Project Id: 2023PJ-IT06

# Synopsis
on

# Sign Language Detection

**Submitted In Partial Fulfillment of the Requirement**

**for the Degree of**

**Bachelor of Technology**

**In**

**Information Technology**

**By**


**Kaustubh Ranjan** **Roll No. 1902900130027**

**Manak Rawal** **Roll No.1902900130034**


**Under the Supervision of**
**Prof. Shivani Sharma**


**ABES INSTITUTE OF TECHNOLOGY**
**GHAZIABAD**



**AFFILIATED TO**
**Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY**
**LUCKNOW, UTTAR PRADESH**
**(SEPTEMBER-2022)**

# INDEX

# OBJECTIVE

The aim of this proposed work is to develop a Sign Language Recognition System that takes video input to track hands, identify its actions, and convert & predict those actions into signs, ultimately giving output as respective letters of the English alphabets.

The Objectives of this proposed work are:

1) To create benchmark datasets with smartphone cameras.
2) To develop a web application for sign language recognition.
3) To achieve higher accuracy than existing solutions.

# PROBLEM STATEMENT

Communication is basic and necessary for an individual's life. But some are born with the lack of ability to either convey or collect thoughts. While others encounter mis happenings that carry away their natural capability to communicate.

Use of Sign Languages is a popular & effective way to facilitate the deaf and dumb community, but it has its own limitations too, primarily being the lack of sign language interpreters. This creates two difficulties, one being the non-availability of the sign language interpreters itself and second being the non-reachability of the available interpreters.

## ABSTRACT

Sign Language is an essential tool to bridge the communication gap between normal and hearing-impaired people. It is one of the communication tools for the hard-of-hearing people community and common people community but, normal people find it hard to understand the sign language and gestures of the deaf and dumb. Many tools can be used to translate the sign language created by the disabled into a form that normal people can understand. The studies are based on various image acquisition, pre-processing, hand gesture segmentation, extraction of features, and classification methods.

The important element used in sign language recognition (SLR) as the input method is the camera. The input data for the SLR is in the form of a live feed that can easily be recorded by a camera. The most existing SLR through image processing uses high quality cameras whereas to reach larger audience, we propose using of normal cameras like of smartphones and webcams, which themselves are getting better as technology evolves, and performs even better in our proposed work due to availability of more suitable dataset.

The proposed work presented here shows the importance of incorporating intelligent solution into the sign language recognition systems and meets the requirement of a sign language recognition (SLR) web portal that is universally and easily available for every needful person having a better accuracy than existing solutions and along with this, a dataset is also prepared for future references. Overall, it is expected that this study will facilitate knowledge accumulation and creation of intelligent-based SLR and provide readers, researchers, and practitioners a roadmap to guide future direction.

# INTRODUCTION

Sign language is the mode of communication which uses visual ways like expressions, hand gestures, and body movements to convey meaning. Sign language is extremely helpful for people who face difficulty with hearing or speaking. Sign language recognition refers to the conversion of these gestures into words or alphabets of existing formally spoken languages. Thus, conversion of sign language into words by an algorithm or a model can help bridge the gap between people with hearing or speaking impairment and the rest of the world.

Vision-based hand gesture recognition is an area of active current research in computer vision and machine learning. Being a natural way of human interaction, it is an area where many researchers are working on, with the goal of making human computer interaction easier and natural, without the need for any extra devices. So, the primary goal of gesture recognition research is to create systems, which can identify specific human gestures and use them. For example, to convey information. For that, vision-based hand gesture interfaces require fast and extremely robust hand detection, and gesture recognition in real time. Hand gestures are a powerful human communication modality with lots of potential applications and in this context, we have sign language recognition, the communication method of deaf and mute people.

