IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 27, 2021, accepted January 8, 2022, date of publication January 13, 2022, date of current version January 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3142918

Helping Hearing-Impaired in Emergency Situations: A Deep Learning-Based Approach

QAZI MOHAMMAD AREEB[1], MARYAM[1], MOHAMMAD NADEEM[1], ROOBAEA ALROOBAEA[2], AND FAISAL ANWER[1]

[1]Department of Computer Science, Aligarh Muslim University, Aligarh 202002, India
[2]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia

Corresponding author: Mohammad Nadeem (mnadeem.cs@amu.ac.in)

This work was supported by the Taif University Researchers Supporting Project, Taif University, Taif, Saudi Arabia, under Grant TURSP-2020/36.

ABSTRACT Hearing-impaired people use sign language to express their thoughts and emotions and reinforce information delivered in daily conversations. Though they make a significant percentage of any population, the majority of people can't interact with them due to limited or no knowledge of sign languages. Sign language recognition aims to detect the significant motions of the human body, especially hands, analyze them and understand them. Such systems may become life-saving when hearing-challenged people are in desperate situations like heart attacks, accidents, etc. In the present study, deep learning-based hand gesture recognition models are developed to accurately predict the emergency signs of Indian Sign Language (ISL). The dataset used contains the videos for eight different emergency situations. Several frames were extracted from the videos and are fed to three different models. Two models are designed for classification, while one is an object detection model, applied after annotating the frames. The first model consists of a three-dimensional convolutional neural network (3D CNN), while the second comprises of a pre-trained VGG-16 and a recurrent neural network with a long short-term memory (RNN-LSTM) scheme. The last model is based on YOLO (You Only Look Once) v5, an advanced object detection algorithm. The prediction accuracies of the classification models were 82% and 98%, respectively. YOLO based model outperformed the rest and achieved an impressive mean average precision of 99.6%.

INDEX TERMS Human–computer interaction, sign language, hand gesture recognition, classification, object detection.

I. INTRODUCTION

Human-Computer interaction is an interdisciplinary research area focusing on designing computational technologies to make the interaction between humans and computers possible. Hand gesture recognition is its sub-field in which computer vision and artificial intelligence have aided to provide nonverbal communication between humans and computers by identifying significant movements of the human hands [1]. Though there are a variety of applications of hand gesture recognition, accurate recognition remains a challenging task [1].

Sign language recognition is a typical application of hand gesture recognition [2]. It is often considered that only deaf people rely on sign languages for conveying their thoughts. However, particular medical problems such as down syndrome, autism, cerebral palsy, trauma, and brain diseases or speech difficulties may require a nonverbal mode of communication [6]. 6,909 spoken languages and 138 sign languages have been identified, but there is no universal sign language [3]. Each has its own syntactical and grammatical structures to provide definitive means of communication, primarily for deaf communities worldwide. Sign languages emphasize on the movement of the hands, arms, head, and body in a conceptually predetermined manner to significantly construct a gesture language. Indian Sign Language (ISL) is the name given to the sign language used in India. According to the 2011 census, 2.7 million people in India cannot speak, and 1.8 million are deaf [4]. They face difficulty communicating with others because most normal people are unfamiliar with sign language. However, communication between them becomes inevitable in emergencies. Sign language

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Hosseinzadeh.

FIGURE 1. Expressing 'Thank you' in a) Indian, and b) American sign languages.

interpreters are required to convert sign language to spoken language and vice versa, but their supply is limited and expensive. As a result, automatic sign language recognition systems are needed to translate signs into corresponding text or voice without the assistance of interpreters [7]. Through human-computer interaction, systems can be built to aid in the development of the deaf and other communities who rely on sign languages.

Like spoken languages, sign languages also develop naturally as a result of groups of people interacting with one another; regions and cultures also play an essential role in their development. People who do not know the same language cannot communicate because most sign languages are not mutually intelligible. Common sign languages are American Sign Language (ASL), British Sign Language (BSL), and French Sign Language (FSL). Each Sign Language has its own syntax and ways to use it. For instance, ASL uses only one hand while languages like ISL and BSL use both hands in the gestures. Because of differences in syntax and expression, results obtained in recognizing one language cannot be used for another. To clearly explain, take an example of a sign representing 'Thank You.' In ASL, it is expressed by starting with the fingers of the dominant hand near the lips and moving the hand forward and a bit down in the direction of the person we are thanking, shown in Figure 1(b). While in ISL, it is represented in the same way but with both hands, as shown in Figure 1(a).

Hand gesture recognition is not only used to facilitate communication among the deaf but also in various other applications. It has inspired new technologies such as gesture-based signaling systems [5], [6], virtual reality [7], [8], and smart television [9], [10] in the computer-vision and design recognition industries. Numerous applications have made significant progress, including the recognition of sign language [11], [12], the control of robots [13], [14], and the playing of virtual musical instruments [15], [16]. The progress in recognizing gestures by hands has also attracted much attention from the industrial sector in manufacturing interactions between human robots [17], [18] and self-driving cars [19]. The recognition of sign language [20]–[22], recognition of sports' specific sign language [23], the accuracy of human activity [24], stance and posture [25], [26], and physical monitoring of exercises [27] are all new and innovative applications of hand gesture recognition.

To address the gesture recognition problem, there are various handmade feature techniques [28], [29] and deep learning-based approaches [30], [31]. Deep learning is a form of machine learning that has gained prominence in the field of sign recognition in recent years. Deep learning has found applications in a variety of areas, including cancer prediction (colon, blood, or lung cancer), tumor detection, medical imaging, Alzheimer and Parkinson disease prediction and modeling, skin lesions detection, optical coherence tomography image processing, abnormalities related to body parts

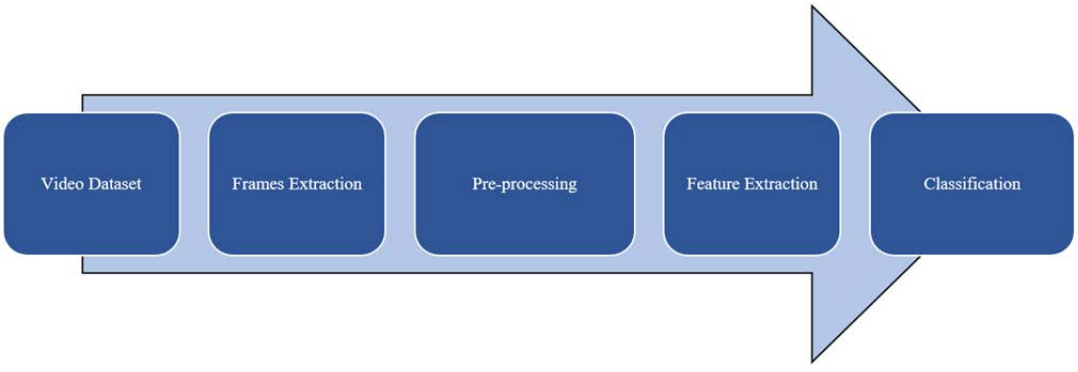


FIGURE 2. General process of sign language recognition system.

(breast, heart, etc.), and so on [71]. Kononenko [72] gives a machine learning-based review of the evolution of intelligent data analysis in medicine, emphasizing the importance of using machine learning in medical diagnostics. Various deep learning procedures are discussed by Zhang *et al.* [73] in order to develop computer-aided medical diagnosis tools. Additional works are being done to improve its utilization in the medical area, such as [74]. Deep learning's progress has allowed academics to explore answers to contemporary concerns, including Covid-19 [75]. As a result, to address the gesture recognition problem, there are various hand-made feature techniques [28], [29] and deep learning-based approaches [30], [31].

CNNs (Convolutional Neural Networks) are a type of deep neural network used to process visual data. Researchers have widely employed them for image classification and detection challenges in recent years. CNNs are also being used for video classification due to their effectiveness in classifying and detecting images and their contents [43]. Deep learning models have been shown to perform recognition and classification tasks fast and correctly, but their implementation in real-time application settings is limited [30]. A general implementation of sign language recognition involves gathering data (images or videos), pre-processing, feature extraction, and classification (Figure 2). For static gestures, images are required, while a sequence of images or videos is needed to extract the spatio-temporal features for a dynamic gesture. Pre-processing entails selecting several frames from the videos, applying different filters if required, and resizing the frames based on the model's input. Following preparation, the input is sent to a combination of convolutional and pooling layers for feature extraction. To provide a label for each frame, the extracted feature output is passed to the fully connected layers and subsequently to the SoftMax layer. To extract temporal characteristics and categorize a sequence of frames, often RNN with long short-time memory (LSTM) is employed.

Emergency communication includes an alert, warning, help, self-protective measures, and other matters involving immediate response and recovery. Sign language-based emergency situations such as feeling pain, calling for help, or a doctor are not easily recognizable by ordinary people [60]. In this study, we developed vision-based gesture identification techniques employing CNNs on a video dataset to recognize ISL based hand movements in emergency scenarios. Three models have been proposed for the same. The first two models classify the frames and extract the spatio-temporal features. Out of these two, one model utilizes 3-D Convolutional Neural Networks (3-D CNN) to retrieve spatiotemporal features directly, while the other model combines pre-trained VGG-16 and a LSTM. Since LSTM is best suited for learning long-term temporal information, the short-term spatiotemporal features are learnt using CNN before using LSTM. The third model is a You Only Look Once (YOLO) v5 algorithm based object detection model that detects the hand in the frame and returns the label. The video dataset is a publicly available dataset containing videos of eight different emergency signs, including accident, call, doctor, help, hot, lose, pain, and thief. The object detection algorithm performs the best of the three models with 99.6% mean average precision. The results are compared with previous works and outperformed them significantly, showing the effectiveness of proposed models. A thorough analysis of results is also presented.

