Computers, Materials & Continua
DOI:10.32604/cmc.2022.019586
Article

Tech Science Press

Dynamic Hand Gesture Recognition Using 3D-CNN and LSTM Networks

Muneeb Ur Rehman[1], Fawad Ahmed[1], Muhammad Attique Khan[2], Usman Tariq[3], Faisal Abdulaziz Alfouzan[4], Nouf M. Alzahrani[5] and Jawad Ahmad[6,*]

[1]Department of Electrical Engineering, HITEC University Taxila, Pakistan
[2]Department of Computer Science, HITEC University Taxila, Pakistan
[3]College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Khraj, Saudi Arabia
[4]Department of Forensic Sciences, College of Criminal Justice, Naif Arab University for Security Sciences, Riyadh, Saudi Arabia
[5]Department of Information Technology, Albaha University, Albaha, Saudi Arabia
[6]School of Computing, Edinburgh Napier University, UK
[*]Corresponding Author: Jawad Ahmad. Email: J.Ahmad@napier.ac.uk

Received: 18 April 2021; Accepted: 27 July 2021

Abstract: Recognition of dynamic hand gestures in real-time is a difficult task because the system can never know when or from where the gesture starts and ends in a video stream. Many researchers have been working on vision-based gesture recognition due to its various applications. This paper proposes a deep learning architecture based on the combination of a 3D Convolutional Neural Network (3D-CNN) and a Long Short-Term Memory (LSTM) network. The proposed architecture extracts spatial-temporal information from video sequences input while avoiding extensive computation. The 3D-CNN is used for the extraction of spectral and spatial features which are then given to the LSTM network through which classification is carried out. The proposed model is a light-weight architecture with only 3.7 million training parameters. The model has been evaluated on 15 classes from the 20BN-jester dataset available publicly. The model was trained on 2000 video-clips per class which were separated into 80% training and 20% validation sets. An accuracy of 99% and 97% was achieved on training and testing data, respectively. We further show that the combination of 3D-CNN with LSTM gives superior results as compared to MobileNetv2 + LSTM.

Keywords: Convolutional neural networks; 3D-CNN; LSTM; spatio-temporal; jester; real-time hand gesture recognition

1 Introduction

Gestures are primary tool of symbolic communication and natural form in which humans express themselves more effectively. They vary from simple to more complex actions which allow us to communicate with others. Due to rapid development in the field of deep learning and computer vision technology, the use of biological characteristics of human beings have become a focus for shifting human-computer interaction from traditional ways to new methods. As the most flexible body part of a human body is hand, therefore, hand gestures can express rich and various form of communication between humans and machines. They are widely used for communication between humans and computers or other electronic devices such as smart phones, robotics, auto-mobile infotainment system, etc. Gesture recognition can replace human-computer interaction from touch or wired-controlled input devices [1].

There are two types of gestures as shown in Fig. 1, static gesture [2], in which there is no change in body pose or arm; only the hand is kept still with some specific pose over time. In the second case, the arm with hand moves and there are a set of poses that vary according to the time interval. This is referred to as dynamic gesture [3]. Hand gesture detection methods generally have three main steps: (i) pre-processing (ii) feature extraction and (iii) gesture recognition. Gesture recognition requires hand movement in a video stream. It is done by first transforming video sequence into frames and then going through feature extraction steps and finally recognizing the hand gestures.

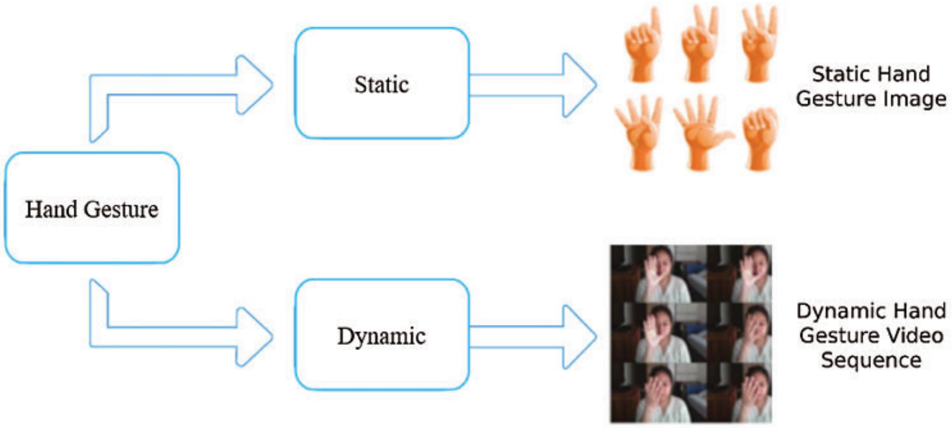

Hand Gesture
Static
Static Hand Gesture Image
Dynamic
Dynamic Hand Gesture Video Sequence

Figure 1: Type of hand gestures

Different sensors have different sensing capabilities. Mostly, a single sensor is used for gesture based interactive technology. Raw data needs to be collected by the sensors before the gesture recognition process start. There can be many other ways to get this data, for example, using a contact-less sensor such as a radar for hand movement detection or a wearable sensor such as a glove, which can measure the pressure applied by the fingers around the wrist [4,5]. Image based approaches mimic the use of eyes to recognize objects in this world. Similarly, robots or human-machine interaction needs cameras to see and recognize things. Initial research in image-based gesture recognition had limitations due to low accuracy, poor real-time recognition and algorithm complexity. With the passage of time, these issues were addressed due to faster computers and advancement in the field of artificial intelligence, especially, deep learning.