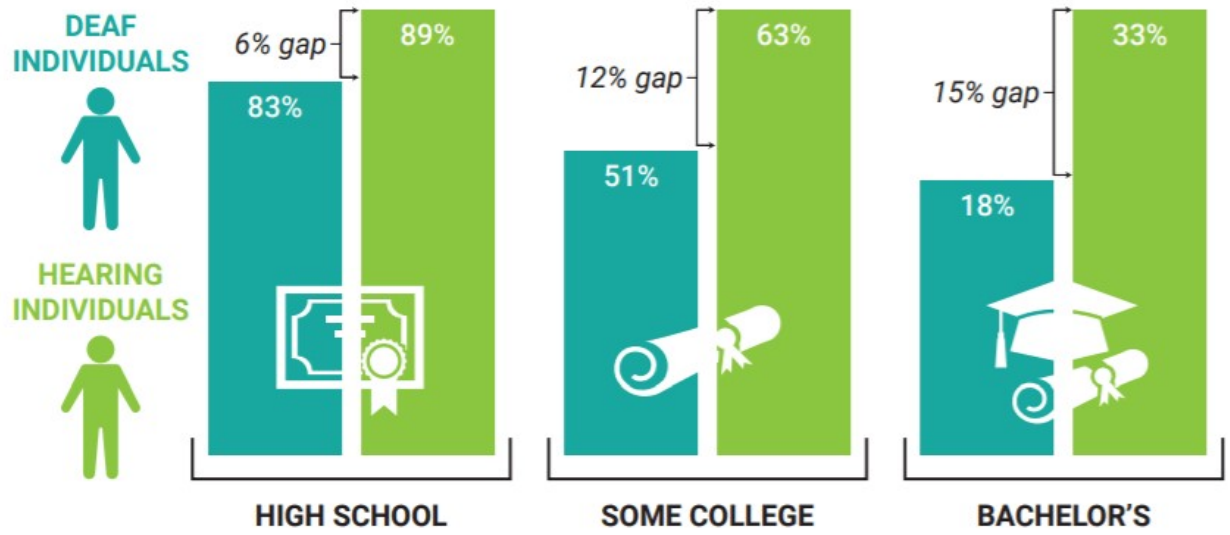



*Figure-1: Academic achievements analysis of DHH in US (2017)*

[1]Figure-1 shows academic achievements of DHH (Deaf or Hard of Hearing) and normal hearing people. Only 18% of DHH students earn heir undergraduate degree compared to 33% of normal hearing individuals. Looking at this graph, we can see the achievement gap doubles after high school.


*Figure-2: Employment analysis of DHH in US (2016)*

[1]Figure-2 depicts employment of DHH (Deaf or Hard of Hearing) and normal hearing people. Additionally, only 48% of individuals who are DHH were employed, compared to 72% of hearing people. While these statistics may not be surprising to professionals who support DHH students, they continue to be alarming and motivating.

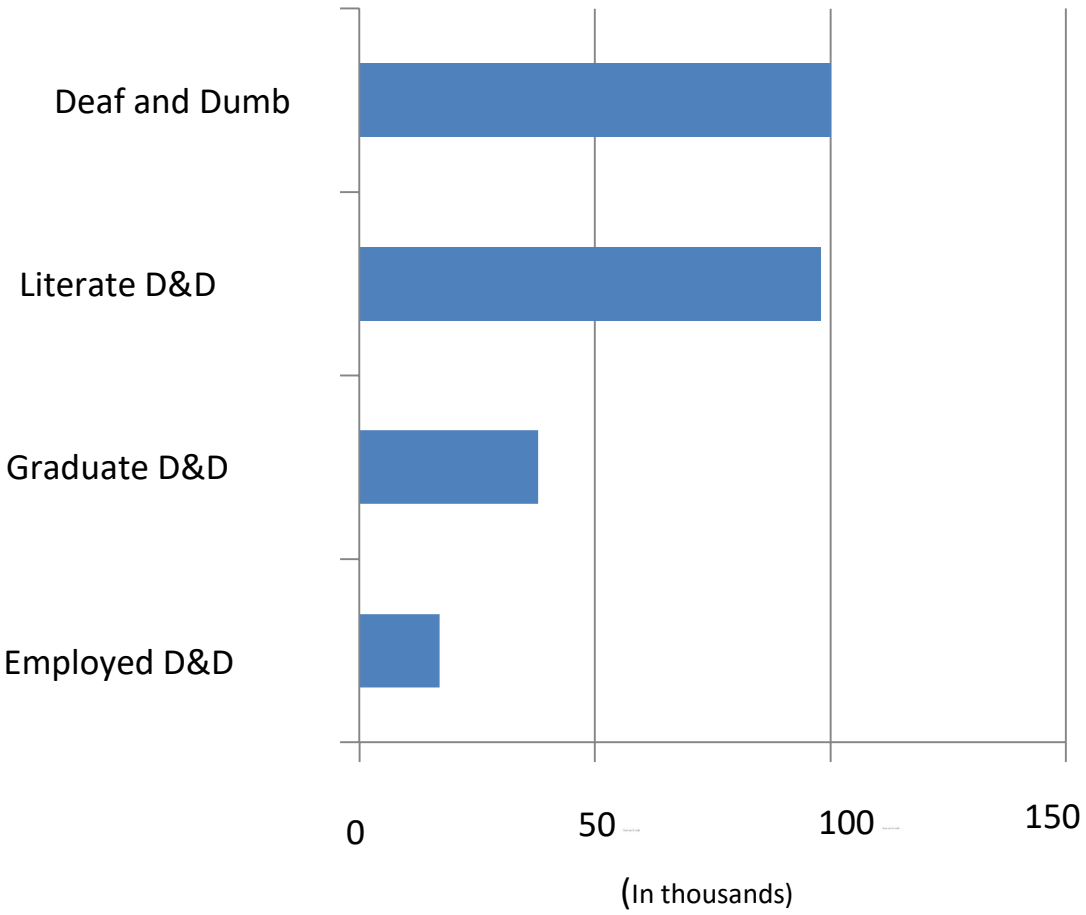



*Figure-3: Employment analysis of deaf and dumb population of India*

[2]Figure-3 allows us to take a notice on the situation of D&D (Deaf and Dumb) in India. Not only are these statistics important to educators and policy makers, but also to the teens we work with. When they turn their graduation tassel from the right to the left, that is the last time they can fully rely on supportive adults in school to advocate for them.