The major contributions of the present work include:

- Implementation of state-of-the-art classification and object detection methods that can be used when hearing-impaired people are in desperate situations and want to convey their thoughts to other people at the earliest.
- A thorough literature review of related studies.
- Based on the first publicly available dataset of the hand gestures of the emergency ISL words.
- Analysis of videos of eight emergency signs.
- Comparison of the performance of proposed methods with each other on various metrics and with the previous studies.
- Establishing the superiority of the VGG + LSTM model over other classification models.
- To unveil an impressive performance of the object-detection model to identify dynamic gestures.

The remainder of the paper is organized in the following manner. The second part briefly examines similar gesture detection algorithms before moving on to different deep learning network approaches. The third portion of this article discusses the design concept and structure of the models. The experiment's outcomes and effects are presented in the fourth part. The final part contains the conclusions.

II. RELATED WORKS

Due to the constantly increasing need for Hand Gesture Recognition, a growing number of academics are concentrating their efforts on dynamic HGR based on video data. Researchers have implemented several tasks involving Classification and Detection aiming to recognize a gesture.

Gestures in Sign languages can be broadly classified as static and dynamic gestures. Static hand gestures can be classified using images, whereas dynamic hand gesture recognition necessitates utilizing a sequence of images or videos. It learns the temporal and spatial features of a gesture. The accomplishment of object detection and classification tasks through deep learning algorithms like CNNs [32]–[36] and others have led to a growing trend towards using them in computer vision applications such as Sign Language Recognition, Games, and Virtual Reality, Human-Computer Interaction, and so on [67]. CNNs are also being used to achieve cutting-edge performance on images and videos [34], [35]. There have been various approaches to classifying hand gestures using CNN. The Convolutional Neural Network (CNN) has also been successfully applied on hand gesture recognition tasks [36], [37]. Rao *et al.* [68] utilized a neural network

classifier for selfie continuous sign language identification. To achieve dynamic sign language recognition, this sort of system primarily employed deep learning technology, such as the techniques of extracting the discriminative characteristics of hand movements with Convolutional Neural Networks (CNNs) in [36].

The inputs in static hand gestures are not time-related or required in chronological order. However, the previous state must be preserved for dynamic motions through a chain-like structure of repeating modules that aid in learning long-term dependencies in sequential data. For hand gestures, video datasets are used to display the dynamics of the gestures easily. Commonly, 2D CNN is used for the recognition of static images. While for dynamic gestures, spatiotemporal features need to be extracted. Despite the exemplary performance of 2D CNNs on static gesture tasks, they are bound to model temporal characteristics and motion sequence. To solve this issue, authors have applied Long Short-Term Memory (LSTM) at the end of 2D CNN for spatio-temporal features recognition [38]–[42]. Do *et al.* [38] proposed a multi-level feature LSTM on a Dynamic Hand Gesture Recognition (DHG) dataset of 14 and 28 classes and achieved96.07 and 94.40% accuracy. Cui *et al.* [12] developed a video-based recurrent convolutional neural network for continuous sign language recognition. Elboushaki *et al.* [39] used two models, Residual Networks (ResNets), for learning the Spatio-temporal information from colored images, and Convolutional Long Short-Term Memory Networks (ConvLSTM) to capture their temporal interdependencies. And 2D-ResNets was then used to extract deep features from the gesture. Liao *et al.* [69] proposed a multimodal dynamic sign language recognition method based on deep 3-dimensional residual ConvNet and bi-directional LSTM networks (BLSTM-3D residual network or B3D ResNet) for the recognition of complex hand gestures. The technique consisted of three main parts. First, the hand object was localized in the video frames, utilizing R-CNN to gather hand position information. The second part was the video sequence features extraction module, which performed the task of long-term spatio-temporal features extraction with inputted segmented video frames. The third part was the dynamic sign language recognition module, which analyzed the long-term temporal dynamics and predicted the hand gesture label. Through which, the video sequence label was predicted. Based on the top label prediction scores, this label would be treated as the video sequence label and outputted as the recognition result. The results on test datasets show that the proposed method obtained recognition accuracy of 89.8% on the DEVISIGN_D dataset and 86.9% on SLR_Dataset.

John *et al.* [40] extracted frames that represent the gestures efficiently from the video dataset before feeding them to long-term recurrent convolution networks (LRCN). This way of extracting frames improved the classification accuracy of the model. Lai and Yanushkevich [41] used both depth and skeleton data for hand gesture recognition by combining the convolutional neural networks (CNN) and the recurrent neural networks (RNN) on the dynamic hand gesture dataset. They attained an overall accuracy of 85.46%. Obaid *et al.* [70] proposed a model comprised of two stages to solve the hand gesture recognition (HGR) problem in Video Sequences. The first stage is pre-processing, while the second is to classify, label the frames and recognize the hand gestures through deep learning. For the later stage, they proposed two models- A convolutional neural network (CNN) and a recurrent neural network with a large-short-term memory were used in the first model (RNN-LSTM). Model I used two types of neural networks in its network architecture: a CNN for spatial feature extraction and an RNN for temporal feature extraction. The second network design consists of a single CNN supplied with grayscale or depth data. The CNNs were fed training and testing data after they had been trained. Individual frames from each gesture were transformed into a sequence and utilized as a dataset for RNN training and testing.

The RNN with long short-time memory (LSTM) was then used for temporal feature extraction and classification of the sequence of frames. The VIVA dataset was used to evaluate the model. The results show that the proposed method obtained Validation Accuracy of 82% with color information, 89% with depth information for Model II, and 93% validation accuracy when color and depth information was combined for Model I. Molchanov *et al.* [42] suggested a combination of 3-D CNN and RNN for gesture recognition.

In addition, 3D CNNs have been utilized to capture discriminative characteristics in both the spatial and temporal dimensions. It takes a series of video frames as inputs. Saqib *et al.* [43] proposed a 3D CNN model to classify static and dynamic gestures of Pakistani sign language and attained an accuracy of 90.79%. Camgoz *et al.* [44] proposed a novel 3D CNN architecture for broad-scale, independent gesture recognition. The architecture uses eight 3D convolutional layers, five 3D max-pooling layers for feature extraction, two fully connected layers, and a SoftMax for classification.

For the hand detection task, [45] proposed a real-time hand gesture recognition model based on YOLO (You Only Look Once) v5 and DarkNet-53 convolutional neural networks. The model was able to successfully detect gestures in low-resolution images and complex environments. They implemented both static and dynamic gestures recorded on a video and attained an accuracy of 97.68%. Mambou *et al.* [46] suggested a sexual assault alert system from various scenes at night. The model was implemented on a combination of YOLO and CNN architecture. Generally, techniques like 3D CNN, a combination of CNN and RNN, have been widely used by researchers. Thus, this study too proposes a comparative study in classifying static as well as dynamic hand gestures through both these methods. Not surprisingly, YOLO has produced state-of-the-art results in various detection tasks. Therefore, we implemented YOLOv5 on our dataset before annotating the images.

Khari *et al.* [47] suggested a static gesture recognition method by using a pre-trained VGG19 model on the ASL Dataset. The model achieved an accuracy of 94.8%.

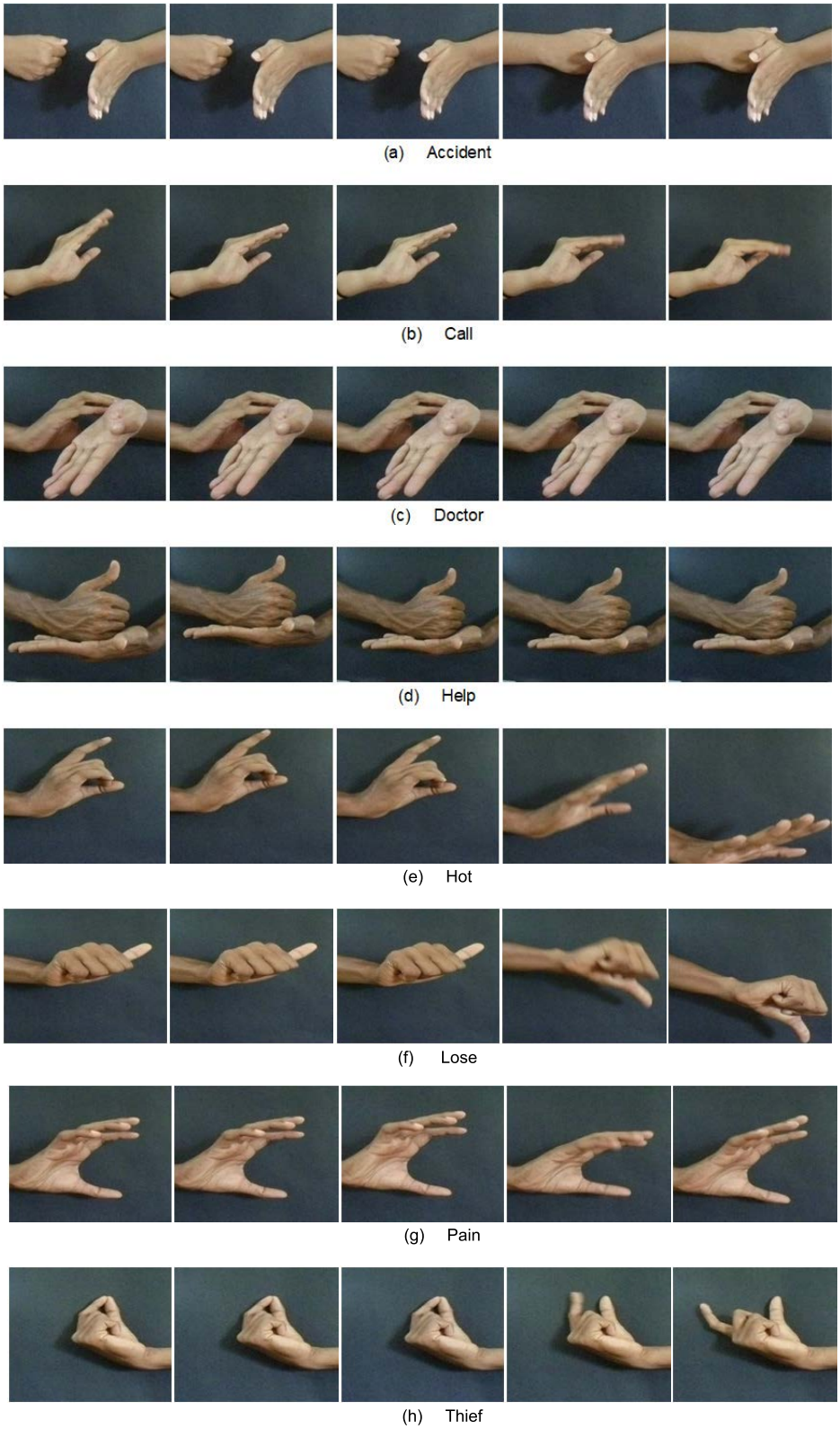


FIGURE 3. Set of frames for different emergency signs of the video dataset.