Deep Learning (DL) is a developing field and is a sub-category of machine learning inspired by the function of human brain and its structure. It uses multiple hidden neural network layers for better learning of a model. It can learn the features of an object accurately and easily under complex surrounding or background. The Convolutional-Neural-Network (CNN) is a very famous DL model which is used in image-based applications. Nowadays, DL is used in visual object detection and recognition, speech recognition and many other applications [6–8]. In recent years, many hand gesture recognition methods have been proposed using deep learning techniques. Fig. 2 shows the basic steps for automatic recognition of hand gestures.

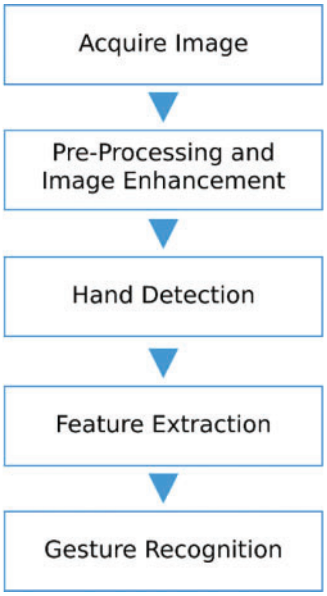


Figure 2: Basic steps for automatic gesture recognition

As shown in Fig. 2, hand gesture recognition process is divided into four steps; image acquisition, image enhancement, hand detection, feature extraction and finally gesture classification. Input images are collected in the form of multiple frames for dynamic gestures, whereas for static gestures, a single image can also be used. Image enhancement techniques are applied to increase the quality of input images. To apply deep learning based gesture recognition, the dataset needs to be large; therefore to enrich the input dataset, data augmentation is employed in this work using scaling, translating, rotating and shearing techniques.

Dynamic gesture recognition falls under the category of video classification since the dataset is mostly available in the form of video frames. Hence both spatial and temporal domains features are used. Dynamic gesture recognition is a difficult task because the images obtained from video recordings does not have consistent pixels; the camera is not fixed at one position and every person performs the same gesture in a different way. Gesture includes different background with hand and arm continuous movement due to which it is not easy for an algorithm to predict the gestures with very high accurately.

In this paper, a deep learning-based model which is combination of 3D-CNN and LSTM is proposed for recognizing dynamic hand gestures. To evaluate the proposed technique, the 20BN-jester dataset [9] is used. The 20BN-jester consists of 148,092 labeled video clips showing different people performing different dynamic hand gestures. The dataset has approximately 5000 video clips per class which are separated into training, validation and test sets. Due to computational restrictions, only 15 classes have been used in this paper. As discussed in the later section of the paper, the proposed model attained an accuracy of 97% on unseen data taken from the test set.

The remaining paper is organized as follows. In Section 2, work of other researchers using deep learning methods for gesture recognition is discussed. In Section 3, the proposed technique which is based on 3D-CNN and LSTM for the recognition of dynamic hand gesture is presented. In Section 4, experimental results are discussed along with different optimizers and hyper parameters used in this work. In addition, the prediction results are also shown for the new unseen data during the testing phase. Finally, Section 5 concluded the paper.

2 Related Work

Learning spatio-temporal features is critical for performance to be stable in human hand gesture or action recognition. Several deep neural networks based models have been introduced recently [10]. However, gesture recognition is significantly different from action recognition. The background information in an action recognition task is helpful for correct prediction of any action, but for gesture recognition, the background may be same for all gestures, which is a challenging issue for accurate prediction. Hand gesture recognition focuses more on the hand movement rather than the background.

Hand gesture recognition methods have been introduced to correctly identify and track hand postures. Many methods have been proposed for hand gesture recognition in recent years. Zhao et al. [11] presented a technique based on computer vision for real-time hand gesture recognition. Adaptive skin color and motion detection is used to identify hand regions. Hand images are extracted using the Histograms of Oriented Gradients (HOG). The characteristic local distribution of edges and intensity gradients are used to describe hand gestures. PCA-LDA is utilized to project the extracted HOG features into a low-dimensional subspace. Later, these features are classified using K-Nearest-Neighbors (KNN). A total of ten different gestures are classified with an accuracy of 91%. Chung et al. [12] proposed a technique for gesture recognition based on CNN. The technique recognizes hand gesture using a webcam. Color space and different morphology operation are used to differentiate gestures from complex background. To track the gesture movement, kernel correlation filters are used. The processed images are then fed into two different models; the VGG-Net and the AlexNet. The VGG-Net attained a better recognition rate as compared to the AlexNet. The recognition accuracy attained by the VGG-Net is 95.61%.

