Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.038211
Article

Appearance Based Dynamic Hand Gesture Recognition Using 3D Separable Convolutional Neural Network

Muhammad Rizwan[1,*], Sana Ul Haq[1,*], Noor Gul[1,2], Muhammad Asif[1], Syed Muslim Shah[3], Tariqullah Jan[4] and Naveed Ahmad[5]

[1]Department of Electronics, University of Peshawar, Peshawar, 25120, Pakistan
[2]Department of Electronics Engineering, Korea Polytechnic University, Siheung, Korea
[3]Department of Electrical Engineering, Capital University of Science and Technology, Islamabad, 44000, Pakistan
[4]Department of Electrical Engineering, University of Engineering and Technology, Peshawar, 25120, Pakistan
[5]Department of Computer Science, Prince Sultan University, Riyadh, 11586, Saudi Arabia
*Corresponding Authors: Muhammad Rizwan. Email: rizwan.haider1995@gmail.com; Sana Ul Haq. Email: sanaulhaq@uop.edu.pk
Received: 02 December 2022; Accepted: 28 March 2023; Published: 09 June 2023

Abstract: Appearance-based dynamic Hand Gesture Recognition (HGR) remains a prominent area of research in Human-Computer Interaction (HCI). Numerous environmental and computational constraints limit its real-time deployment. In addition, the performance of a model decreases as the subject's distance from the camera increases. This study proposes a 3D separable Convolutional Neural Network (CNN), considering the model's computational complexity and recognition accuracy. The 20BN-Jester dataset was used to train the model for six gesture classes. After achieving the best offline recognition accuracy of 94.39%, the model was deployed in real-time while considering the subject's attention, the instant of performing a gesture, and the subject's distance from the camera. Despite being discussed in numerous research articles, the distance factor remains unresolved in real-time deployment, which leads to degraded recognition results. In the proposed approach, the distance calculation substantially improves the classification performance by reducing the impact of the subject's distance from the camera. Additionally, the capability of feature extraction, degree of relevance, and statistical significance of the proposed model against other state-of-the-art models were validated using t-distributed Stochastic Neighbor Embedding (t-SNE), Mathew's Correlation Coefficient (MCC), and the McNemar test, respectively. We observed that the proposed model exhibits state-of-the-art outcomes and a comparatively high significance level.

Keywords: 3D separable CNN; computational complexity; hand gesture recognition; human-computer interaction

1 Introduction

The vision-based HGR is a prominent area of research in non-verbal HCI [1]. The importance of vision-based HGR stems from its wide range of applications, which include sign language interpretation [2], virtual reality, augmented reality [3], and desktop applications. The first vision-based HGR system was introduced in the early 80s, but due to environmental and computational constraints, the approaches proposed in that period were unable to attain the desired outcomes [4].

Recent technological advancements such as high-resolution cameras, Graphics Processing Units (GPUs), and 3D cameras have resulted in the development of numerous HGR models. The HGR models can be grouped into either depth-based or appearance-based approaches, depending on the technology utilized for the data acquisition. In the depth-based approach, a camera with an embedded distance sensor is used to acquire the raw data. The object's depth details are provided through the distance sensor, while the camera is employed to concentrate on the hands. Kinect-1.0 [5], Real Sense [6], LiDAR [7], and Leap Motion Sensor [8] are examples of depth-based technologies. On the other hand, the appearance-based approach obtains raw data through an inexpensive 2D camera, which is usually embedded in numerous devices including desktops, laptops, tablets, and smartphones. The acquired data is then processed through different algorithms for gesture recognition. For better recognition accuracy, the respective approach can make use of multiple 2D cameras. The proposed research utilizes the appearance-based approach since it is cost-effective and can be easily deployed on various devices.

The appearance-based approach can be further classified into static and dynamic HGR. In the static HGR data is recorded in the form of still pictures which requires the subject to hold a specific stance for an instant so that it can be accurately captured. In the case of dynamic HGR, the sequence of gesture movements is provided as the model's input. This approach is more practical because a gesture is performed by making certain movements. For this reason, the dynamic HGR has great significance as it incorporates the behavior of Human-to-Human Interaction (HHI).

The appearance-based HGR faces several obstacles due to diverse environmental variables that affect the overall performance of the model. These problems include varying gesture velocity, illumination, skin color, and the subject's distance from the camera. In addition, the computational complexity of a model is a major challenge. A model with high computational complexity requires expensive computing resources, which not only increases the model's cost but also limits its implementation on a target device.

Researchers have proposed numerous HGR techniques, including the Hidden Markov Model (HMM), Templet Matching [9], Dynamic Time Warping (DTW) [10], k-Nearest Neighbour (k-NN), Support Vector Machine (SVM), and CNN [11]. CNN is the best-known deep learning approach because of its efficient classification and feature extraction capabilities. The architecture was introduced by Lecun et al. [12] in 1998 but remained off-sight because of high computational expenses and limited resources. Later, in 2012, a CNN variant surpassed other models in Kaggle's competition by achieving the highest accuracy in image classification tasks. Since then the competition has been dominated by CNN every year [13]. Furthermore, the CNN model can be deployed using a GPU but it is computationally expensive. For this reason, we must choose between a model's performance and its computational complexity. The proposed work employs the CNN architecture in novel ways to achieve an improved generalization and recognition rate without compromising the model's computational

complexity. In addition, the proposed model was deployed in real-time while taking into account the subject's attention, the instant of performing a gesture, and the subject's distance from the camera.

This paper is organized as follows. Section 2 provides an overview of the related research work in HGR. Whereas, Sections 3 and 4 discuss the dataset and the pre-processing approaches used in this research, respectively. Sections 5 and 6 provide details about the base model and proposed model, respectively. It is followed by the real-time implementation of HGR models in Section 7. The experimental setup for training, validation, and testing is discussed in Section 8. The behavioral analysis of the model is given in Section 9, which is followed by the experimental results in Section 10. Finally, a detailed discussion of the experimental results and the conclusion is provided in Section 11 and Section 12.

2 Related Work

The advent of HCI in the early 80s paved the way for vision-based HGR. The initial vision-based HGR attempts were either performed with colored gloves or hand markers. The techniques, however, were unable to attain the required accuracy due to various technological restrictions including camera resolution and computational power. Besides color-glove and hand marker-based approaches [4], the true vision-based approach for HGR was first reported by Rehg et al. [14] in 1993. A 3D model was developed that acquired data using a pair of cameras positioned at different angles. Though the method eliminated self-occlusion that occurs with a single camera is used, it was achieved by restricting motion to a confined area.

In the year 1993, the vision-based HGR drew the attention of most researchers. Since then, numerous models have been developed including SVM, HMM, DTW, and Templet Matching. In addition to these machine learning models, CNN has been proven to be the most efficient feature extraction and classification model. Molchanove et al. [15] proposed two parallel streams of the 4-layered 3D-CNN model. One of the streams was trained on high-resolution input, while the other was trained on low-resolution input. The model was trained on the VIVA dataset for 19 hand gesture classes. The model attained an average accuracy of 77.5% with comparatively less computational complexity. Singha et al. [11] introduced a hybrid approach for HGR using SVM, k-NN, and Artificial Neural Network (ANN) classifiers. The model was trained using NITS hand gesture database IV, consisting of 40 hand gesture classes. The overall accuracy of the model was 92.2%, but it had high computational complexity. Hussain et al. [16] utilized a pre-trained VGG-16 deep learning model and fine-tuned it on 6 static and 8 dynamic hand gestures. The dynamic HGR has high feature complexity. Therefore, the Hue Saturation Value (HSV) skin algorithm followed by blob area elimination was used to pre-process the input frames. The model obtained a maximum offline accuracy of 93.1%. Since the feature extraction approach was dependent on skin color, the real-time performance of the model was degraded. In addition, the model utilized 16 convolutional layers which led to a computationally complex design and high costs in real-time implementation. Zhang et al. [17] developed a 4-layered 3D convolutional model that was trained on 27 dynamic hand gesture classes using the 20BN-Jester dataset. The model has attained an average accuracy of 90.0%, but the real-time performance of the model degraded as the distance between the subject and the camera was increased [16,18].

Kopuklu et al. [19] proposed a real-time dynamic HGR model by fine-tuning ResNet101 [20] on the 20BN-Jester dataset for 27 gesture classes. The model obtained the best recognition accuracy of 97.0%, but it was computationally costly due to the greater number of layers. Sarma et al. [21] utilized

the model ensemble approach by fusing 2D and 3D CNN models to achieve an average accuracy of 99.0%. In their approach, 2D-CNN input was pre-processed through an optical flow-guided motion template. The technique obtained high accuracy, but the real-time implementation of the approach may be slower due to the high pre-processing time and the model's complexity. The model was trained and tested on the Graffiti dataset, consisting of 10 hand gesture classes. Al-Hammadi et al. [2] proposed a technique of recording the whole-body motions and then extracting the upper arm and hand data using an open pose framework. The extracted data was used as input to the proposed two-stream autoencoder for gesture recognition. The proposed approach achieved an overall accuracy of 87.7% for 40 hand gesture classes. In real-time, the overall performance of the model degraded because some gestures with similar angles of movements got confused with each other. In addition, due to the parallel streams of network architecture, the respective model had high computational complexity.

A 3D separable CNN model was proposed by Hu et al. [3], which was trained on a self-collected dynamic hand gesture dataset of 6 classes. The computational complexity of the model was reduced by employing the MobileNet [22] and ShuffleNet [23] approaches. The model obtained an average accuracy of 98.8% and was computationally efficient. However, the real-time performance of the model was very poor.

The majority of the models proposed so far were developed to acquire the input data while only concentrating on the subject's hands. Whereas, the proposed research work incorporates the upper half of the subject's body, which includes the head, face, hands, and arms. In order to extract the key feature from the acquired data, several head and face detection algorithms are proposed by researchers. Saqib et al. [24] fine-tuned Faster Region-Based CNN (F-RCNN) on the publicly available dataset of Hollywood movies for real-time detection of subjects' heads. The model outperformed VGG16 and YOLO V2 by gaining a comparatively better mean average precision (mAP) of 79.1%. Khan et al. [25] designed a Fully Convolutional Network (FCN) for head and face detection. The model was trained and tested on four different datasets, i.e., Hollywoodheads, Casablanca, SHOCK, and WIDERFACE. The proposed approach was able to achieve a comparatively better F1 score of 83% for the WIDERFACE dataset. In addition, the proposed model was further evaluated for three different categories based on the size of subjects' heads, i.e., small, medium, and large, using the WIDERFACE dataset. The approach outperformed the rest of the models by achieving a comparatively better mAP of 72.34%, 82.41%, and 83.94% for the respective datasets. Zhang et al. [26] proposed a Multitask Cascaded Convolutional Network (MTCNN) for the detection of subjects' faces. The model consists of three different stages: Proposed Network (P-Net), Refined Network (R-Net), and Output Network (O-Net). The MTCNN was trained and tested on the WIDERFACE dataset and was able to achieve a comparatively high average accuracy of 95.4%. Viola et al. [27] developed a face detection model using the Adaptive Boost (AdaBoost) algorithm. The proposed model has a comparatively lower level of computational complexity and an average accuracy of 97%. In addition, Viola Jones' algorithm has become so prominent that most devices use it for face detection, including smartphones and digital cameras. While considering the real-time performance of Viola Jones' algorithm, we used it in the proposed research work.

HGR, regardless of the number of remarkable research contributions, still encounters certain computational and performance barriers. We observed that the deep learning architectures developed so far either have high performance but are computationally complex or vice versa. Hence, a give-and-take situation exists between the model's complexity and performance. The objective of the proposed research was to develop a 3D CNN architecture with comparatively better recognition accuracy and less computational complexity. Besides this, the other factor that highly impacts the model's

performance is the subject's distance from the camera. The proposed work developed an approach for real-time deployment of the model while considering the subject's distance from the camera.

3 Dataset

The proposed model was designed for training on a dataset that accounts for the most realistic aspects, including varying illumination, gesture velocity, and skin color. Another important factor was that a gesture sample should include the subject's entire body above the waist rather than just their hands. This consideration was necessary to enable the proposed model for natural HHI. The 20-BN Jester dataset [28] meets all these requirements. It is comprised of 148,000 video samples that are grouped into 27 classes. The details of these 27 classes along with sample distribution are given in Fig. 1. The dataset also considered a variety of cultural and environmental aspects, such as varying gesture velocity, illumination, and skin color, as shown in Fig. 2. The data was acquired by developing a platform that interacted with a crowdsourcing platform like Amazon Mechanical Turk (MTurk) to find the crowd workers to accept the task and then redirect them to the developed platform. The crowd workers were given samples of the gestures to be performed via the platform. Each gesture's data was acquired in the form of a 3-s video, which was recorded using the front camera of the computer. The distance between the subject and the camera was not restricted but was observed to be in the range of 70 to 90 cm. The recorded video was reviewed on the platform to respond to workers with either a successful or unsuccessful status. If the worker received a successful status, they were paid; otherwise, they were allowed to redo the task rather than be rejected immediately. The total number of subjects recorded with the respective setup was 1376, i.e., an average of 43 video samples were acquired per subject. It should also be noted that some samples in the dataset were recorded more often than others, thus increasing the overall sample size.

