IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

SIGN LANGUAGE AND HAND GESTURE RECOGNITION SYSTEM

[1]Mrinal M Prasad, [2]Sreelekshmi B, [3]Akhila M, [4]Rejeesh R, [5]Soorya Narayan
[1]UG Scholar, [2]Assistant Professor, [3 4 5]UG Scholars
[1]Department of Computer Science and Engineering,
[1]Mahaguru Institute of Technology, Pallickal PO, Mavelikara, Koickal – Kattachira Rd, Kattachira, Kerala 690503

Abstract: A current field of study in computer vision and machine learning is the recognition of hand gestures and sign language for human-computer interaction. Making systems that can recognize certain gestures and utilize them to transmit information or control devices is one of its main objectives. However, hand postures are the static structure of the hand, but gestures are the dynamic movement of the hand, and gestures need to be described in the spatial and temporal domains. The two main methods used to recognize hand gestures are vision-based methods and data glove methods. A vision-based system capable of real-time sign language recognition is the major goal of this effort. The rationale behind opting for a vision-based system is that it offers a more straightforward and natural means of interaction between a person and a computer. Given that the human hand is one of the most crucial communication tools in daily life and thanks to the ongoing development of image and video processing methods, research on human-machine interaction through gesture recognition has led to the use of such technology in a huge variety of applications, including touch screens, video game consoles, virtual reality, medical applications, and sign language recognition. Deaf individuals tend to communicate with one another most naturally using sign language, however it has been noted that they have trouble interacting with hearing people in a regular fashion. In the same way that spoken language has a vocabulary of words, sign language has a vocabulary of signs.

***Index Terms* - Computer Vision, Machine Learning, Sign Language, Hand Gesture Recognition**

I. INTRODUCTION

1.1 Introduction

The deaf and hard-of-hearing community has access to essential communication tools like sign language and hand gestures that help them engage with others. The creation of Hand Gesture Recognition Systems and Sign Language has been made possible by technological breakthroughs that aim to close the communication gap. In order to interpret and comprehend hand gestures and sign language, these systems use computer vision, machine learning, and pattern recognition algorithms. This helps people with hearing impairments and those who are not sign language experts communicate with one another. In order to improve accessibility and inclusion for people with hearing and speech impairments[6]. Sign Language and Hand Gesture Recognition System's main objective is to enable effective and efficient communication. Through the translation of sign language and hand motions into text. These methods enable communication with a larger spectrum of individuals whether by voice or writing, dismantling obstacles and promoting equality. Understanding and interpreting the complex hand, finger, and facial expression movements and postures forms the basis of a Sign Language and Hand Gesture Recognition System. These systems may record and examine hand gestures by utilizing cameras, sensors, and specialized algorithms. They can do this by identifying the particular sign language symbols or hand gestures connected to words, sentences, or orders. Systems for recognizing hand gestures and sign language have a huge range of uses. Deaf and hard-of-hearing students can utilize them to successfully communicate with their peers and teachers in educational institutions by using them to enhance inclusive learning environments[14].

What it can do is

- Recognition of hand signs
- Enable effective communication
- Interpret and recognise sign language gestures
- Real-time processing

In healthcare facilities, these technologies help to increase patient and healthcare professional contact, guarantee correct information transmission, and offer a more individualized level of treatment[9].

1.2 Motivation

The motivation behind a Sign Language and Hand Gesture Recognition System is its potential to increase accessibility, enable natural human-computer interaction, improve assistive technologies, facilitate education, foster communication, and increase the capabilities of robotics and virtual reality systems[3]. We can build a more welcoming and engaging world for people with different communication requirements by using technology to decipher and comprehend sign language and hand gestures.

1.3 Problem Statement

A sign language and hand gesture recognition system solves the issue of the requirement for an automated and precise technique of understanding sign language and hand gestures. Hand gestures are frequently utilized for a variety of reasons, including human-computer interface, virtual reality, and robotics. Sign language is an essential tool for those with hearing problems. Manual interpretation and recognition of these motions, however, can be laborious, arbitrary, and prone to mistakes. The problem statement calls for the creation of a system that can accurately and effectively identify and interpret hand and sign language motions[8][13].