# LITERATURE SURVEY

*Table-1:Literature Survey on Sign Language Detection System*

| Author | Proposed Work | Advantages | Limitations |
|---|---|---|---|
| [3](Adeyanju et al., 2021) | Using Camera/Webcam as Image Acquisition Technique | The device does not require wearing other external devices, and users only need to use their hands within the camera collection range. Low cost. Convenient and easy to use. | It is greatly affected by environmental factors such as light, skin colour, and occlusion. It requires several image processing techniques which might affect the recognition accuracy. |
| [3](Adeyanju et al., 2021) | Using Data Glove as Image Acquisition Technique | These devices are not Affected by the external environment when collecting data. It provides an improved recognition accuracy. Extraction of features with sensor based is relatively easier. | It reduces the naturalness of interaction. Inconvenient to use by user. Very expensive |
| [3](Adeyanju et al., 2021) | Using Kinetic as Image Acquisition Technique | It is useful for many human computer interaction applications. The depth of distance detection is limited. | It can be affected by lighting conditions, hand and face segmentation, complex background, and noise. Kinect not suitable for outdoor applications and sensitivity to sunlight. |
| [3](Adeyanju et al., 2021) | Using Leap motion controller as Image Acquisition Technique | It has high recognition accuracy and faster processing speed around 200 frames per second. It can detect and track hands, fingers, and finger-like objects. | Due to it highly sensitive, accuracy of the recognition might be affected with small movement in sign position. |

| | | | |
|---|---|---|---|
| [4](Verma et al., 2017) | Using Histogram Equalization(HE) as Image Enhancement Technique | It's simple to implement and highly effective for grayscale images. | It may increase the contrast of background noise. It is difficult to distinguish between noise and the desired features. It changes the brightness of an image. |
| [5](Kamal et al., 2019) | Using Adaptive Histogram Equalization(AHE) as Image Enhancement Technique | It is suitable to enhance local contrast and edges in every region of an image. It outperforms the histogram equalization technique. | It has an adverse effect on desired output due to its noise-amplification behaviour. It fails to retain the Brightness on the input image. |
| [6](Suharjito et al., 2017) | Using Contrast Limited Adaptive Histogram Equalization (CLAHE) as Image Enhancement Technique | It has a reduced noise compared to AHE and HE. It provides local output response and avoids brightness saturation | It produces an unsatisfactory result when the input image has an unbalanced contrast ratio and increased brightness |
| [7](Chourasiya & Khare, 2019) | Using as Logarithmic transformation Image Enhancement Technique | It is used to reduce higher intensities pixel values into lower intensities pixel values | Applying the technique to a higher pixel value will enhance the image more and cause loss of actual information in the image. It does not apply to all kinds of images. |
| [8](Dhanushree et al., 2019) | Using Median Filter as an image filtering technique. | It preserves thin edges and sharpness from an input image. Both of the problems of the mean filter are tackled by the median filter. | It is relatively expensive and Complex to compute. It is good only for removing salt and Pepper noise. It is less effective at removing the gaussian type of noise from the image. |

| | | | |
|---|---|---|---|
| [9](Kasmin, 2020) | Using Mean Filter as an image filtering technique. | Easy to implement. | A single wrongly represented pixel value can significantly impact the mean value of all pixels in their immediate neighborhood. It blurs an edge when the filter neighborhood crosses a boundary. |
| [10](Basu, 2002) | Using Gaussian Filter as an image filtering technique. | It is effective for removing the gaussian type of noise. The weights give higher significance to pixels near the edge. | It has high computational time and sometimes removes edges details in an image. |
| [11], [12](Kaluri & Pradeep Reddy, 2016a, 2016b) | Using Adaptive Filter as an image filtering technique. | It preserves edges and other high-frequency parts better than a similar linear filter | It is computational complexity. There are still some visible distortions available in the image using an adaptive filter. . |
| [13], [14](Maru et al., 2017; Tania & Rowaida, 2016) | Using Wiener Filter as an image filtering technique. | It is a popular filter used for image restoration. It is not sensitive to noise. Suitable to exploit the statistical properties of the image. The small window size can be used to prevent blurring of edges. | Prior knowledge of the power spectral density of the original image is unavailable in practice. It is comparatively slow to apply because it works in the frequency domain. The output image is very blurred. |
| [15](Lee et al., 1990) [16](Cheng et al., 2002) [17](Xu et al., 2013) [18](Dong et al., 2008) | Using Thresholding Method as segmentation technique. | It is a fast and straightforward approach. It does not require prior information to operate. It has a low computation cost. | It is highly dependent on peaks, while spatial details are not considered. Sensitive to noise. Selection of an optimal threshold value is difficult. |
| [19], [20](Bhardwaj & Mittal, 2012; Rashmi et al., 2013) | Using Edge based Method as segmentation technique. | Suitable for images having better contrast between objects. | It is not suitable for images with too much noise or too many edges. |