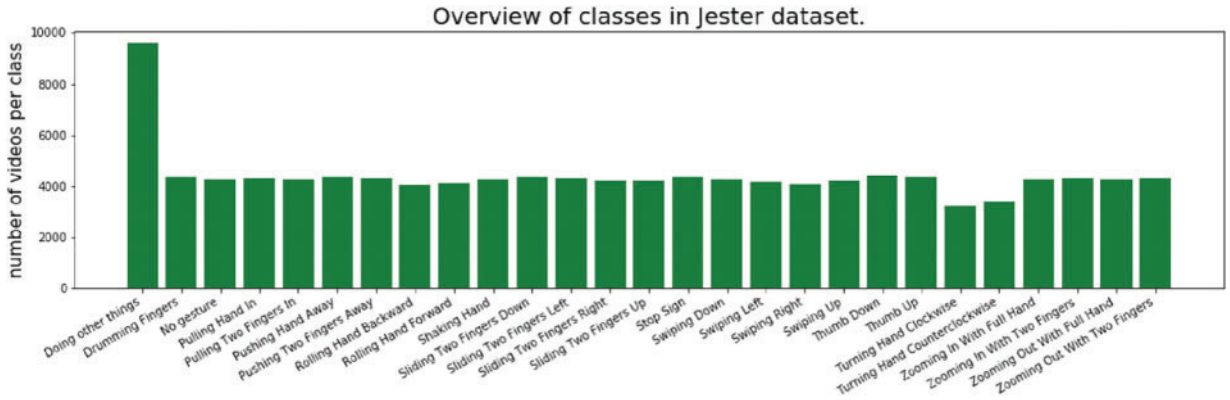

Figure 1: Distribution of 27 gesture classes in the 20-BN Jester dataset

In the proposed study, we aimed to operate the basic features of a desktop using gestures, i.e., sliding left, sliding right, sliding up, sliding down, and terminating the active window on a computer. Therefore, 6 out of 27 classes from the 20BN-Jester dataset were utilized, i.e., swiping left, swiping right, swiping up, swiping down, stop sign, and no gesture class. The desktop operations were performed in correspondence to these gesture classes, except for the no-gesture class.

Figure 2: Samples from the 20-BN Jester dataset

4 Data Pre-Processing

The deep learning architecture, in contrast to machine learning, can be trained on data samples without any pre-processing techniques. The approach reduces human effort to some extent, but it requires a comparatively longer learning time due to its high dimensionality. This problem can be addressed through pre-processing algorithms such as motion history images [29], optical flow [30,31], and frame differencing [32]. In this work, key features from the input were efficiently extracted through either optical flow or frame differencing as compared to the motion history image. In addition, we observed that the average processing time consumed between two frames by the optical flow was 0.06 s, and for the frame differencing it was 0.004 s. The real-time deployment of the proposed model requires consideration of several factors, including the pre-processing time and performance, so the frame differencing algorithm was used for this purpose.

The procedure for the frame differencing algorithm depends on the following two steps:

1. Firstly, the input video frames were converted from RGB to grayscale, as shown in Figs. 3a and 3b, respectively. The numbers on the left side of the frames are used to indicate their sequence.
2. Secondly, the subject's movement from the first to the last frame is extracted by subtracting each frame from the subsequent frame, as shown in Fig. 3c. The two successive grayscale frames utilized in the frame differencing are shown by the numbers on the left side of the frames in Fig. 3c.

Since each video sample has a distinct number of input frames, we standardized the number of input frames to 8 after the frame differencing. The frames were standardized to 8 by either adding the difference between the previous frame with itself or removing the additional frames. The removal of extra frames, i.e., generally 1 or 2 frames, caused no loss since the key gesture features were observed to end earlier, as shown in Fig. 3c.

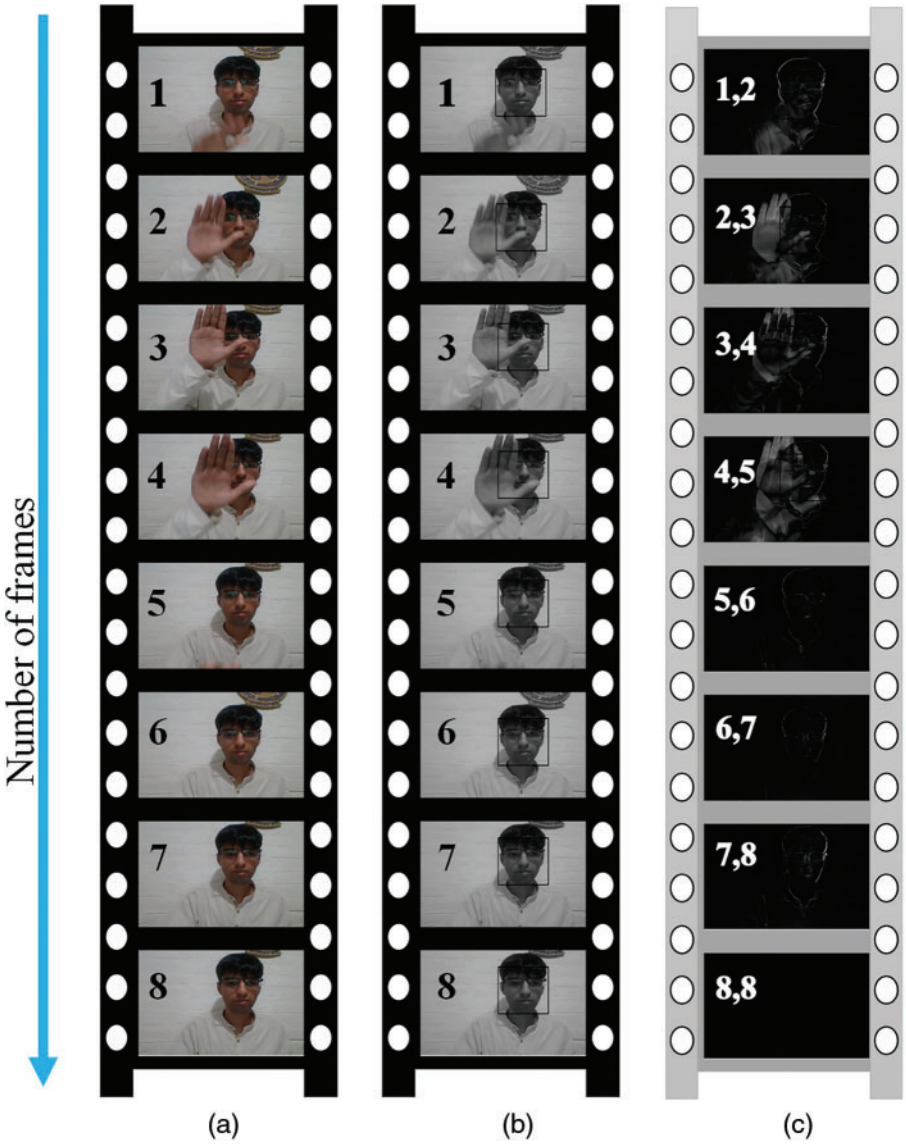

Figure 3: Frame differencing by converting the (a) raw RGB frames to (b) grayscale frames, and taking the (c) difference between two consecutive frames

5 Base Model

The dynamic HGR requires the extraction of spatial as well as temporal features, which results in high-dimensional complex data. Hu et al. [3] proposed a 3D separable CNN for dynamic HGR, as shown in Fig. 4. The model was developed for the application of augmented reality goggles. This model is used as a base model in the proposed work. The 3D CNN, besides spatial features, also extracts temporal features for the dynamic HGR. The depth-wise separable CNN [22] approach is utilized in the design of the 3D CNN model, as the standard approach has high computational complexity. To obtain better recognition accuracy, the ShuffleNet [23] and ResNet [20] approaches were used to train the deeper layers as efficiently as the shallow layers. The structural features of the base model include:

1. A convolution kernel size of $3 \times 3 \times 3$ was used to reduce the number of weights,
2. The temporal dimension was down-sampled at the end (i.e., convolution blocks 8 and 9) to learn more features,
3. To learn the downsampling process, the greater value of stride was used instead of pooling, and
4. To extract better information, more channels were added before the downsampling.
5. The convolution blocks 2 to 10 performed 3D separable convolution, while convolution block 1 utilized the standard 3D convolution [23].

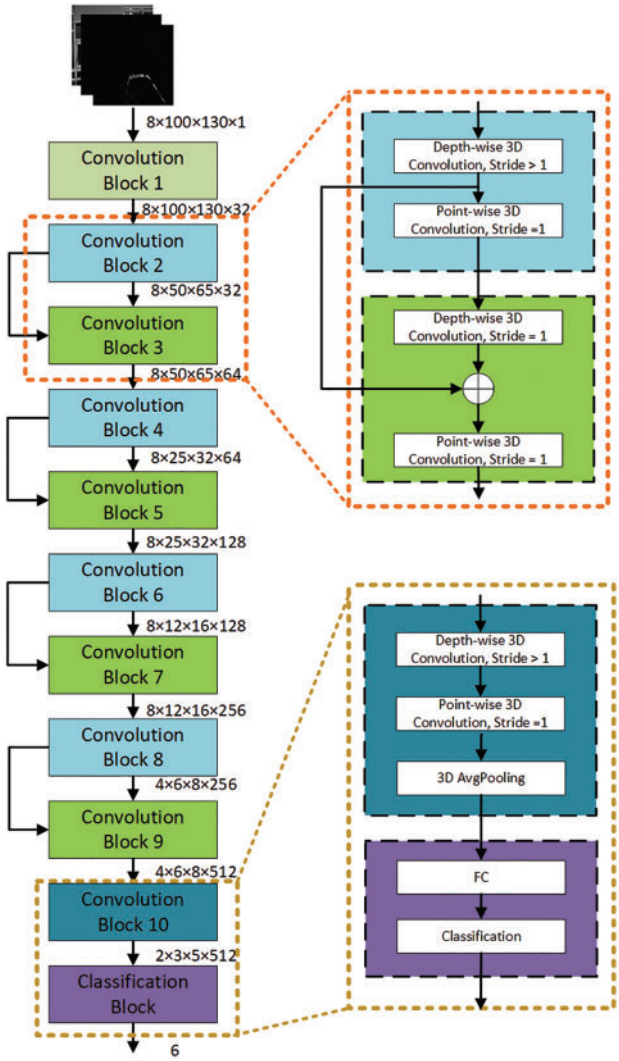

Figure 4: Base model architecture for hand gesture recognition

The base model was trained on 6 hand gestures from a dataset collected through HoloLens. The data set contains a total of 110,000 samples, with an average of 22,000 samples per class. The frame differencing algorithm was used to discard the complex background before training the model. The log function was used as a loss function, while the RMSprop was used for the model's optimization. In addition, the layer-wise learning approach [33] was used to speed up the training process. The base model was trained for 5 epochs with an average accuracy of 95.7%. Furthermore, it should be

noted that the aforementioned experimental setup and their respective outcomes are summarized in [3]. Whereas, the respective arrangements utilized in the proposed work are discussed in Sections 8 and 10.

6 Proposed Model

The proposed model was structured with enhanced generalization and feature extraction capabilities. These fundamental characteristics have significantly improved the proposed model's performance in comparison to other state-of-the-art models. The steps taken to accomplish the desired outcome are discussed in the following subsections.

6.1 Enhancing Generalization

The base model proposed in [3] utilizes a separable CNN in which each depth-wise convolution layer is followed by an activation function, as depicted in Fig. 5a. Whereas in the proposed approach, the activation function was used after each convolution layer, i.e., depth-wise and point-wise layers of the separable CNN. The respective arrangements are illustrated in Fig. 5b. The model's generalization behavior and rate of convergence were analyzed through experiments for each of the corresponding arrangements rather than just validation accuracy, as in [34] and [35]. The Leaky Rectified Linear Unit (Leaky ReLU) was used as an activation function for the experiments. It is observed from the experimental results that the use of an activation function after each depth-wise and point-wise convolution layer results in enhanced generalization. The outcome of the experiments is discussed in Section 11.

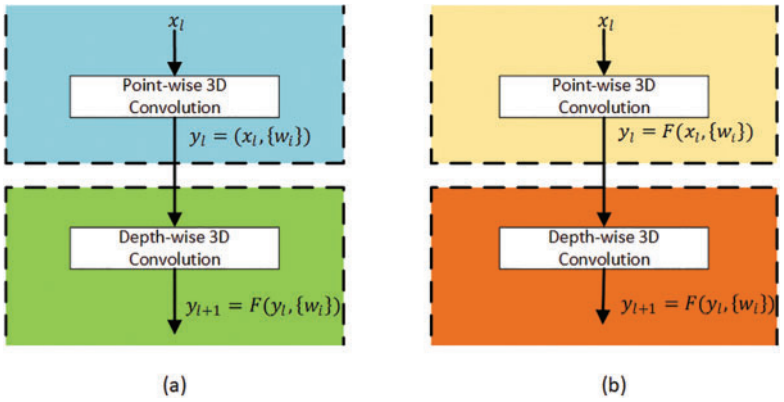

Figure 5: Model's generalization of (a) the base model, and (b) the proposed model

In Fig. 5, x_l is the input to a layer, $\{w_i\}$ are the weights, and y_l is the output of the respective layer. The activation function is represented by $F(\ )$.

6.2 Enhancing Feature Extraction

The stride was used for the downsampling in the base model to automatically learn the process, as shown in Fig. 6a. However, a stride is just a hyperparameter that is used to specify the kernel's step size. When stride one is used, the kernel step is reduced to a minimum value that results in enhanced feature extraction. The experiments were conducted using the base model with two different techniques of downsampling, i.e., stride and max pooling, as shown in Fig. 6. The maximum pooling was preferred over the average pooling because of the better representation of the input samples. To analyze the feature extraction capabilities of the proposed model in comparison to the base model, we utilized

the t-distributed Stochastic Neighbour Embedding (t-SNE) approach [36]. The t-SNE is a machine learning approach used for the visual analysis of high-dimensional features by mapping them to low-dimensional features. The visual outcomes of the t-SNE approach are shown in Fig. 7. The outcomes for the proposed and base models were obtained using the same test set as mentioned in Section 8.1. We observed that the proposed model was able to learn the distinct features far more effectively than the base model. Fig. 7 shows that in the case of the proposed model with enhanced feature extraction, each cluster class was easily distinguishable from others as compared to the base model. We also observed that the convergence rate of the proposed model was improved, as discussed in Section 9.

