WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

Sign Language Training Tool Using Machine Learning Techniques

Mahalakshmi V, Asst. Prof. Dr. E. Ranjith

Krishnasamy College of Engineering and Technology, Cuddalore.

ABSTRACT

Sign language recognition has been widely used for communication amongst the hearing-impaired and non-verbal community. From early electric signal-based sign language recognition to modern-day recognition using machine/deep learning techniques, researchers all over the world have tried to automate this task. This project mainly aim to carry out key point detection based sign language recognition(SLR).Used different machine learning algorithm like random forest, support vector machine and neighbour to train the model. Finally the best model is selected from the model testing using evaluation metrics namely f1score, precision and recall. Simple GUI is designed to get the user input and the prediction is carried out by the best machine learning model.Media Pipe is used to predict the hand key points from web camera. Using OpenCV the video is captured using webcam and it get compared with the model and the model will predict the sign letter, once the first letter A is predicted it will pass to 2nd B letter and the time is calculated for letter a to z.

1. INTRODUCTION

Sign language is an essential tool to bridge the communication gap between normal and hearing-impaired people. However, the diversity of over 7000 present-day sign languages with variability in motion position, hand shape, and position of body parts making automatic sign language recognition (ASLR) a complex system. In order to overcome such complexity, researchers are investigating better ways of developing

ASLR systems to seek intelligent solutions and have demonstrated remarkable success. Each method has its own strength compare to other methods and researchers are still using different methods in developing their own Sign Language Recognition. Each method also has its own limitations compared to other methods. The aim of this paper is to review the sign language recognition approaches and design a best training tool for visually challenged people.

Sign language recognition is the process of recognizing and interpreting human hand gestures, movements, and poses used in sign languages, which are used by deaf and hard-of-hearing individuals to communicate with each other and with the hearing community. Sign languages are complex visual languages that use a combination of hand gestures, facial expressions, and body movements to convey meaning. They are not simply a visual representation of spoken languages, but are distinct and independent languages with their own grammar, syntax, and vocabulary. There are many different sign languages around the world, each with its own regional variations and dialects.

2. LITERATURE SURVEY

In the recent years there has been tremendous research done on the hand gesture recognition.

With the help of literature survey done we realized the basic steps in hand gesture recognition are :-

- Data acquisition
- Data preprocessing
- Feature extraction
- Gesture classification

Data acquisition

The different approaches to acquire data about the hand gesture can be done in the following ways:

1. Use of sensory devices

It uses electromechanical devices to provide exact hand configuration, and position. Different glove based approaches can be used to extract information .But it is expensive and not user friendly.

2. Vision based approach

In vision based methods computer camera is the input device for observing the information of hands or fingers. The Vision Based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices. These systems tend to complement biological vision

Describing artificial vision systems that are implemented in software and/or hardware.

The main challenge of vision-based hand detection is to cope with the large variability of human hand's appearance due to a huge number of hand movements, to different skin-colour possibilities as well as to the variations in view points, scales, and speed of the camera capturing the scene.

Data preprocessing and Feature extraction for vision based approach

• In the approach for hand detection combines threshold-based color detection with background subtraction. We can use Adaboost face detector to differentiate between faces and hands as both involve similar skin-color.

• We can also extract necessary image which is to be trained by applying a filter called Gaussian blur. The filter can be easily applied using open computer vision also known as OpenCV.

• For extracting necessary image which is to be trained we can use instrumented gloves. This helps reduce computation time for preprocessing and can give us more concise and accurate data compared to applying filters on data received from video extraction.

• We tried doing the hand segmentation of an image using color segmentation techniques but as mentioned in the research paper skin color and tone is highlighting on the lighting conditions due to which output we got for the segmentation we tried to do were no so great. Moreover we have a huge number of symbols to be trained for our project many of which look similar to each other like the gesture for symbol 'V' and digit '2', hence we decided that in order to produce better accuracies for our large number of symbols, rather than segmenting the hand out of a random background we keep background of

hand a stable single color so that we don't need to segment it on the basis of skin color . This would help us to get better results.