| | | | |
|---|---|---|---|
| [21](Garcia-Lamont et al., 2018) | Using Region-based Method as segmentation technique. | It is less susceptible to noise and more useful when defining similarity criteria is easy. | It is quite expensive in terms of computation time and memory consumption. |
| [22], [23](Cebeci & Yildiz, 2015; Ghosh & Dubey, 2013) | Using Clustering Method as segmentation technique. | It's more useful for real-world challenges due to the fuzzy partial membership employed. | Determining membership functions is not easy. |
| [24](Khan, 2014) | Using Artificial Neural-Network based Method as segmentation technique. | It does not require a complex program to work. Less prone to noise. | Computational time in training is higher. |

# ARCHITECTURAL DIAGRAM

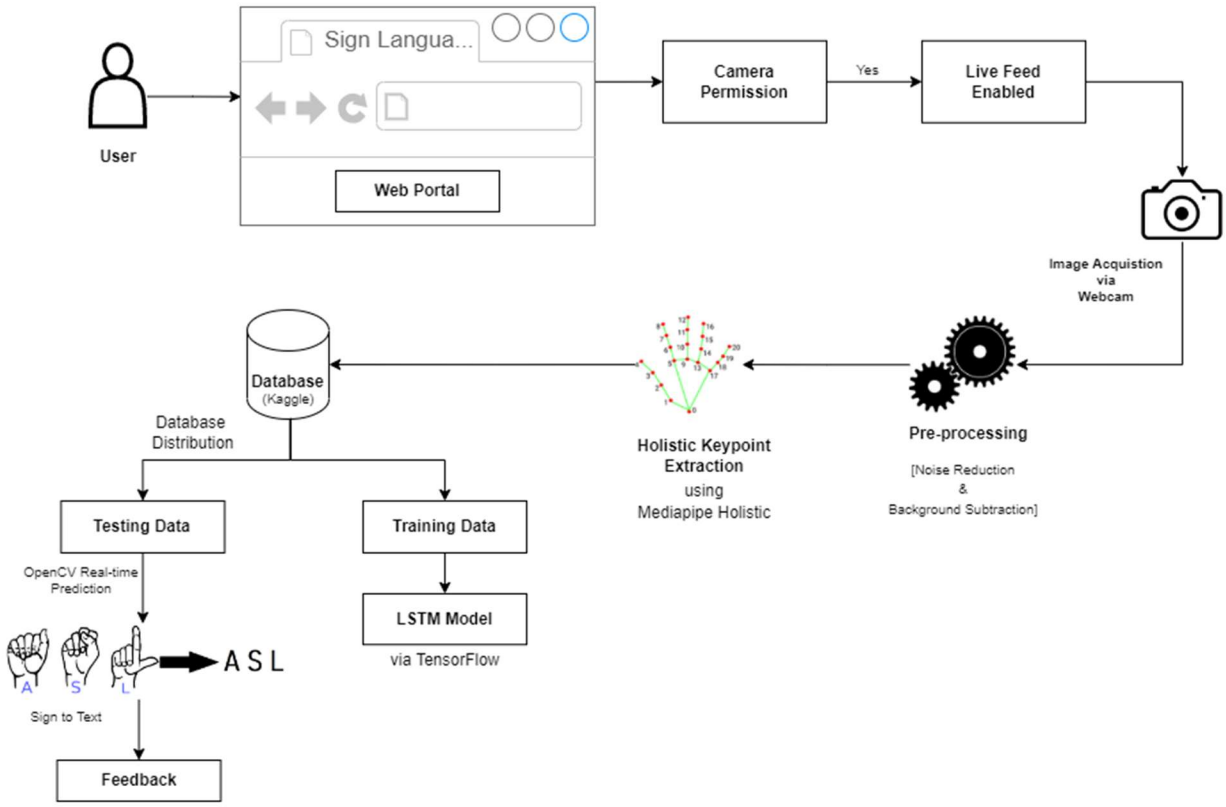



*Figure-4: Architectural Diagram for Sign Language Detection*

The architectural diagram in Figure-4 describes working infrastructure of the proposed work. It depicts a user logging in to the web portal provided for the user interaction and after accepting requests for required permissions, it briefly introduces to the functioning of camera as the primary means of interplay between detection system and user. Fleeting operations including pre-processing, key points extraction and database accessing takes place. Data from database is used for training and testing, where testing data is also the feeded input data that is converted from gestures to signs and finally the user experience is received as feedback.