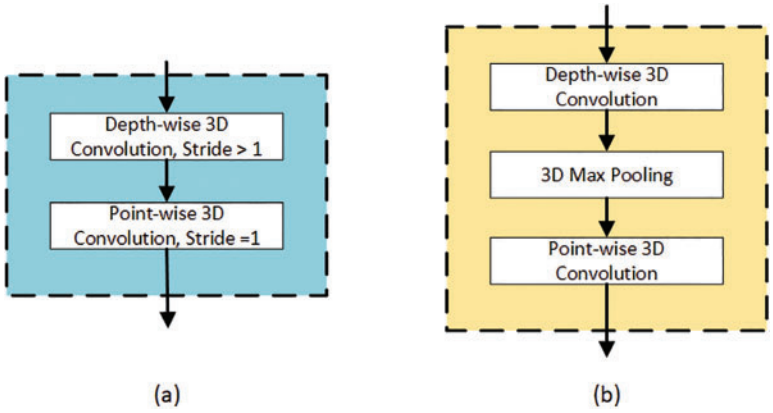

Figure 6: Model's features extraction using (a) the stride, and (b) the max pooling

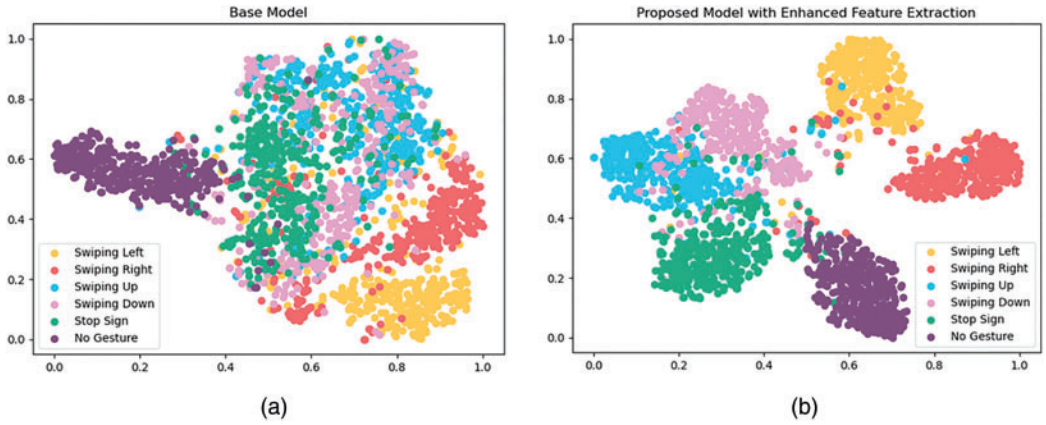

Figure 7: Visualization of the extracted features using t-SNE for the (a) base model, and (b) proposed model with enhanced feature extraction

6.3 Enhancing Model's Parameters

The parameters of the base model were reduced up to eight times before the fully connected layers, as shown in Fig. 4. For this reason, the layers near the input learned more features than the later layers. In the proposed technique, rather than abruptly losing the parameters before the fully connected layers, downsampling was applied in steps, as shown in Fig. 6b. This approach slightly enhances the proposed model's parameters as compared to the base model but reduces the model's computation time and complexity. In addition, it has resulted in better recognition accuracy, as discussed in Section 10.

The proposed model architectures without and with parameters enhancement are shown in Fig. 8. These architectures exhibit the following common features:

1. The models are comprised of 10 convolution blocks and a classification block,
2. The Convolution Block 1 performs the standard 3D convolution,
3. The Convolution Blocks 2 to 10 utilize the 3D separable convolution [23] for computational efficiency, and
4. The ResNet [20] approach is used for 4 pairs of the Convolution Blocks 2 to 9 to train the deeper layers as efficiently as the shallow layers.

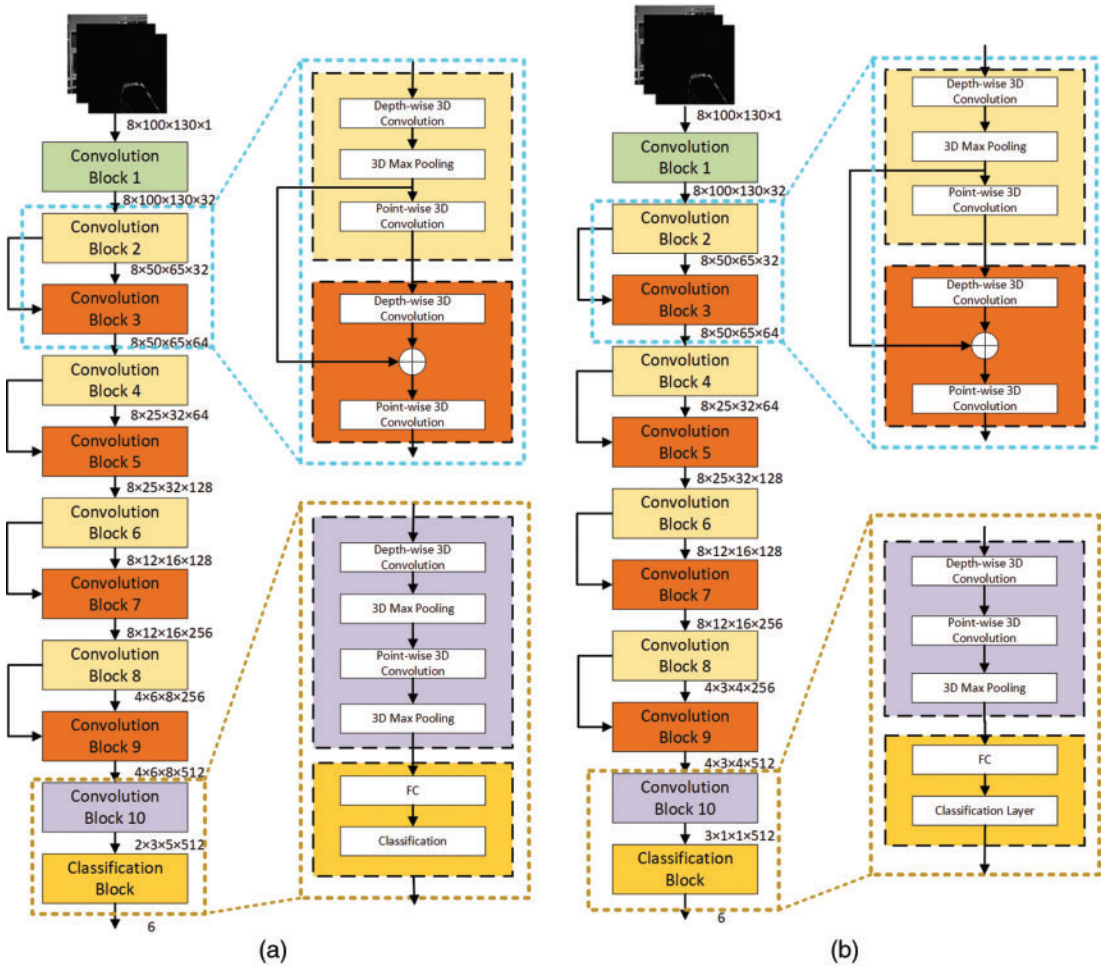

Figure 8: Proposed model (a) without parameters enhancement, and (b) with parameters enhancement

The only difference between the proposed models without and with parameter enhancement is at Convolution Blocks 8 and 10, as can be observed from the output dimensions of the respective blocks in Fig. 8. The rest of the arrangements for both the proposed models are identical.

7 Real-Time Deployment of the HGR Model

The real-time deployment of the proposed HGR model was based on the following three factors:

1. Attention of the subject towards the device,
2. The instant of performing a gesture, and
3. The subject's distance from the camera.

These factors are critical in real-time deployment because ignoring them would result in significant wastage of computing resources. In addition, ignoring the distance between the subject and the camera results in a decline in the model's performance. The following subsections discuss the real-time deployment of the model.

7.1 Detection of Subject's Attention

The subject's attention can be detected from the face by determining whether it is directed towards or away from the camera. This was accomplished with the help of a face detection algorithm. Researchers have proposed various face detection algorithms MTCNN [26], F-RCNN [37], and the Viola Jones algorithm [27]. The Viola Jones face detection algorithm is the most commonly used algorithm and has been deployed for numerous applications. The reason for such a high significance is the higher rate of detection, which is approximately 0.06 s with 97% average accuracy [27]. Besides Viola Jones, the other face detection algorithms especially those based on deep learning architecture, provide better detection accuracy but have a slower response time, e.g., 0.5 s for MTCNN. In addition, since the Viola Jones algorithm uses Haar features, it can only detect the face when it is directed towards the camera, while the other deep learning architectures can detect the face even from the side. This drawback of the Viola Jones algorithm has made it suitable for detecting a subject's attention, as the algorithm will only detect a face when directed toward the camera.

In order to ensure the precise detection of the subject's attention, we utilized four sequential frames in real-time. The subject is considered attentive only when the algorithm detects the face in each of these four frames. The proposed approach is explained with the help of a pseudocode, as given in Algorithm 1.

Algorithm 1: Pseudocode for the detection of subject's attention.

```
Face_detected = 0
for i=1 to 4 do
        A(i) = Input rgb frames
        A1(i) = Convert rgb frames A(i) to grayscale frame
        B = Detect face from A1(i) using Viola Jones algorithm
        B_shape = find Shape of B
        if B_shape not empty then
                Face_detected = Face_detected + 1
                        if Face_detected =4 then
                                Final = "Positive"
                        end if
        end if

        else do
                Face_detected = 0
                Final = "Negative"
        end else
end for
Results ⟶ Final provides the Subject's Attention detected through four frames
```

7.2 Instant of Performing Gestures

The consideration of the subject's attention preserves some computational resources; however, a subject may be attentive but still not be performing any gesture. For this purpose, a frame differencing approach was adopted, which was based on four consecutive frames. The algorithm detects any movement below the neck based on a threshold. This approach efficiently utilizes the resources to a greater extent and also helps to compute the gesture's duration. The pseudocode for the respective approach is given in Algorithm 2.

Algorithm 2: Pseudocode for the detection of instant of gesture being performed.

```
Y = Processed frames to grayscale
for i=1 to 4 do
        A(i) = Face detected in Y through Viola Jones algorithm
        A1(i) = Extract A from Y
        A2(i) = Store A1 for further processing
end for

B = Extract features from A2 using frame differencing
B1 = Count extracted features in B
if B1 > threshold then
        Final = "Motion Detected"
end

else do
        Final = "No Motion Detected"
end else
Results ⟶ Final provides the instant of motion being detected
```

7.3 Subject's Distance from the Camera

The model's performance downgrades as the subject's distance from the camera increases [16,18]. This is due to the blending of the subject's features into the background. To reduce the effect of distance variation on the model's performance, the following steps were taken:

1. In the first step, the facial coordinates x, y, w and h were detected from the frame, F_r, having height H_f and width W_f using the Viola Jones algorithm. Where the x and y are the top left coordinates of the face, and, w and h represent the width and height of the face, respectively.
2. Secondly, the threshold value of w was adjusted. It was observed from the experiments that the value of w was inversely related to the subject's distance from the camera. A total of 100 samples of w were recorded for the subject's distance greater than the usual, i.e., 75 cm from the camera. These real-time samples of w were analyzed to set the threshold value of w.
3. In the case of the w having a value less than the threshold, then the mid-point M_B of the subject's body was calculated using (1).

$$M_B = x + \frac{w}{2} \tag{1}$$

4. An offset coefficient α was added to y, so that the subject's head becomes part of the resulting frame. The resultant offset value of the subject's y-coordinate is given by

$$Y_{offset} = y + \alpha \tag{2}$$

5. The subject's body width W_B and height H_B were calculated using (3) and (4), respectively.

$$W_B = M_B \pm 2w \tag{3}$$

$$H_B = \left(\frac{H_f - Y_{offset}}{2}\right) + \beta \tag{4}$$

The coefficient β was used in (4) in order to avoid the body height from being too small in the case of the subject's face laying close to the lower border of the input frame.

6. Finally, the desired frame F_E was extracted from the frame F_r as given in (5). The frame F_E was resized according to the input dimensions of the HGR model.

$$F_E = F_r\,(H_B, W_B\,) \tag{5}$$

The desired outcome of (5) is shown in Fig. 9. We observed that the proposed approach can efficiently detect the subject's body within a frame.

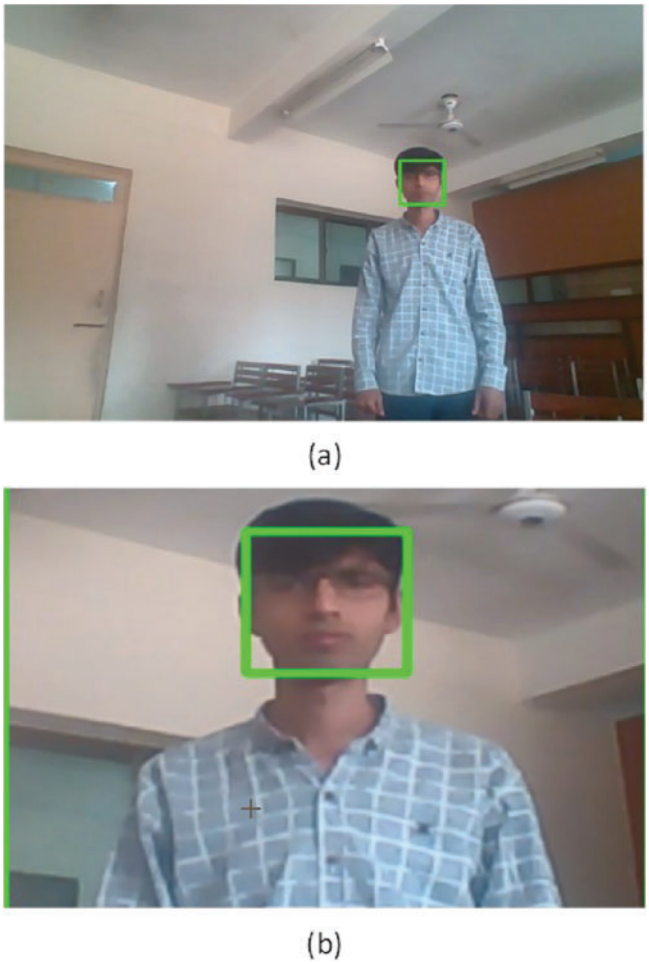

Figure 9: Detection of subject's position from (a) raw frame to (b) processed frame

The proposed approach was validated for a maximum distance of 183.1 cm between the subject and the camera. The model's performance degraded drastically for a distance greater than 183.1 cm. The overall workflow of the real-time deployment of the proposed HGR model is shown in Fig. 10.