Bao et al. [13] proposed a two-dimensional CNN model for recognition of gestures. A nine-layer CNN is used to directly categorize hand gesture present in the images without preprocessing segmentation of the region of interest. The presented technique is able classify seven different types of hand gestures in real-time. The system achieved 97.1% accuracy with simple background and 85.3% accuracy was attained when the images had complex background. Neethu et al. [14] have also used CNN based classification technique for gesture recognition. The hand, which is the region of interest, is first separated from the background followed by adaptive histogram equalization to increase the contrast of the input image. Further, to segment fingers, connected component analysis is used. To classify different hand gestures, the segmented finger tips are fed to the CNN. The proposed technique attains an accuracy of 96.2% for gesture recognition with complex background.

Apart from 2D-CNN approaches for the effective recognition of hand gestures, 3D-CNNs are also used by researchers. In [15], a three-dimensional convolutional network (3D-ConvNets) with an attention mechanism technique is proposed for learning of spatio-temporal features. The model was trained on the UCF-101 and HMDB-51 datasets. The authors claim that 3DConvNets are better than simple 2D-CNNs for spatial-temporal learning of features. The 3D-CNN with Resnet101 architecture and softmax classifier achieved an accuracy of 95.5% on the UCF-101 dataset.

Molchanov et al. [16] proposed a robust hand gesture classification algorithm which uses 3D-CNN with augmentation techniques in spatio-temporal domain for reducing overfitting. Their model when used on the VIVA challenge dataset, attained classification precision of 77.5%. In [17], the researchers introduced two models for hand gesture recognition. The first model consists of CNN and an RNN-LSTM network. When the model was fed with color channel, it achieved an accuracy of 83%, whereas when the depth channel data was introduced, the accuracy was 89%. The second model consists of two parallel merged CNN and RNN with LSTM fed by RGB-depth dataset. This second model achieved 93% accuracy. Hakim et al. [18] used a 3D-CNN model followed by LSTM to extract the spatial-temporal features of 23 hand gestures, which includes 13 static and 11 dynamic. After the classification stage, a finite-state machine (FSM) is fused with the 3D-CNN+LSTM model to supervise the categorical decision. The dataset used was a combination of RGB and depth data. The model achieved an accuracy rate of 97.8% on subset class of eight gestures while recognition with the FSM model improved to 91% from 85% in real-time.

Nguyen et al. [19] proposed a two-stream convolution network model on 6 classes out of 25 using the 20BN-jester dataset. MobileNet-V2 followed by LSTM was used for spatio-temporal features extraction. The MobileNet-V2 is used because of its smaller number of training parameters due to which it took less time for training than other models mentioned in the paper. This model achieved precision of 91.25% which is a bit less than other models, but it greatly reduced the execution time and memory resources. In [20], the authors designed a low memory and power budget architecture for hand gestures recognition from video streams. The model has two parts: (1) A light-weight CNN architecture for extracting features and, (2) A deep CNN classifier for the classification of detected hand gestures. The authors used Levenshtein distance as the evaluation metric to classify hand gestures in real-time. ResNeXt-101 model is used on two publicly available datasets—the NVIDIA Hand Gesture and Ego-Gesture dataset. The model achieved an accuracy of 94.04% and 83.82%, respectively.

In [21], the authors proposed a deep deformable 3D-CNN with an impressive accuracy and real-time dynamic hand gesture recognition processing. The authors proposed three types of 3D-CNN models; Modified C3D, deformed ResNext3D-101 and InceptionResNet3D-v2 to learn the spatio-temporal information from video sequences. A spatio-temporal deformable CNN module is considered for three different datasets, Jester, Ego-Gesture and Chalearn-IsoGD. The model pays attention to learn more discriminative portions in a video sequence in both spatial and temporal domains. The 3D-CNN models have more training parameters which makes it computationally expensive and time consuming. Pigou et al. [22] emphasis that learning of temporal information is crucial for dynamic real-time gesture recognition. They proposed a model which consists of a residual network, batch normalization and exponential linear units (ELUs) for simple RGB dataset. It is concluded from their work that temporal information and LSTM is very important for getting accurate gesture prediction while dealing with dynamic gestures. However, RNN-based models can lead to some difficulties during training such as exploding or vanishing gradient. When using LSTM alone for hand gesture recognition, the model can ignore the low-level spatial or temporal information. It is because of this reason; a 3D-CNN coupled with LSTM is used in this work.

3 The Proposed Model

It is a challenging task to learn temporal and spatial information for gesture recognition with only one model [23]. To address this problem, a new architecture which consists of a 3D-CNN followed by LSTM and a Softmax classifier is proposed in this paper as shown in Fig. 3. This architecture has several steps such as data-loading, data-augmentation, training and testing.