Furthermore, Paul *et al.* [48] proposed two CNNs to categorize 24 static signs from ASL. The work is based on the ASL Finger Spelling dataset and attained an accuracy of 86.52% and 85.88% on RGB images. Masood *et al.* [49] used Inception-v3 pre-trained CNN, with a combination of LSTM, to propose three models of various LSTM units for the Argentinean video dataset (LSA). They attained the best accuracy of 95.2%. Using pre-trained models like VGG-16 or

inception before LSTM has achieved excellent results in the past. In model II proposed in this work, we also took the same approach of using VGG-16 before LSTM on the emergency dataset. In addition, the method proposed by Adithya and Rajesh [60], who employed two models on the same emergency dataset, also suggested pre-trained GoogleNet [65] with LSTM. The other approach by Adithya and Rajesh [60] was using a Multi-class SVM.

ASL is based on a single-hand sign language, whereas ISL uses both hands, which makes it difficult to recognize as compared to ASL. In addition, most of the signs are dynamic, for which classification and detection tasks require image sequence input. Furthermore, hands involved in gesturing may involve complicated motion in ISL. The non-availability of the benchmark dataset has also hindered the developments in automatic sign recognition. Due to these issues, comparatively less research work has been carried out on the ISL recognition system [50]. In the present work, we tried to fill this research gap with the help of advanced deep learning methods. The emergency signs studied are mentioned in Figure 3.

III. METHODOLOGY

This section discusses the dataset used and the methods adopted for sign language recognition.

A. DATASET

Sign Language recognition has not been well-researched for ISL due to the non-availability of a standard publicly available dataset. For other languages, especially for ASL, there are many standard datasets. The three such word-level ASL datasets are Purdue RVL-SLLL ASL Database [51], Boston ASLLVD [52], and RWTH-BOSTON-50 [53]. LSA64 [54] is an Argentine word-level dataset, PSL Kinect 30 [55] is a Polish word-level dataset, DEVISIGN [56] is a Chinese, GSL [57] is Greek, DGS Kinect [58] is German, and LSE-sign [59] is Spanish Sign Language dataset. In this study, a video-based ISL dataset is used that contains 412 videos [60]. Researchers working on vision-based sign language recognition and hand gesture recognition will get benefitted from this dataset. The dataset's primary goal is to advance in the field of sign recognition since it has several applications in society, such as providing a platform for the deaf to communicate essential messages to authorities. Furthermore, the dataset may be used as a basic benchmark database for a collection of emergency ISL hand gestures. The dataset included eight hand gestures representing ISL words such as 'accident,' 'call,' 'doctor,' 'help,' 'hot,' 'lose,' 'pain,' and 'thief'; often used to transmit information or request help in emergency scenarios (Figure 3). Out of a total of 412 videos, each sign is represented in 50 different videos on average. The complete list is provided in Table-1. The video was shot on 26 adult individuals, 12 males and 14 females aged 22 and 26 years.

TABLE 1. Dataset description and number of extracted frames per video per class.

Class Label	Sign	Number of Videos	Number of Frames Extracted
0	Accident	52	260
1	Call	52	260
2	Doctor	52	260
3	Help	52	260
4	Hot	52	260
5	Lose	50	250
6	Pain	52	260
7	Thief	50	250
	Total	412	2060

B. PRE-PROCESSING

The present study utilizes a short subsequence rather than the whole video to recognize the sign words. To efficiently teach the model the dynamics of the movements, we first implemented our strategy by extracting 20 frames from each video. To shorten the training period, we began reducing the numbers of extracted frames (1-10) and discovered that 5 frames from each video sequence at equal intervals were sufficient to reflect the dynamics of the gestures without compromising the prediction accuracy. Furthermore, previous researches demonstrate that human action recognition may be accomplished by just a few frames (1–7 frames) [76], [70].

Images were graded from 0 to 7 to represent eight different classes. Table-1 shows the class labels or scores, corresponding hand gestures, and class size. For the object detection purpose, the dataset was labeled before applying the model. The YOLO format was used to label data in a text file format and store information such as class ID and the class to which it belongs. The extracted frames have been resized from (500 by 600) to (150 by 150) pixels. Data normalization was also used to verify that the data distribution in each input pixel is uniform, resulting in faster model training convergence. The overall dataset (100%) is divided into three subsets: training (60%), validation (20%), and testing (20%).

C. PROPOSED METHODS

We utilized two models for the classification task, Model I and Model II. Model I consist of 3D CNN, while Model II uses a combination of pre-trained CNN and LSTM. Model III, an object detection model, is based on the YOLO (You Only Look Once) v5 algorithm for detecting hand gestures.

1) CLASSIFICATION

Artificial Neural Network such as convolutional neural network (CNN) is widely employed to analyze pixel input [41]. CNN is applied for image recognition and processing, object detection, feature learning, and sequence prediction with RNN. CNN uses a variation of the multilayer perceptron model and carries at least one convolutional layer that can be wholly connected or pooled. The convolutional layers

generate feature maps to record a portion of an image before being split down into boxes and sent out for non-linear processing. The input data is passed through the network, and each layer extracts features further to pass them to the subsequent layers [43].

A convolution layer is a tool that allows image feature extraction based on various filters trained by the network itself. The convolution function is defined in Equation (1).

$$x_{h,v}^{l} = f\left(\sum_{h}\sum_{v} x_{h+m,v+n}^{l-1}.k_{h,v}^{l} + b\right) \tag{1}$$

where k represents the feature map of a layer l, is the convolution kernel, f is the activation function of the hidden layer, is the map of features of the previous layer, and b is the biases. Moreover, an Activation function is added to let the network learn complex patterns. The most commonly used activation function is ReLu (Rectified Linear Unit) and is mathematically defined in Equation (2).

$$f(x) = \max(o, x) \tag{2}$$

where x is the input of a neuron. The SoftMax activation function is used in the final fully-connected layer of a network. It is mathematically defined in Equation (3).

$$y' = \frac{e^{z}}{\sum_{r=1}^{nc} e^{z_r}} \tag{3}$$

where nc denotes the number of classes and z is the output value(probability) for the current patch after passing through the convolution neural network. The loss function is a metric for evaluating the model during training and should ideally decrease over iterations. We used categorical cross-entropy loss in our work, defined in Equation (4).

$$L(y, y') = \frac{-1}{N}\left(\sum_{i=1}^{N} y.\log y'\right) \tag{4}$$

where y′ is the label predicted by our classifier, and y is the ground truth label.

Pooling layers are used to reduce the complexity of the model. It reduces the learning parameters and the number of computations performed on the network. Max pooling is used to summarise the strongest activations throughout a region by taking the largest input value within a filter and dropping the remaining values. Furthermore, normalization and dropout layers are used to avoid over-fitting and to make the model learn independently.

Model I make use of 3D CNN for feature extraction and dense layers at the end for classification as shown in Figure 4. The architecture of model I comprises of two Convolutional blocks, each consisting of a 3D convolutional layer, Max-Pooling 3D layer, Batch Normalization and Dropout Layer. The size of all convolution kernels is $3 \times 3 \times 3$ with stride $1 \times 1 \times 1$. The number of convolution kernels is 32 and 64, respectively. The size of all pooling kernels is $1 \times 2 \times 2$. Furthermore, there is a 50% and 40% dropout respectively at the end of each convolutional block. ReLu was used as the activation function in each block. A Flatten layer is placed after the convolutional block to convert the extracted data into one dimension followed by a fully connected layer of 256 neurons. Finally, an output layer with SoftMax activation function of 8 neurons corresponds to the total number of classes in the hand gesture dataset.

All inputs are regarded as independent in a conventional neural network. This technique is restricted by problems where the network has to recall events from past information, such as predicting one word in a phrase or predicting a video framework. Recent neural networks (RNNs) are built particularly for identifying patterns in data streams. The result depends on prior calculations in these networks. In addition, these networks contain a ''memory'' to collect information on the calculations made thus far. The most popular RNN network is LSTM [61]. As illustrated in Figure 1, a general approach to sign language recognition on video datasets starts from extracting frames from videos followed by pre-processing of the frames. Features are then extracted from the processed frames before being classified.