1.3.1 Objectives

- Bridges the communication gap between people who cannot speak and the general public.
- Making a computer understand speech, facial expressions and human gestures are the main objective in this project.
- Output in which result can be altered image or report that is based on image analysis.

II. Literature reviews

Literature Review of Sign Language and Hand Gesture Recognition System:

1. **Zhang, J., Li, M., and Li, Y. (2019). a study on hand gesture identification. 20(6), 763-776, Frontiers of Information Technology & Electronic Engineering.**

This review presents an overview of several hand gesture detection technologies, including deep learning and conventional computer vision-based systems. Features including feature extraction, classification techniques, datasets, and assessment measures are among the subjects it covers.

2. **Starner, T., Weaver, J., and Pentland, A. (1998. Using hidden Markov models, real-time recognition of American Sign Language from video. IEEE Transactions on, Pattern Analysis and Machine Intelligence, 20(12), 1371–1375.**

A real-time system for American Sign Language (ASL) recognition utilizing Hidden Markov Models (HMMs) is presented in this fundamental study. The technology was highly accurate in identifying ASL gestures from video clips.

3. **L. Pigou, S. Dieleman, and P. J. Kindermans (2018). employing convolutional neural networks to recognize signs in sign language. (Pp. 232-239) In International Conference on Image Analysis and Recognition.**

The use of convolutional neural networks (CNNs) for sign language recognition is investigated in this paper. It examines several CNN designs and training methods and shows how well CNNs work at detecting both static and moving gestures.

4. **Betke, M., Sclaroff, S., & Athitsos, V. (2003). the video-based gesture recognition system called Poseidon. 53(3), 45–69, International Journal of Computer Vision.**

Real-time hand gesture detection from video sequences is the primary goal of the Poseidon system, which is discussed in this study. For effective recognition, it combines motion-based characteristics with appearance-based features.

These chosen literature sources offer information on the most recent approaches, strategies, and developments in sign language and hand gesture recognition systems. They cover a variety of subjects, including as conventional methods for computer vision, deep learning-based techniques, multimodal fusion, real-time recognition, and particular applications like ASL identification. These publications can be consulted by researchers to obtain a thorough grasp of the subject and to look into potential directions for more study and advancement[6][11].

2.1 Proposed System

Our suggested method is a convolution neural network-based sign language recognition system that identifies different hand motions by recording video and turning it into frames. Following the segmentation of the hand pixels, the picture is obtained and forwarded for comparison with the trained model. Our method is hence better at obtaining accurate text labels for letters.

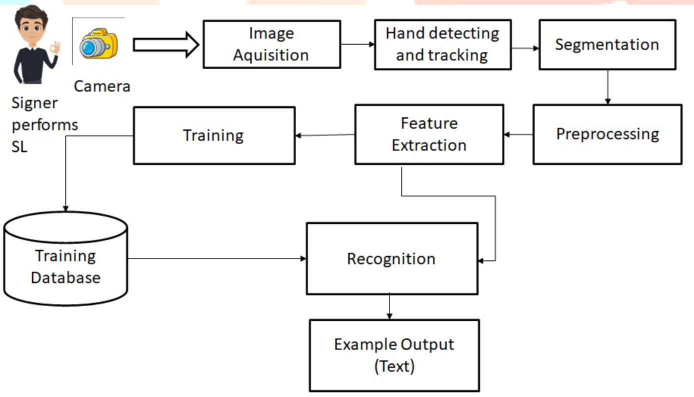

Fig 2.1: Architecture of Sign Language Recognition System

III. METHODOLOGY

3.1 Training Module:

Supervised machine learning: It is one of the ways of machine learning where the model is trained by input data and expected output data. To create such model, it is necessary to go through the following phases:

1. Model construction
2. Model training
3. Model testing
4. Model evaluation

3.1.1 Model Construction:

It depends on machine learning algorithms. In this projects case, it was neural networks.