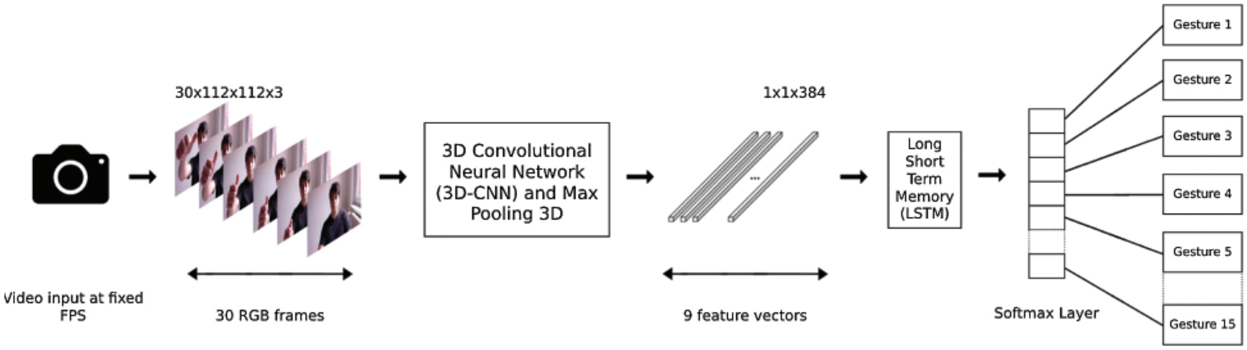


Figure 3: The proposed model pipeline

The proposed model pipeline consists of fusing two models, a 3D-CNN and an LSTM. The original frame size of the dataset has a height of 100 pixels and variable width; therefore, the input frames are first resized to 112 × 112 pixels during data loading. The 3D-CNN network is used to learn spatial information from successive video frames. The output of the 3D-CNN produces feature maps which are converted into vector that has 9 time-steps; also known as length of sequence and 384 features, it is then fed into the LSTM model as it accepts input with number of samples, time steps, and features information. In this model the number of sample taken per sequence is 1. To learn the temporal information from video frames, an LSTM network is used for the classification of hand gestures.

3.1 Dataset

Various video datasets are available publicly, however for this research, the 20BN-Jester, which is a very large-scale real-world dataset has been used. This dataset is generated by 1376 different actors in different unconstrained environments. It contains over 148,092 short video clips of 3 s length. Each video has 27 or more frames, which makes this the largest hand gesture dataset with more than 5 million frames in total. Due to time constraints and memory resources restriction, only 15 out of 27 hand gestures are used in this work. The original dataset has more than 4000 videos per class, however, 2000 random videos per class are chosen, which are further divided into 80% training and 20% validation set.

3.2 Data-Preprocessing

Video sequences of the 20BN-jester dataset have different length. For data preparation, the first step is to unify all the video-clips. Every video is limited to 30 frames per video. The dataset has videos of different length varying from 27 to 46 frames. Only 30 frames for each video clip have been adopted to train the model. The overall dataset contains 30,000 folders for 15 classes and each class has 2000 folders or samples. The dataset is separated into 80% training and 20% validation sets, respectively. All video frames are resized to 112 × 112 pixels during loading of data for training.

3.3 Data-Augmentation

Deep learning models need more data for improved training and subsequent performance. To achieve this, data augmentation techniques are used to modify the current dataset and create more variations of the images which will improve the model learning. The data augmentation techniques used in this work are explained below.

Image augmentation uses affine transformation to modify the geometric structure of images, preserving the ratios of distances and collinearity. It is often used in deep learning to increase training data quantity. In this work, each frame is first translated by −20% to +20% per axis. Then images are scaled by 80% to 120% of their original size. In addition, shearing and rotation operations are also performed for each frame. Besides affine transformations, contrast normalization and additive Gaussian noise are also applied to each frame. Contrast normalization is applied uniformly for each per image.

Adding noise to small dataset can increase the dataset and reduce overfitting and has a regularizing effect. When the neural network tries to learn very high frequency spatio-temporal features or patterns that occur a lot, the model is usually over fitted. To avoid such a scenario, data augmentation is performed by using the additive Gaussian noise with zero mean. Eq. (1) below shows the PDF distribution of Gaussian noise.

$$f(x) = \sqrt{\frac{1}{2\pi\sigma^2}} \cdot e^{\frac{-(x-\mu)^2}{2\sigma^2}}, \tag{1}$$

where x is the gray value (0 to 255), σ is the standard deviation and μ is the mean. We can improve the learning capabilities of the model by adding the right amount of noise to the image. In this work, Gaussian noise between ranges 0.0 to 0.05 per channel is added. Tab. 1 shows that before applying data augmentation, the model was over-fitting. On the other hand, after data augmentation, a significant improvement in training-validation accuracy and loss can be observed.

Table 1: Effect of data augmentation on training and validation results

Epoch #	Before augmentation				After augmentation			
	Training		Validation		Training		Validation	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
1–10	54%	0.94	64%	0.75	79%	0.63	81%	0.60
11–20	70%	0.51	76%	0.71	89%	0.42	86%	0.43
21–30	84%	0.45	86%	0.62	92%	0.33	90%	0.34
31–40	85%	0.33	87%	0.45	95%	0.13	94%	0.19
41–50	88%	0.32	90%	0.45	99%	0.04	97%	0.09

3.4 Learning Spatio-Temporal Features

Nowadays, deep learning based techniques are being widely used to perform gesture recognition tasks more accurately. A number of researchers have used CNN models to classify static gestures, however, for dynamic gesture recognition, CNN based models do not have high accuracy. The combination of learning both spatial and temporal features is a necessary requirement for dynamic gesture classification. To achieve this, a six-layer 3D-CNN model is used in this work which can extract temporal features by preserving the spatial information of the video frames. It is pertinent to mention that similar networks have been used in video classification problems, for example, [24] to extract short-term temporal information from input video frames. Merely using a 3D-CNN model for dynamic hand gesture recognition is not good enough to learn long-term spatio-temporal information from video datasets. Therefore, another network which can learn the long-term temporal information is needed. In this work, a combination of 3D-CNN followed by an LSTM network is used, as shown in Fig. 4.