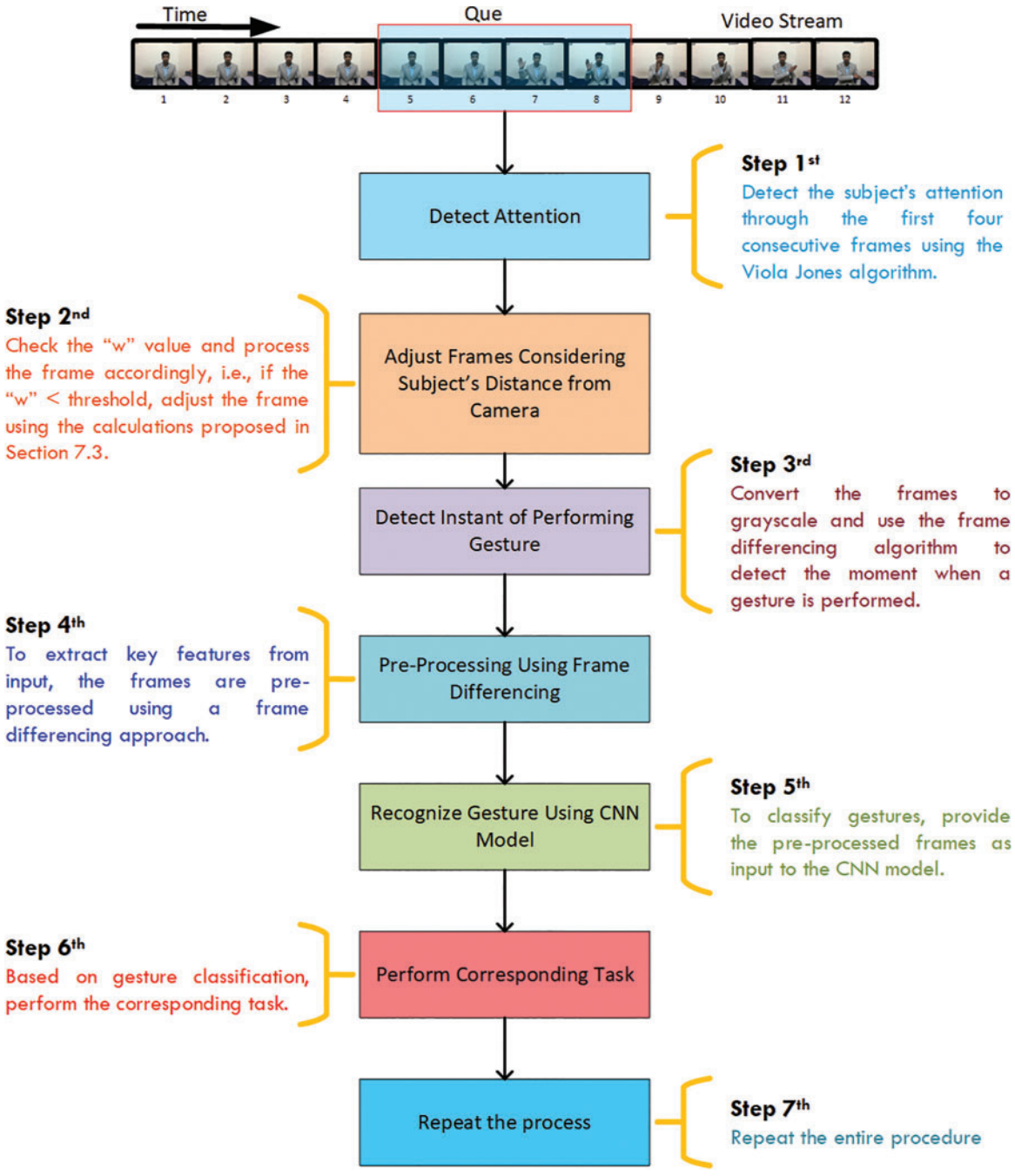

Figure 10: Workflow for the real-time deployment of the hand gesture recognition model

8 Experimental Setup

The proposed model's performance was evaluated and compared with other state-of-the-art models, i.e., 3D Separable CNN [3], 3D CNN [17], and C3D [38]. The other state-of-the-art models, such as P3D [39], I3D [40], and R(2 + 1)D [41], were not considered for comparison because each of these models is comparatively deeper than the others and requires input samples with comparatively high resolution, which leads to high computation cost. The experimental setup used in this study is discussed in the following subsections.

8.1 Training, Validation and Testing

The proposed and comparative models were trained on 25,340 samples, validated on 2512 samples, and tested on 2000 samples. While training the models, the loss for each batch was calculated using the negative log function as defined in (6).

$$Loss_{categorical} = -\sum_{i} t_i \log(y_i) \tag{6}$$

where t_i and y_i denote the true and predicted outputs, respectively.

The loss optimization for each model was performed using the ADAM optimizer [42], except for the C3D [38]. The C3D model could not be optimized using the ADAM optimizer. For this reason, the gradient decent optimizer was used in the case of C3D with a momentum of 0.9. The training and validation procedures were run for 20 epochs with a batch size of 6 and a learning rate of 10^{-4} to 10^{-6}. We trained the models for 20 epochs to ensure that they achieve their optimum performance, even though we were able to attain the best validation accuracy at or before 15 epochs, as shown in Fig. 11. In addition, the optimization functions and hyperparameters such as batch size, epochs, and learning rate were determined through a trial-and-error approach until the best results were achieved.

8.2 Real-Time Testing

The real-time evaluation of the HGR models was performed on a total of 360 samples at distances of 75.3, 123.7, and 183.1 cm from the camera. The performing gestures were randomly generated on a computer screen at each individual distance with the same frequency. The real-time samples were acquired using a 2D computer camera, followed by the approach discussed in Section 7. The state-of-the-art and the proposed models were loaded in parallel so that each model obtained the same input. For comparison, the models were also evaluated without distance calculation at distances of 123.7 and 183.1 cm from the camera. Since 75.3 cm is the usual distance, therefore no distance calculation was performed at this distance. The distance was measured with a laser distance meter LDM-60, as shown in Fig. 12.

8.3 Evaluation Metrics

The models were evaluated based on the recognition accuracy, computational cost, computation time, and the impact of the distance between the subject and the camera on recognition accuracy. The recognition accuracy of the models was evaluated using the average accuracy, precision, recall, and F1 score. In addition, the confusion matrices were used to obtain the individual class's recognition accuracy. The computational costs of various models were compared based on floating-point operands (FLOPs), number of parameters, computation time using GPU, and the model's size.

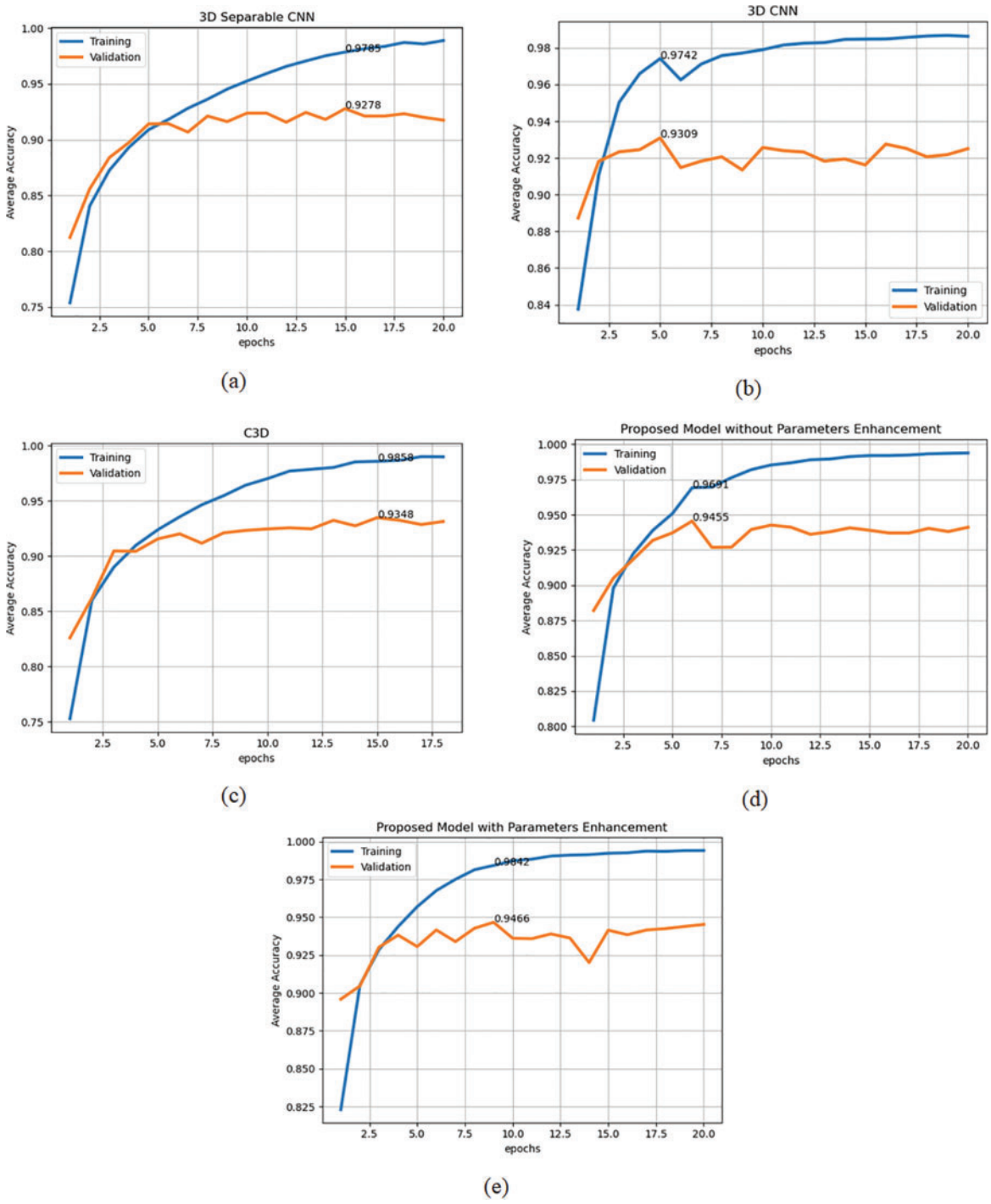

Figure 11: Training and validation accuracy of the (a) 3D separable CNN, (b) 3D CNN, (c) C3D, (d) proposed model without parameters enhancement, and (e) proposed model with parameters enhancement

LDM-60 Specification	
Measurement Range	0.05~60 meter
Measurement Accuracy	$\pm 1.5mm$
Laser Classification	Class II
Laser Specification	635nm, P<1mW

Figure 12: Laser distance meter

Furthermore, the relevancy of the model was tested using the MCC. The MCC ranges from -1 to $+1$, where a value near ± 1 denotes the best agreement and disagreement, while 0 denotes the random predictions according to the actual outcome. We preferred MCC over the Cohen's Kappa and Brier Score as it is observed to be more informative [43]. In addition, we also performed the significance test using the McNemar [44]. The respective approach utilizes a 2×2 contingency table, as shown in Fig. 13. The elements in Fig. 13 provide the following information:

1. The first element "a" denotes the number of samples that are correctly predicted by the models under consideration,
2. The second element "b" denotes the number of samples that are correctly predicted by the first model but are incorrectly predicted by the second,
3. The third element "c" denotes the number of samples predicted correctly by the second model but incorrectly predicted by the first, and,
4. The last element "d" denotes the number of samples that are wrongly predicted by the models under consideration.

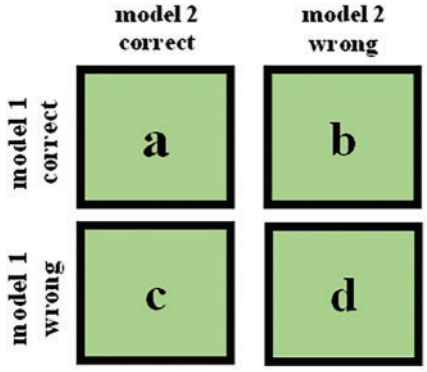

Figure 13: Contingency table for analyzing the significance between two models

The model's significance can be computed using the formula given (7).

$$\chi^2 = \frac{(|b-c|-1)^2}{b+c} \tag{7}$$

where χ^2 is the McNemar's test statistics (chi-squared), while b and c are the number of samples predicted correctly by the models 1 and 2, respectively.

The significance of the proposed model was calculated with a 95% confidence interval ($\alpha = 0.05$) and a threshold of 2.015, i.e., if $\chi^2 > 2.015$, then the proposed model is significantly better than the model under consideration.

9 Analysis of the Proposed Model Behavior

The enhancement of the proposed model in terms of generalization, feature extraction, and the model's parameters as discussed in Section 6 was analyzed in comparison to the base model [3] using a subset of the 20-BN Jester dataset containing 4200 training and 820 validation samples. The hyperparameters were the same as discussed in Section 8. The response of the proposed model for each configuration is described in the following subsections.

