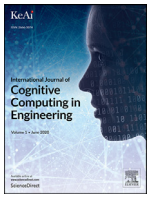
International Journal of Cognitive Computing in Engineering 4 (2023) 170–178

Contents lists available at ScienceDirect

International Journal of Cognitive Computing in Engineering

journal homepage: https://www.keaipublishing.com/en/journals/international-journal-of-cognitive-computing-in-engineering/



Continuous word level sign language recognition using an expert system based on machine learning

R Sreemathy*, MP Turuk, S Chaudhary, K Lavate, A Ushire, S Khurana

Department of Electronics and Telecommunication Engineering, Pune Institute of Computer Technology, Pune, India

ARTICLE INFO

Keywords:
Sign language recognition
Continuous gesture recognition
Static gesture recognition
YOLOv4
Mediapipe
SVM
Expert system

ABSTRACT

The study of sign language recognition systems has been extensively explored using many image processing and artificial intelligence techniques for many years, but the main challenge is to bridge the communication gap between specially-abled people and the general public. This paper proposes a python-based system that classifies 80 words from sign language. Two different models have been proposed in this work: You Only Look Once version 4 (YOLOv4) and Support Vector Machine (SVM) with media-pipe. SVM utilizes the linear, polynomial and Radial Basis Function (RBF) kernels. The system does not need any additional pre-processing and image enhancement operations. The image dataset used in this work is self-created and consists of 80 static signs with a total of 676 images. The accuracy of SVM with media-pipe is 98.62% and the accuracy of YOLOv4 obtained is 98.8% which is higher than the existing state-of-the-art methods. An expert system is also proposed which utilizes both the above models to predict the hand gesture more accurately in real-time.

1. Introduction

Today, about 5% of the people in the world suffer from a hearing disability. Sign language is the primary source of communication for these people. Sign language greatly facilitates communication in the specially-abled community. Sign language is a language that communicates and expresses emotions based on visual gestures. There exists a communication gap when the specially-abled person wants to express their opinions and thoughts to the general public. Currently, these two groups rely primarily on human interpreters, which can be costly and inconvenient.

Many sign languages exist around the world such as American Sign Language, Indian Sign Language, British Sign Language and Japanese sign Language. Each language has its unique features, such as the use of one or both hands and the use of facial expressions to convey an emotion. Indian Sign Language uses both hands for most of the signs. This makes the recognition of Indian Sign Language a challenging task as the position and orientation of both hands of a person need to be tracked. A full-fledged expert system is essential to ease day-to-day communication for specially-abled people. This type of system may play an important role in developing an automatic sign-to-text translator.

Different image processing and computer vision-based approaches (Ismail et al., 2021; Sreemathy et al., 2022) have been developed in recent years, but unfortunately, these models fail to achieve a sufficiently high accuracy to develop real-time applications. Some of the challenges with these methods are the presence of occlusion, disruption and noisy background. Various deep learning techniques (Elboushaki et al., 2020; Konstantinidis et al., 2018) are being utilized to overcome these challenges. Development in the field of deep learning and computer vision has led researchers to develop a variety of automated sign language recognition methods to interpret sign language gestures. This will help to reduce the communication gap between people with disabilities and the general public. It also allows specially-abled people to have equal opportunities and promote personal growth.

Due to the unavailability of a standardized dataset for Indian Sign Language, most researchers have created their dataset or tested their methodologies on pre-existing datasets or datasets from American Sign Language. Hence, the creation of a dataset of words from Indian Sign Language was necessary. Like other languages, sign language has a very large vocabulary of words. As more research is done in this field, the vocabulary of words that can be detected and classified increases. One of the main needs for this work is to increase the corpus of words that are available to be used in the future work to make them capable of recognizing a larger vocabulary of words with a higher accuracy. The dataset of 676 images from 80 classes has been created using words from the day-to-day vocabulary.

This dataset is then used with YOLOv4 and SVM with Mediapipe to classify hand signs from the Indian Sign Language. This work proposes a system which can accurately classify and recognize hand gestures from a large vocabulary of words from Indian Sign Language. The main motivation of this work is that it enables the hearing impaired community to

* Corresponding author.
E-mail address: rsreemathy@pict.edu (R. Sreemathy).

https://doi.org/10.1016/j.ijcce.2023.04.002
Received 18 November 2022; Received in revised form 20 April 2023; Accepted 20 April 2023
Available online 23 April 2023

easily communicate with the general public and the model can be easily deployed on edge devices to create other solutions which will greatly help the hearing impaired community. The use of such devices will also help with the education of specially enabled people and improve their cognition. The novelty provided with this work is the development of an expert system which combines two of the proposed methods to achieve a higher overall accuracy.

The contributions of this work are as follows -

1. Most earlier works have focused on static hand gesture recognition. This work focuses on both static and continuous hand gesture recognition. Continuous hand gesture recognition is implemented using the Gramformer module. The models produce an output from recognizing the individual static hand gestures which are then used in conjunction with the Gramformer module to recognize the continuous hand gesture and create fully formed sentences.
2. A database of 80 Indian Sign Language words with a total of 676 images is created. There has not been much work done on increasing the number of classes used for training the various sign language recognition models. Most of the work done is focused on the recognition of words and numbers.
3. YOLOv4 and SVM with Mediapipe is used for the recognition and classification of words from Indian Sign Language. YOLOv4 is used for detection and classification of the hand signs. On the other hand, Mediapipe is used as a feature extractor and SVM is used for classification.
4. An expert system is proposed combining the two models mentioned earlier to achieve an overall higher real time accuracy. This eliminates the small inaccuracies present in the individual models which helps in achieving a higher recognition rate.