Begin with its object: model = Sequential()
Then consist of layers with their types: model.add(type_of_layer())
After adding a sufficient number of layers the model is compiled. At this moment Keras communicates with TensorFlow for construction of the model. During model compilation it is important to write a loss function and an optimizer algorithm. It looks like:

model.comile(loss = ‘name_of_loss_function’, optimizer= ‘name_of_opimazer_alg’)

The loss function shows the accuracy of each prediction made by the model.

3.1.2 Model Training:

After model construction it is time for model training. In this phase, the model is trained using training data and expected output for this data. It’s look this way: model.fit(training_data, expected_output). Progress is visible on the console when the script runs. At the end it will report the final accuracy of the model.

3.1.3 Model Testing:

During this phase a second set of data is loaded. This data set has never been seen by the model and therefore it’s true accuracy will be verified. After the model training is completeand it is understood that the model shows the right result, it can be saved by:

model.save(“name_of_file.h5”). Finally, the saved model can be used in the real world. The name of this phase is model evaluation. This means that the model can be used to evaluate new data.

3.2 Pre-processing:

3.2.1 Understanding aspect ratios:

An aspect ratio is a proportional relationship between an image's width and height. Essentially, it describes an image's shape. Aspect ratios are written as a formula of width to height, like this: For example, a square image has an aspect ratio of 1:1, since the height and width are the same. The image could be 500px × 500px, or 1500px × 1500px, and the aspect ratio would still be 1:1.As another example, a portrait-style image might have a ratio of 2:3. With this aspect ratio, the height is 1.5 times longer than the width. So the image could be 500px × 750px, 1500px × 2250px, etc.

3.2.2Cropping to an aspect ratio:

Aside from using built in site style options , you may want to manually crop an image to a certain aspect ratio. For example, if you use product images that have same aspect ratio, they'll all crop the same way on your site.

Option 1 - Crop to a pre-set shape

Use the built-in Image Editor to crop images to a specific shape. After opening the editor, use the crop tool to choose from preset aspect ratios.

Option 2 - Custom dimensions

To crop images to a custom aspect ratio not offered by our built-in Image Editor, use a third-party editor. Since images don’t need to have the same dimensions to have the same aspect ratio, it’s better to crop them to a specific ratio than to try to match their exact dimensions. For best results, crop the shorter side based on the longer side.

- For instance, if your image is 1500px × 1200px, and you want an aspect ratio of 3:1, crop the shorter side to make the image 1500px × 500px.
- Don't scale up the longer side; this can make your image blurry.

Image scaling:

In computer graphics and digital imaging, image scaling refers to the resizing of a digital image. In video technology, the magnification of digital material is known as upscaling or resolution enhancement. When scaling a vector graphic image, the graphic primitives that make up the image can be scaled using geometric transformations, with no loss of image quality. When scaling a raster graphics image, a new image with a higher or lower number of pixels must be generated. In the case of decreasing the pixel number (scaling down) this usually results in a visible quality loss. From the standpoint of digital signal processing, the scaling of raster graphics is a two- dimensional example of sample-rate conversion, the conversion of a discrete signal from a sampling rate (in this case the local sampling rate) to another.