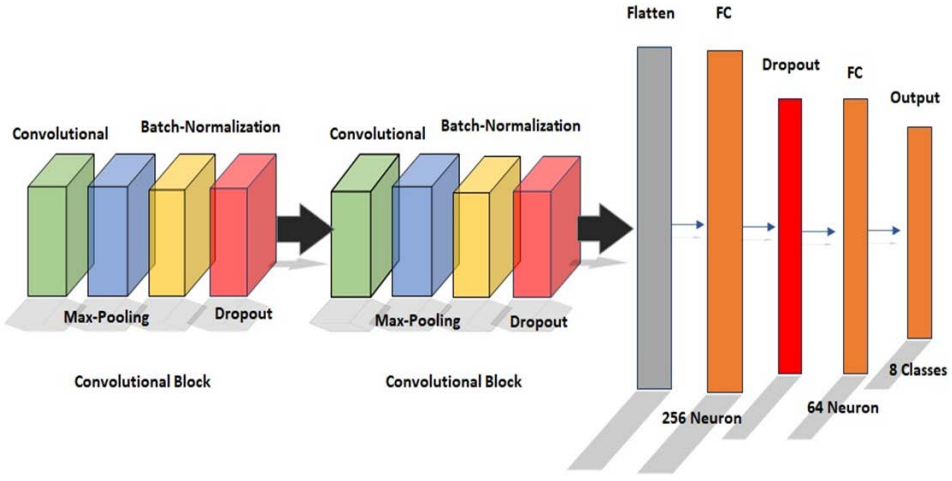


FIGURE 4. The detailed overall architecture of Model I.

TABLE 2. Layers, output shapes and parameters of Model I.

Layers	Output Shape	Parameters
Conv3D_1	(None, 3, 148, 148, 32)	2624
MaxPooling3D_1	(None, 3, 74, 74, 32)	0
Batch-Normalization_1	(None, 3, 74, 74, 32)	128
Dropout_1	(None, 3, 74, 74, 32)	0
Conv3D_2	(None, 1, 72, 72, 64)	55360
MaxPooling3D_2	(None, 1, 36, 36, 64)	0
Batch-Normalization_2	(None, 1, 36, 36, 64)	256
Dropout_2	(None, 1, 36, 36, 64)	0
Flatten	(None, 82944)	0
Dense_1	(None, 256)	21233920
Dropout_3	(None, 256)	0
Dense_2	(None, 64)	16448
Dense_3	(None, 8)	520
Total Parameters		21,309,256

The second model, i.e., Model II is based on LSTM on top of a pre-trained CNN (VGG-16) to classify the video sequence as shown in Figure 6. It uses transfer learning and LSTM for learning spatial and temporal features respectively. Each frame is passed onto the CNN (VGG-16) to extract spatial features. The architecture of VGG-16 is shown in Figure 5. The outputs are then sent into a LSTM to find temporal characteristics in the image stream. Finally, the extracted features are sent to a fully connected layer that predicts the classification for the whole input sequence.

The VGG 16 network was originally trained on the ImageNet dataset, which had over 14 million high-resolution images with 1000 distinct classifications. In order to utilize it for our work, we deleted the classification layers that were trained on the ImageNet dataset and modified the pre-trained model's input shape to meet our needs. As a consequence, our model includes almost 14 Million learned parameters and concludes with a maxpooling layer from the network's Feature Learning section. After that, we used a 256-unit LSTM layer to learn the temporal characteristics, followed by a fully connected layer of 1024 neurons to classify the features. Finally, a Softmax layer was added to give the output, as shown in Table 3. The training is carried out using the Adam optimizer, with a learning rate = 0.001 and batch size of 32 for both models.

2) DETECTION

Model III uses YOLO v5 to detect the hand movements and classifies the gesture with a confidence score. YOLO v5 is an advanced real-time object detection algorithm with top performances on two official object detection datasets: Pascal VOC [32] and Microsoft COCO [33]. It is a state-of-the-art detector that predicts the coordinates of objects in the image and the class label's confidence score (probability). The architecture of the network is shown in Figure 7. YOLO v5 comprises of three segments; CSPDarknet network, PANet network, and YOLO Layer, often referred to as backbone, neck, and head of the architecture, respectively. Initially, the data is sent to the CSPDarknet for feature extraction, followed by PANet for feature fusion before outputting the class, score, location, and object's size through the Yolo layer.

The input data needs to be annotated in the specific YOLO format that labels the data into text file format having information as object class, object coordinates, height, and width. For our study, hands in the images are annotated into the required YOLO format.

We did not employ the YOLO model to compensate for the classification model's shortcomings; instead, we developed it as a detection model that can provide better performance in real-time deployment [62]. With multiple objects in the frame, the classification model fails to classify while detection model can efficiently detect and classify the target object due to the bounding box. Hence, the classification and detection models are not meant to be compared as they demonstrate different concepts.

IV. RESULTS AND DISCUSSIONS

In this section, the results of all three models are presented. We have used Google Colab to train the proposed models,

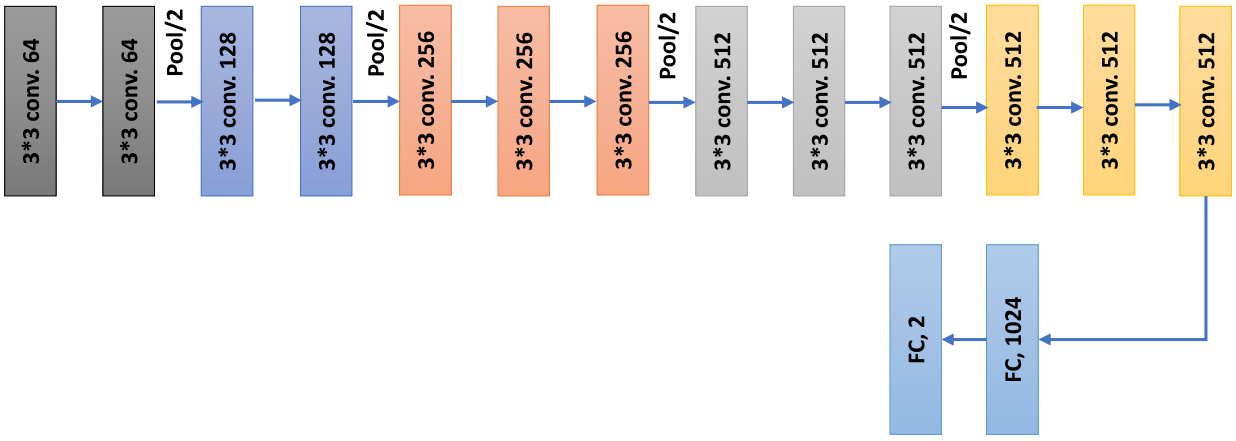


FIGURE 5. Layer-wise architecture of VGG-16 model.

a free environment that runs on the cloud and provides GPU based computation facility. The Keras Library [63] and TensorFlow [64] backend were utilized to implement the models.

A. EVALUATION METRICS

To evaluate the models, commonly used measures such as accuracy, precision, recall, and confusion metrics are considered. To assess model performance, object detection models such as YOLO also rely on a specific metric, i.e., mean average precision (mAP). Average precision is defined in Equation (5):

$$AP = \int_0^1 p\,(r)\,dr \tag{5}$$

In addition, Mean Average Accuracy (mAP) is the sum of average precision of all classes by the number of classes and is defined in Equation (6),

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{6}$$

where N is the number of objects for classification. Two medium accuracy ranges, map@0.5 and map@0.5:0.95, are used. The mAP@0.5 shows the average confidence level accuracy of 50%. The mean value of average precision in the range of 50% to 95% is the mAP@0.5:0.95.

B. PERFORMANCE OF CLASSIFICATION MODELS

The classification models were trained and tested on 2060 frames from 412 videos. We trained both the models on a common training set and tested them on the same testing set. The models classify various emergency signs from ISL as 'accident,' 'call,' 'doctor,' 'help,' 'hot,' 'lose,' 'pain' and 'thief.' The results for each sign from both models are presented in Table 4.

TABLE 3. Layers, output shapes and parameters of Model II.

Layers	Output Shape	Parameters
Input Layer	(None, 5, 150, 150, 3)	0
VGG-16	(None, 5, 512)	14714688
LSTM	(None, 256)	787456
Dense_1	(None, 1024)	263168
Dense_2	(None, 8)	8200
Total Parameters		15,773,512

The accuracy and loss calculated during training for Model I are shown in Figure 8. The architect and parameters of the models were tuned during training to achieve better results. 3D CNN-based model was able to achieve a maximum of 82% accuracy on the test set. The difference between the training and testing accuracy infers that the model was still over-fitting. Furthermore, the fluctuating testing accuracies and testing loss suggest that the model could not generalize, which could be solved if the available dataset was large. The Confusion matrix of Model I is shown in Figure 9. It was also noticed that 3D CNN could not learn hands that had excessive shifting. Since the signs 'Doctor' and 'Thief' had the least movement among all, the model achieved relatively

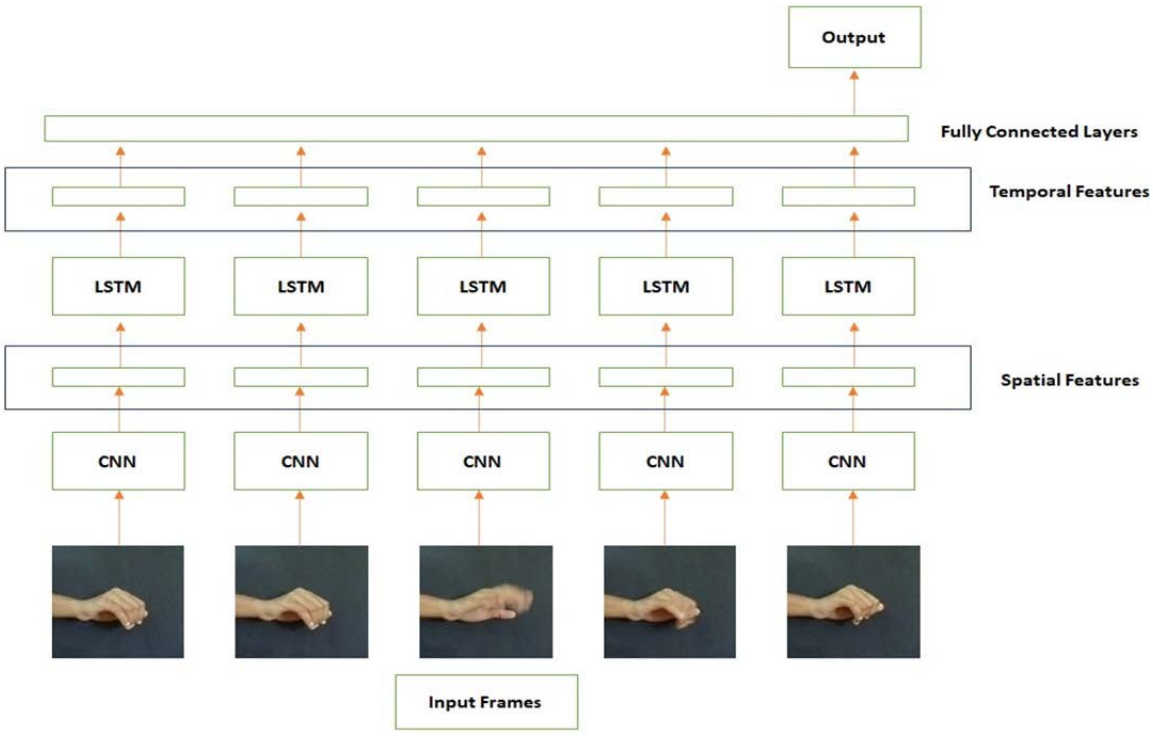


FIGURE 6. Combining VGG-16 and LSTM networks to design the framework of Model II.