This work is divided into four sections. Section 2 discusses the earlier methods and techniques proposed for sign language recognition and classification. Section 3 consists of the proposed method using YOLOv4, SVM with Mediapipe and the expert system combining the two approaches. Section 4 discusses the results of the proposed method and comparison with the state-of-art methods. Section 5 concludes by summarizing all the contributions and results of this work.

2. Related work

As this domain expands, there have been numerous studies conducted on hand gesture recognition, and there are numerous implementations involving both machine learning and deep learning methods aimed at recognizing human hand gestures. Hardware-based approaches are popular among researchers to accurately recognize sign language by tracking the position and orientation of the hands. One of the approaches involves the use of the Microsoft Kinect Sensor (Rioux-Maldague and Giguere, 2014) to capture intensity and depth data to classify hand signs of American Sign Language. The reported accuracy was 99% for known users and 79% for unknown users. Another hardware-based approach involves the use of a glove fitted with sensors (Kushwah et al., 2017) to measure the analog voltage based on the orientation of the hands and classify hand signs based on these values. Image processing approaches are popular because they are computationally less expensive compared to machine learning or deep learning-based approaches and can be used on devices that have lower computational capabilities. One approach involves the use of thresholding using OpenCV and the Haar-cascade classifier (Ismail et al., 2021) to detect and classify hand signs from video frames. Another method involves the use of HOG features extracted from images (Sreemathy et al., 2022) which are then used to train a back-propagation network to achieve an accuracy of 89.5%. A novel approach involving the use of radar has also been proposed (Lee et al., 2020) which recognizes hand gestures by very accurately tracking the user's gestures for classification and claims to achieve an overall accuracy of 98.8%.

Natural Language processing (NLP) plays an important role in the translation of sign language. NLP is used to convert multiple individual signs into fully formed sentences. Extensive work has been done on detecting parts of speech from a variety of sentences (Samantra et al., 2022) using neural networks, for example recurrent neural networks (RNN), long short term memory (LSTM) networks and bi-directional LSTM networks. Finding semantic similarities between a set of sentences also plays a vital role in the accurate detection of sign language. Methods for determining the semantic similarity between sentences has also been developed (Ahmad and Faisal, 2022) using multiple metrics like word vector, wordnet, word order and final sentence similarity. A RNN based attention model is also proposed (Chen et al., 2020) which utilizes the meaning and position of words in a sentence to accurately create a language model. This work focuses on accurately identifying signs and combining them to form semantically correct sentences. Due to the lesser extent of the vocabulary and lower complexity of sentences used in this work, the use of other algorithms like Gramformer is preferred.

Deep learning has greatly increased the ability to perform complex classification tasks which might not be possible with traditional machine learning algorithms. Deep learning-based approaches to the identification and classification of sign language have gained popularity in recent years, especially with the development of powerful hardware to train complex deep learning models. The use of LSTM-based networks is becoming increasingly popular for sign language recognition. The use of ConvLSTM with ResNets (Elboushaki et al., 2020) to capture skin color and conventional LSTM models to detect skeletal structure (Konstantinidis et al., 2018) for static hand gesture recognition has been proposed. A conjunction of ConvLSTM and 3D residual networks to capture local and global feature information (Peng et al., 2020) to detect dynamic hand gestures is suggested. Multi-stream networks to extract hand pose and spatial hand relation features which are then passed to LSTM networks (Rastgoo et al., 2020) and Multiple Extraction and Multiple Prediction (MEMP) networks consisting of alternating 3D CNN and convolutional LSTM layers (Zhang and Li, 2019) are popular approaches to the recognition of sign language. The different versions of YOLO have also yielded good results in recent years. YOLO v3 with DarkNet53 (Mujahid et al., 2021) to detect objects and hand gestures with an accuracy of 97.68% has been proposed. YOLOv4 has also been used for the recognition of 40 different words with 200 images per class (Ali et al., 2022) with different lighting conditions, with the two sets each achieving an average precision of 96.44% and 98.01%. YOLOv5 was used for its object detection and classification capability to detect and classify signs from the American Sign Language dataset (Dima and Ahmed, 2021; Kim et al., 2018). YOLO has also been used for detecting hand regions from an image (Bankar et al., 2022) which are then passed to a 5-layer CNN for classification. Some of the other methods involve the use of Faster RCNN with a region proposal network (RPN) to detect the signs from the images and then pass them to the VGG-16 and ResNet18 architectures (Alawwad et al., 2021) for classification, achieving an overall accuracy of 93.2% and 93.4% respectively. VGG-16 is also popular for the use of skin color detection and classification (Huang et al., 2019), achieving an accuracy of 94.7% for 9 gestures. Pretrained VGG-16 was also used for the detection of ASL gestures (Masood et al., 2018) which achieved an accuracy of 96%. 3D CNNs are also widely used, especially for the detection of skeletal pose structures (Duan et al., 2022) for dynamic action and gesture recognition. Convolutional Neural Networks (CNNs) have played a large role in the previous years in the field of computer vision, image and object classification. CNNs have been used for detecting gestures and fingerspelling from videos (Ashiquzzaman et al., 2020) using spatial pyramid pooling. A lightweight semantic segmentation network is also proposed (Benitez-Garcia et al., 2021) called FASSDNet which is based on Temporal Segment Networks (TSN) and Temporal Shift Modules (TSM) to classify 13 gestures in real-time. CNNs and stacked autoencoders have been used to classify hand gestures (Pinto et al., 2019) with the help of numerous image processing techniques with an accuracy of 94.7%.

Table 1
Summary of Related Works.