Gesture classification

• In Hidden Markov Models (HMM) is used for the classification of the gestures .This model deals with dynamic aspects of gestures. Gestures are extracted from a sequence of video images by tracking the skin-colour blobs corresponding to the hand into a body– face space centered on the face of the user. The goal is to recognize two classes of gestures: deictic and symbolic. The image is filtered using a fast look–up indexing table. After filtering, skin colour pixels are gathered into blobs. Blobs are statistical objects based on the location (x,y) and the colour metry (Y,U,V) of the skin colour pixels in order to determine homogeneous areas.

• In Naïve Bayes Classifier is used which is an effective and fast method for static hand gesture recognition. It is based on classifying the different gestures according to geometric based invariants which are obtained from image data according to geometric based invariants which are obtained from image data after segmentation. Thus, unlike many other recognition methods, this method is not dependent on skin colour. The gestures are extracted from each frame of the video, with a static background. The first step is to segment and label the objects of interest and to extract geometric invariants from them. Next step is

the classification of gestures by using a K nearest neighbor algorithm aided with distance weighting algorithm (KNNDW) to provide suitable data for a locally weighted Naïve Bayes‟ classifier.

- According to paper on "Human Hand Gesture Recognition Using a Convolution Neural Network" by Hsien-I Lin , Ming-Hsiang Hsu, and Wei-Kai Chen graduates of Institute of Automation Technology National Taipei University of Technology Taipei, Taiwan, they construct a skin model to extract the hand out of an image and then apply binary threshold to the whole image. After obtaining the threshold image they calibrate it about the principal axis in order to center the image about it. They input this image to a convolutional neural network model in order to train and predict the outputs. They have trained their model over 7 hand gestures and using their model they produce an accuracy of around 95% for those 7 gestures.

3. PROPOSED SYSTEM

The proposed system provides an application where all types of workers, organizational sectors, companies can use the system to satisfy their requirement.

System Modules

- DATA COLLECTION
- PREPROCESSING
- MODEL TRAINING
- MODEL TESTING
- PERFORMANCE EVALUATION
- OUTPUT PREDICTION

Module Description

1.DATA COLLECTION

- The data set is a collection of data in pickle file format from kaggle website.
- Contain 747 row and 29 columns.
- Here we converted the pickle file to csv file format. Pkl to . csv is to **use the to_csv() method from the pandas library**

2. PREPROCESSING

- Checking the units of data.
- Assign dependent variable and independent variable.
- splitting the data into training data and testing data.
- The code sets the value of "units_in_data" to 28, meaning there are 28 units in the data. Then, it creates an empty list "titles" and uses a for loop to iterate over the range of units. For each iteration, it creates a string "unit-i" (where "i" is the loop index) and appends it to the "titles" list. The final list "titles" will contain 28 strings, with each string being "unit-0", "unit-1", "unit-2", ..., "unit-27".

3. MODEL TRAINING

Three Algorithm is used random forest , support vector machine and k nearest neighbour.

Random forest

Random Forest is known for its ability to handle noisy data and its ability to estimate feature importance, making it a popular algorithm for many real-world applications. Random Forest is an ensemble machine learning algorithm that operates by constructing a multitude of decision trees and aggregating their predictions. It is used for both regression and classification problems.

The code is created for Random Forest Classifier algorithm using the scikit-learn library. It sets the number of trees in the forest to 30 using the "n_estimators" parameter. The ".fit" method trains the classifier on the training data "X_train" and corresponding target values "y_train".The ".predict" method is then used to generate predictions for the test data "X_test".Finally, the code prints the accuracy of the predictions using the "metrics.accuracy_score" function from scikit-learn, which compares the actual target values "y_test" to the predicted values "y_pred". The accuracy score is the proportion of correct predictions in the test data.

Support vector machine

Support Vector Machine (SVM) is a type of machine learning algorithm that can be used for sign language recognition. SVM works by finding a boundary between classes in the feature space that maximizes the margin, which is the distance between the boundary and the closest data points from each class. The data points closest to the boundary are known as support vectors and have the greatest impact on the boundary. SVM can handle both linear and non-linear decision boundaries, making it a versatile algorithm for sign language recognition. Additionally, SVM can handle high-dimensional data, which is often the case in sign language recognition where multiple features such as hand shape, orientation, and motion need to be considered.

