and picture processing, which lengthens the processing time. Yolo, on the other hand, was created especially for real-time systems. In our project, we utilised YOLO's most recent version, version 5.

A. CNN

Convolutional neural network, or CNN, is a type of deep learning neural network, or machine learning. The neural connection pattern of the human brain is analogous to a convolutional network architecture [7]. The algorithm that may assign a value to a picture as input to an object and then be able to tell one object from another in addition to others. CNN utilises features extracted from the pictures Every CNN consists of three layers: an input layer, which is a greyscale image, an output layer, which is a binary or multiclass label, and a third hidden layer, which includes a convolution layer, a RELU layer, and a pooling layer. The classification is then done by an artificial neural network. An example of a form of neural network that is frequently used for image categorization, detection, and other computer vision tasks is the convolution neural network [8].

In the CNN architecture, an input image comes first. After that, we must turn it into pixels. Let's simplify things by using an 8x8 image size. Convolution layers, which apply a series of filters to the input image to extract features, are the fundamental components of a CNN [9]. After that, we can do convolution by running the image through a 3x3 convolutional filter. This convolution filter will examine each and every aspect of the image—we referred to them as steps—and will extract all of the crucial details. There are other counts for these convolutional layers, however in this case we have only taken into account 35. Thus, it will extract 35 distinct features from the image. A new image with a 6x6 resolution will result from this convolution. This convolutional result is also known as a feature map. The RELU activation function is then used to provide nonlinearity to our model. As a result, it will simply map any negative values on our feature map to 0 using our feature map as a starting point. And if the values are more than 0, they will remain unchanged.

B. YOLO ALGORITHM

Ultralytics created the object detection model known as YOLOv5. It is the most recent model in the YOLO (You Only Look Once) series, which are well-known for their speedy object identification. A single convolutional neural network is used by YOLOv5's new architecture to recognise objects in pictures and videos. A sizable dataset of photos with annotations identifying the location and class of items in each image is used to train the YOLOv5 model. The model predicts the bounding boxes and class probabilities for each object in the input image during inference. The four versions of YOLOv5 are YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x.

The size and the amount of parameters, the accuracy of these versions vary, with YOLOv5x being the biggest and most precise. Medical imaging, surveillance systems, and driverless vehicles are just a few of the uses for YOLOv5. Roboflow, is a well-known object detecting programme.

The image is first separated into a variety of grids. The size of each grid is m × m. Each hand is checked for presence of grid units. The usage of bounding boxes, feature extraction networks, and detection at multiple scales are modifications from YOLO [10]. Bounding boxes are created when an object is found in the grid. Each bounding box has four parameters: height, width, box centre, and type of object being identified. This causes numerous bounding boxes to form Hence, after calculating IOU (Intersection over Union) for each box, the boxes with the highest IOU are chosen. For instruction we have provided 6 lessons, one for each hand gesture.

In order to anticipate the gesture represented by the hand sign in the supplied image, the algorithm is taught to recognise hands. The short processing time of YOLO is its main benefit, which is important for creating a computer vision model.

Mohammed et al. [11] explained that the hand region in a depth image was located and separated using the Microsoft Kinect sensor. This technique works well when the skin and hand colours match the face. In this instance, ISL features are automatically generated by convolutional neural networks (CNN). The approach can read ISL alphabets accurately in real time, with a model accuracy of 99.3%.

Joudaki and Rehman [12] suggested a geometric neural model that can recognize the alphabet of sign language. The alphabet of sign language is utilised for communication. Hand motions are recorded by these cameras. So, it uses its in-depth skills. Geometric Sign Recognition System is known as GSLR. There are many uses for simple design. This technique takes advantage of traits that remain constant despite hand motion. Both elements increase precision. The recognition accuracy of a neural network can be increased by precise hand movements. It might even pick up on the infant's sign language.

Singh & Sharma [13] designed a convolutional neural network that recognise sign language based on movement. Because to its simpler representation and fewer parameters, this model outperforms earlier CNN methods. Also, they assessed the model's performance using VGG-11 and 16. To evaluate performance, two datasets were employed. An RGB camera was used to record ISL motions, and then an ASL dataset was added. ISL datasets are 99.96% accurate while ASL datasets are 100% accurate. Modern approaches are contrasted with it. The resilience was influenced by the numerous elements. The proposed method is superior to current approaches in characterising large volumes of motions. This data collection is not affected by scale or rotation.