Reference	Model	No. of Classes	Accuracy
Mujahid et al. (2021)	Lightweight YOLOv3	5	97.68
Zhang and Li (2019)	MEMP network	64	97.01
Konstantinidis et al. (2018)	Body and hand skeletal features using LSTM	64	98.09
Rastgoo et al. (2020)	Hand, pose and hand relation features with LSTM	100	98.42
Pinto et al. (2019)	CNN and stacked denoising autoencoder	24	98.32
Alawwad et al. (2021)	Faster R-CNN and ResNet-18	28	93.40
Masood et al. (2018)	VGG-16	40	95.54
Ashiquzzaman et al. (2020)	Spatial Pyramid Pooling Deep CNN	29	94.00

A method is proposed for the classification of Chinese Sign Language with CNNs using extracted video frames (Yang and Zhu, 2017) which have been transformed to the HSV color space achieving an accuracy of 98.6% using the Adadelta algorithm and 99.4% using the Adagrad algorithm. Artificial Neural Networks (ANNs) are like CNNs and are used to classify sign language gestures (Rao and Kishore, 2018) from videos captured on a phone. These images are pre-processed and augmented and given to the ANN to obtain a word match score of 90%. CNNs are used in conjunction with stacked autoencoder networks (Oyedotun and Khashman, 2017) to classify images from the Thomas Moeslund dataset to obtain an accuracy of 92.83%. Standard CNN architectures such as AlexNet and GoogleNet are used to classify Indian Sign Language (Pardeshi et al., 2019) which obtained an accuracy of 98.61% and 91.9% respectively. CNNs have also been used to classify gestures of American Sign Language (Tolentino et al., 2019) using a dataset of 1200 images to obtain an accuracy of 90.04%, 93.44% and 97.52% for numbers, alphabets and words respectively. Hidden Markov Models (HMMs) are used to classify Chinese Sign Language gestures from videos using data collected from 3D trackers and cyber gloves (Ma et al., 2000). This consisted of a database of 220 words and 80 sentences and achieved an accuracy of 94.7%. HMMs are also used to classify 22 gestures (Vogler and Metaxas, 2004) with an accuracy of 87.88% using a database of static and continuous gestures.

A summary of some of the related works is shown in Table 1. Some of the methods proposed above use various algorithms to develop the Sign Language Recognition System for static words. But these models are trained over a miniature dataset of over 20–30 words while some require external hardware to collect the data as well as a few image processing operations which are time-consuming. Day-to-day life communication is mostly in the form of sentences. But very few systems are developed which recognize static gesture words and convert them into meaningful sentences. The proposed system uses two methods, YOLOv4 and SVM with Media-pipe as a feature extractor. This reduces additional image pre-processing operations and does not require any external hardware like cyber gloves to collect data.

3. Proposed methodology

The proposed method has been divided into two main subsections -

1. Hand Gesture Recognition System using SVM with Media-pipe
2. Hand Gesture Recognition System using YOLO

3.1. Hand gesture recognition system using SVM with media-pipe

3.1.1. Dataset

The dataset consists of 80 classes. A total of 13,000 images are obtained by augmenting the available images by rotating them between −3° and +3° and the brightness of each image is varied between −25% to 25% to avoid the constraint of light. Some of the words created are shown in Fig. 1. Some of the words in the created dataset are cold, good, happy, book, etc. These words are used as they are some of the most commonly used words in day-to-day conversation. The dataset is created with the help of a professional signer.

Fig. 1. Created Dataset.

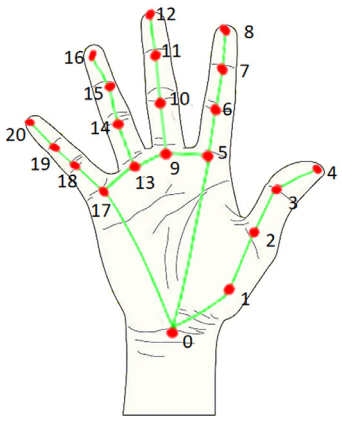
Fig. 2. RGB (left) and BGR (right) image comparison.

3.1.2. Image pre-processing

The only pre-processing step that is carried out is the conversion of the image from BGR to RGB. The only difference between BGR and RGB is the order of the bits representing the red, green and blue components of the image. Fig. 2 demonstrates the difference between BGR and RGB images. The default color format in OpenCV is referred to as BGR, and thus it must be converted to RGB before utilizing it as an input to Media-pipe.

3.1.3. Hand detection and tracking

Mediapipe has been used to concentrate solely on the creation of an algorithm for recognizing signs. The frame is initially extracted from the video and sent to the Mediapipe system. Within the Mediapipe pipeline, there are two main models. The first one is a palm detection model



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Fig. 3. Media-pipe hand landmarks.