# ACTIVITY DIAGRAM

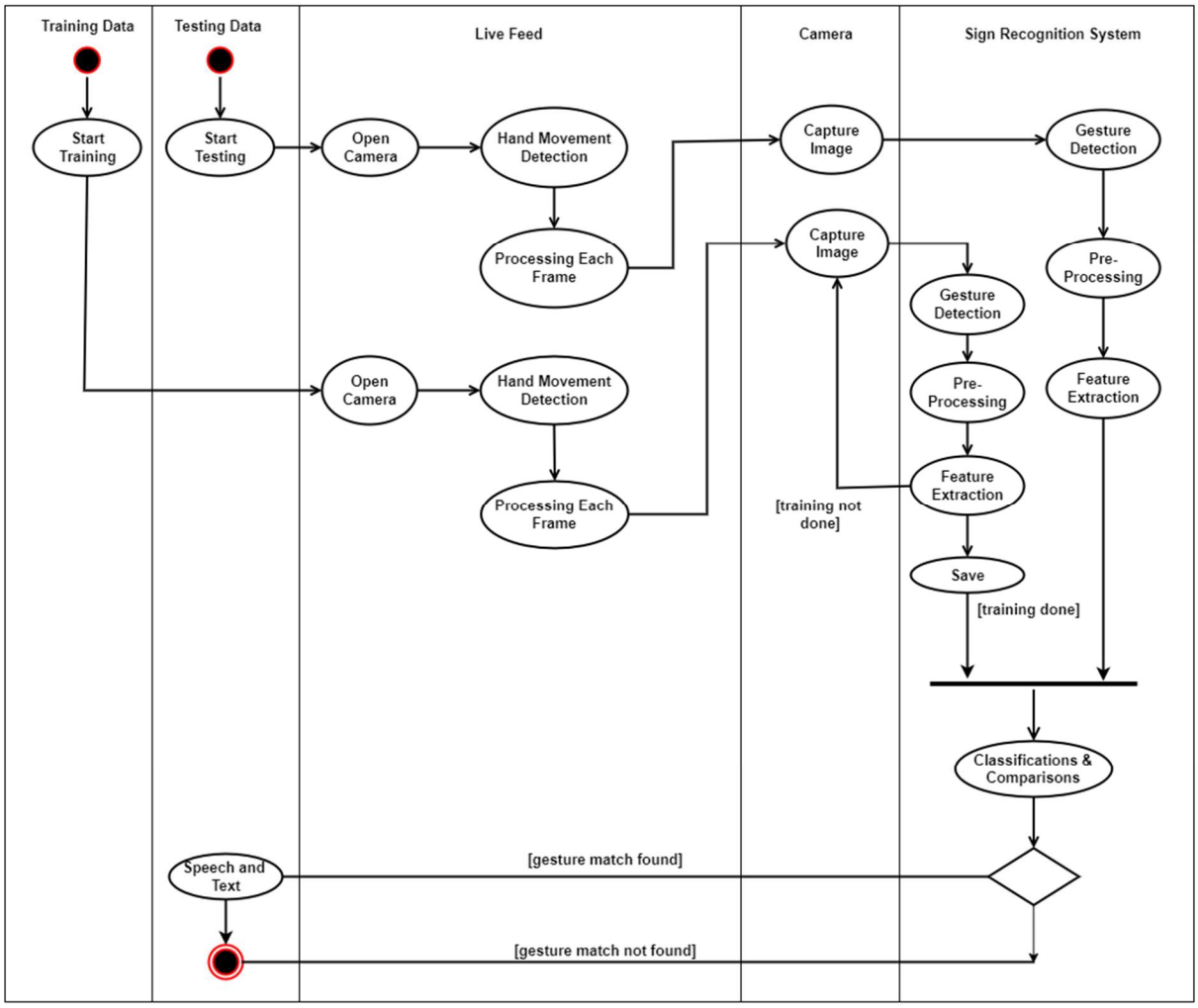



*Figure-5: Activity Diagram for Sign Language Detection*

Shown above in Figure-5 is the activity diagram which elaborates the working progression of the proposed work. Initially, the data into the Sign Language Detection System (SLDS) is inputted via the database. Two different types of data i.e., training and testing data is used and both follow the same advancement when live feed is switched on, on opening of camera. Processing each frame, hand movement is detected and images captured are sent for gesture detection. It is done after the captured images have been pre-processed with operations like noise removal and features like its key points are extracted. For the training data, the processed images are stored and sent for classification, where the testing data is directly entered. Finally, after the model compares the gestures, if suitable gesture is found, it is converted into text and if not, the process is stopped, ultimately bringing an end to the SLDS.