Datasets used for training

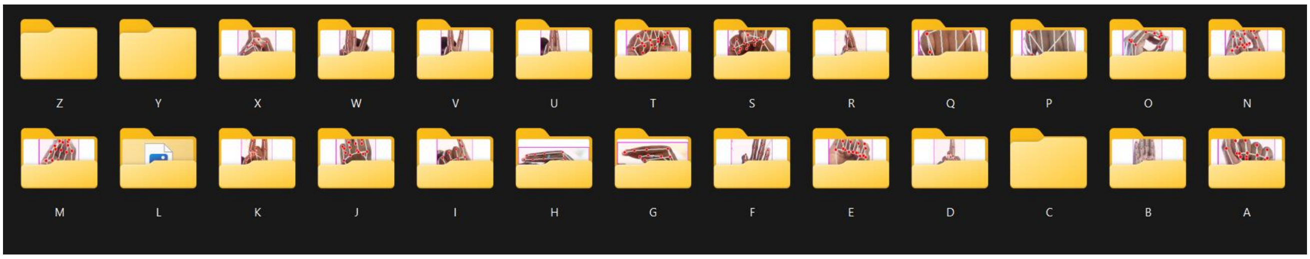

Fig 3.1: Dataset used for training the model

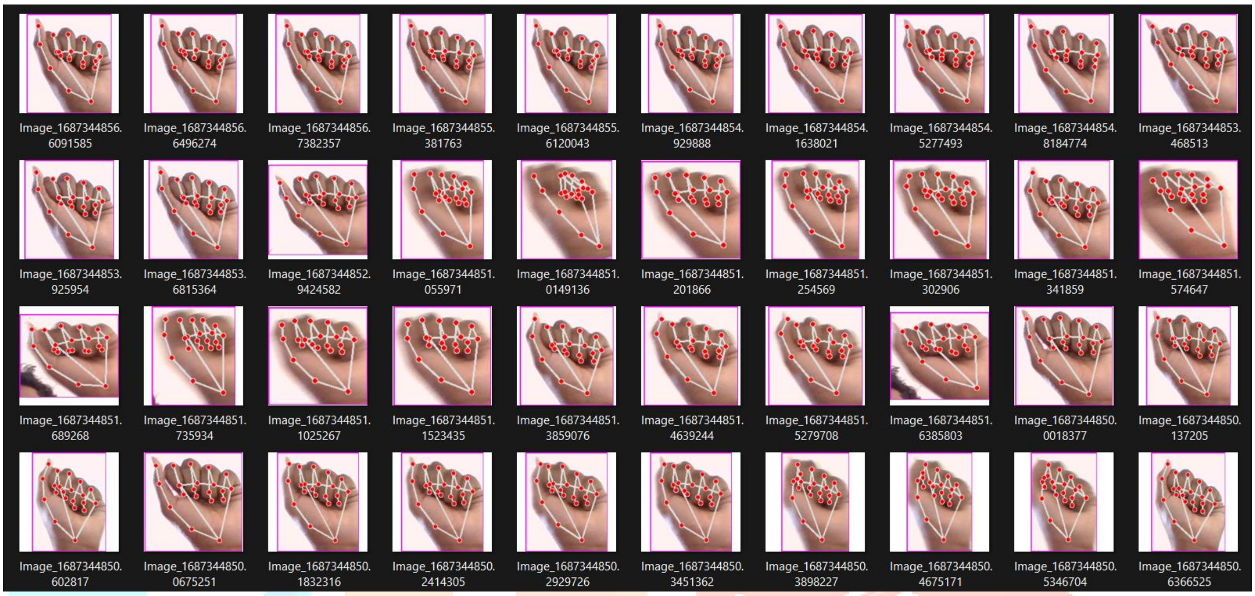

Fig 3.2: Training data given for Letter A

3.3 Convolution Operation:

In purely mathematical terms, convolution is a function derived from two given functions by integration which expresses how the shape of one is modified by the other.

Convolution formula:

$$(f * g)(t) \overset{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

Here are the three elements that enter into the convolution operation:

- Input image
- Feature detector
- Feature map

Steps to apply convolution layer:

- You place it over the input image beginning from the top-left corner within the borders you see demarcated above, and then you count the number of cells in which the feature detector matches the input image.
- The number of matching cells is then inserted in the top-left cell of the feature map. You then move the feature detector one cell to the right and do the same thing. This movement is called a and since we are moving the feature detector one cell at time, that would be called a stride of one pixel.
- What you will find in this example is that the feature detector's middle-left cell with the number 1 inside it matches the cell that it is standing over inside the input image. That's the only matching cell, and so you write “1” in the next cell in the feature map, and so on and so forth.