KNEAREST NEIGHBOUR

K-Nearest Neighbors (KNN) is another machine learning algorithm that can be used for sign language recognition. KNN operates by classifying a data point based on the majority class of its K nearest neighbors in the feature space. The value of K is a hyperparameter that can be tuned for optimal performance. In sign language recognition, KNN can be used to identify the sign language gesture that is closest to a given test sample based on its features such as hand shape, orientation, and motion. KNN is a simple and intuitive algorithm that is easy to implement, but it can be computationally expensive for large datasets and may not perform well with high-dimensional data.

4. MODEL TESTING

Model testing is an essential step in evaluating the performance of machine learning models. In the context of sign language recognition using random forest, k-nearest neighbors (KNN), and support vector machine (SVM), the following steps can be taken for model testing:

Model Evaluation: Evaluate the performance of the models using the testing set. Common performance metrics for classification tasks include accuracy, precision, recall, and F1 score. It is also essential to examine the confusion matrix to identify which classes are often confused by the models.

1. Confusion matrix
2. Accuracy
3. Precision
4. Recall
5. F1 score

For simplicity, we will mostly discuss things in terms of a binary classification problem where let's say we'll have to find if an image is of a cat or a dog. Or a patient is having cancer (positive) or is found healthy (negative). Some common terms to be clear with are:

True positives (TP): Predicted positive and are actually positive.

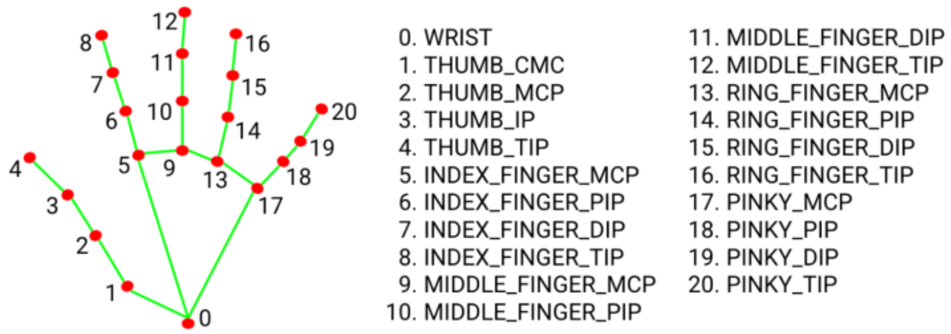
False positives (FP): Predicted positive and are actually negative.

True negatives (TN): Predicted negative and are actually negative.

False negatives (FN): Predicted negative and are actually positive.

5. OUTPUT PREDICTION

- Sign language recognition module: A module that recognizes the sign language gestures performed by the user and provides feedback on accuracy.



Mediapipe hands object as input and returns a dictionary containing the distance data and the original image with annotated landmarks.

1.The function first converts the image to RGB and processes it with the Mediapipe hands object.

2.It then extracts the hand landmark data from the Mediapipe output and calculates the distances between various landmarks on the hand.

3.The resulting distance data is stored in a list and returned along with the annotated image.

4. SYSTEM REQUIREMENT

The purpose of system requirement specification is to produce the specification analysis of the task and also to establish complete information about the requirement, behavior and other constraints such as functional performance and so on. The goal of system requirement specification is to completely specify the technical requirements for the product in a concise and unambiguous manner.