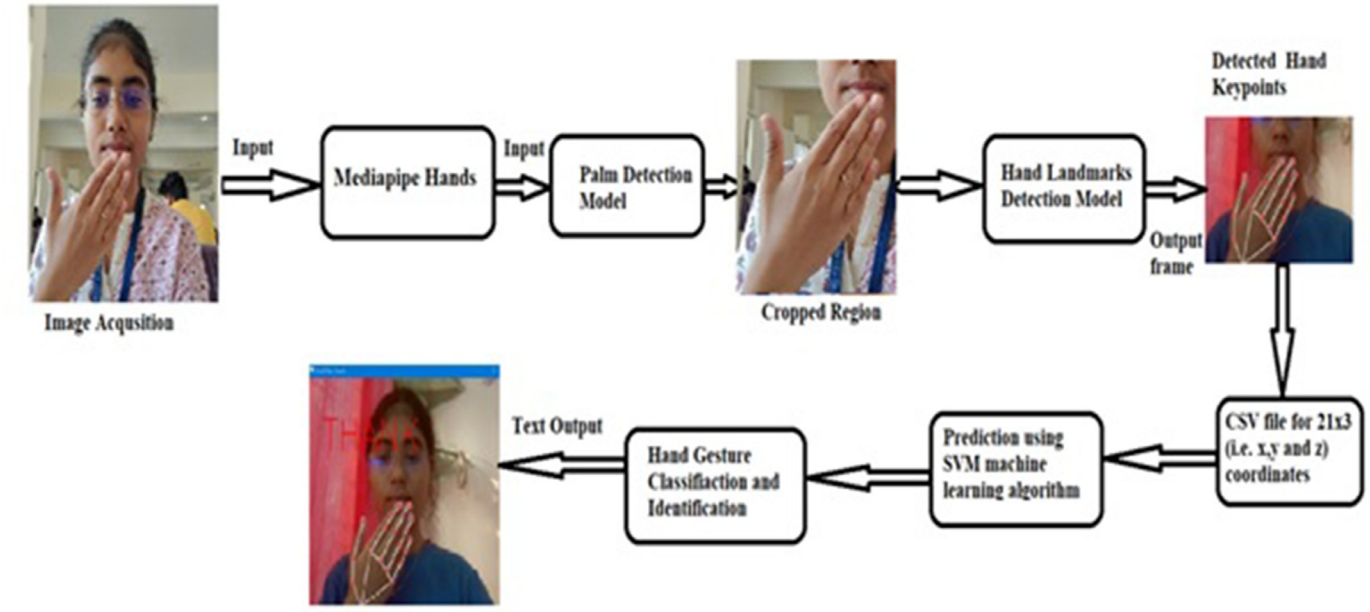


Fig. 4. Block diagram of the proposed system using SVM with Media-pipe.

that recognizes the hand within the video frame and crops the hand portion, this cropped region is fed to the second model that recognizes the position of the hands' point of interest. With the recognized hand region, this model accurately localizes 21 3D hand coordinates (x, y, and z-axes) and saves them in a CSV file. The Hand Landmark model detects 21 landmark sites, as shown in Fig. 3. This results in a total of 63 datapoints which are used to train the SVM model.

The flow of the algorithm is shown in Fig. 4. Initially, the images from the dataset are given to the Mediapipe algorithm. Mediapipe uses a palm detection model to detect the palm of the person in the image and extract the key points from the palm. These key points are then stored in a CSV file which is used to train the SVM algorithm. Once the SVM model is trained, a test image is then given to Mediapipe to extract the key points. These are then given to the trained SVM model to accurately test it and predict the hand gesture. The SVM model is then used in the implementation of the proposed expert system.

3.1.4. Data cleaning and normalization

Each image in the dataset is fed into Mediapipe, which compiles all the data into a single record. This record is then run via the pandas library to ensure there are no null sections. Sometimes, due to a hazy image, Mediapipe is unable to distinguish the hand, resulting in an incorrect section in the dataset. As a result, cleaning those rows is critical. Rows containing these incorrect sections are eliminated. The data file is then separated into two sets: training and validation. 80% of the data is used to train our model using various optimization and loss functions, while the remaining 20% is used to validate the model.

3.1.5. Model training

SVM works well in situations where the number of samples is greater than the number of dimensions. When there is a clear margin of distinction between classes, SVM performs well. As a result, SVM is used to classify multiple sign classes. The optimization problem that SVMs are used to solve is expressed as follows:

$$min(w, b, d) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} d_i \tag{1}$$

$$y_i(w^T \phi(x_i) + b) \geq 1 - d_i \tag{2}$$

where,

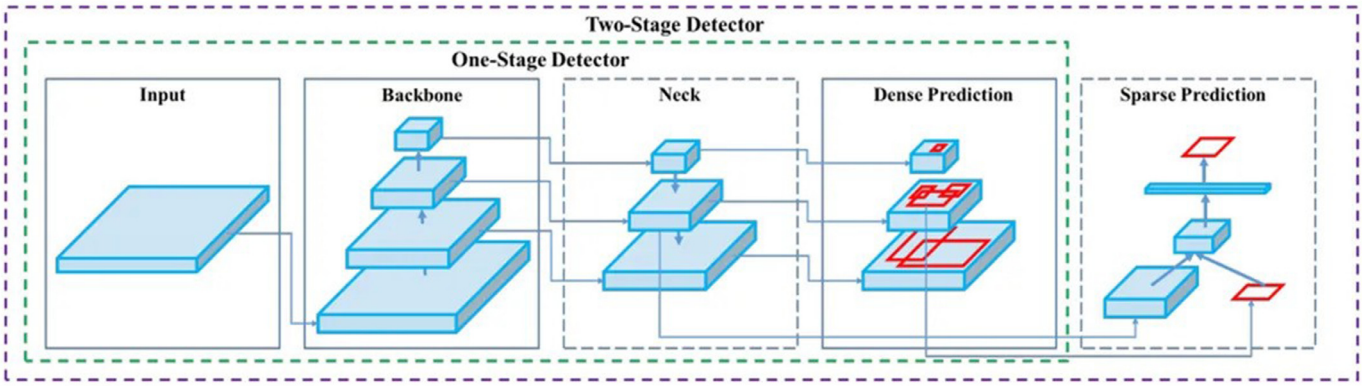


Fig. 5. YOLOv4 Architecture.

d_i = distances to correct margin with $d_i \geq 0$ and i=1,2,...,n
C = regularization parameter
$w^T w$ = normal vector
$\Phi(x_i)$ = transformed input space vector
y_i = ith target value

3.2. Hand gesture recognition system using YOLOv4

3.2.1. YOLOv4

YOLO is a different approach for real-time object detection. YOLO uses frame object detection as a regression problem to spatially separate bounding boxes and associated class probabilities. This causes the single neural network to predict the bounding boxes and class probabilities directly from the full image in one evaluation. YOLO is extremely useful because it is also trained on full images and directly optimizes the detection performance. The three main components of YOLO architecture are the backbone, neck and head as shown in Fig. 5

- Backbone: Backbone is a deep learning architecture that fundamentally acts as a feature extractor. The CSPDarknet53 model is used for this step.
- Neck: Neck is a subset of a bag of specials that collects feature maps from different stages of the backbone. It is a feature aggregator. The Path Aggregation Network (PAN) is used as the neck.
- Head: Head is also called an object detector. It finds the region where the object might be present, but it does not talk about which object is present in that region.