IV. SYSTEM ANALYSIS

4.1 TenserFlow

A library for dataflow and differentiable programming used for a variety of applications called TensorFlow is free and open source software.

4.2 OpenCV

A collection of programming functions with a focus on real-time computer vision is called OpenCV (Open Source Computer Vision collection).It was initially created by Intel, then backed by Willow Garage and Itseez (which Intel eventually purchased). Under the terms of the open-source BSD license, the library is free to use and cross-platform.

4.3 Keras

Python-based Keras is an open-source library for neural networks. It may function on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML, among other frameworks. It focuses on being user-friendly, modular, and extendable in order to enable quick experimentation with deep neural networks. Its major inventor and maintainer is François Chollet, a Google engineer, and it was created as a component of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). The XCeption deep neural network model was also created by Chollet.

4.4 NumPy

A library for the Python programming language called NumPy adds support for big, multidimensional arrays and matrices as well as a ton of high-level mathematical operations that can be performed on these arrays. Jim Hugunin and a number of other developers worked together to produce Numeric, the predecessor to NumPy. By heavily altering Numeric and combining features from the rival Numarray, Travis Oliphant built NumPy in 2005. Numerous people contribute to NumPy, an open-source program[1][4].

4.5 Hardware Requirements:

- Camera:Good quality,3MP
- Ram: Minimum 8GB or higher
- GPU: 4GB dedicated
- Processor: Intel Pentium 4 or higher
- HDD: 10GB or higher
- Monitor: 15” or 17” colour monitor
- Mouse: Scroll or Optical Mouse or Touch Pad
- Keyboard: Standard 110 keys keyboard

4.6 Software Requirements:

- Operating System: Windows, Mac, Linux
- SDK: TensorFlow,OpenCV, Keras, NumPy

V. PROPOSED METHODOLOGY

5.1 Data Flow Diagram

A data flow (DFD) diagram enhances the process characteristics by graphically outlining the movement of information or data. In order to create a holistic overview without getting into too much detail—which may be described later—DFD is frequently utilized as the first stage. Data processing is visualized using DFDs. The DFD outlines the kinds of data that will be input into and retrieved from the system, as well as how and where the data will be created inside the system. In contrast to a systematic flow chart that emphasizes control flow or the UML function flow diagram, which introduces both control and data flow, this graphic does not include information on the process time or whether the processes will operate sequentially or similarly.

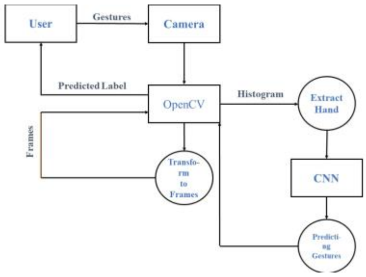

Fig 5.1: Data Flow Diagram for Sign Language Recognition

5.2 Use Case Diagram

Use case for representing system performance between the explanation and analysis of requirements. Use case examples to describe how the system works and how it helps the actor see outcomes. By instructing on the system's boundaries and representing the work they have done as well as the entire environment, actors and their issue cases may be identified. Actors are not included in the system, but instances are. The use case explains the system using the actor's behavior as an example. It represents the work done by the system as a series of actions that give the actor tangible outcomes.