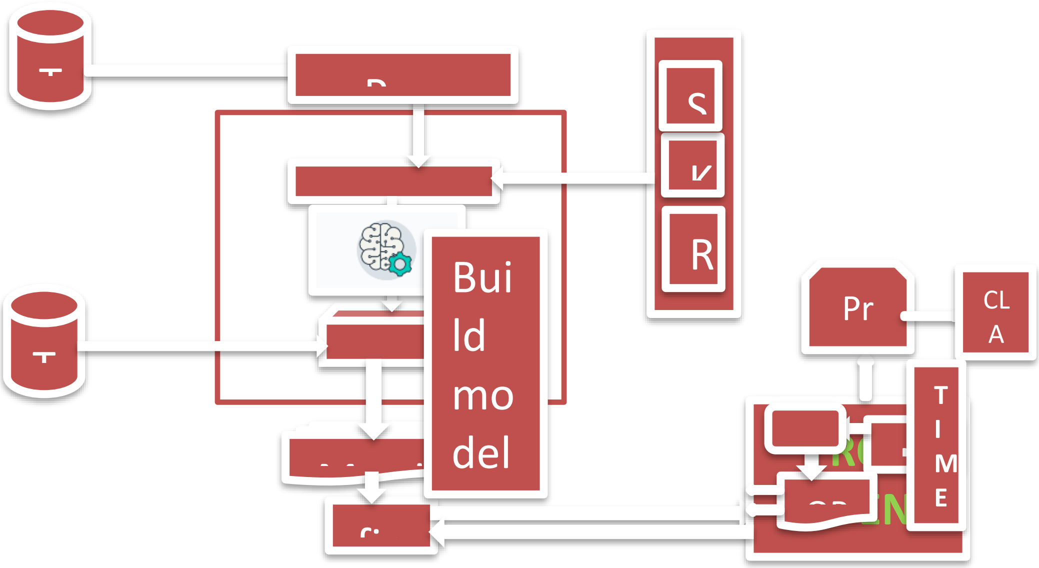
Hardware Requirements

- Processor - Intel i3 above
- Speed - 3.19 GHZ
- RAM - minimum 4GB
- SSD - 10 GB above

Software Requirements

- Operating System - Windows 10
- Front End - HTML & CSS
- Back End - PYTHON
- Tool - Anaconda ,Pycharm

5. ARCHITECTURE DIAGRAM



6. RESULT

In this section, analyze the results of the proposed system. The screenshots are the results of the system.

Figure 6.1: Confusion matrix of Random forest

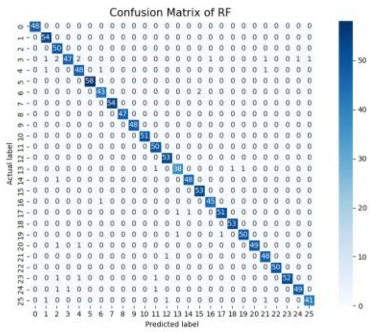


Figure 6.2: Learning sign language

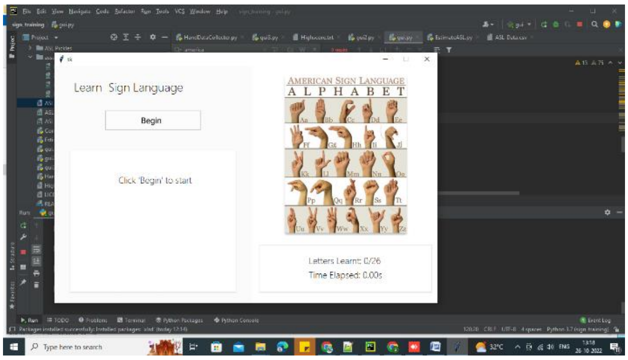
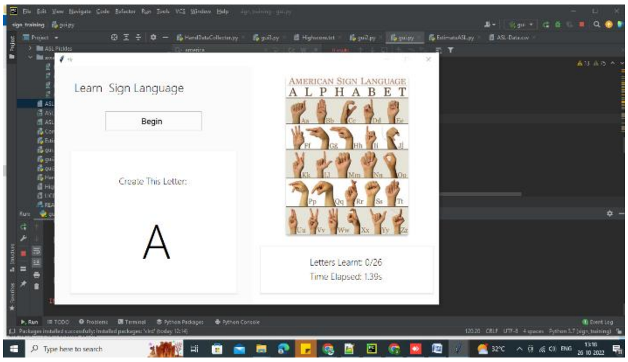


Figure 6.3: Detecting sign language



The development and implementation of a sign language recognition system using machine learning and computer vision techniques have the potential to greatly benefit the hearing-impaired and non-verbal community. By accurately detecting and interpreting sign language gestures, the system can facilitate communication and bridge the gap between individuals with hearing disabilities and the general population. To ensure the system's performance and accuracy, extensive performance evaluation measures, such as accuracy assessment, confusion matrix, evaluation metrics, cross-validation, and user feedback, need to be implemented. This evaluation process helps identify areas for improvement, addresses false positives or negatives, and ensures the system's effectiveness in recognizing sign language gestures.

Overall, the successful implementation of the sign language recognition system, supported by a user-friendly interface, accurate prediction, and efficient time calculation, holds great potential in promoting inclusivity and enhancing communication for individuals with hearing disabilities. Ongoing system maintenance, updates, and user support are essential to address any issues, incorporate user feedback, and ensure the system's long-term functionality and effectiveness.