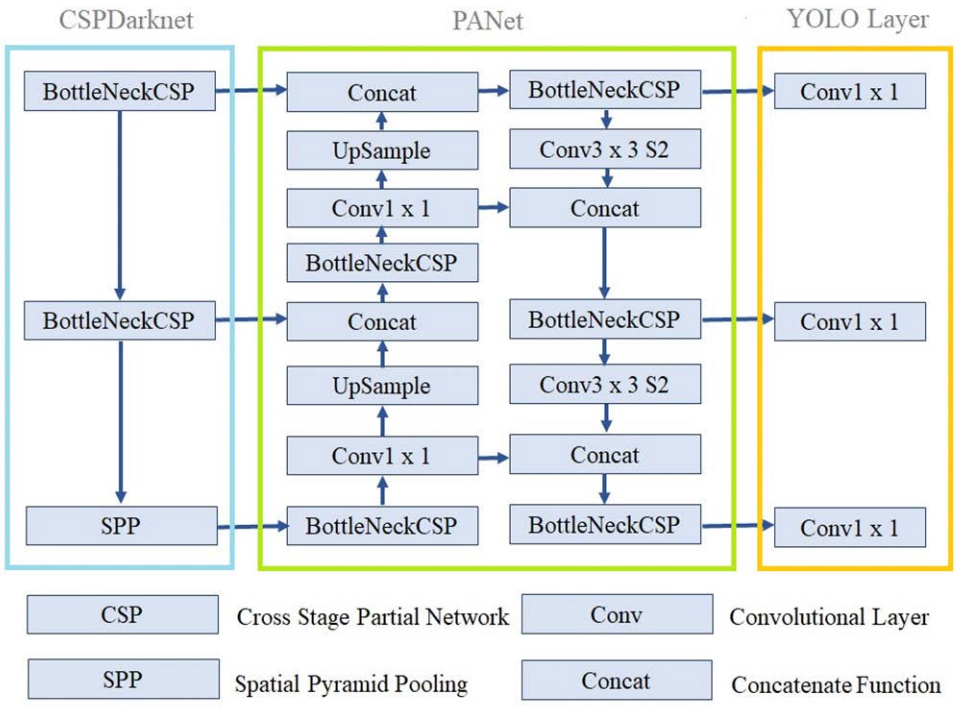


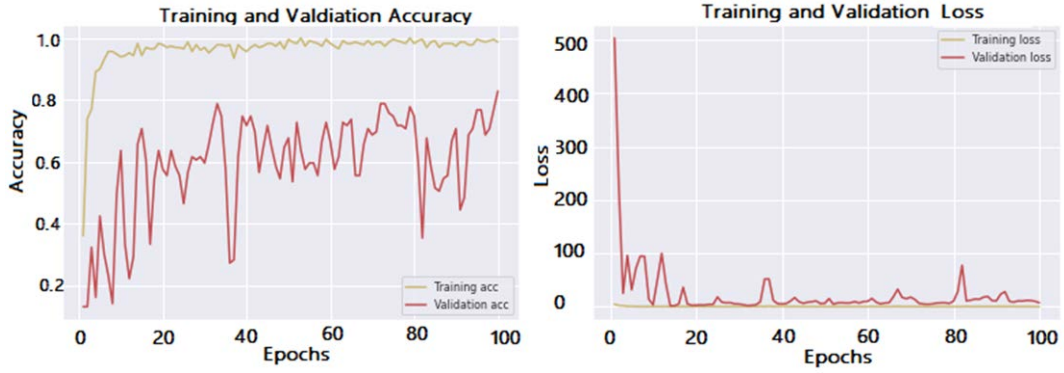
FIGURE 7. Detailed architecture of YOLO v5 algorithm [62].

better results on them while the other signs such as 'Help,' 'Lose' and 'Call' weren't classified with accuracy. The reason could be that the model was not able to learn temporal features in the sequence. Furthermore, it was observed that double-handed signs were classified more accurately than single-handed ones. Some of the misclassified instances of Model-I are shown in Figure 10.

On the other hand, Model II, i.e., VGG-16, combined with LSTM achieved an accuracy of 98%. The accuracy and loss evaluated during training for Model II are shown in Figure 11. The accuracy gradually improved throughout training. Since the feature extraction block was pre-trained, the model was able to learn in very quickly. After around ten epochs, the performance in terms of accuracy and loss became steady. It is evident from the confusion matrix (Figure-12) that the model successfully learned both the spatial and temporal features of each sign and differentiated each one of them. Due to the similarity of movement of the hands, Model II showed some error in differentiating the signs 'Pain' and 'Call' as shown in Figure 13. The same can be inferred from the corresponding confusion matrix too.

Because of the benefit of feature extraction from a pre-trained CNN before LSTM, the results achieved using Mode lII outperformed those obtained using Model I, as seen in the graphs above.

Adithya and Rajesh's technique [60] was used to compare the outcomes of both models, who employed two models on the same dataset. One approach used a Multi-class SVM, and the other was a pre-trained GoogleNet [65] with LSTM. The comparison showed that our Model II outperformed their



(a) Accuracy (b) Loss

FIGURE 8. The accuracy and loss trends of Model I.

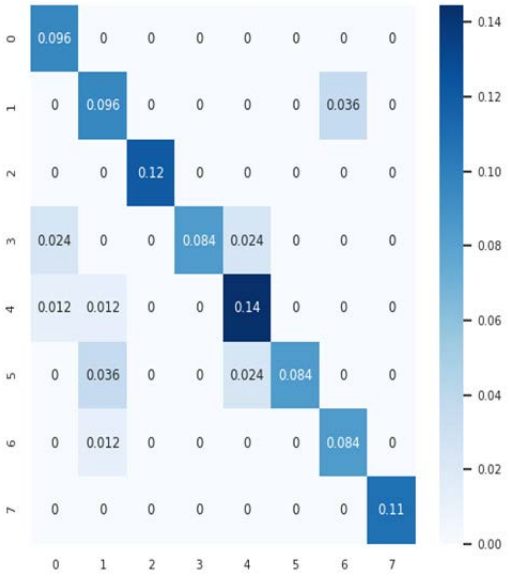


FIGURE 9. Confusion matrix of Model I.

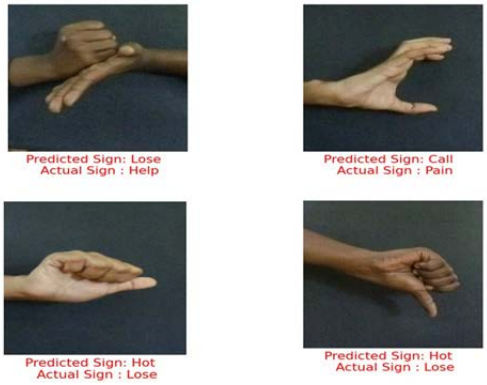


FIGURE 10. Some misclassified frames of Model I.

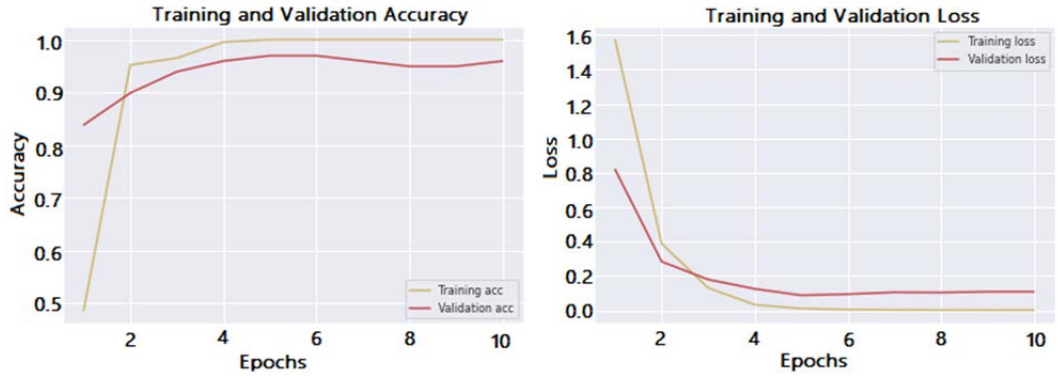
methods. However, Model-I did not produce better results than theirs. Table 5 gives a comparative overview of the same.

C. PERFORMANCE OF DETECTION MODELS

Table 6 shows the results obtained from YOLO Model. Since the dataset contains a constant background, the bounding box was intentionally left a bit large to cover the maximum area of the gesture without fearing the model learning anything from the background. The model returns the class ID and the confidence score of the gesture.