3.2.2. Gramformer

Gramformer is an open-source framework for detecting, highlighting, and correcting grammar mistakes in natural language text. It takes the sentence as the input and returns the grammatically correct sentence. The framework consists of three models: error correction, error detection, and error highlighter model. An error correction model is used to convert the continuous sentences to grammatically correct sentences.

Fig. 6 shows the block diagram of the proposed algorithm. The steps in the algorithm are as follows:

1. The dataset was created for the various classes.
2. The images for each sign were annotated using Roboflow to indicate the region of interest (ROI).
3. The existing images in the dataset were augmented by rotating them and changing their brightness to increase the total number of images in the dataset.
4. These images were then fed to the YOLOv4 algorithm to train the classifier.
5. The algorithm was then tested with static gestures and these gestures were then fed into Gramformer to convert them into a grammatically correct sentence.

Table 2
Parameters and Values for SVM.

Parameter	Value
Learning Rate	0.001
Number of Classes	80
Regularization Parameter(C)	50
Gamma	0.1
Kernel	Linear, Poly, RBF

3.2.3. Implementation

The images from the dataset are labeled using Roboflow. Roboflow is used to store and organize image data and annotate images. The data is labeled in text format. The annotated data is used for training the YOLOv4 model. The training process begins with the collection of the dataset, the next step is labeling and augmenting the images using Roboflow. After the images are augmented, they are fed to the DarkNet-53 model which is trained according to the defined configuration. The trained YOLOv4 model is used for the implementation of the proposed expert system.

3.3. Expert system

Once the SVM and YOLOv4-based classifiers were trained separately, an expert system is proposed using the combined accuracies of both individual systems. This expert system utilizes the best of both models by comparing the accuracy and confidence of the detection of signs and deciding in real-time while predicting the gesture. The system passes the input through both models and produces a result based on which model has a higher accuracy while training and the confidence of detection in real-time.

The flow of the algorithm and a pseudo code for the implementation of the expert system are shown in Fig. 7.

Algorithm for the Expert System -

1. An image is taken as input from the camera.
2. The input image is passed to both SVM and YOLOv4 models.
3. Both models produce an output prediction.
4. The confidence of the predicted hand sign is compared.
5. The output of the model with the higher confidence for the particular sign is displayed as the final output of the system. If the models have the same confidence for an input, the predicted output is displayed.

4. Results

Python 3.8 was used to train YOLO and SVM models on a workstation with an NVIDIA RTX 3090 GPU with 24GB of VRAM. The parameters for SVM with Mediapipe are presented in Table 2. The hyperparameters C and gamma depict the error control and curvature of the SVM decision