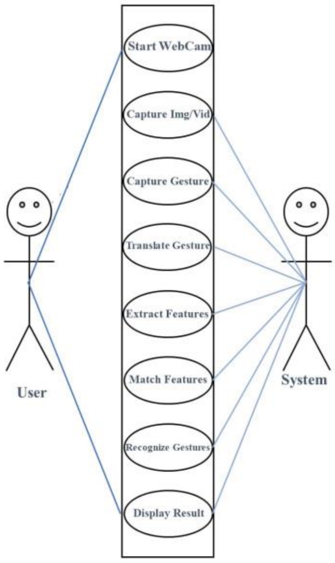

Fig 5.2: Use Case Diagram for Sign Language Recognition

5.3 Class Diagram

Create the class's structure and content components using the class drawing, package, and object marked designs. When building the technique, concept, result, and outcome, the class outlines how it approached the figure. The three components of a class are its name, its attributes, and its actions. Relationships like inheritance, cohabitation, and so on are also shown in class diagrams. The most typical relation in a class diagram is relationship. The term "association" describes the connection between class instances.

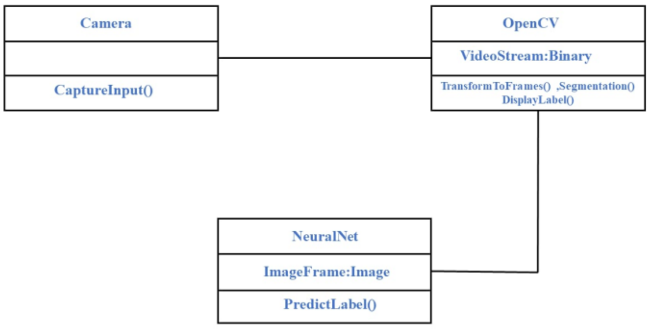

Fig 5.3: Class Diagram for Sign Language Recognition

5.4 Sequence Diagram

The sequence diagram depicts how several things interact when they are chronologically presented. It refers to the sequence in which messages must be sent and received by view-included objects and other objects required to accomplish class and visual functionality. In the case of a logical method, the system under development k is obtained and sequence diagrams are frequently employed. Events, or events that happen, are happenings. Interactions are shown when messages are entered with horizontal arrows, with the message name above. Asynchronous communications are represented by open arrows, synchronous calls are represented by solid arrow tops, and reply messages are represented by dashed lines. If the caller delivers a synchronous message, one must wait for the message to finish before doing an action, such issuing a subroutine instruction. It can go on processing without stopping if the caller delivers an asynchronous message. Applications with several threads, those that are event-driven, and middleware that uses messages all use asynchronous calls. An execution statement (message response in UML) is being processed when an opaque rectangle called an activation box or a method-call box is drawn on the lifeline.

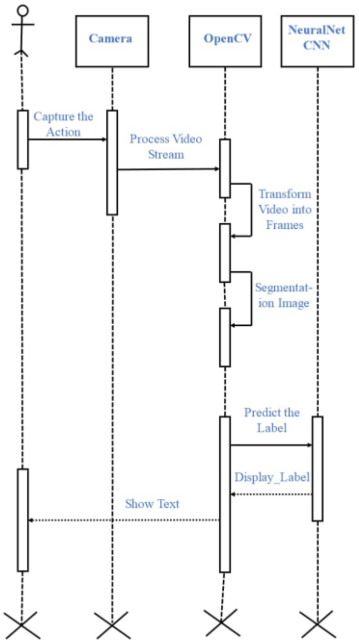

Fig 5.4: Sequence Diagram for Sign Language Recognition

5.4 State Diagram

To signal the level of the next process, object calling methods add additional activation boxes to the second vertex and utilize messages. The lifeline will have an X and a dash line written beneath it if an item is destroyed (removed from memory). It must be the outcome of the communication, whether it came from the item or anything else. Extrinsic message sequences (gates in the UML) or circles (queries in the UML) can be used to exhibit extraordinary messages. Extrinsic message sequences (gates in the UML) or circles (queries in the UML) can be used to exhibit extraordinary messages. Links between several pieces are used to simulate parallelism, conditional branching, and alternative interactions.