Model III is trained on 500 epochs with a batch size of 16. The learning rate is 0.01, and the decay is 0.0005. It took around 4 hours to train the model. Precision and recall are considered to evaluate the object detection model as they can provide valuable insight into the model performance at various confidence values. Generally, as the confidence threshold increases, the precision also increases while the recall decreases. Therefore, the F1 score is especially helpful in deciding the optimum confidence that equally poises the precision and recall values for the model. The confidence score and various metrics were plotted to find the appropriate confidence score. Figure 14 illustrates that before a confidence value of 0.8, the model predicts everything accurately. In addition, the confusion matrix on the test set is given in Figure 15.

A high precision value indicates a highly confident model in classifying a given sample as positive, and a high recall value indicates a model's ability to correctly classify positive samples as positive. Furthermore, as the recall increases, the precision decreases since as the number of positive samples increases (high recall), the accuracy of classifying each sample correctly decreases (low precision). So, in order to find the optimum values of precision and recall, a precision-recall curve is plotted to easily determine the value at which precision as well as recall, both are high. The average precision (AP) is a metric that represents the mean of all precision values by summarizing the whole precision-recall curve with the help of a single value. In other words, the AP is calculated as the weighted sum of precisions at each threshold where the weight is the increase in recall. The PR graph is shown in Figure 16. The region below the curve is utilized to determine the AP value of the object. As seen in the figure, our model manages to get high precision and recall, resulting in greater AP. The precision, recall, and mAP for each gesture class are presented in Table 7. The high values of all the performance



(a) Accuracy (b) Loss

FIGURE 11. The accuracy and loss trends of Model II.

TABLE 4. Performance comparison of classification Models I and II.

Sign	Class Label	Model I			Model II		
		Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)
Accident	0	73	100	84	100	100	100
Call	1	62	73	67	100	82	90
Doctor	2	100	100	100	100	100	100
Help	3	100	64	78	100	100	100
Hot	4	75	86	80	100	100	100
Lose	5	100	58	74	100	100	100
Pain	6	70	88	78	80	100	89
Thief	7	100	100	100	100	100	100
Testing Accuracy			82			98	

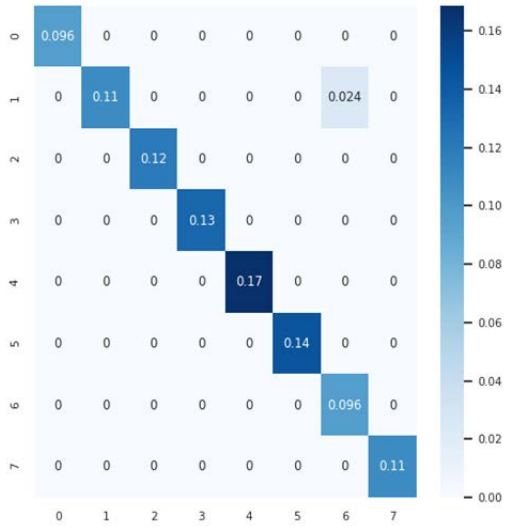


FIGURE 12. Confusion matrix of Model II.

FIGURE 13. Some misclassified frames of Model II.

measures indicate the effectiveness of Model III in identifying the hand gestures of various signs.

D. PERFORMANCE COMPARISON AND DISCUSSION

In the present study, we developed models for classifying and detecting hand gestures for ISL. The dataset utilized includes hand gestures, double and single-handed motions, and dynamic and static movements, which adds realism to the suggested models. We compared the performance of

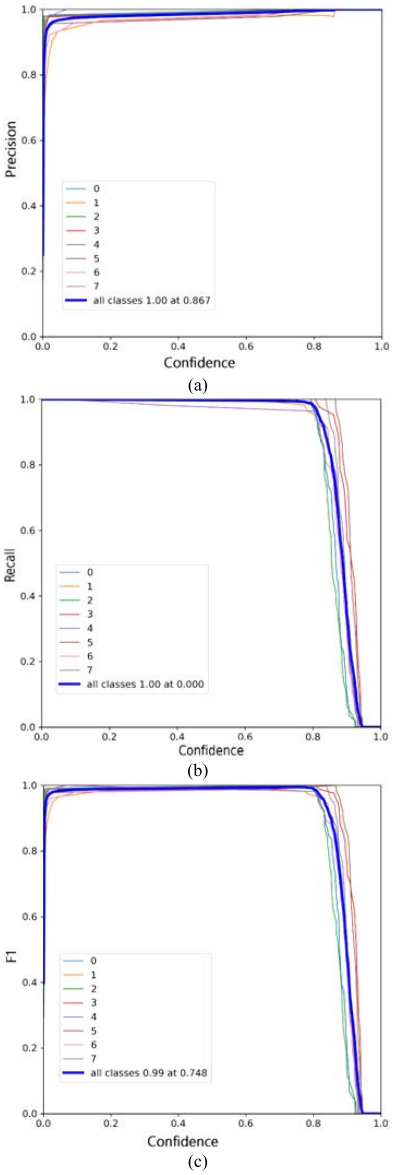


FIGURE 14. Confidence vs a) Precision b) Recall and c) F-1 score.

classification models with the related works [60] (Table 5), and YOLO v5 was used as the detection model.

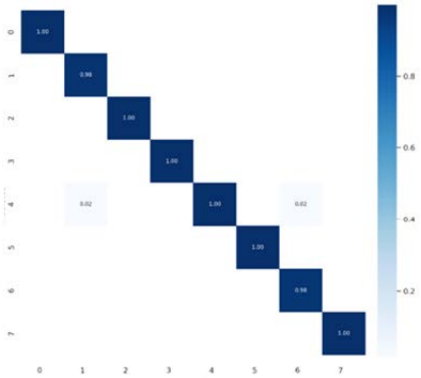


FIGURE 15. Confusion matrix of Model III.

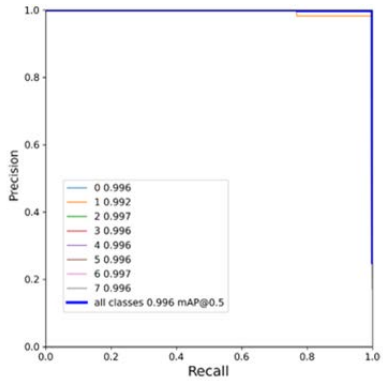


FIGURE 16. Precision vs Recall.

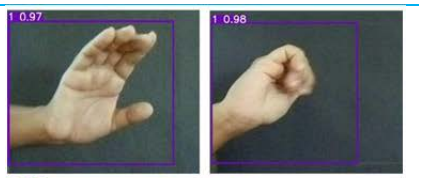
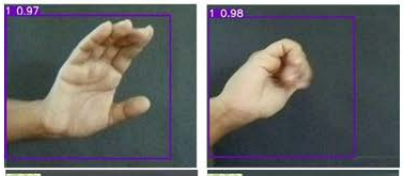
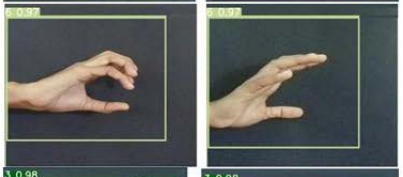
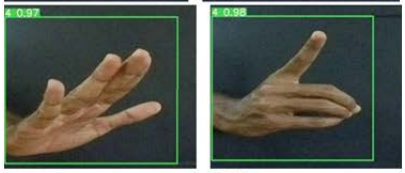
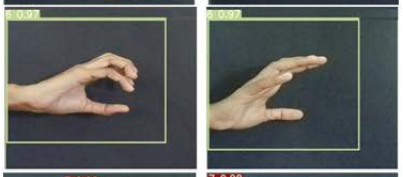
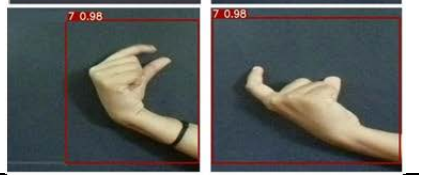
TABLE 5. Comparison of our classification models with that of V. Adithya [60].

Models	Method	Testing Accuracies (%)
V. Adithya [60]	Multi-Class SVM	90
V. Adithya [60]	Pre-trained GoogleNet+LSTM	96
Proposed Model I	3D CNN	82
Proposed Model II	Pre-trained VGG16 +LSTM	98

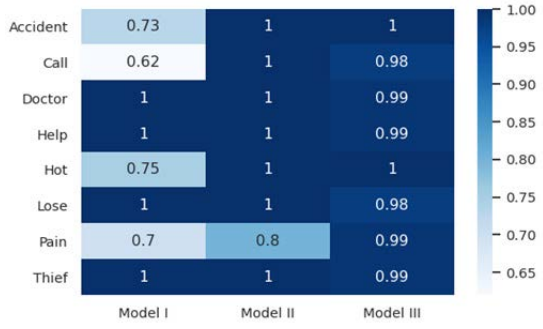
To compare the performance of models, their respective precision and recall values are shown in Figure 17. It can be seen that signs representing 'Accident,' 'Doctor,' 'Help' and 'thief' are classified accurately by all the models. Also, these signs require both hands in making the gesture, leading to the conclusion signs including both hands are relatively easier to classify. Next, 'Call,' and 'Pain' were the most difficult signs to be categorized accurately by any model. Both signs have dynamic gestures, and at a point in their movement, they seem to have a common hand position, which might have been challenging for models to differentiate. The model I could not yield satisfactory results in identifying dynamic signs such as Accident, Call, Hot, and Pain, implying that 3D CNN did not correctly learn all of the temporal characteristics.