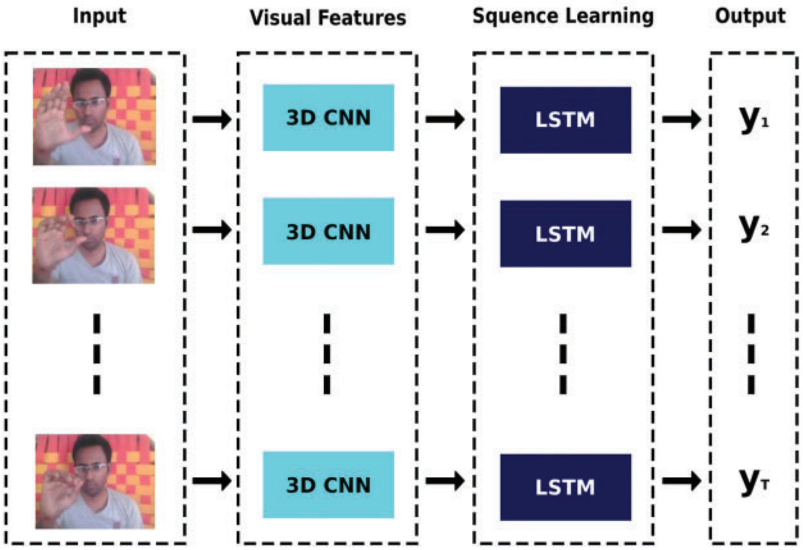


Figure 4: The proposed model general diagram

The structure of an LSTM unit consists of input/output and forget/cell gates which controls the learning process as shown in Fig. 5. These gates are adjusted with the help of sigmoid functions to control the opening and closing during the learning process. The long-term memory in LSTM is known as the Cell state. It controls the information to be stored within an LSTM cell from the previous intervals. The remembering vector is called the forget gate which modifies the cell gate. If the forget gate output state is 0, it tells the cell gate to forget the information, and if 1, it tells the cell gate to keep it in the cell state.

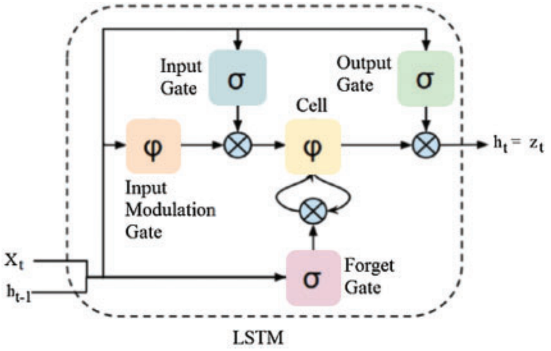


Figure 5: A typical long short-term memory unit

Eqs. (2) to (7) illustrate the learning process inside an LSTM unit [25].

$$i_t = \sigma(x_t w_{xi} + h_{t-1} w_{hi} + c_{t-1} w_{ci} + w_{ibais}). \tag{2}$$

$$f_t = \sigma(x_t w_{xf} + h_{t-1} w_{hf} + c_{t-1} w_{cf} + w_{fbais}). \tag{3}$$

$$z_t = tanh(x_t w_{xz} + h_{t-1} w_{hz} + w_{zbais}). \tag{4}$$

$$c_t = z_t \otimes i_t + c_{t-1} \otimes f_t. \tag{5}$$

$$o_t = \sigma(x_t w_{xo} + h_{t-1} w_{ho} + c_{t-1} w_{co} + w_{obais}). \tag{6}$$

$$h_t = o_t + tanh(c_t). \tag{7}$$

where, "i_t" is the input gate and "f_t", "o_t", "z_t" are the forget, output and cell gates, respectively. Whereas c_t and h_t are output memory activation functions at time 't'. Eqs. (3), (4), (6) and (7) are the formulas for forget cell, output gates and hidden state. For learning visual features of all the frames, we have used six convolution layers and 4 pooling layers. The features from the 3D-CNN are then fed into the LSTM network, which learns the sequence of the time series frames. The sequence of layers which makes up the 3D-CNN + LSTM architecture is shown in Fig. 6.