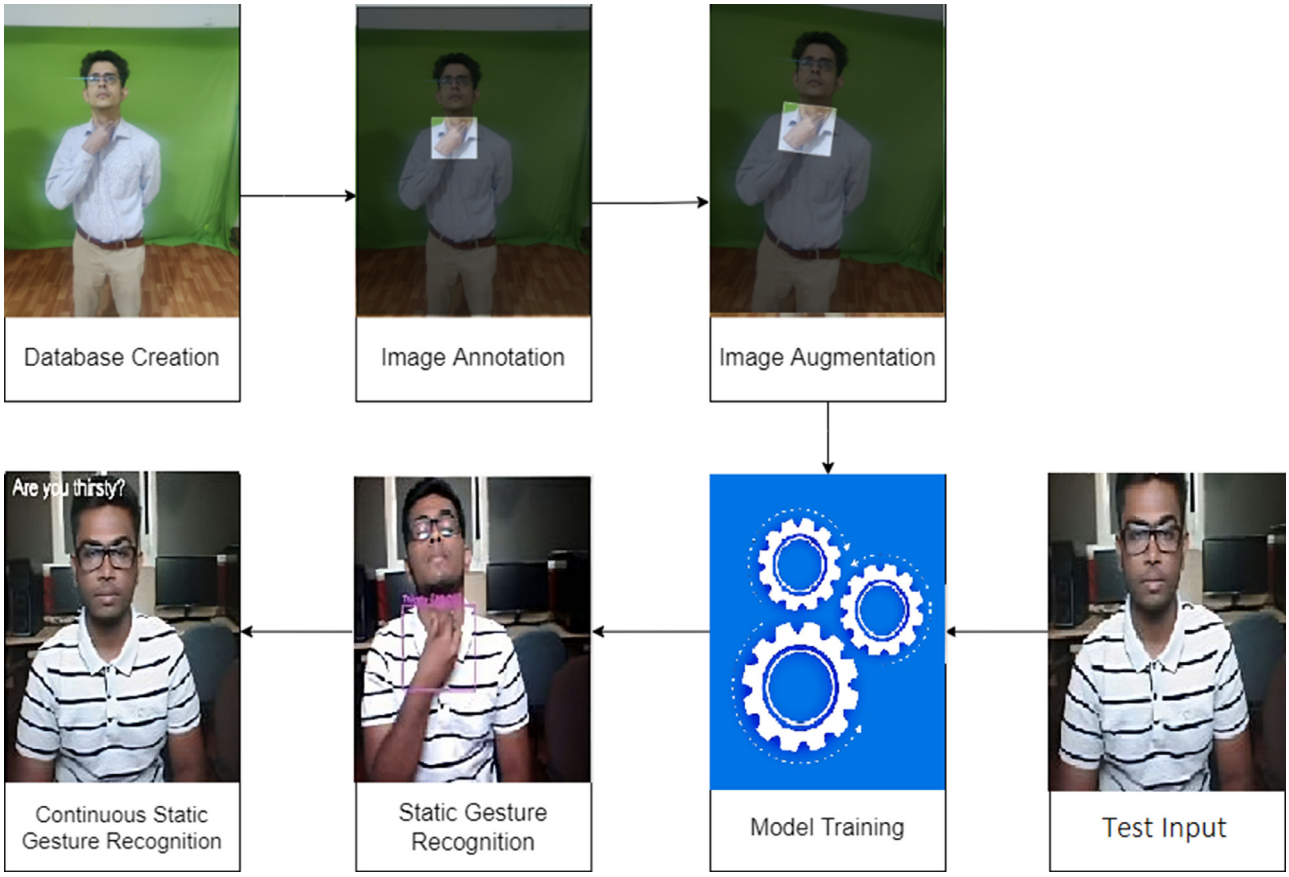


Fig. 6. Block diagram of the proposed system using YOLOv4.

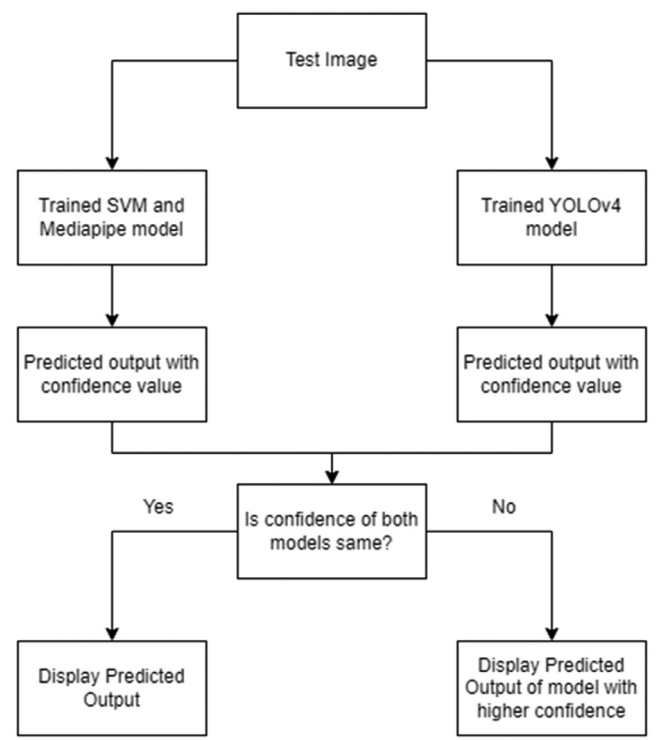


Fig. 7. Flow chart for the proposed Expert System.

Table 3
Parameters and Values for YOLOv4.

Parameter	Value
Learning Rate	0.001
Number of Classes	80
Optimizer	Adam
Activation	Linear, Leaky ReLU
Filter Size	32,64,128,256,512,1024
Mask	0–8
Decay	0.0005
Batch Size	64

boundary, respectively. The value of C is high to get a higher overall accuracy and gamma is low to get a clearer semantic boundary between the classes. The other important parameters for YOLOv4 are presented in Table 3.

The output of the SVM model is shown in Fig. 8 for different hand gestures such as you, hello, drink and thank you. The accuracy of the SVM system is around 98.55%. The accuracy of each class of the SVM model is presented in Table 4.

The output of the YOLOv4 model is shown in Fig. 9. The hand gestures shown are hungry, phone and worry. The accuracy of the system is 98.8%. The Mean Average Precision (mAP) of each class of the YOLO model is presented in Table 5.

The output of the continuous hand gesture recognition system is presented in Figs. 10, 11 and 12. The static hand gestures are individually identified and are fed into the Gramformer module to obtain grammatically correct sentences. Fig. 10 shows the output of the gesture, “I am angry”. Fig. 11 shows the output of the gesture, “Are you thirsty?”. Fig. 12 shows the output of the gesture, “I have fever”.

Fig. 8. SVM Output.

Fig. 9. YOLOv4 Output.

Table 4
Accuracy of the SVM model for all classes.

Class	Accuracy	Class	Accuracy	Class	Accuracy	Class	Accuracy
Afternoon	1	Fever	0.94	Name	0.95	So Much	0.97
A Lot	1	Fine	1	Not	0.98	Sorry	1
Angry	1	Food	1	Open	1	Speak	1
Bag	1	Football	0.95	Paints	1	Sports	0.97
Bench	1	Free	1	Pen	1	Stand	1
Book	1	Friend	1	Pencil	1	Teacher	1
Break	1	From	1	Phone	0.95	Team	1
Brush	1	Glass	0.88	Play	1	Thank You	1
Chat	1	Good	0.98	Prayer	1	Thirsty	0.93
Colors	0.90	Happy	1	Project	0.95	Tie	1
Comb	0.95	Hello	1	Promise	0.94	Tiffin	1
Computer	1	Help	1	Quiet	1	Time	1
Congratulations	1	Hungry	1	Read	1	Understand	1
Cry	1	Hurt	1	Repeat	1	Up	1
Dance	1	I	1	Ring	1	Water	0.95
Do	1	Keep	1	School	1	Welcome	1
Doubt	1	Light	0.97	Shoes	1	What	1
Down	0.90	Love	1	Sit	0.91	You	1
Duster	1	Morning	1	Sleep	1		
Feeling	1	Music	0.90	Slow	1		

Table 5
Mean Average Precision (mAP) of the YOLO model for all classes.