In contrast, Model II was able to learn both the spatial and temporal features more comfortably because of the presence of a pre-trained VGG16 feature extractor. Compared to the 3D-CNN network, the combination of a VGG16 network

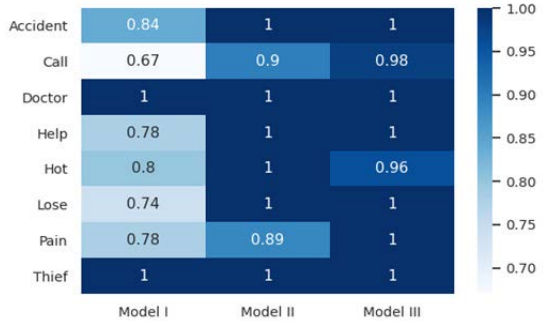
TABLE 6. Gesture detected using the proposed YOLOv5-based methodology for each class.

Sign	Class Label	Model's Output
Accident	0	
Call	1	
Doctor	2	
Help	3	
Hot	4	
Lose	5	
Pain	6	
Thief	7	

with an LSTM architecture achieved considerably greater precision and recall rates. On the other hand, YOLO v5 successfully detected each sign in the dataset with an overall precision and recall of 99.5%. YOLO offers several benefits over other techniques, making it an advanced detector. YOLO uses a single CNN for both classification and localization instead of utilizing a two-step approach for object classification and localization. Second, YOLO is fast and processes images at around 40-90 frames per second [66]. It implies that streaming video can be handled in real-time, with only a few



(a) Precision values of all three models



(b) Recall values of all three models

FIGURE 17. Performance comparison of all three models on the basis of a) Precision and b) Recall.

TABLE 7. Results from Model III.

Sign	Class Label	Precision (%)	Recall (%)	mAP
Accident	0	100	100	99.6
Call	1	98.2	98.5	99.2
Doctor	2	99.9	100	99.7
Help	3	99.8	100	99.6
Hot	4	100	96.6	99.6
Lose	5	98.5	100	99.6
Pain	6	99.4	100	99.7
Thief	7	99.8	100	99.6
	Total	99.5	99.4	99.6

milliseconds of delay. It suggests that Model III may be used to recognize emergency gestures in real-time.

V. CONCLUSION

The present work proposes classification and detection models on an emergency ISL dataset. The best classification model uses a combination of pre-trained VGG-16 and LSTM, while the detection model is based on the YOLO v5. The classification model achieved 98% accuracy, and the detection model achieved 99.6% mean average precision. The developed hand gesture recognition system classifies and detects both static as well as dynamic hand gestures from video frames. Furthermore, we found that even a smaller set of images are enough to recognize the dynamic gesture. For deaf people, sign language provides a means of emergency communication that helps them deal with difficult times. Situations like feeling pain, calling for help, or a doctor may arise anytime and anywhere. The present study opened the door to develop applications based on proposed methods that can be used when hearing-impaired people are in desperate situations and want to convey their thoughts to other people at the earliest.

COMPLIANCE WITH ETHICAL STANDARDS

- Disclosure of potential conflicts of interest: We declare that there is no conflict of interests among authors.
- Research involving human participants and/or animals (Ethical approval): This article does not contain any studies with human participants or animals performed by any of the authors.
- Informed consent: Informed consent was obtained from all individual participants included in the study.

ACKNOWLEDGMENT

The authors are grateful to the Taif University Researchers Supporting Project, Taif University, Taif, Saudi Arabia, under Grant TURSP-2020/36.

REFERENCES

[1] S. Escalera, V. Athitsos, and I. Guyon, "Challenges in multi-modal gesture recognition," *J. Mach. Learn. Res.*, vol. 17, no. 2, pp. 1–54, 2016.
[2] B. Garcia and S. A. Viesca, "Real-time American sign language recognition with convolutional neural networks," *Convolutional Neural Netw. Vis. Recognit.*, vol. 2, pp. 225–232, 2016.
[3] C. A. Padden, "Sign language geography," in *Deaf Around the World: The Impact of Language*. New York, NY, USA: Oxford Univ. Press, 2011, pp. 19–37.
[4] D. Tirthankar, S. Sambit, K. Sandeep, D. Synny, and B. A. Anupam, "A multilingual multimedia Indian sign language dictionary tool," in *Proc. 6th Workshop Asian Lang. Resour.*, 2008, pp. 11–12.
[5] Y. Fang, K. Wang, J. Cheng, and H. Lu, "A real-time hand gesture recognition method," in *Proc. IEEE Int. Conf. Multimedia Expo*, Beijing, China, Jul. 2007, pp. 995–998.
[6] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *J. Imag.*, vol. 6, no. 8, p. 73, 2020.
[7] H. Grant and C.-K. Lai, "Simulation modeling with artificial reality technology (SMART): An integration of virtual reality and simulation modeling," in *Proc. Winter Simulation Conf.*, Washington, DC, USA, Dec. 1998, pp. 437–441.
[8] T.-D. Tan and Z.-M. Guo, "Research of hand positioning and gesture recognition based on binocular vision," in *Proc. IEEE Int. Symp. VR Innov.*, Singapore, Mar. 2011, pp. 311–315.
[9] S.-H. Lee, M.-K. Sohn, D.-J. Kim, B. Kim, and H. Kim, "Smart TV interaction system using face and hand gesture recognition," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2013, pp. 173–174.
[10] H. Dong, A. Danesh, N. Figueroa, and A. E. Saddik, "An elicitation study on gesture preferences and memorability toward a practical hand-gesture vocabulary for smart televisions," *IEEE Access*, vol. 3, pp. 543–555, 2015.
[11] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3D convolutional neural networks," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Turin, Italy, Jun. 2015, pp. 1–6.
[12] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7361–7369.
[13] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: A review," *Int. J. Ind. Ergonom.*, vol. 68, pp. 355–367, Nov. 2018.
[14] D. Xu, X. Wu, Y.-L. Chen, and Y. Xu, "Online dynamic gesture recognition for human robot interaction," *J. Intell. Robot. Syst.*, vol. 77, nos. 3–4, pp. 583–596, Mar. 2015.