9.1 Enhanced Generalization

The response of the proposed model with enhanced generalization in comparison to the base model is shown in Fig. 14. It is observed that the difference between the training and validation accuracy is reduced for the proposed model as compared to the base model and hence the model's generalization is improved. A model with better generalization has a test accuracy closer to that of training accuracy [45], and therefore it is more desirable.

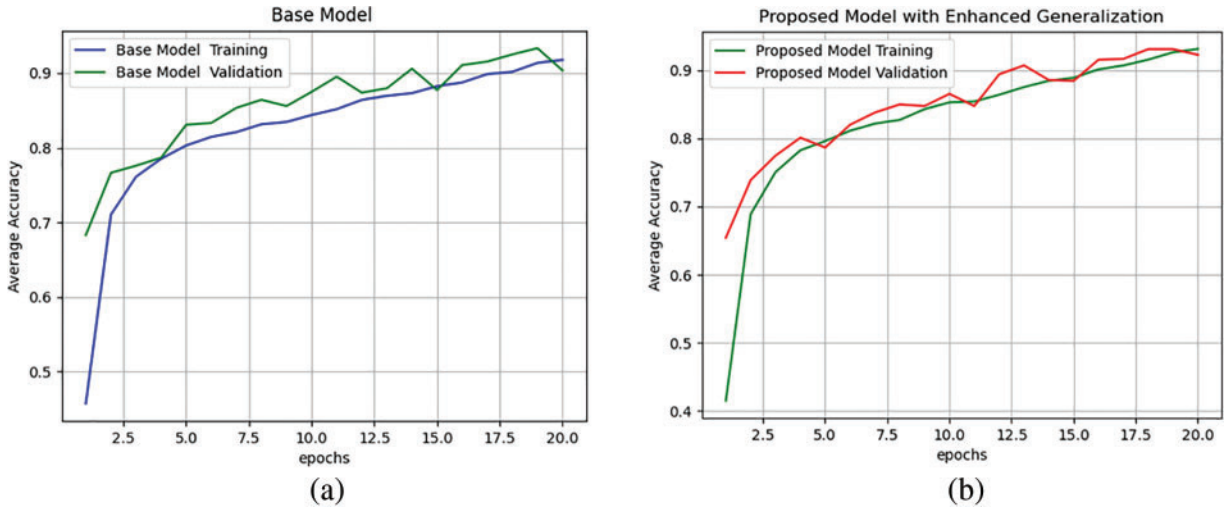

Figure 14: Analysis of model's generalization of (a) the base model, and (b) proposed model

9.2 Enhanced Feature Extraction

The enhancement of the model's generalization has resulted in a low convergence rate, as can be observed from the gradient plots in Fig. 15. The rate of gradient convergence is related to gradient dispersion. The gradient dispersion results in the gradient value being very low for the layers near the input and very high for the layers near the output. In the case of gradient dispersion, the shallow layers require the learning rate to be large, whereas the deeper layers require the learning rate to be small. If this fact is ignored, then the model will not be well-trained. We observed from Fig. 15b that the model with greater gradient dispersion has low convergence rate in comparison to the base model in

Fig. 15a. To improve the model's convergence rate, the feature extraction capability of the proposed model was enhanced, as discussed in Section 6.2. The response of the respective arrangements is shown in Fig. 15c. We observed that the rate of convergence of the proposed model with enhanced feature extraction was far better than the base model and the proposed model with enhanced generalization. The extraction of efficient features from the input data not only improves the convergence rate but also ensures promising results for real-time testing.

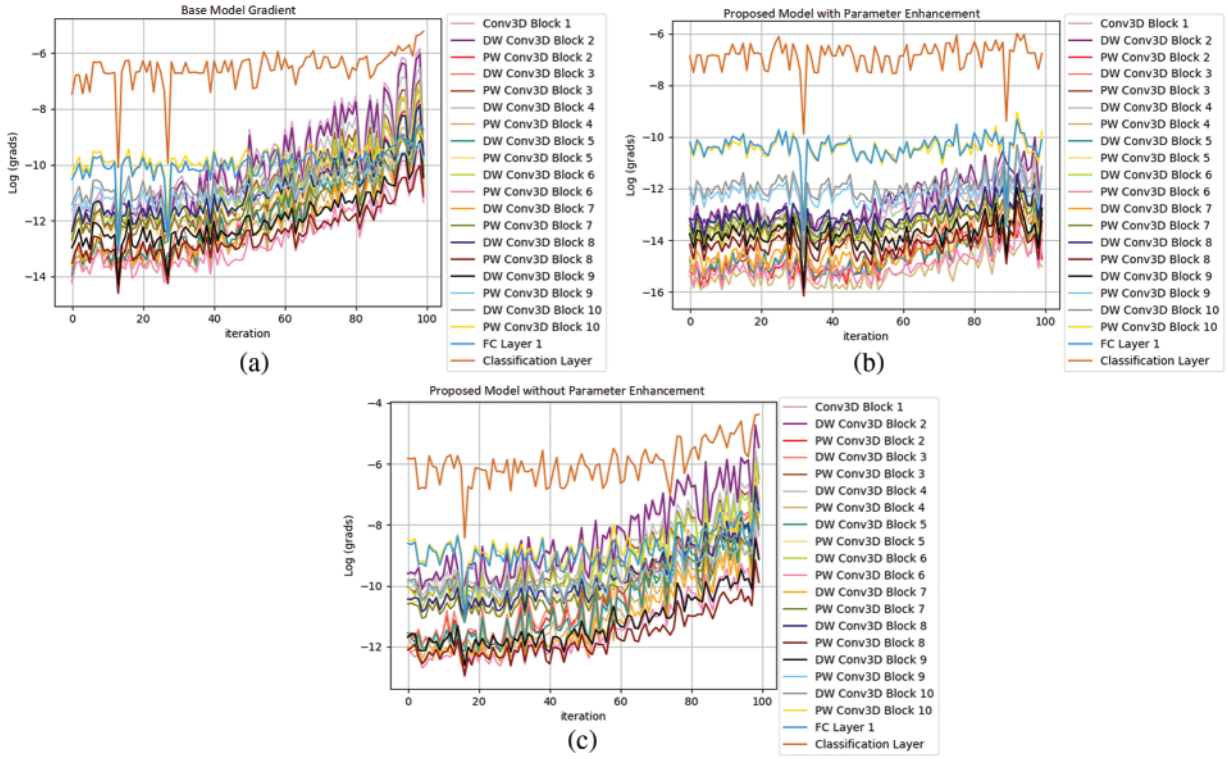

Figure 15: Gradient plots of 3D convolution layers, including depth-wise and point-wise convolution of (a) base model, (b) proposed model with enhanced generalization, and (c) proposed model with enhanced feature extraction

9.3 Model's Parameters Enhancement

The proposed model without and with parameter enhancement was trained and tested on the dataset as mentioned in Section 8.1. The performance of the proposed model was compared with that of other state-of-the-art models, i.e., 3D separable CNN, 3D CNN, and C3D. These models provide better recognition accuracy for the HGR. However, the majority of existing models ignore the computational cost and real-time deployment of models on devices with limited resources. In this research, the proposed model and other state-of-the-art models were evaluated in both offline and real-time scenarios.

10 Experimental Results

10.1 Offline Testing

The offline testing of the proposed model and different state-of-the-art models was performed using 2000 samples of the 20BN-Jesture dataset. The performance of the models in terms of different metrics is given in Table 1. We observed that each of the models demonstrated a correlation between the predicted and actual outcome because their MCC value was closer to +1 except for the 3D Separable CNN. In addition, the statistical analysis showed that the proposed model was significantly better than other state-of-the-art models as the chi squared value was greater than the threshold, i.e., 2.015. The contingency tables for the respective analyses are shown in Fig. 16. We observed that the proposed model with enhanced parameters had high significance when there was a considerable difference between b and c, i.e., b was greater than c, as in Figs. 16a–16c. Whereas, if the difference between b and c was not sufficient enough, as in the case of Fig. 16d, then in various situations, one of the two models may lead or perform similarly. The models' performance in Table 1 also shows agreeable results to those of statistical analysis. We observed that the proposed model without parameter enhancement provided better classification performance, and an average accuracy of 93.20% and an F1 score of 93.25% were obtained. The performance of the proposed model with parameter enhancement was slightly lower than that of the one without parameter enhancement. The other state-of-the-art models lagged in performance, and F1 scores of 56.36%, 90.84%, and 91.29% were obtained for the 3D Separable CNN, 3D CNN, and C3D, respectively. The 3D Separable CNN had a validation accuracy of 92.78%, as shown in Fig. 11a, but the test results showed that the model was significantly overfitted and was unable to extract the key features from the input samples, as can be observed from Fig. 7a. The confusion matrices in Fig. 17 show the average accuracy of the models for each class. We observed that the proposed model had resulted in comparatively better accuracy for each gesture class. However, when considering the class-wise performance, lower classification performance was achieved for the swiping up gesture for all models. This is due to the similarity in features between the swiping up and the swiping down gestures, as shown in Fig. 18. We observed that while performing the swiping up and swiping down gestures, the subject has to move his or her hand from the bottom to the top side of the frame, due to which these classes are mostly confused with each other and hence lead to poor accuracy.

Table 1: Comparison of different hand gesture recognition models in the offline scenario

Models	Accuracy	Precision	Recall	F1	MCC	McNemar
3D separable CNN [3]	0.6115	0.6935	0.6115	0.5635	0.5493	541.03
3D CNN [17]	0.9085	0.9086	0.9085	0.9084	0.8902	5.39
C3D [38]	0.9130	0.9130	0.9130	0.9129	0.8956	3.58
Proposed model without parameters enhancement	0.9320	0.9344	0.9320	0.9325	0.9187	1.43
Proposed model with parameters enhancement	0.9245	0.9258	0.9245	0.9248	0.9095	–

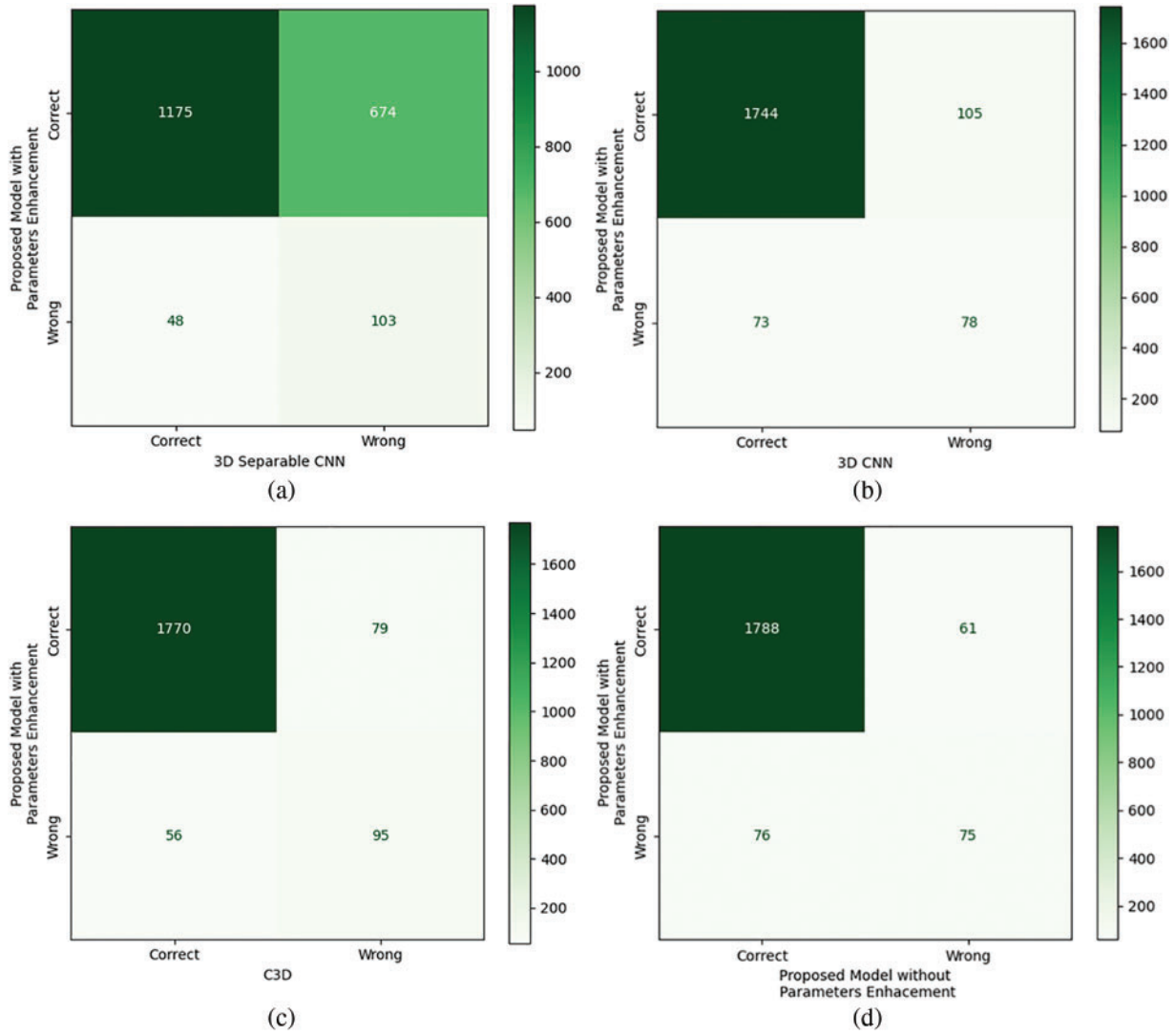

Figure 16: Contingency table obtained from the statistical analysis of the proposed model with parameters enhancement, and (a) 3D separable CNN, (b) 3D CNN, (c) C3D, and (d) proposed model without parameters enhancement

In addition, the average accuracy attained by different models using the 20-BN Jester dataset is shown in Table 2 for comparison purposes. Other than the proposed model, the models in Table 2 were trained and tested on a complete dataset of 27 gesture classes. Note that the comparison of the proposed work with the previous models is not the objective of this study due to the different scenarios. However, we can observe that some of the models are comparatively better than the proposed model, but the real-time deployment of these models is limited due to their comparatively higher computational complexity.