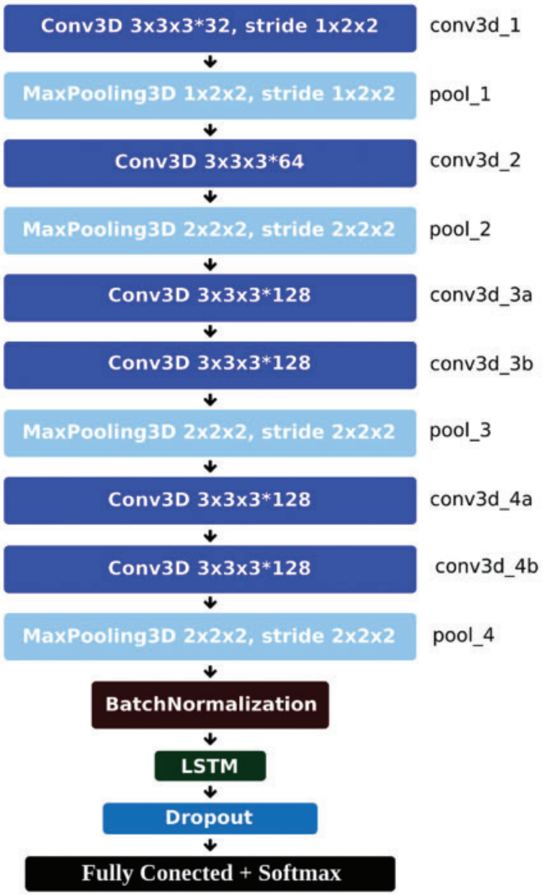


Figure 6: The proposed 3D-CNN + LSTM architecture

The features obtained from the 3D-CNN layers passes to the L2 batch normalization layer and are then fed to the LSTM layer. This is followed by a dropout layer to avoid overfitting and finally the fully connected layer followed by output Softmax layer as shown in Fig. 6. L2 batch normalization is applied to obtain higher learning rates and to accelerate the initialization process for training and to reduce overfitting of the model. Batch Normalization as shown by Eq. (8) is carried out by using mean and variance of training data batches before the activation layer on the input.

$$Output_{NN}\left(X;w,b,\alpha,\beta\right)=activation\left(\frac{\left(Xw-\mu\right)}{\sigma\left(Xw\right)}\alpha+\beta\right), \tag{8}$$

where, μ is the mean and σ is the standard deviation. These parameters are computed with respect to the batch size of the training data, 'X'.

L2 regularization is a method used in deep learning with sum of square of scale weights which are added to the loss function as a penalty condition to be minimized as shown by Eq. (9). L2 regularization ensures that the scale of weights should be close to zero. L2 regularization is also known as the "weight decay regularization".

$$L\lambda\left(w\right)=L\left(\mathrm{w}\right)+\lambda\left\|w\right\|\frac{2}{2}. \tag{9}$$

Each Conv3D has a kernel size (3 × 3 × 3), stride and pooling size (2 × 2 × 2) except for the first layer which is 1 × 2 × 2. This layer preserves the temporal details. Feature maps have three different filter depths; 32, 64 and 128 which reduces the training parameters to approximately 3 million. Features are extracted by the 3D-CNN model and are then fed to the LSTM first layer with a unit size of 512. A dropout layer is added after the LSTM layer with a value 0.5 and then the probability results are computed using the softmax function.

4 Experimental Results and Discussion

This section presents the experimental results of the proposed scheme. Simulation is carried out using Google Colab GPU Tesla T4 with 16 GB memory and RAM of 25 GB. The deep learning framework Keras has been used to implement the proposed architecture. Tab. 2 shows a comparison of the proposed 3D-CNN + LSTM model with other models in terms of accuracy, precision and recall using the 20BN-jester dataset for 15 classes. For the MobileNet-V2 + LSTM model, pre-trained weights of the ImageNet dataset [26] were used which gave a validation accuracy of 84% at 20 epochs. The results however did not improve further with validation loss not going below 0.25. The accuracy was reasonable but the real-time gestures prediction through a webcam was not accurate. After this, L2 batch normalization was introduced to MobilNet-V2+LSTM model and the accuracy improved to 87%, which was better but not acceptable as compared to other techniques proposed in the literature. A light-weighted model consisting of 3D-CNN+LSTM with L2-batch normalization was used which had 3.7 million training parameters. The combination of two models and normalization technique for sequential video dataset produce competitive results as shown in table below.

Further, we have implemented our model with three optimizers: Adam, SGD (stochastic gradient descent) and Adadelta and the results obtained from these experiments are shown in Fig. 7.

Table 2: Accuracy, precision, recall using different models

Method	Accuracy	Precision	Recall
MobileNet-V2 + LSTM	84%	83.7%	84.3%
MobileNet-V2 + L2Norm + LSTM	87%	85%	86%
3D-CNN + L2Norm + LSTM	97.5%	97.1%	98.2%

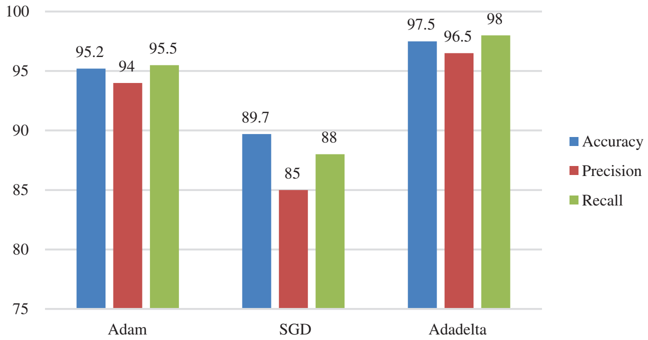


Figure 7: Comparison of different optimizers

As shown in Fig. 7, the Adam optimization technique achieved a validation accuracy of 95.2%, whereas the Stochastic Gradient Descent (SGD) optimizer achieved a lower accuracy. Adadelta optimizer with our proposed 3D-CNN+LSTM model achieved the best accuracy. The learning rate for all three optimizers is '0.00001'. In Figs. 8 and 9, the accuracy and loss curves are shown with 'Adadelta' optimizer.