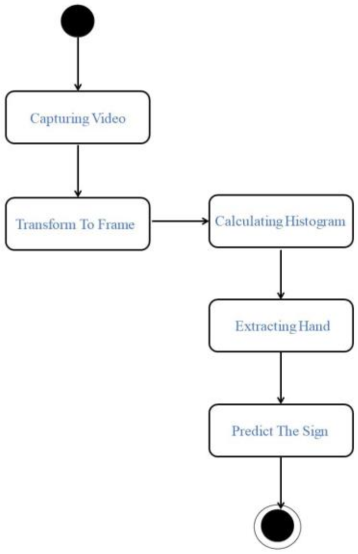

Fig 5.5: State Diagram for Sign Language Recognition

VI. RESULT AND DISCUSSION

Accurate recognition is the key indicator of a sign language and hand gesture recognition system's performance. This indicator shows how accurately the system can recognize and comprehend gestures. Higher accuracy rates are preferred since they provide efficient user-system interaction and communication. The quality and selection of the datasets used for training and assessment have a significant impact on how well the system performs. Datasets featuring a variety of gestures, variations, and environmental factors are more useful for simulating real-world situations and increase the resilience of the system[12]. The effectiveness of the system is significantly influenced by the use of suitable categorization methods. Deep learning models like Convolutional Neural Networks (CNNs) or Random Forests, as well as machine learning algorithms like Support Vector Machines (SVM), k-Nearest Neighbours (k-NN). Successful applications of recurrent neural networks (RNNs) have been made. Comparisons between various approaches might shed light on which ones work best for a certain gesture detection job[15]. Interactive applications require real-time processing. The system's applicability for real-time applications depends on how quickly and effectively its recognition process, which includes pre-processing, feature extraction, and classification, works. To achieve quicker inference times, optimizations like hardware acceleration or parallel processing might be used. The usage of sign language and hand gestures varies greatly depending on the audience, the situation, and the surroundings. In order to provide accurate recognition in a variety of settings, a robust system should be able to manage changes in hand forms, motions, speeds, occlusions, and lighting conditions.

In conclusion, a variety of pictures are now required by apps as information sources for clarification and analysis. To conduct a variety of applications, certain characteristics must be extracted. When a picture is changed from one form to another, such as when digitizing, scanning, sharing, storing, etc., deterioration takes place. As a result, the final picture must go through a process known as image enhancement, which entails a variety of techniques aimed at enhancing an image's visual presence. The main goal of image enhancement is to improve the input for other autonomous image processing systems while also improving the interpretability or awareness of information in pictures for human viewers. To make the image more understandable by the computer, the image is then subjected to feature extraction utilizing a variety of techniques. A strong instrument for preparing expert knowledge, edge detection, and the synthesis of false information from several sources is a sign language recognition system. Convolution neural networks' goal is to achieve the correct categorization[7][10].

VII. FUTURE SCOPE OF THE PROJECT

The Sign Language and Hand Gesture Recognition System has a broad range of potential uses and developments in the future. Key areas for growth in the future include:

- **Greater Recognition Accuracy**: Systems for recognizing gestures still have space for development. Exploring more sophisticated deep learning approaches, such as using larger and more varied datasets, cutting-edge network topologies, and improved feature representations, can lead to improvements. For certain sign languages or user-specific movements, recognition performance can be improved using strategies like transfer learning and domain adaptation.

- **Low-Latency and Real-Time Processing**: Interactive applications require real-time performance. Future research might concentrate on streamlining the processing pipeline to improve the speed and decrease latency of gesture recognition systems. Real-time performance on devices with limited resources can be achieved by using hardware acceleration, parallel processing, or edge computing.

- **Continuous Gesture Recognition**: A key component of natural and uninterrupted engagement is the ability to recognize gestures continuously. Future systems can concentrate on identifying continuous, dynamic sign language motions while also recording the temporal elements and changes in gesture. This would make it easier for people and the system to communicate effectively and continuously.

- **Multimodal Fusion**: The robustness and accuracy of gesture recognition systems may be increased by combining many modalities, such as vision, depth, audio, and wearable sensors. The capacity of the system to manage changes in illumination, occlusions, and various user settings can be improved by fusing several modalities to offer complementing information.