III. PROPOSED WORK

A deep learning-based strategy using YOLOv5 can also be used to recognise signs in real-time for the deaf and dumb. YOLOv5 is an object identification method that recognises and categorises items in real-time using deep neural networks. The suggested model would employ YOLOv5 to recognise and categorise sign language motions in real-time, which could subsequently be converted for the user into text or speech.

IV. METHODOLOGY

DATA COLLECTION:

The OpenCV package, which is imported in the library section of a desktop or laptop, automatically starts enabling the camera to start capturing videos of the person performing the sign language sentences with one hand or both hands. This is how the sign language detection data set is customised .Only pictures with the proper sign language are taken from the subject's movies after they have been gathered. The data set consists of 150 photos, around 30 for each of the 5 symbols, which make up the entire collection.

DATA ANNOTATION:

Images of the gesture are included in the data set. The data collection must contain labels and the annotating bounding box in order to train in YOLOv5. It is necessary to normalise the value of the annotation box coordinate between 0 and

1. An open-source graphical image annotation programme called LabelImg is widely used for labelling and annotating images for machine learning applications. LabelImg can be used to label and annotate the images of sign language motions for the deaf and dumb, which can then be used to train machine learning models for real- time sign language detection.

DATA UPLOADING:

Update the label map when the images have been labelled. The annotations are stored as XML files in the PASCAL VOC format and as JPG pictures. Training and testing data sets can be created from the currently gathered photos. You can get over the bounding box fitting issue with the help of this dataset partition. The height and depth of the image, as specified in the bounding box, are contained in the XML files for these images.

Upload the customised dataset into github repository in zip file format and then Use the curl command with the -L option to download the zip file from the Github link and then Use the unzip command to extract the contents of the zip file into a new directory, then we can use the extracted dataset in YOLOv5 for training and testing your model.

TRAINING MODEL:

The strategy is to make minor adjustments to the YOLOv5x model that has already been trained using the COCO data set. In order to train on the newly relatively tiny data set, transfer learning is used. Also, a number of augmentation techniques, including HSV, colour spacing, mosaic, and picture scaling, were used to improve the initial image. Here, we employ the fine-tuned hyperparameters from the COCO data set, such as the SGD optimizer, 0.01 learning rate, 0.0005 weight decay, and 1500 epoch on batch size 16 and picture size 416. At 900 epoch, the model is fairly steady and accurate.

The subsequent adjustments are barely noticeable.A confidential threshold is used to perform the experimental outcome of 0.5 in python 3.8 PyTorch 1.8.1, a 12GB NVIDIA Tesla K80 GPU, and Colab Notebook make up the majority of the training software. The training took two hours to complete.

Fig:1 augmentation on the training set EVALUTION AND SAVE MODEL:

The YOLOv5 model's weights and biases are saved in the "best.pt" and "last.pt" files, which are frequently created while training. The model can be loaded with these checkpoint files to continue training or to make inferences about new data. The checkpoint file called "best.pt" provides the model's parameters that produced the best validation loss during training. As the model with the greatest performance on the validation set, it is frequently utilised for evaluation on a held-out test set. The model with the lowest validation loss, which represents the model's propensity to generalise to new data, is chosen as the best one.

There are two models created for best.pt and last.pt. As yolov5 has already built the detect.py for us, all we need to do is pass the path of the best.pt, define the image size and confidence as 0.5, and utilise the text pictures as the source images. It has preserved our outcomes inside the run and will forecast all text images**.**

REAL TIME PREDICTION:

Python may be used to make real-time predictions using a variety of tools and frameworks, including PyTorch. The trained model should first be loaded into your Python environment. Input data must be continuously streamed for real-time prediction.

Video frames will represent this info. To make this data compliant with the input specifications of your model, pre-processing will be necessary.

The prepared input data can then be given to the loaded model for inference. Based on the

input data, the model will generate predictions and output the outcomes.

V. RESULTS

A confidence level of 0.5 is applied to the findings. Initial results showed that we could average 0.987 map@0.5 and 0.985 even with a minimal amount of data.