Class	mAP	Class	mAP	Class	mAP	Class	mAP
Afternoon	1	Fever	1	Name	1	So Much	1
A Lot	1	Fine	1	Not	1	Sorry	1
Angry	1	Food	1	Open	1	Speak	1
Bag	1	Football	1	Paints	1	Sports	0.666
Bench	1	Free	1	Pen	1	Stand	1
Book	0	Friend	0.733	Pencil	1	Teacher	1
Break	1	From	1	Phone	0.833	Team	1
Brush	1	Glass	1	Play	1	Thank You	1
Chat	1	Good	1	Prayer	1	Thirsty	1
Colors	1	Happy	1	Project	1	Tie	1
Comb	1	Hello	1	Promise	1	Tiffin	1
Computer	1	Help	1	Quiet	1	Time	1
Congratulations	1	Hungry	1	Read	1	Understand	1
Cry	1	Hurt	1	Repeat	1	Up	1
Dance	1	I	1	Ring	1	Water	1
Do	1	Keep	1	School	1	Welcome	1
Doubt	1	Light	1	Shoes	1	What	1
Down	1	Love	1	Sit	1	You	1
Duster	1	Morning	1	Sleep	1		
Feeling	1	Music	1	Slow	1		

Fig. 10. I am Angry.

Fig. 11. Are you thirsty?

Fig. 12. I have fever.

Table 6
Comparison of the Proposed Algorithms.

Models	Precision	Recall	F-1 Score	Accuracy
SVM(linear)	96.59	96.59	96.59	96.77
SVM(poly)	98.46	98.46	98.46	98.62
SVM(rbf)	94.10	94.10	94.10	94.77
YOLOv4	97.20	99.17	98.17	98.8

Table 6 presents the accuracy of different proposed models. The performance of different models is evaluated based on the values of precision, F-1 score, accuracy and recall.

Table 7 compares the accuracy of the state-of-the-art work of Mujahid et al. (2021), Zhang and Li (2019), Konstantinidis et al. (2018), Rastgoo et al. (2020), Pinto et al. (2019) with our proposed model which produces better results and accuracy.

Table 7
Comparison of the state-of-the-art work with our proposed model.

Reference	Model	Accuracy(%)
Mujahid et al. (2021)	Lightweight YOLOv3	97.68
Zhang and Li (2019)	MEMP network	97.01
Konstantinidis et al. (2018)	Body and Hand skeletal features using LSTM	98.09
Rastgoo et al. (2020)	Hand, pose and hand relation features with LSTM	98.42
Pinto et al. (2019)	CNN and stacked denoising autoencoder	98.32
Proposed Method	SVM with mediapipe	98.62
Proposed Method	YOLOv4	98.80

5. Conclusion

In this work, two models for sign language recognition: YOLOv4 and SVM with media-pipe are proposed. The trained models were extended to be able to recognize continuous gestures with high accuracy with Gramformer. Both these models have their advantages and disadvantages. The SVM model with Mediapipe is trained quickly but is vulnerable to noise as gesture recognition depends on hand key points extracted using Mediapipe. YOLOv4 requires high computational power but the problem of transition gesture recognition is eliminated. The overall accuracy obtained using the Mediapipe and SVM was 98.62% and using YOLO was 98.8%. Furthermore, the proposed system was compared with different models and it was observed that the proposed model obtained better accuracy compared to the other state-of-the-art models. An expert system is also proposed which uses the best of both models to produce real-time results with even higher accuracy than both models individually. This system can be implemented in different public places to help specially-abled people to communicate with the general public with ease.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Rajiv Gandhi Science & Technology Commission, Government of Maharashtra, India. We would also like to acknowledge the support provided by the NVIDIA GPU Grant Program. We would like to thank Pune Institute of Computer Technology for their constant support and dedication to this project. Our special thanks to the signers.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ijcce.2023.04.002.

References

Ahmad, F., & Faisal, M. (2022). A novel hybrid methodology for computing semantic similarity between sentences through various word senses. *International Journal of Cognitive Computing in Engineering, 3*, 58–77.

Alawwad, R. A., Bchir, O., & Ismail, M. M. B. (2021). Arabic sign language recognition using faster R-CNN. *International Journal of Advanced Computer Science and Applications, 12*(3).

Ali, A.-S., ÇEVİK, M., & ALQARAGHULI, A. (2022). American sign language recognition using YOLOv4 method. *International Journal of Multidisciplinary Studies and Innovative Technologies, 6*(1), 61–65.

Ashiquzzaman, A., Lee, H., Kim, K., Kim, H.-Y., Park, J., & Kim, J. (2020). Compact spatial pyramid pooling deep convolutional neural network based hand gestures decoder. *Applied Sciences, 10*(21), 7898.

Bankar, S., Kadam, T., Korhale, V., & Kulkarni, M. A. (2022). Real time sign language recognition using deep learning. *International Research Journal of Engineering and Technology, 9*(4), 955–959.

Benitez-Garcia, G., Prudente-Tixteco, L., Castro-Madrid, L. C., Toscano-Medina, R., Olivares-Mercado, J., Sanchez-Perez, G., & Villalba, L. J. G. (2021). Improving real-time hand gesture recognition with semantic segmentation. *Sensors, 21*(2), 356.