In the next step, the models were tested in real-time with and without incorporating the distance calculation, and the results were analyzed.

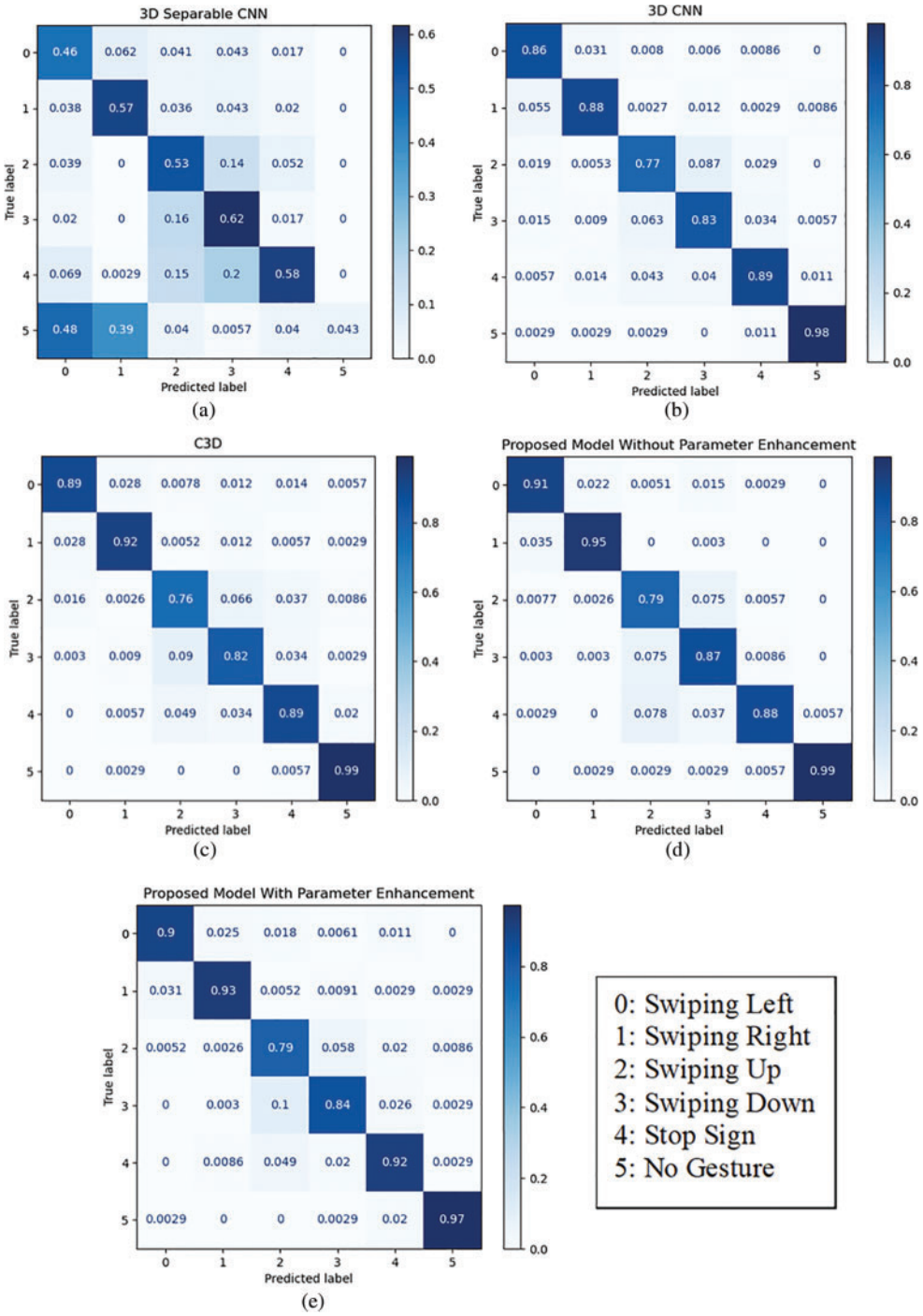

Figure 17: Confusion matrices of the offline testing for the (a) 3D separable CNN, (b) 3D CNN, (c) C3D, (d) proposed model without parameters enhancement, and (e) proposed model with parameters enhancement

Figure 18: Gestures exhibiting similar features: (a) swiping up, and (b) swiping down

10.2 Real-Time Testing

The real-time performance of the proposed models and other state-of-the-art models for three distinct positions of the subject from the camera are given in Table 3. Based on the distance between the subject and the camera, the following observations are made:

1. The proposed model with parameter enhancement at a usual distance of 75.3 cm achieved a comparatively better recognition accuracy of 91.66%, which was followed by the proposed model without parameter enhancement and 3D CNN with recognition accuracies of 90.83% and 90.00%, respectively.

2. At distances greater than the usual, i.e., 75.3 cm, the performance of all the HGR models drastically degrades without the distance calculation. At a distance of 123.7 cm, the best classification performance of 67.50% was obtained for the proposed model with parameter enhancement, which was followed by the proposed model without parameter enhancement and the 3D CNN with recognition accuracies of 63.33% and 55.83%, respectively. Increasing the distance to 183.1 cm further degraded the performance of all models, and the proposed model without parameter enhancement provided the best accuracy of 55.00%. The 3D Separable CNN performed poorly for both distances.
3. The inclusion of the distance calculation has substantially improved the overall performance of all the HGR models. At a distance of 123.7 cm, the best accuracy of 95.00% was obtained for the proposed model with parameter enhancement, which was followed by the proposed model without parameter enhancement and the 3D CNN with classification accuracies of 88.33% and 82.50%, respectively. Increasing the distance to 183.1 cm degraded the performance of all models, and the best accuracy of 89.16% was obtained for the proposed model with parameters enhancement.

Table 2: Average accuracy of various models over the 20-BN Jester dataset

Models	Average accuracy
ResNet101 [19]	97.0%
3D CNN [17]	90.0%
PAN ResNet101 [46]	97.4%
X3D MobileNet-V3 [47]	95.56
3D-Squeeze Net [48]	90.77%
3D-Shuffle Net V2 [48]	86.91%
3D Mobile Net V2 [48]	86.43%
Proposed model without parameters enhancement	93.20%
Proposed model with parameters enhancement	92.45%

In general, the proposed model with parameter enhancement provided the best performance in all scenarios, followed by the proposed model without parameter enhancement and the 3D CNN. It was also observed that the 3D Separable CNN performed poorly due to its inefficient feature extraction capability, leading to a high rate of overfitting. The confusion matrices for the HGR models at each distinct position mentioned in Table 3 are provided in Figs. 19 to 23, respectively. We can observe from Figs. 19 to 21 that increasing the subject's distance from the camera results in an abrupt degradation of the model's performance without the proposed distance calculation. On the other hand, the results in Figs. 22 and 23 show a remarkable improvement in the performance as the proposed distance calculation prevents the blending of the subject's feature into the background. The results indicate that the performance of each HGR model was remarkably improved due to the inclusion of the proposed distance calculation.

Table 3: Comparison of different hand gesture recognition models for the real-time scenario with variable distance between the subject and the camera without and with the inclusion of the distance calculation

Models	Distance between subject and camera (cm)	Accuracy	Precision	Recall	F1
Real-time model testing at a normal distance					
3D separable CNN [3]		0.5916	0.5046	0.5916	0.5337
3D CNN [17]		0.9000	0.9119	0.9000	0.8985
C3D [36]	75.3	0.8833	0.9024	0.8833	0.8852
Proposed model without parameters enhancement		0.9083	0.9169	0.9083	0.9074
Proposed model with parameters enhancement		0.9166	0.9276	0.9166	0.9164
Real time model testing without inclusion of distance calculation					
3D separable CNN [3]		0.4500	0.3436	0.4500	0.3751
3D CNN [17]		0.5583	0.5986	0.5583	0.5300
C3D [36]	123.7	0.5583	0.4695	0.5583	0.4724
Proposed model without parameters enhancement		0.6333	0.7573	0.6333	0.6039
Proposed model with parameters enhancement		0.6750	0.8796	0.6750	0.6835
3D separable CNN [3]		0.3833	0.4988	0.3833	0.3197
3D CNN [17]		0.3250	0.2086	0.3250	0.2342
C3D [36]	183.1	0.4833	0.4615	0.4833	0.3720
Proposed model without parameters enhancement		0.5500	0.6029	0.5500	0.4672
Proposed model with parameters enhancement		0.5000	0.4988	0.5000	0.3908
Real time model testing with inclusion of distance calculation					
3D separable CNN [3]		0.5166	0.4440	0.5166	0.4726
3D CNN [17]		0.8250	0.8774	0.8250	0.8119
C3D [36]	123.7	0.7000	0.8744	0.7000	0.6974
Proposed model without parameters enhancement		0.8833	0.9093	0.8833	0.8844
Proposed model with parameters enhancement		0.9500	0.9522	0.9500	0.9499
3D separable CNN [3]		0.3833	0.2695	0.3833	0.3023
3D CNN [17]		0.6250	0.6000	0.6250	0.5834
C3D[36]	183.1	0.5916	0.6626	0.5916	0.0.549
Proposed model without parameters enhancement		0.7583	0.8638	0.7583	0.7536
Proposed model with parameters enhancement		0.8916	0.8992	0.8916	0.8930

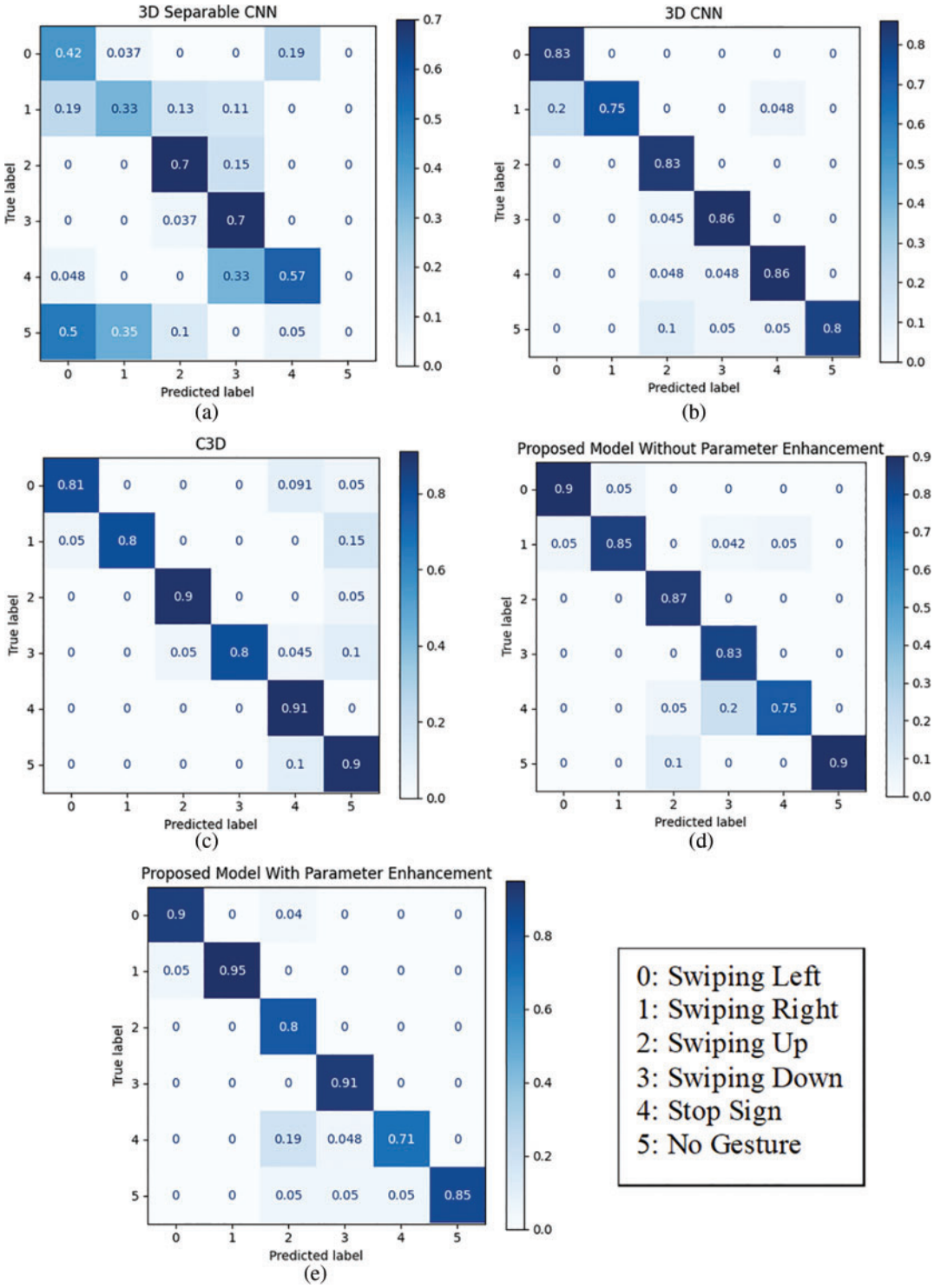

Figure 19: Confusion matrices of the real-time testing at a distance of 75.3 cm for the (a) 3D separable CNN, (b) 3D CNN, (c) C3D, (d) proposed model without parameters enhancement, and (e) proposed model with parameters enhancement