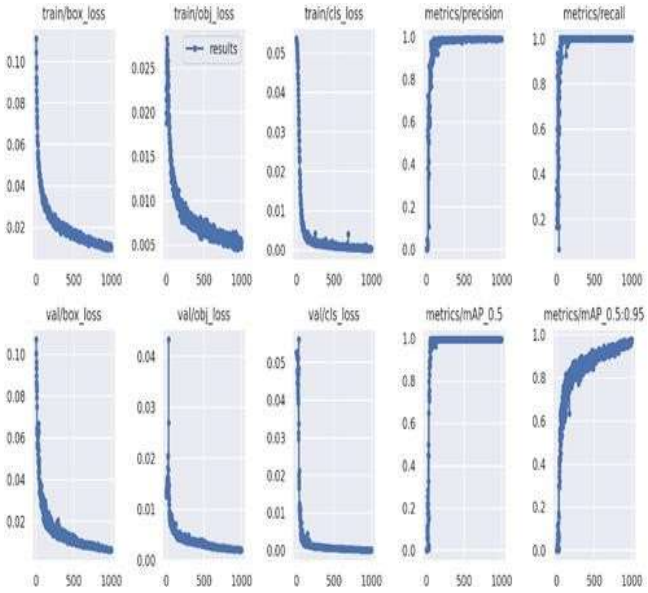

Fig 2. Evaluation Graphs

VI. CONCLUSION

Even with a tiny data set, the findings are still on average 0.98 F1 scores in the identification of 6 distinct categories of hand motions. This finding suggests that YOLOv5 has a good chance of successfully recognising the hand gesture data set. The pre-trained weight for the YOLOv5x solution is just 167MB of memory, making it light enough to use on any mobile device**.** Moreover, YOLOv5x has a high frame rate

Both the accuracy and the frame rate must be at their best for real-time sign language detection. Pretrained can be installed on an AI computing platform because it is light and quick. As a result, it makes for the best choice for real-time sign language recognition**.**

REFERENCES

[1]. L. C. Barczak, N. H. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, “A new 2d static hand gesture colour image dataset for asl gestures,” Research Letters in the Information and Mathematical Sciences, vol. 15, pp. 12–20, 2011.

[2]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779– 788

[3]. Chen R C 2019 Automatic license plate recognition via sliding -window darknet -yolo deep *learning image and vision computing 87 pp 47-56*

[4]. H. Qu, T. Yuan, Z. Sheng, and Y. Zhang, “A pedestrian detection method based on yolov3 model and image enhanced by retinex,” in 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). IEEE, 2018, pp. 1–5.

[5]. Zaki P S,William M M,Soliman B K, Alexsan K G, Khalil K and El- Moursy M 2020 Traffic signs detection and recognition system using deep learning Preprint arXiv:2003.03256

[6]. Podder, Kanchon Kanti, et al. "Bangla Sign Language (BdSL) Alphabets and Numerals Classification Using a Deep Learning Model." Sensors 22.2 (2022): 574.

[7]. D.Michie ,D. J,&C.C. Taylor, “Machine learning. Neural and statistical classification”.(1994).

[8]. Zhang J, Hui L, Lu J, Zhu Y (2018)Attention- basrd neural network for detection.International Conference on pattern Recognisation.

[9]. Valentyn Sichkar, Sergey A.Kolyubin. “effect of various dimension convolution layer filter on traffic sign classification accuracy”.

[10]. Redmon J and Farhadi A 2018 Yolov3: An incremental improvement Preprint arXiv: 1804.0276.

[11]. Mohammed A. A. Q, Lv J, Islam M. S & Sang Y, Multi-Model Ensemble Gesture Recognition Network for High-Accuracy Dynamic Hand Gesture Recognition, Journal of Ambient Intelligence and Humanized Computing. (2022). https://doi.org/10.1007/ s12652-021- 03546-6

[12]. Joudaki S & Rehman A, Dynamic Hand Gesture Recognition of Sign Language Using Geometric Features Learning, International Journal of Computational Vision and Robotics. 12(1) (2022) 1 https://doi.org/10.1504/ijcvr.2022.119239

[13]. Sharma S & Singh S, Vision-Based Hand Gesture Recognition Using Deep Learning for the Interpretation of Sign Language, Expert Systems with Applications. 182(2021) 115657. https://doi.org/10.1016/j.eswa.2021.115657