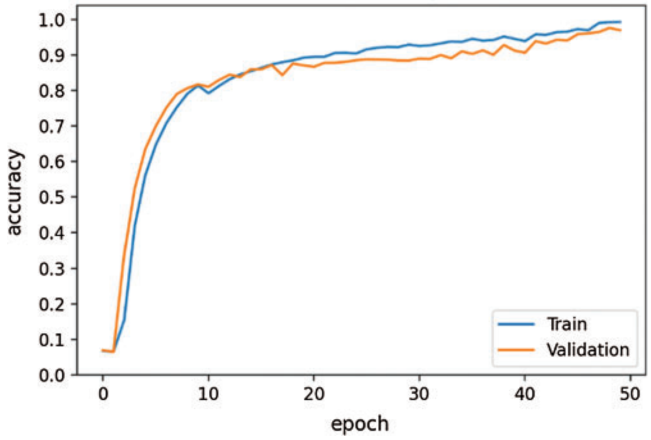


Figure 8: Model accuracy

From the model accuracy and model loss curves, it takes 50 epochs to reach the desired loss as the model was trained from scratch. For the first 4 epochs, the accuracy remained unchanged with very high loss. Later, after 6 epochs, the model achieved higher accuracy. The batch size kept for this training was 32 due to which it took many hours for training since the model was loading dataset in batches of 32. In addition, data augmentation was also carried out due to which it took almost 36 h to train the whole dataset for 50 epochs. With early stopping technique, training was stopped when the validation accuracy of 97.5% and loss of 0.09 was achieved. To avoid overfitting, L2 regularization together with batch normalization were used.

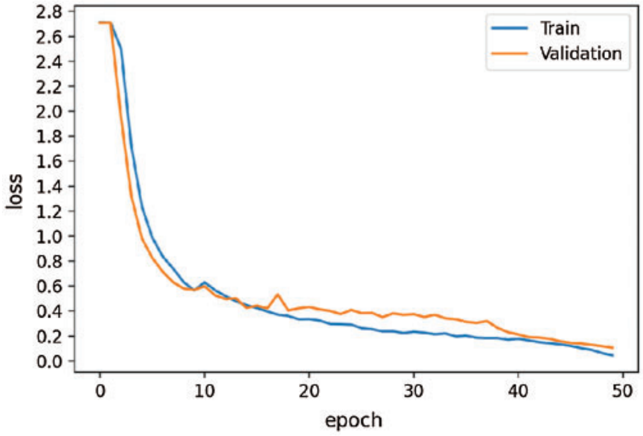
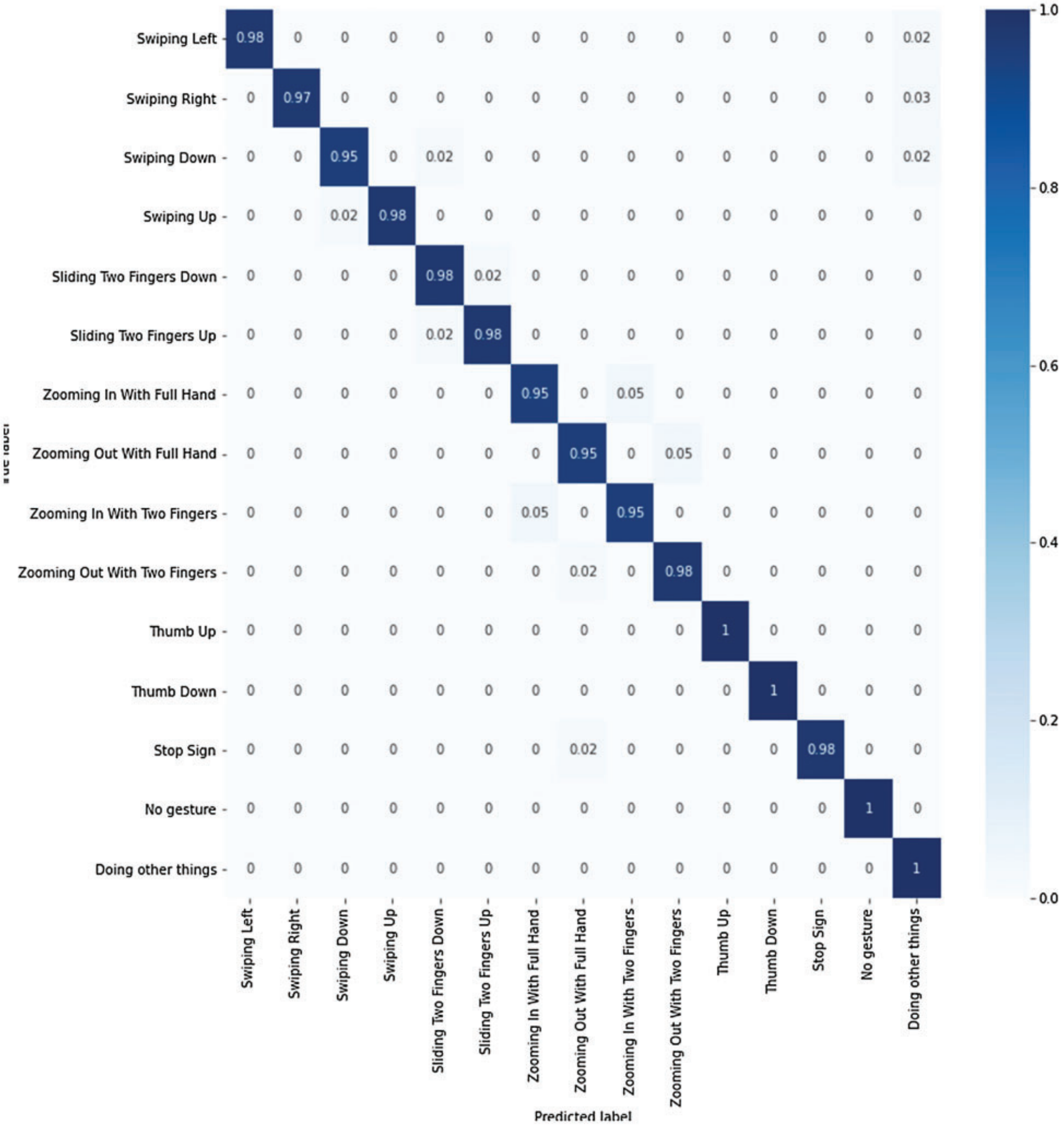