# TOOLS AND TECHNIQUES

The domain analysis that we have done for the project mainly involved understanding the pattern recognition.

- **TensorFlow**

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

Features: TensorFlow provides stable Python (for version 3.7 across all platforms) and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (early release). Third-party packages are available for C#, Haskell Julia, MATLAB, R, Scala, Rust, OCaml, and Crystal. "New language support should be built on top of the C API. However, not all functionality is available in C yet." Some more functionality is provided by the Python API.

Application: Among the applications for which TensorFlow is the foundation, are automated image-captioning software, such as DeepDream.

- **OpenCV**

OpenCV (Open-Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.[1] Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.
OpenCV's application areas include:

- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition

To support some of the above areas, OpenCV includes a statistical machine earning library that contains:

- k-nearest neighbour algorithm
- Naive Bayes classifier
- Artificial neural networks

- Support vector machine (SVM)
- Deep neural networks (DNN) Image-Processing:

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.
8
If we talk about the basic definition of image processing then “Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality”.

Digital-Image:

An image may be defined as a two-dimensional function f (x, y), where x and y are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.
In another word an image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f (x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be.
Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.
Image processing basically includes the following three steps:
Importing the image
Analysing and manipulating the image

Output in which result can be altered image or report that is based on image analysis

Applications of Computer Vision:

Here we have listed down some of major domains where Computer Vision is heavily used.

- Robotics Application
- Localization − Determine robot location automatically
- Navigation
- Obstacles avoidance

- **Numpy**

NumPy is a library for the Python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open- source software and has many contributors. Features: NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and

functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using NumPy.

Limitations: Inserting or appending entries to an array is not as trivially possible as it is with Python's lists. The np.pad(...) routine to extend arrays actually creates new arrays of the desired shape and padding values, copies the given array into the new one and returns it. NumPy'snp.concatenate([a1, a2]) operation does not actually link the two arrays but returns a new one, filled with the entries from both given arrays in sequence. Reshaping the dimensionality of an array with np.reshape(...) is only possible as long as the number of elements in the array does not change. These circumstances originate from the fact that NumPy's arrays must be views on contiguous memory buffers. A replacement package called Blaze attempts to overcome this limitation.

- **MediaPipe Holistic**

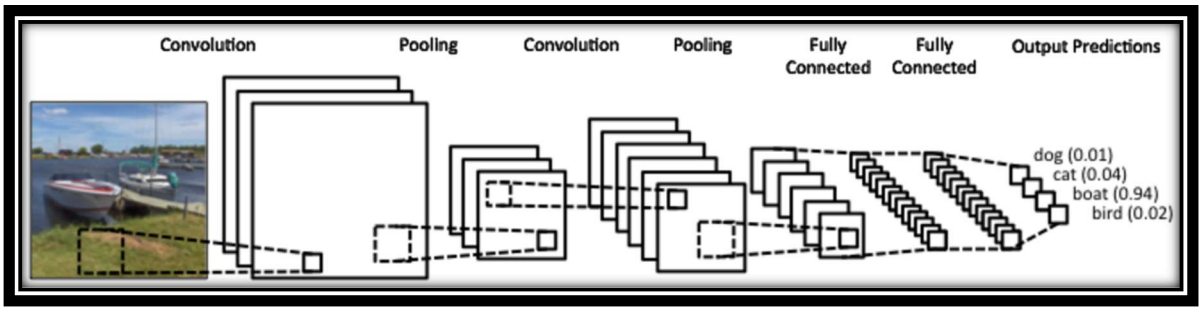
The MediaPipe Holistic pipeline integrates separate models for pose, face and hand components, each of which are optimized for their particular domain. However, because of their different specializations, the input to one component is not well-suited for the others. The pose estimation model, for example, takes a lower, fixed resolution video frame (256x256) as input. But if one were to crop the hand and face regions from that image to pass to their respective models, the image resolution would be too low for accurate articulation. Therefore, we designed MediaPipe Holistic as a multi-stage pipeline, which treats the different regions using a region appropriate image resolution.

The pipeline is implemented as a MediaPipe graph that uses a holistic landmark subgraph from the holistic landmark module and renders using a dedicated holistic

renderer subgraph. The holistic landmark subgraph internally uses a pose landmark module , hand landmark module and face landmark module.

- **Long Short-Term Memory (LSTM)**

Long Short-Term Memory (LSTM) networks were designed specifically to overcome the long-term dependency problem faced by recurrent neural networks RNNs (due to the vanishing gradient problem). LSTMs have feedback connections which make them different to more traditional feedforward neural networks. This property enables LSTMs to process entire sequences of data (e.g. time series) without treating each point in the sequence independently, but rather, retaining useful information about previous data in the sequence to help with the processing of new data points.