7. FUTURE WORK

Expanded Sign Language Vocabulary: Currently, the system may be designed to recognize a limited set of sign language gestures or individual letters. A future enhancement could involve expanding the vocabulary to include more complex phrases or words, allowing for more comprehensive communication. Real-Time Translate: Integrating real-time translation capabilities can be a valuable addition to the system. This would involve converting the recognized sign language gestures into text or spoken language, facilitating communication between sign language users and non-sign language speakers. Improved Gesture Recognition: Enhancing the accuracy and robustness of gesture recognition is an ongoing area of research. This can involve exploring advanced machine learning techniques, such as deep learning, and leveraging larger and more diverse datasets to train the models. Multi-modal Input: Currently, the system may rely on video input from a webcam to detect sign language gestures. However, incorporating other input modalities, such as depth sensors or wearable devices, can provide additional information and improve the system's performance and accuracy. Adaptability to User Preferences: Allowing users to customize and adapt the system to their specific sign language dialect or preferences can be valuable. This may involve incorporating user profiles and preferences into the system to personalize the recognition process.

REFERENCES

1. Duarte, S. Palaskar, L. Ventura, D. Ghadiyaram, K. DeHaan, F. Metze, J. Torres, and X. Giro-i-Nieto. “How2Sign: A Large-scale Multimodal Dataset for Continuous American Sign Language”. In: Conference on Computer Vision and Pattern Recognition (CVPR). 2021.

2. World Health Organization 2021 Deafness and hearing loss. 2021. url: https : / / www . who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (visited on 12/15/2021).

3. Graves, A.-r. Mohamed, and G. E. Hinton. “Speech Recognition with Deep Recurrent Neural Networks”. In: CoRR abs/1303.5778 (2013). arXiv: 1303 . 5778. url: http : //arxiv.org/abs/1303.5778.

4. H. Brashear, T. Starner, P. Lukowicz, and H. Junker. “Using multiple sensors for mobile sign language recognition”. In: Nov. 2005, pp. 45–52. isbn: 0-7695-2034-0. doi: 10.1109/ISWC.2003.1241392. [

5. D. Uebersax, J. Gall, M. van den Bergh, and L. V. Gool. “Real-time sign language letter and word recognition from depth data”. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops) (2011), pp. 383–390.

6. S. A. Mehdi and Y. N. Khan. “Sign language recognition using sensor gloves”. In: Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP ’02. 5 (2002), 2204–2206 vol.5.

7. **Z**. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. “American sign language recognition with the kinect”. In: Proceedings of the 13th international conference on multimodal interfaces. 2011, pp. 279–286.

8. R. Sutton-Spence and B. Woll.The Linguistics of British Sign Language: An Introduction. Cambridge University Press, 1999. isbn: 9781107494091. url: https://books.google. de/books?id=0betAQAAQBAJ.
9. **P**. Boyes-Braem, R. Sutton-Spence, and R. te Leiden. The Hands are the Head of the Mouth: The Mouth as Articulator in Sign Languages. International studies on sign language and the communication of the deaf. Gallaudet University Press, 2001. isbn: 9783927731837. url: https://books.google.de/books?id=EyeLQgAACAAJ.
10. **10.** O. M. Sincan, J. C. S. J. Junior, S. Escalera, and H. Y. Keles. ChaLearn LAP Large Scale Signer Independent Isolated Sign Language Recognition Challenge: Design, Results and Future Research. 2021. arXiv: 2105.05066 [cs.CV].
11. R. L. McKinley and J. W. Rohrer, "A Machine Learning Approach to American Sign Language Recognition," in IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, vol. 42, no. 6, pp. 1069-1078, Nov. 2012, doi: 10.1109/TSMCC.2011.2167896.
12. M. A. Garg and S. Kumar, "ASL Recognition Using Machine Learning and Computer Vision Techniques," in IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2017, pp. 1-5, doi: 10.1109/ICECCT.2017.8117854.
13. N. Kumar and A. C. Kesavan, "American Sign Language Recognition Using Machine Learning and Image Processing Techniques," in IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, India, 2015, pp. 1-6, doi: 10.1109/ICCIC.2015.7435782.
14. S. M. Chakraborty and S. K. Mukhopadhyay, "Real-time American Sign Language Recognition System Using Machine Learning Approach," in IEEE Transactions on Consumer Electronics, vol. 61, no. 2, pp. 205-213, May 2015, doi: 10.1109/TCE.2015.