[15] N. L. Hakim, S.-W. Sun, M.-H. Hsu, T. K. Shih, and S.-J. Wu, "Virtual guitar: Using real-time finger tracking for musical instruments," *Int. J. Comput. Sci. Eng.*, vol. 18, no. 4, pp. 438–450, 2019.
[16] N. Dawar and N. Kehtarnavaz, "Real-time continuous detection and recognition of subject-specific smart TV gestures via fusion of depth and inertial sensing," *IEEE Access*, vol. 6, pp. 7019–7028, 2018.
[17] W. Kaczmarek, J. Panasiuk, S. Borys, and P. Banach, "Industrial robot control by means of gestures and voice commands in off-line and on-line mode," *Sensors*, vol. 20, no. 21, p. 6358, Nov. 2020.
[18] P. Neto, M. Simão, N. Mendes, and M. Safeea, "Gesture-based human-robot interaction for human assistance in manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 101, nos. 1–4, pp. 119–135, Mar. 2019.
[19] G. Young, H. Milne, D. Griffiths, E. Padfield, R. Blenkinsopp, and O. Georgiou, "Designing mid-air haptic gesture controlled user interfaces for cars," in *Proc. ACM Hum.-Comput. Interact.*, vol. 4, Honolulu, HI, USA, Apr. 2020, Art. no. 81.
[20] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, T. S. Alrayes, and M. A. Mekhtiche, "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192527–192542, 2020.
[21] A. Vaitkevičius, M. Taroza, T. Blažauskas, R. Damaševičius, R. Maskeliunas, and M. Woźniak, "Recognition of American sign language gestures in a virtual reality using leap motion," *Appl. Sci.*, vol. 9, no. 3, p. 445, Jan. 2019.
[22] T. M. Rezende, S. G. M. Almeida, and F. G. Guimarães, "Development and validation of a Brazilian sign language database for human gesture recognition," *Neural Comput. Appl.*, vol. 33, no. 16, pp. 10449–10467, Aug. 2021.
[23] J. Žemgulys, V. Raudonis, R. Maskeliunas, and R. Damaševičius, "Recognition of basketball referee signals from real-time videos," *J. Ambient Intell. Hum. Comput.*, vol. 11, no. 3, pp. 979–991, Mar. 2020.
[24] F. Afza, M. A. Khan, M. Sharif, S. Kadry, G. Manogaran, T. Saba, I. Ashraf, and R. Damaševičius, "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image Vis. Comput.*, vol. 106, Feb. 2021, Art. no. 104090.
[25] A. Nikolaidis and I. Pitas, "Facial feature extraction and pose determination," *Pattern Recognit.*, vol. 33, no. 11, pp. 1783–1791, Nov. 2000.
[26] A. Kulikajevas, R. Maskeliunas, and R. Damaševičius, "Detection of sitting posture using hierarchical image composition and deep learning," *PeerJ Comput. Sci.*, vol. 7, p. e442, Mar. 2021.
[27] K. Ryselis, T. Petkus, T. Blažauskas, R. Maskeliunas, and R. Damaševičius, "Multiple Kinect based system to monitor and analyze key performance indicators of physical training," *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, p. 51, Dec. 2020.
[28] S. P. K. Arachchi, N. L. Hakim, H.-H. Hsu, S. V. Klimenko, and T. K. Shih, "Real-time static and dynamic gesture recognition using mixed space features for 3D virtual world's interactions," in *Proc. 32nd Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Kraków, Poland, May 2018, pp. 627–632.
[29] H. Cheng, L. Yang, and Z. Liu, "Survey on 3D hand gesture recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1659–1673, Sep. 2016.
[30] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Comput. Appl.*, vol. 28, no. 12, pp. 3941–3951, Dec. 2017.
[31] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Boston, MA, USA, Jun. 2015, pp. 1–7.
[32] P. Ahmad, H. Jin, R. Alroobaea, S. Qamar, R. Zheng, F. Alnajjar, and F. Aboudi, "MH UNet: A multi-scale hierarchical based architecture for medical image segmentation," *IEEE Access*, vol. 9, pp. 148384–148408, 2021.
[33] M. Masud, M. D. Alshehri, R. Alroobaea, and M. Shorfuzzaman, "Leveraging convolutional neural network for COVID-19 disease detection using CT scan images," *Intell. Automat. Soft Comput.*, vol. 29, no. 1, pp. 1–3, 2021.
[34] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
[35] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[37] L. Pigou, S. Dieleman, P. J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 572–578.
[38] N.-T. Do, S.-H. Kim, H.-J. Yang, and G.-S. Lee, "Robust hand shape features for dynamic hand gesture recognition using multi-level feature LSTM," *Appl. Sci.*, vol. 10, no. 18, p. 6293, Sep. 2020.
[39] A. Elboushaki, R. Hannane, K. Afdel, and L. Koutti, "MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112829.
[40] V. John, A. Boyali, S. Mita, M. Imanishi, and N. Sanma, "Deep learning-based fast hand gesture recognition using representative frames," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2016, pp. 1–8.
[41] K. Lai and S. N. Yanushkevich, "CNN+RNN depth and skeleton based dynamic hand gesture recognition," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3451–3456, doi: 10.1109/ICPR.2018.8545718.
[42] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4207–4215.
[43] S. Saqib, A. Ditta, M. A. Khan, S. A. R. Kazmi, and H. Alquhayz, "Intelligent dynamic gesture recognition using CNN empowered by edit distance," *Comput., Mater. Continua*, vol. 66, no. 2, pp. 2061–2076, 2021.
[44] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Using convolutional 3D neural networks for user-independent continuous gesture recognition," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 49–54, doi: 10.1109/ICPR.2016.7899606.
[45] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius, R. Maskeliunas, and K. H. Abdulkareem, "Real-time hand gesture recognition based on deep learning YOLOv3 model," *Appl. Sci.*, vol. 11, no. 9, p. 4164, May 2021, doi: 10.3390/app11094164.
[46] S. Mambou, O. Krejcar, P. Maresova, A. Selamat, and K. Kuca, "Novel hand gesture alert system," *Appl. Sci.*, vol. 9, no. 16, p. 3419, Aug. 2019.
[47] M. Khari, A. K. Garg, R. Gonzalez-Crespo, and E. Verdú, "Gesture recognition of RGB and RGB-D static images using convolutional neural networks," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 5, no. 7, p. 22, 2019.
[48] P. Paul, M.-A.-U.-A. Bhuiya, M. A. Ullah, M.-N. Saqib, N. Mohammed, and S. Momen, "A modern approach for sign language interpretation using convolutional neural network," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, 2019, pp. 431–444, doi: 10.1007/978-3-030-29894-4_35.
[49] S. Masood, A. Srivastava, H. Thuwal, and M. Ahmad, "Real-time sign language gesture (word) recognition from video sequences using CNN and RNN," in *Intelligent Engineering Informatics*, vol. 695, V. Bhateja, C. C. Coello, S. Satapathy, and P. Pattnaik, Eds. Singapore: Springer, 2018, pp. 623–632, doi: 10.1007/978-981-10-7566-7_63.
[50] D. Tewari and S. K. Srivastava, "A visual recognition of static hand gestures in Indian sign language based on Kohonen self-organizing map algorithm," *Int. J. Eng. Adv. Technol.*, vol. 2, no. 2, pp. 165–170, Dec. 2012.
[51] R. B. Wilbur and A. C. Kak, "Purdue RVL-SLLL American sign language database," School Elect. Comput. Eng., Purdue University, West Lafayette, IN, USA, Tech. Rep. TR-06-12, 2006.
[52] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali, "The American sign language lexicon video dataset," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2008, pp. 1–8.
[53] M. Zahedi, D. Keysers, T. Deselaers, and H. Ney, "Combination of tangent distance and an image distortion model for appearance-based sign language recognition," in *Proc. Joint Pattern Recognit. Symp.* Berlin, Germany: Springer, 2005, pp. 401–408.
[54] F. Ronchetti, F. Quiroga, C. A. Estrebou, L. C. Lanzarini, and A. Rosete, "LSA64: An Argentinian sign language dataset," in *Proc. XXII Congreso Argentino de Ciencias de la Computacion (CACIC)*, 2016, pp. 794–803.
[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[56] X. Chai, H. Wanga, M. Zhoub, G. Wub, H. Lic, and X. Chena, "DEVISIGN: Dataset and evaluation for 3D sign language recognition," Inst. Comput. Technol., Chin. Academy Sci., Beijing, China, Tech. Rep., 2015.
[57] E. Efthimiou and S.-E. Fotinea, "GSLC: Creation and annotation of a Greek sign language corpus for HCI," in *Proc. HCI*, 2007, pp. 657–666.
[58] *Kinect Gesture*. Accessed: Jul. 16, 2019. [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=52283

[59] E. Gutierrez-Sigut, B. Costello, C. Baus, and M. Carreiras, "LSE-Sign: A lexical database for Spanish sign language," *Behav. Res. Methods*, vol. 48, no. 1, pp. 123–137, Mar. 2016.
[60] V. Adithya and R. Rajesh, "Hand gestures for emergency situations: A video dataset based on words from Indian sign language," *Data Brief*, vol. 31, Aug. 2020, Art. no. 106016, doi: 10.1016/j.dib.2020.106016.
[61] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[62] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, Feb. 2021, doi: 10.3390/f12020217.
[63] F. Chollet. (2017). *Keras*. [Online]. Available: http://keras.io
[64] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.
[65] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.
[66] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
[67] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 1, pp. 131–153, Jan. 2019, doi: 10.1007/s13042-017-0705-5.
[68] G. A. Rao, P. V. V. Kishore, A. S. C. S. Sastry, D. A. Kumar, and K. Kumar, "Selfie continuous sign language recognition with neural network classifier," in *Proceedings of 2nd International Conference on Micro-Electronics, Electromagnetics and Telecommunications*. Singapore: Springer, 2018, pp. 31–40.
[69] Y. Liao, P. Xiong, W. Min, W. Min, and J. Lu, "Dynamic sign language recognition based on video sequence with BLSTM-3D residual networks," *IEEE Access*, vol. 7, pp. 38044–38054, 2019, doi: 10.1109/ACCESS.2019.2904749.
[70] F. Obaid, A. Babadi, and A. Yoosofan, "Hand gesture recognition in video sequences using deep convolutional and recurrent neural networks," *Appl. Comput. Syst.*, vol. 25, no. 1, pp. 57–61, May 2020, doi: 10.2478/acss-2020-0007.
[71] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: A survey," *Evol. Intell.*, pp. 1–22, Jan. 2021, doi: 10.1007/s12065-020-00540-3.
[72] I. Kononenko, "Machine learning for medical diagnosis: History, state of the art and perspective," *Artif. Intell. Med.*, vol. 23, pp. 89–109, Aug. 2001, doi: 10.1016/S0933-3657(01)00077-X.
[73] Y.-D. Zhang, Z. Dong, S.-H. Wang, and C. Cattani, "IEEE access special section editorial: Deep learning for computer-aided medical diagnosis," *IEEE Access*, vol. 8, pp. 96804–96810, 2020, doi: 10.1109/ACCESS.2020.2996690.
[74] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nature Commun.*, vol. 11, no. 1, p. 3923, Dec. 2020, doi: 10.1038/s41467-020-17419-7.
[75] T. Javaheri *et al.*, "CovidCTNet: An open-source deep learning approach to diagnose COVID-19 using small cohort of CT images," *NPJ Digit. Med.*, vol. 4, no. 1, p. 29, Dec. 2021, doi: 10.1038/s41746-021-00399-3.
[76] K. Schindler and L. van Gool, "Action snippets: How many frames does human action recognition require?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587730.

QAZI MOHAMMAD AREEB is currently pursuing the degree with the Department of Computer Science, Aligarh Muslim University. He has worked as a Research Assistant at the Department of IT, University Technology Petronas (UTP), Malaysia. He is an Active Member of the Computer Science Society Research Group, where he has worked on various research problems. He has experience in the field of artificial intelligence, machine learning, and cybersecurity. He has published several research papers in international conferences and peer-reviewed journals. His main research interests include developing serviceable technologies through artificial intelligence and computer vision.

MARYAM graduated from AMU, in 2018. She received the master's degree in computer science and applications from the Department of Computer Science, AMU, Aligarh, in 2021. She is also a Certified Microsoft Technology Associate in Python Programming. Her research interests include machine learning and natural language processing. For her master's degree, she was awarded with the University Gold Medal.

MOHAMMAD NADEEM received the bachelor's and master's degrees from AMU, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from IIT (ISM), Dhanbad, Jharkhand, in 2017. He is currently working as an Assistant Professor with the Department of Computer Science, AMU, Aligarh. In June 2016, he joined the department. Before that, he has also worked as an Assistant System Engineer at Tata Consultancy Services (TCS), from January 2012 to February 2013. His research interests include machine learning and soft computing. He was also awarded University Medal in Graduation.

ROOBAEA ALROOBAEA received the bachelor's degree (Hons.) in computer science from King Abdul-Aziz University (KAU), Saudi Arabia, in 2008, and the master's degree in information systems and the Ph.D. degree in computer science from the University of East Anglia, U.K., in 2012 and 2016, respectively. He is currently an Associate Professor with the College of Computers and Information Technology, Taif University, Saudi Arabia. His research interests include human–computer interaction, software engineering, cloud computing, the Internet of Things, artificial intelligence, and machine learning.

FAISAL ANWER received the master's degree in computer application and the Ph.D. degree in information security from Jamia Millia Islamia, New Delhi. He is currently working as an Assistant Professor with the Department of Computer Science, AMU, Aligarh. Prior to joining AMU, he worked as a Senior Software Engineer at Computer Science Corporation (CSC), Noida. He has also worked with CSC, U.K., from 2009 to 2010. He published several research papers in international/national conferences and journals. His research interests include software security, cryptography, program robustness, and machine learning.