e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

REAL TIME SIGN LANGUAGE DETECTION WITH YOLOV5

Karthika P[*1], Dharani M[*2], Guna S[*3], Abirami R[*4], Vasanthi R[*5]

[*1,2,3,4]UG student, Department of Computer Science and Engineering, R P Sarathy Institute of Technology, Salem, Tamil Nadu, India.

[*5]Professor Department of Computer Science and Engineering, R P Sarathy Institute of Technology, Salem, Tamil Nadu, India.

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.

Keywords: Real-time Gesture Detection, OpenCV Pre-processing, YOLOv5 Object Detection, Convolutional Neural Networks (CNN), Webcam Input Processing, Deep Learning for Sign Language.

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. 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.

II. LITERATURE REVIEW

The literature review presents three significant contributions towards real-time sign language recognition using the YOLOv5 algorithm. Firstly, Wong et al. (2022) proposed a method integrating Python programming, OpenCV, and deep learning techniques to create a real-time sign language identification system.

Secondly, Potdar (2015) focused on using the YOLOv5 algorithm specifically for detecting and recognizing American Sign Language (ASL) gestures. They highlighted the importance of understanding ASL gestures to bridge the communication gap between sign language users and the verbal community. Achieving high precision and recall rates, their model demonstrated promising results for real-time gesture recognition. Lastly, Hsieh and Cheng (2021) emphasized the significance of real-time sign language recognition systems in aiding communication with hearing-impaired individuals. Their study particularly targeted Arabic Sign Language (ArSL), which presents unique challenges due to its complex grammar and structure. By employing YOLOv5 and computer vision techniques, they aimed to address these challenges and enhance accessibility for Arabic-speaking deaf communities.

Overall, these studies underscore the potential of YOLOv5-based algorithms in developing practical, efficient, and real-time sign language recognition systems, thereby facilitating inclusive communication for individuals with hearing and speech impairments across different linguistic contexts.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

III. EXISTING SYSTEM

To extract features from the required image, the Deep Learning SSD ML algorithm is used. For the detection, Tensor Flow Object Detection API is used where the extracted features from the pictures taken are passed onto the Tensor Flow module which goes to create comparisons with the real time video present within the frame. On detection of any of those features it's visiting, generate a bounding box round the gesture and make the prediction. The prediction goes to be identical because of the label of the image. We are going to be ready to detect linguistic communication in real time using OpenCV.

IV. PROPOSED SYSTEM

A proposed system for sign language detection would involve multiple components working together to recognize and interpret sign language gestures. Here is a possible overview of the components: Video capture: A camera or webcam captures video footage of the signer's hand gestures and facial expressions. Hand detection: Computer vision techniques are used to identify the signer's hand(s) in the video footage, even if they are partially obscured or moving quickly. and tracking: Once the hand(s) have been detected, computer vision algorithms are used to track their movements over time, allowing the system to recognize when a sign has started and ended.

V. METHODOLOGY

For the proposed real-time sign language detection system using the YOLOv5 algorithm, the methodology can be outlined as follows:

1. Data Collection:

- Obtain a diverse dataset of sign language gestures captured through video recordings or live streams.
- Ensure the dataset covers a wide range of gestures, including different alphabets, numbers, and common signs used in sign language communication.
- Collect data from various sources to capture different lighting conditions, hand orientations, and backgrounds, ensuring robustness of the model.

2. Data Preprocessing:

- Convert video data into frames for processing.
- Resize the frames to a standardized resolution to ensure consistency in the input data.
- Normalize the pixel values to improve model convergence and performance.
- Augment the dataset with techniques such as rotation, scaling, and flipping to increase the diversity of training examples and enhance model generalization.

3. Data Cleaning:

- Remove any corrupted or irrelevant frames from the dataset to ensure data quality.
- Handle any missing or erroneous annotations associated with the sign language gestures.
- Perform outlier detection and removal to eliminate anomalies that may adversely affect model training and performance.

4. Visualization:

- Visualize the preprocessed data to gain insights into the distribution of sign language gestures within the dataset.
- Plot histograms or density plots to analyze the frequency of different gestures and ensure a balanced representation across classes.
- Display sample frames with annotated gestures to verify the correctness of data preprocessing and annotation processes.

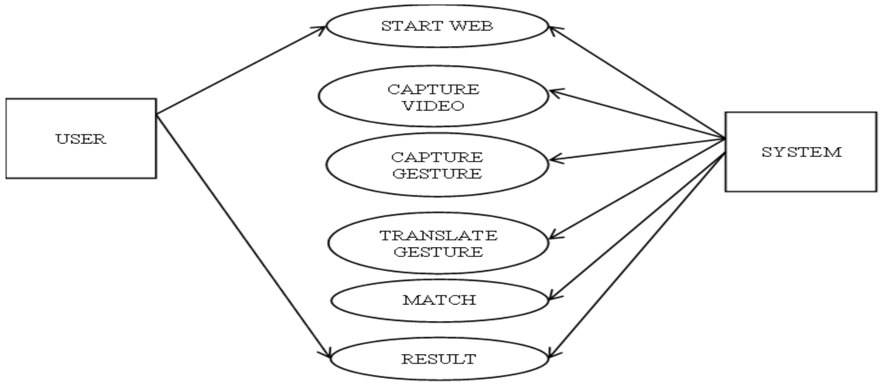

Figure 1: USE CASE DIAGRAM

By following this methodology, researchers can effectively collect, preprocess, clean, and visualize the data necessary for training and evaluating the real-time sign language detection system using the YOLOv5 algorithm. This ensures the quality and integrity of the dataset, leading to more accurate and reliable model performance.

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.

VII. FUTURE ENHANCEMENT

Future enhancements for the real-time sign language detection system using the YOLOv5 algorithm include expanding its capabilities to interpret sequences of gestures, accommodating variations in hand shapes and movements, and integrating real-time feedback mechanisms. Additionally, incorporating gesture translation modules for seamless communication, optimizing for mobile deployment, and designing user-friendly interfaces tailored to sign language users' needs are crucial. Continuous learning mechanisms can adapt the system to evolving sign language usage patterns, ensuring long-term relevance and effectiveness. These advancements aim to improve accuracy, robustness, and accessibility, ultimately fostering better communication and inclusion for individuals with speech or hearing impairments.

VIII. REFERENCE

[1] W. Rongben, G. Lie, T. Bingliang, and J. Lisheng, “Monitoring mouth movement for driver fatigue or distraction with one camera,” in Proc. 7th Int. IEEE Conf. on Intell. Transp. Syst. (ITSC ’04),Washington,USA,Oct.2004, pp.314–19,doi: 10.1109/ITSC.2004.1398917.

[2] L. Bergasa, J. Nuevo, Sotelo, R. Barea, and M. Lopez, “Real-time system for monitoring driver vigilance,” IEEE Trans. Intell. Transp. Syst., vol. 7, no. 1, pp. 63– 77, Mar. 2006, doi: 10.1109/TITS.2006.869598.

[3] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li, “An Overview on Study of Identification of Driver Behavior Characteristics for Automotive Control,” Math. Prob. Eng., Mar. 2014, doi: 10.1155/2014/569109.

[4] J. Angwin, and J. Valentino-DeVries. (Apr. 22, 2011). Apple, Google Collect UserData. The Wall Street Journal. [Online]. Available:
https://www.wsj.com/articles/SB10001424052748703983704576277101723453610

[5] J. Cheng. (Apr. 20, 2011). How Apple Tracks Your Location Without Consent, and Why It Matters. Arstechnica. [Online] http://arstechnica.com/apple/2011/04