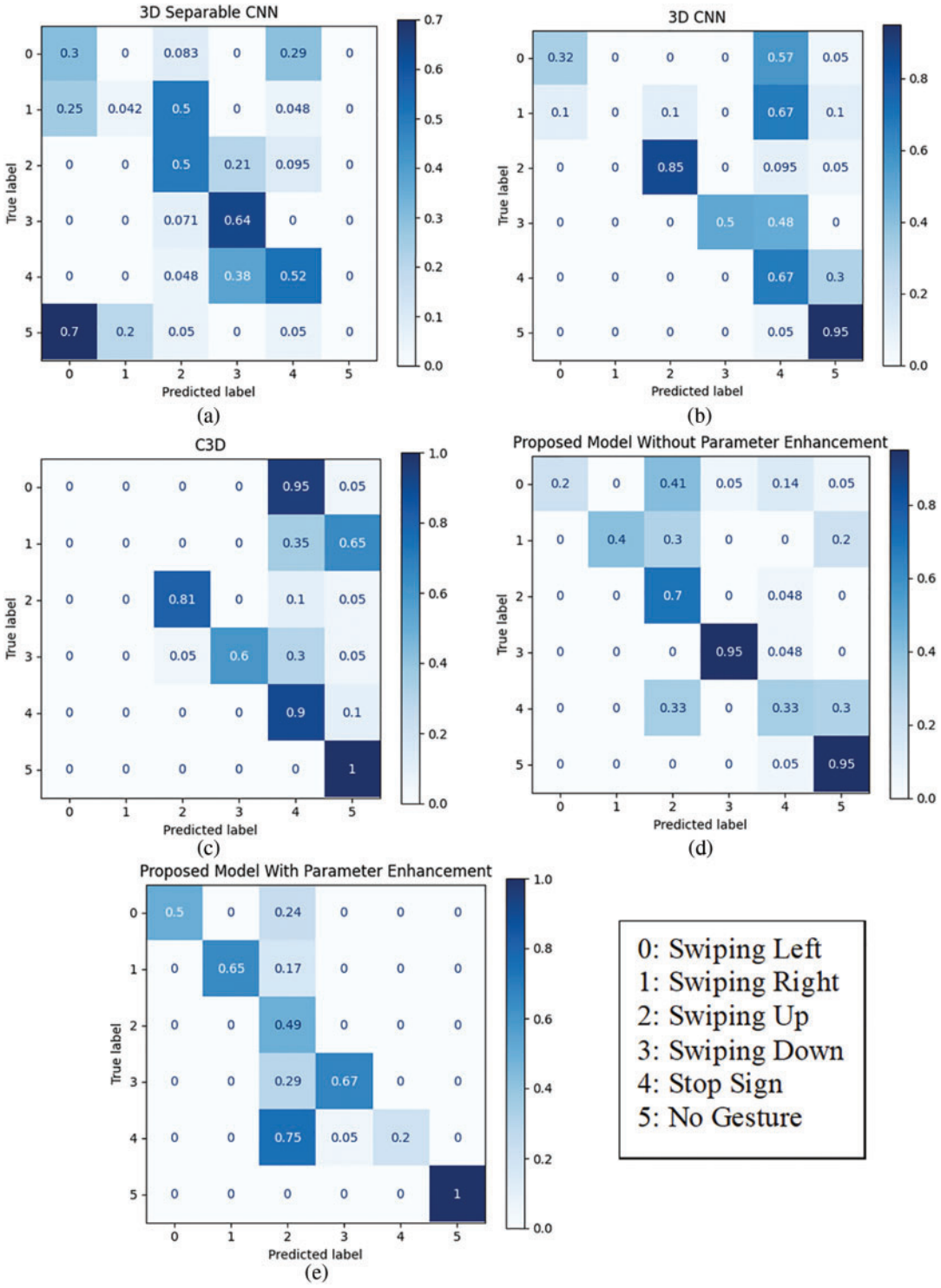

Figure 20: Confusion matrices of the real-time testing at a distance of 123.7 cm without distance calculation for the (a) 3D separable CNN, (b) 3D CNN, (c) C3D, (d) proposed model without parameters enhancement, and (e) proposed model with parameters enhancement

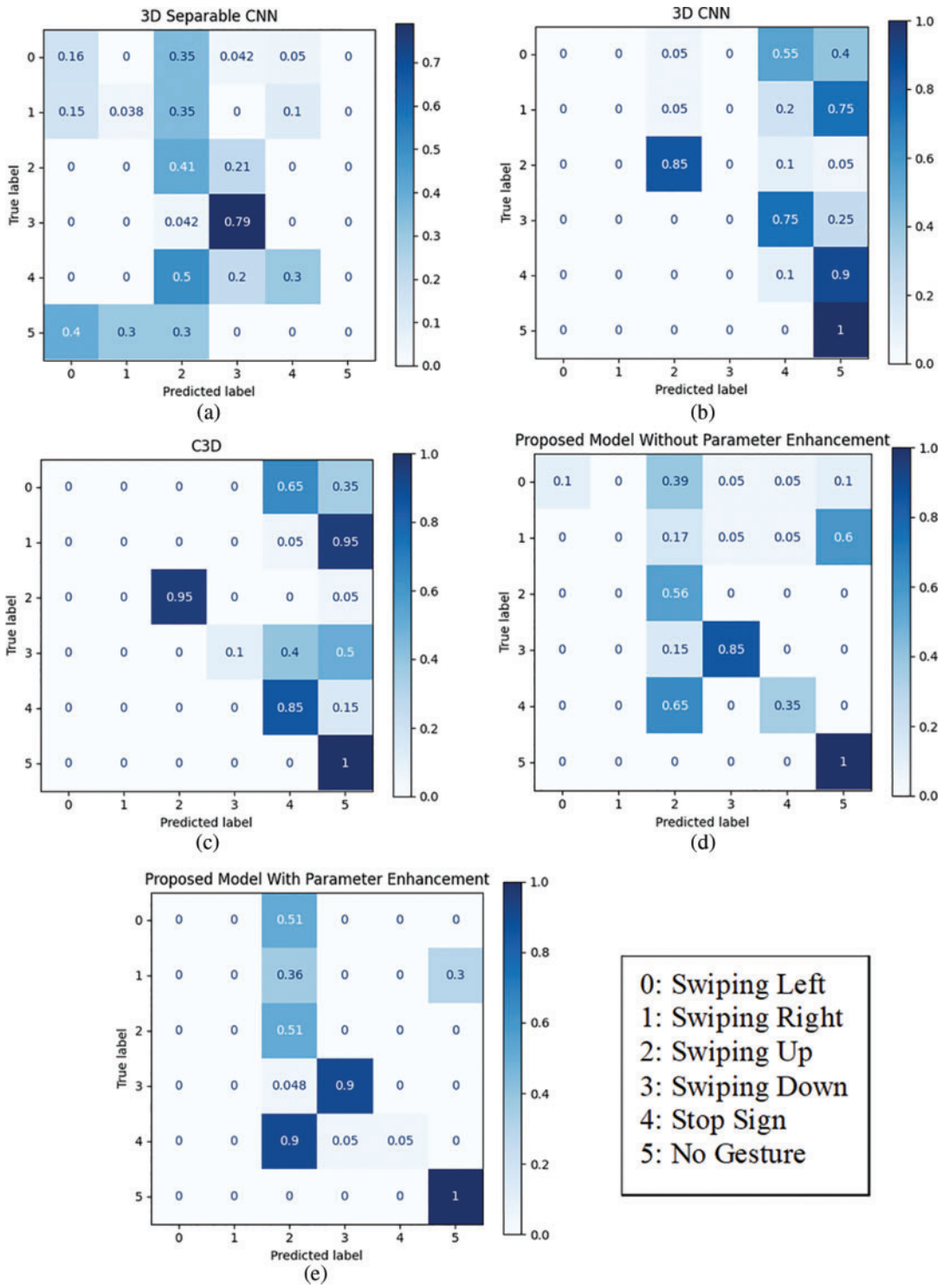

Figure 21: Confusion matrices of the real-time testing at a distance of 183.1 cm without distance calculation for the (a) 3D separable CNN, (b) 3D CNN, (c) C3D, (d) proposed model without parameters enhancement, and (e) proposed model with parameters enhancement

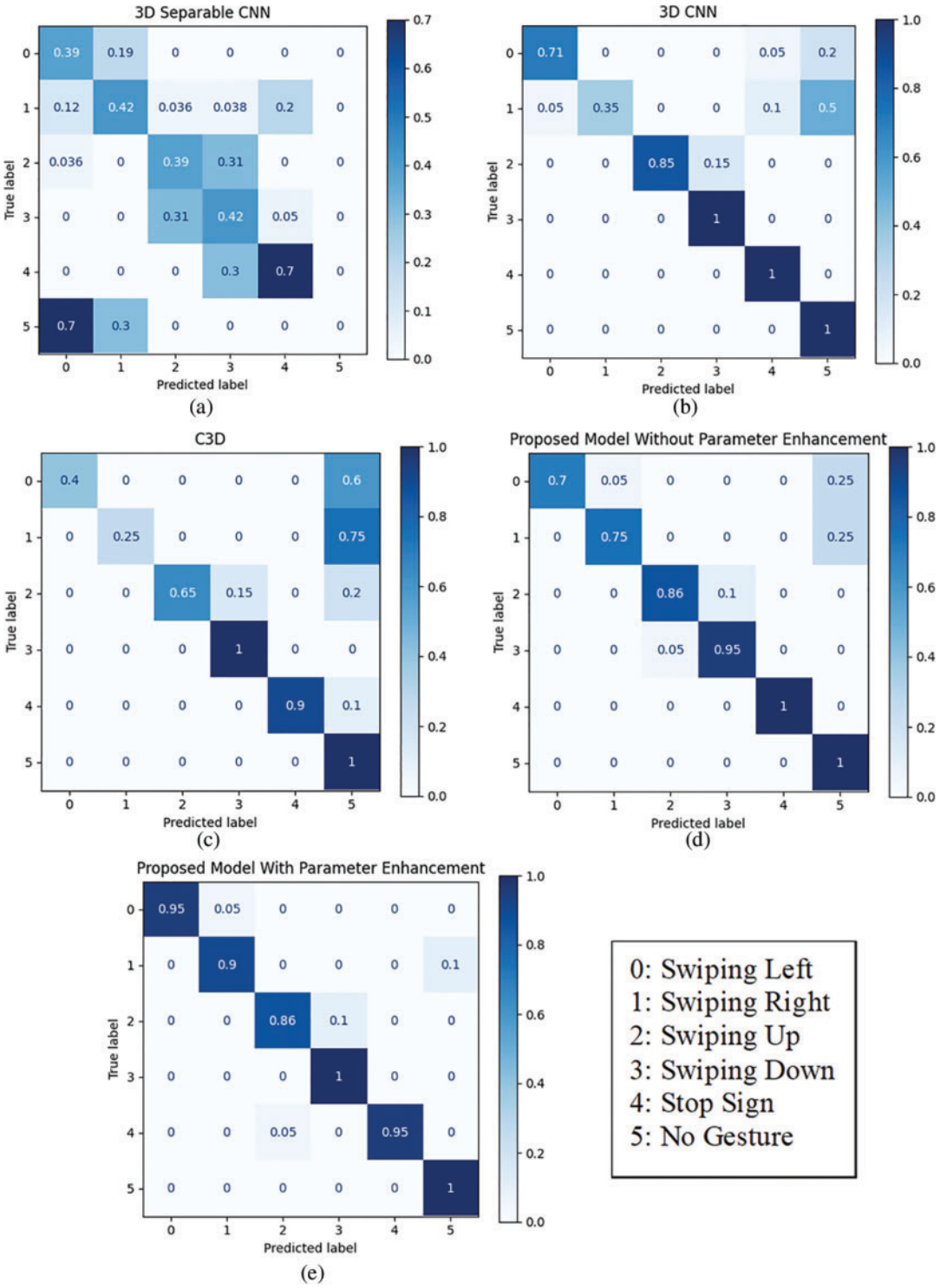

Figure 22: Confusion matrices of the real-time testing at a distance of 123.7 cm with distance calculation for the (a) 3D separable CNN, (b) 3D CNN, (c) C3D, (d) proposed model without parameters enhancement, and (e) proposed model with parameters enhancement