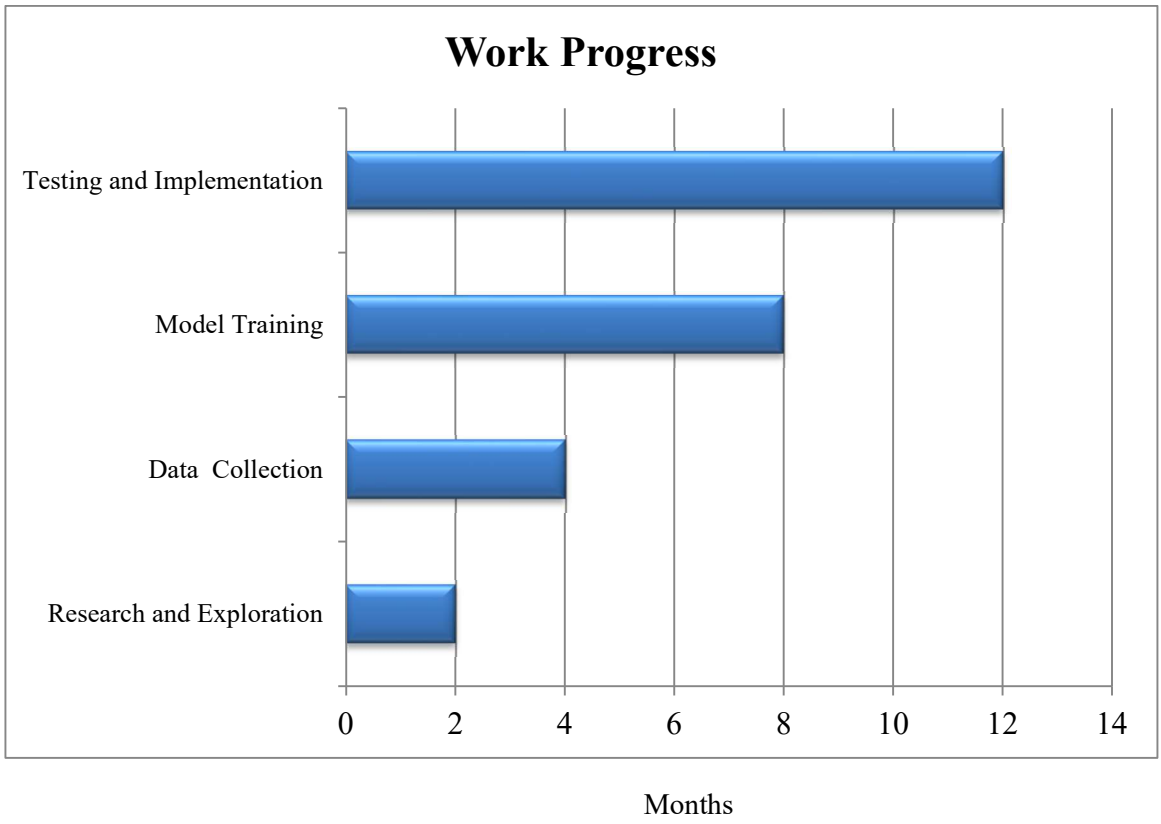

As a result, LSTMs are particularly good at processing sequences of data such as text, speech and general time-series.

LSTM networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

This is a behavior required in complex problem domains like machine translation, speech recognition, and more.

LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field.

# PROJECT PLAN



# REFERENCES


[1] "Helping teens with hearing loss succeed | Audiology Blog." https://audiologyblog.phonakpro.com/helping-teens-with-hearing-loss-succeed/ (accessed Dec. 13, 2022).

[2] S. Mondal, T. Lohar, and A. Naik, "Interactive System for Deaf and Dumb People using Hand Gestures," *International Research Journal of Engineering and Technology*, 2020, Accessed: Dec. 13, 2022. [Online]. Available: www.irjet.net

[3] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis," *Intelligent Systems with Applications*, vol. 12, p. 56, 2021, doi: 10.1016/j.iswa.2021.20.

[4] N. Verma, M. D.-J. of Engineering, T. (IRJET), and undefined 2017, "Contrast enhancement techniques: A brief and concise review," *academia.edu*, 2017, [Online]. Available: https://www.academia.edu/download/54173429/IRJET-V4I7375.pdf

[5] S. M. Kamal, Y. Chen, S. Li, X. Shi, and J. Zheng, "Technical Approaches to Chinese Sign Language Processing: A Review," *IEEE Access*, vol. 7, pp. 96926–96935, 2019, doi: 10.1109/ACCESS.2019.2929174.