Chen, M.-Y., Chiang, H.-S., Sangaiah, A. K., & Hsieh, T.-C. (2020). Recurrent neural network with attention mechanism for language model. *Neural Computing and Applications, 32*, 7915–7923.

Dima, T. F., & Ahmed, M. E. (2021). Using YOLOv5 algorithm to detect and recognize American sign language. In *2021 International conference on information technology (ICIT)* (pp. 603–607). IEEE.

Duan, H., Zhao, Y., Chen, K., Lin, D., & Dai, B. (2022). Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2969–2978).

Elboushaki, A., Hannane, R., Afdel, K., & Koutti, L. (2020). MultiD-CNN: a multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences. *Expert Systems with Applications, 139*, 112829.

Huang, H., Chong, Y., Nie, C., & Pan, S. (2019). Hand gesture recognition with skin detection and deep learning method. In *Journal of physics: Conference series: vol. 1213* (p. 022001). IOP Publishing.

Ismail, A. P., Abd Aziz, F. A., Kasim, N. M., & Daud, K. (2021). Hand gesture recognition on python and opencv. In *IOP conference series: Materials science and engineering: vol. 1045* (p. 012043). IOP Publishing.

Kim, S., Ji, Y., & Lee, K.-B. (2018). An effective sign language learning with object detection based ROI segmentation. In *2018 second IEEE international conference on robotic computing (IRC)* (pp. 330–333). IEEE.

Konstantinidis, D., Dimitropoulos, K., & Daras, P. (2018). Sign language recognition based on hand and body skeletal data. In *2018-3DTV-conference: The true vision-capture, transmission and display of 3D video (3DTV-Con)* (pp. 1–4). IEEE.

Kushwah, M. S., Sharma, M., Jain, K., & Chopra, A. (2017). Sign language interpretation using pseudo glove. In *Proceeding of international conference on intelligent communication, control and devices: ICICCD 2016* (pp. 9–18). Springer.

Lee, H. R., Park, J., & Suh, Y.-J. (2020). Improving classification accuracy of hand gesture recognition based on 60 GHz FMCW radar with deep learning domain adaptation. *Electronics, 9*(12), 2140.

Ma, J., Gao, W., Wu, J., & Wang, C. (2000). A continuous chinese sign language recognition system. In *Proceedings fourth IEEE international conference on automatic face and gesture recognition (Cat. No. pr00580)* (pp. 428–433). IEEE.

Masood, S., Thuwal, H. C., & Srivastava, A. (2018). American sign language character recognition using convolution neural network. In *Smart computing and informatics: Proceedings of the first international conference on SCI 2016: vol. 2* (pp. 403–412). Springer.

Mujahid, A., Awan, M. J., Yasin, A., Mohammed, M. A., Damaševičius, R., Maskeliūnas, R., & Abdulkareem, K. H. (2021). Real-time hand gesture recognition based on deep learning YOLOv3 model. *Applied Sciences, 11*(9), 4164.

Oyedotun, O. K., & Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. *Neural Computing and Applications, 28*(12), 3941–3951.

Pardeshi, K., Sreemathy, R., & Velapure, A. (2019). Recognition of indian sign language alphabets for hearing and speech impaired people using deep learning. In *Proceedings of international conference on communication and information processing (ICCIP)*.

Peng, Y., Tao, H., Li, W., Yuan, H., & Li, T. (2020). Dynamic gesture recognition based on feature fusion network and variant ConvLSTM. *IET Image Processing, 14*(11), 2480–2486.

Pinto, R. F., Borges, C. D., Almeida, A. M., & Paula, I. C. (2019). Static hand gesture recognition based on convolutional neural networks. *Journal of Electrical and Computer Engineering, 2019*, 1–12.

Rao, G. A., & Kishore, P. (2018). Selfie video based continuous Indian sign language recognition system. *Ain Shams Engineering Journal, 9*(4), 1929–1939.

Rastgoo, R., Kiani, K., & Escalera, S. (2020). Video-based isolated hand sign language recognition using a deep cascaded model. *Multimedia Tools and Applications, 79*, 22965–22987.

Rioux-Maldague, L., & Giguere, P. (2014). Sign language fingerspelling classification from depth and color images using a deep belief network. In *2014 Canadian conference on computer and robot vision* (pp. 92–97). IEEE.

Samantra, A., Sa, P. K., Nguyen, T. N., Sangaiah, A. K., & Bakshi, S. (2022). On the usage of neural pos taggers for shakespearean literature in social systems. *IEEE Transactions on Computational Social Systems*, 1–11.

Sreemathy, R., Turuk, M., Kulkarni, I., & Khurana, S. (2022). Sign language recognition using artificial intelligence. *Education and Information Technologies*, 1–20.

Tolentino, L. K. S., Juan, R. S., Thio-ac, A. C., Pamahoy, M. A. B., Forteza, J. R. R., & Garcia, X. J. O. (2019). Static sign language recognition using deep learning. *International Journal of Machine Learning and Computing, 9*(6), 821–827.

Vogler, C., & Metaxas, D. (2004). Handshapes and movements: Multiple-channel American sign language recognition. In *Gesture-based communication in human-computer interaction: 5th International gesture workshop, GW 2003, Genova, Italy, April 15–17, 2003, selected revised papers 5* (pp. 247–258). Springer.

Yang, S., & Zhu, Q. (2017). Video-based chinese sign language recognition using convolutional neural network. In *2017 IEEE 9th international conference on communication software and networks (ICCSN)* (pp. 929–934). IEEE.

Zhang, X., & Li, X. (2019). Dynamic gesture recognition based on MEMP network. *Future Internet, 11*(4), 91.