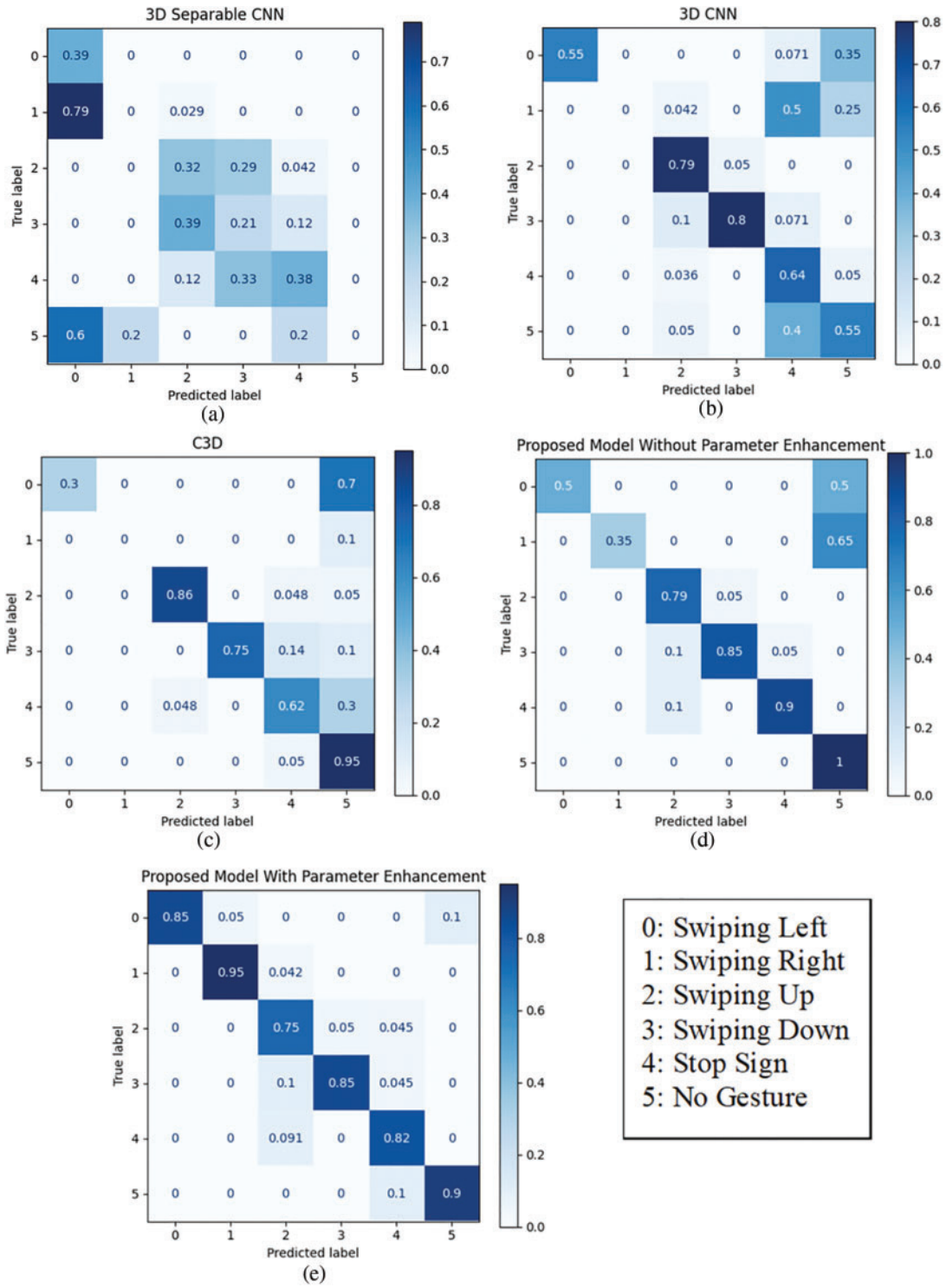

Figure 23: Confusion matrices of the real-time testing at a distance of 183.1 cm with distance calculation for the (a) 3D separable CNN, (b) 3D CNN, (c) C3D, (d) proposed model without parameters enhancement, and (e) proposed model with parameters enhancement

10.3 Computational Complexity

The computational complexity of a model significantly depends on its architectural design and input dimensions. The proposed approach is based on a 3D separable CNN [3], which has less computational complexity than the other existing models. However, in our case, the 3D separable CNN performed poorly in all scenarios. In the proposed approach, the computation parameters and FLOPs were reduced by decreasing the input dimensions from $8 \times 128 \times 128 \times 1$ [3] to $8 \times 100 \times 130 \times 1$. A comparative analysis of the FLOPs, model parameters, computation time, and model size for the proposed model and other models is presented in Table 4. The computation time for each model was calculated over 1000 randomly generated samples having a dimension of $8 \times 100 \times 130 \times 1$. From Table 4, we observed that the FLOPs of the model were directly related to the computation time. The 3D separable CNN [3] had the least number of FLOPs but performed poorly in all scenarios. The computational time of the proposed model is less than 100 ms, which is the maximum tolerable computation time for HGR [15].

Table 4: Computational complexity of the different hand gesture recognition models

Models	MFLOPs	MParameters	Computation time (msec.)	Model size (MB)
3D separable CNN [3]	499	1.1	47.7	4.19
3D CNN [17]	9479	9.03	82.5	34.4
C3D [36]	32496	52.85	209.6	201
Proposed model without parameters enhancement	616	1.1	68.0	4.19
Proposed model with parameters enhancement	591	1.36	62.5	5.19

11 Discussion

The proposed study developed two models for the HGR. The proposed HGR model without parameter enhancement is computationally more complex than the one with parameter enhancement. The McNemar test showed no significant difference between the two models. In the offline scenario, the proposed model outperformed other state-of-the-art models in terms of accuracy, precision, recall, and F1 score. Besides the offline scenario, the models were tested for real-time scenarios as well. The real-time tests were conducted at three distinct positions, both without and with consideration of the proposed distance calculation. The test results showed that the proposed model achieved better performance than other state-of-the-art models. In addition, we observed that considering the proposed distance calculation significantly improved the model's performance and hence ensures the real-time deployment of the model at a maximum distance of 183.1 cm from the camera. In the case where the distance exceeded 183.1 cm, the model's performance degraded as the subject's features began to blend with the frame's background. The 3D Separable CNN performed poorly in all scenarios due to its inefficient feature extraction capability.

The architecture design of the proposed model, besides the consideration of performance, also considers the model's computational complexity. Other than the 3D Separable CNN, the proposed model was observed to have comparatively less computational complexity. The computational

complexity of the model highly impacts the real-time performance, i.e., computationally complex models are unable to attain the desired outcome in time, and vice versa.

12 Conclusion

The proposed study developed a deep learning architecture for HGR with comparatively enhanced generalization and feature extraction capabilities while utilizing less computational resources. The comparative results in the offline scenario showed that the proposed model outperformed other state-of-the-art models in terms of accuracy, precision, recall, and F1 score. In addition, the evolution results showed higher significance for the proposed model in comparison to other state-of-the-art models. Furthermore, a novel approach was proposed for the real-time implementation of the HGR model while considering the subject's attention, the instant of performing a gesture, and the subject's distance from the camera. The real-time performance of the proposed model and other state-of-the-art models was evaluated at the distances of 75.3, 123.7, and 183.1 cm, with and without the consideration of the distance factor. The evaluation results showed that avoiding the distance factor resulted in significantly lower model's performance. The inclusion of the proposed distance calculation substantially improved the performance of all the HGR models. Based on the experimental results, we can conclude that the proposed model resulted in state-of-the-art performance for both the offline and real-time scenarios.

The proposed approach is defined for a single subject within the camera's vision range with a static background. The prediction of the proposed model will be randomized if there are two or more subjects in view of the camera or if the computational device is mobile because the model will not be able to extract the key gesture features. In the future, we aim to develop an approach that can be used for multiple subjects with a dynamic background.

Funding Statement: The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication. They would also like to thank Prince Sultan University for its support.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. S. Al-Shamayleh, R. Ahmad, M. A. M., Abusharia, K. A. Alam and N. Jomhari, "A systematic literature review on vision based gesture recognition techniques," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28121–28184, 2018.

[2] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif *et al.,* "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192527–192542, 2020.

[3] Z. Hu, Y. Hu, J. Liu, B. Wu, D. Han *et al.,* "3D separable convolutional neural network for dynamic hand gesture," *Neurocomputing*, vol. 318, no. 1, pp. 151–161, 2018.

[4] P. Premaratne, "Historical development of hand gesture recognition," in *Human Computer Interaction Using Hand Gestures*, New York: Springer, pp. 5–28, 2014.

[5] D. A. Nicora, "Microsoft kinect," Patent US 9440134B2, 13 Sep. 2016.

[6] D. H. Zhang and R. Hicks, "Depth sensing auto focus multiple camera system," Patent US 828964, 23 Feb 2017.

[7] C. Stoppel, A. Buettner, P. Carve and R. J. Schwarz, "Lidar sensor," Patent WO2017045816A1, 23 March 2017.

[8] D. Holz, "Systems and methods for capturing motion in three-dimensional space," Patent US 9153028B2, 28 Jan 2014.
[9] J. Farooq and M. B. Ali, "Real time hand gesture recognition for computer interaction," in *Int. Conf. on Robotics and Emerging Allied Technologies in Engineering*, Islamabad, Pakistan, pp. 73–77, 2014.
[10] J. L. Raheja, M. Minhas, D. Prashanth, T. Shah and A. Chaudhary, "Robust gesture recognition using kinect: A comparison between DTW and HMM," *Optik*, vol. 126, no. 12, pp. 1098–1104, 2015.
[11] J. Singha, A. Roy and R. H. Laskar, "Dynamic hand gesture recognition using vision-based approach for human computer interaction," *Neural Computing and Applications*, vol. 29, no. 4, pp. 1129–1141, 2016.
[12] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[13] F. Chollet, "What is deep learning?," in *Deep Learning with Python*, 1st ed., Shelter Island, New York, United States: Manning Publication Co., pp. 17, 2018.
[14] J. Rehg and T. Kanade, "DigitEyes: Vision-based hand tracking for human-computer interaction," in *Proc. of 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, USA, vol. 43, no. 1, pp. 16–22, 1994.
[15] P. Molchanov, S. Gupta, K. Kim and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Boston, Massachusetts, USA, pp. 1–7, 2015.
[16] S. Hussain, R. Saxena, X. Han and J. A. Khan, "Hand gesture recognition using deep learning," in *IEEE Int. SoC Design Conf.*, Seoul, South Korea, pp. 48–49, 2017.
[17] W. Zhang and J. Wang, "Dynamic hand gesture recognition based on 3D convolutional neural network models," in *IEEE Int. Conf. on Networking, Sensing and Control*, Banff, Alberta, Canada, pp. 224–229, 2019.
[18] S. S. Kakkoth and S. Gharge, "Survey on real time hand gesture recognition," in *Int. Conf. on Current Trends in Computer, Electrical, Electronics and Communication*, Mysore, India, pp. 948–954, 2017.
[19] O. Kopuklu and N. Kose, "Online dynamic hand gesture recognition including efficiency analysis," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 2, pp. 85–97, 2020.
[20] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770–778, 2016.
[21] D. Sarma and M. B. V. Kavyasree, "Two-stream fusion model for dynamic hand gesture recognition using 3D-CNN and 2D-CNN optical flow guided motion template," *Innovations in Systems and Software Engineering*, vol. 18, no. 4, pp. 1–14, 2022.
[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang *et al.,* "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017. https://arxiv.org/pdf/1704.04861.pdf
[23] X. Zhang, X. Zhou, M. Lin and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp. 6848–6856, 2018.
[24] M. Saqib, S. D. Khan, N. Sharma and M. Blumenstein, "Person head detection in multiple scales using deep convolutional neural networks," in *Int. Joint Conf. on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, pp. 1–7, 2018.
[25] S. D. Khan, Y. Ali, B. Zafar and A. Noorwali, "Robust head detection in complex videos using two-stage deep convolution framework," *IEEE Access*, vol. 8, no. 1, pp. 98679–98692, 2020.
[26] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
[27] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, pp. 511–518, 2001.
[28] J. Materzynska, G. Berger, I. Bax and R. Memisevic, "The Jester dataset: A large-scale video dataset of human gesture," in *IEEE/CVF Int. Conf. on Computer Vision Workshop*, Seoul, South Korea, pp. 2874–2882, 2019.

[29] M. A. R. Ahad, J. K. Tan, H. Kim and S. Ishikawa, "Motion history image: Its variants and applications," *Machine Vision and Applications*, vol. 23, no. 1, pp. 255–281, 2010.
[30] B. Lekshmi and T. Safuvan, "Optical flow based real-time moving object detection in unconstrained scenes," *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 17, pp. 1–5, 2016.
[31] S. D. Khan, A. B. Altamimi, M. Ullah, H. Ullah and F. A. Cheikh, "TCM: Temporal consistency model for head detection in complex videos," *Journal of Sensors*, vol. 1, pp. 1–13, 2020.
[32] Y. Zhanga, X. Wang and B. Qu, "Three-frame difference algorithm research based on mathematical morphology," *Procedia Engineering*, vol. 29, pp. 2705–2709, 2012.
[33] B. Singh, S. De, Y. Zhang, T. Goldstein and G. Taylor, "Layer-specific adaptive learning rates for deep network," in *IEEE 14th Int. Conf. on Machine Learning and Applications (ICMLA)*, Miami, FL, USA, pp. 364–368, 2015.
[34] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 1800–1807, 2017.
[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2818–2826, 2015.
[36] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
[37] X. Sun, P. Wu and S. C. Hoi, "Face detection using deep learning: An improved faster RCNN approach," *Neurocomputing*, vol. 299, pp. 42–50, 2018.
[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3D convolutional network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 4489–4497, 2015.
[39] Z. Qiu, T. Yao and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 5533–5541, 2017.
[40] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 4724–4733, 2017.
[41] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun *et al.,* "A closer look at spatiotemporal convolutions for action recognition," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 6450–6459, 2018.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. https://arxiv.org/pdf/1412.6980.pdf
[43] D. Chicco, M. J. Warrens and G. Jurman, "The Matthews correlation coefficient (MCC) is more informative than Cohen's kappa and brier score in binary classification assessment," *IEEE Access*, vol. 9, no. 1, pp. 78368–78381, 2021.
[44] A. L. Edwards, "Note on the "correction for continuity" in testing the significance of the difference between correlated proportions," *Psychometrika*, vol. 13, no. 1, pp. 185–187, 1948.
[45] P. Kim, "Overfitting," in *Matlab Deep Learning*, 1st ed., New York, United States: Apress, pp. 6, 2017.
[46] C. Zhang, Y. Zou, G. Chen and L. Gan, "PAN: Persistent appearance network with an efficient motion cue for fast action recognition," in *Proc. of the 27th ACM Int. Conf. on Multimedia*, New York, NY, United States, pp. 500–509, 2019.
[47] E. Izutov, "LIGAR: Lightweight general-purpose action recognition," 2021. https://arxiv.org/pdf/2108.13153.pdf
[48] O. Kopuklu, N. Kose, A. Gunduz and G. Rigoll, "Resource efficient 3D convolutional neural networks," in *IEEE/CVF Int. Conf. on Computer Vision Workshop (ICCVW)*, Seoul, Korea, pp. 1910–1919, 2019.