[6] Suharjito, R. Anderson, F. Wiryana, M. C. Ariesta, and G. P. Kusuma, "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output," in *Procedia Computer Science*, 2017, vol. 116, pp. 441–448. doi: 10.1016/j.procs.2017.10.028.

[7] A. Chourasiya and N. Khare, "A Comprehensive Review Of Image Enhancement Techniques," *International Journal of Innovative Research and Growth*, vol. 8, no. 6, May 2019, doi: 10.26671/ijirg.2019.6.8.101.

[8] M. Dhanushree, R. Priyadharsini, and T. S. Sharmila, "Acoustic image denoising using various spatial filtering techniques," *International Journal of Information Technology 2019 11:4*, vol. 11, no. 4, pp. 659–665, Nov. 2019, doi: 10.1007/S41870-018-0272-3.

[9] F. Kasmin, "A Comparative Analysis of Filters towards Sign Language Recognition," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 4772–4782, Aug. 2020, doi: 10.30534/ijatcse/2020/84942020.

[10] M. Basu, "Gaussian-based edge-detection methods - A survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 32, no. 3, pp. 252–260, Nov. 2002, doi: 10.1109/TSMCC.2002.804448.

[11] R. Kaluri and C. Pradeep Reddy, "A framework for sign gesture recognition using improved genetic algorithm and adaptive filter," *Cogent Eng*, vol. 3, no. 1, p. 1251730, Oct. 2016, doi: 10.1080/23311916.2016.1251730.

[12] R. Kaluri and C. H. Pradeep Reddy, "Sign gesture recognition using modified region growing algorithm and Adaptive Genetic Fuzzy Classifier," *International Journal of Intelligent Engineering and Systems*, vol. 9, no. 4, pp. 225–233, 2016, doi: 10.22266/ijies2016.1231.24.

[13] S. Tania and R. Rowaida, "A Comparative Study of Various Image Filtering Techniques for Removing Various Noisy Pixels in Aerial Image," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 3, pp. 113–124, Mar. 2016, doi: 10.14257/ijsip.2016.9.3.10.

[14] M. Maru, P. G. Scholar, and M. C. Parikh, "Image Restoration Techniques: A Survey," 2017.

[15] S. U. Lee, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," *Comput Vis Graph Image Process*, vol. 52, no. 2, pp. 171–190, Nov. 1990, doi: 10.1016/0734-189X(90)90053-X.

[16] H. D. Cheng, X. H. Jiang, and J. Wang, "Color image segmentation based on homogram thresholding and region merging," *Pattern Recognit*, vol. 35, no. 2, pp. 373–393, Nov. 2002, doi: 10.1016/S0031-3203(01)00054-1.

[17] W. Xu, Q. Li, H. J. Feng, Z. H. Xu, and Y. T. Chen, "A novel star image thresholding method for

effective segmentation and centroid statistics," *Optik (Stuttg)*, vol. 124, no. 20, pp. 4673–4677, Nov. 2013, doi: 10.1016/J.IJLEO.2013.01.067.
[18] L. Dong, G. Yu, P. Ogunbona, and W. Li, "An efficient iterative algorithm for image thresholding," *Pattern Recognit Lett*, vol. 29, no. 9, pp. 1311–1316, Nov. 2008, doi: 10.1016/J.PATREC.2008.02.001.
[19] Rashmi, M. Kumar, and R. Saxena, "Algorithm and Technique on Various Edge Detection : A Survey," *Signal Image Process*, vol. 4, no. 3, pp. 65–75, Jun. 2013, doi: 10.5121/sipij.2013.4306.
[20] S. Bhardwaj and A. Mittal, "A Survey on Various Edge Detector Techniques," *Procedia Technology*, vol. 4, pp. 220–226, 2012, doi: 10.1016/j.protcy.2012.05.033.
[21] F. Garcia-Lamont, J. Cervantes, A. López, and L. Rodriguez, "Segmentation of images by color features: A survey," *Neurocomputing*, vol. 292, pp. 1–27, Nov. 2018, doi: 10.1016/J.NEUCOM.2018.01.091.
[22] Z. Cebeci and F. Yildiz, "Comparison of K-Means and Fuzzy C-Means Algorithms on Different Cluster Structures," *Journal of Agricultural Informatics*, vol. 6, no. 3, Nov. 2015, doi: 10.17700/JAI.2015.6.3.196.
[23] S. Ghosh and S. K. Dubey, "Comparative Analysis of K-Means and Fuzzy C-Means Algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 4, 2013, doi: 10.14569/IJACSA.2013.040406.
[24] W. Khan, "Image Segmentation Techniques: A Survey," *Journal of Image and Graphics*, pp. 166–170, 2014, doi: 10.12720/JOIG.1.4.166-170.