- **User Adaptation and Personalization**: Adapting and personalizing systems to each user's unique hand shapes, motions, and signing styles can improve identification accuracy. It is possible to investigate user-specific models or online learning techniques to dynamically update the system depending on user feedback and enhance performance for specific users over time.

- **Real-World Deployment and Accessibility**: The future scope involves implementing sign language and gesture detection systems in actual settings, therefore ensuring their widespread accessibility and use. For those with hearing loss, language problems, or in different human-machine interaction settings, integration with assistive technology, communication devices, virtual reality systems, or robots can be useful.

- **Gesture-Based Control and Interaction**: Gesture recognition systems may be improved to allow sophisticated gesture-based control in interactive systems including robotics, virtual reality, gaming, smart homes, and others. This covers the deft and natural operation of robotic systems, the manipulation of virtual objects, immersive games, and the seamless integration with smart devices and Internet of Things (IoT) platforms.

- **Multilingual and Cultural Adaptation**: For universal accessibility, gesture recognition systems must be expanded to handle a variety of sign languages and cultural gestures. Research may concentrate on creating models that can generalize across various sign languages and adjust to cultural variances in hand movements.

The future potential of Sign Language and Hand Gesture Recognition System includes improvements in accuracy, real-time processing, flexibility, multimodal fusion, user personalization, real-world deployment, and broadening the range of applications and accessibility. More effective, efficient, and inclusive gesture recognition and communication systems will be developed with further study and innovation in these fields.

VIII. REFERENCES

[1] Salem Ameen, Sunil Vadera University of Salford 2017, Classify American Sign Language Finger spelling from Depth and Colour Images.

[2] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[3] https://www.iosrjen.org/Papers/vol3_issue2%20(part-2)/H03224551.pdf

[4] Faria, B.M., et al., Evaluation of distinct input methods of an intelligent wheelchair in simulated and real environments: a performance and usability study. Assistive technology : the official journal of RESNA, 2013. 25(2): p. 88-98.

[5] Blum, A. and T. Mitchell, Combining labeled and unlabeled data with co-training, in Proceedings of the eleventh annual conference on Computational learning theory1998, ACM: Madison, Wisconsin, United States. p. 92-100.

[6] Zhang, D. and G. Lu, A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures. Journal of Visual Communication and Image Representation, 2003(14 (1)): p.41-60.

[7] Arabic sign language recognition with 3D convolutional neural networks, 2017.

[8] Andersen, M.R., et al., Kinect Depth Sensor Evaluation for Computer Vision Applications, in Technical report ECE-TR-6 2012, Department of Engineering – Electrical and Computer Engineering, Aarhus University. p.37.

[9] Kauppinen, H., T. Seppanen, and M. Pietikainen, An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995. 17(2): p. 201-207.

[10] Zafrulla, Z., et al., American sign language recognition with the kinect, in 13th International Conference on Multimodal Interfaces2011, ACM: Alicante, Spain. p.279-286.

[11] Andersen, M.R., et al., Kinect Depth Sensor Evaluation for Computer Vision Applications, in Technical report ECE-TR-6 2012, Department of Engineering – Electrical and Computer Engineering, Aarhus University. p.37.

[12] Madankar, M., Chandak, M. B., & Chavhan, N. (2016). Information retrieval system and machine translation: A review. Procedia Computer Science, 78, 845-850. https://doi.org/10.1016/j.procs.2016.02.071

[13] Lieberman, Z., T. Watson, and A. Castro. Open Frameworks. 2004 10 October 2013 [cited 2011; open Frameworks is an open source C++ toolkit designed to assist the creative process by providing a simple and intuitive framework for experimentation. Available from: http://www.openframeworks.cc/

[14] OpenNI. The standard framework for 3D sensing. 2013; Available from: http://www.openni.org/

[15] Miner, R. RapidMiner: Report the Future. December 2011. Available from: http://rapid-i.com/