Figure 9: Model loss

Testing Results: For testing, 600 video-clips were chosen from the test folder of the 20BN-jester dataset. Each class has 40 or less video clips. Prediction results of the proposed model for unseen data are shown in Tab. 3. The results show that the model achieved 97% test accuracy on the unseen data. The model produced good results on 15 most difficult classes taken from 20BN-jester dataset.

Table 3: Prediction results

Classes	Precision	Recall	Fl-score	Number of video clips
Swiping left	1	0.98	0.99	41
Swiping right	1	0.97	0.99	39
Swiping down	0.97	0.95	0.96	40
Swiping up	1	0.98	0.99	41
Sliding two fingers down	0.95	0.98	0.96	41
Sliding two fingers up	0.97	0.97	0.97	41
Zooming in with full hand	0.95	0.95	0.95	40
Zooming out with full hand	0.95	0.95	0.95	38
Zooming in with two fingers	0.95	0.95	0.95	40
Zooming out with two fingers	0.95	0.97	0.96	40
Thumb up	1	1	1	40
Thumb down	1	1	1	41
Stop sign	1	0.97	0.99	40
No gesture	1	1	1	40
Doing other things	0.93	1	0.96	39
Accuracy			0.97	600
Micro avg	0.98	0.97	0.97	600
Weighted avg	0.98	0.97	0.98	600



	Swiping Left	Swiping Right	Swiping Down	Swiping Up	Sliding Two Fingers Down	Sliding Two Fingers Up	Zooming In With Full Hand	Zooming Out With Full Hand	Zooming In With Two Fingers	Zooming Out With Two Fingers	Thumb Up	Thumb Down	Stop Sign	No gesture	Doing other things
Swiping Left	0.98	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02
Swiping Right	0	0.97	0	0	0	0	0	0	0	0	0	0	0	0	0.03
Swiping Down	0	0	0.95	0	0.02	0	0	0	0	0	0	0	0	0	0.02
Swiping Up	0	0	0.02	0.98	0	0	0	0	0	0	0	0	0	0	0
Sliding Two Fingers Down	0	0	0	0	0.98	0.02	0	0	0	0	0	0	0	0	0
Sliding Two Fingers Up	0	0	0	0	0.02	0.98	0	0	0	0	0	0	0	0	0
Zooming In With Full Hand	0	0	0	0	0	0	0.95	0	0.05	0	0	0	0	0	0
Zooming Out With Full Hand	0	0	0	0	0	0	0	0.95	0	0.05	0	0	0	0	0
Zooming In With Two Fingers	0	0	0	0	0	0	0.05	0	0.95	0	0	0	0	0	0
Zooming Out With Two Fingers	0	0	0	0	0	0	0	0.02	0	0.98	0	0	0	0	0
Thumb Up	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Thumb Down	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Stop Sign	0	0	0	0	0	0	0	0.02	0	0	0	0	0.98	0	0
No gesture	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Doing other things	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 10: Confusion matrix

From a total 600 video-clips, 41 were classified as "Swiping Left" gesture. In actual, 40 video clips belong to swiping left class, hence the model predicted 40 clips correctly but 1 video clip was predicted false positive, therefore the recall is 98% for this class. Similarly for all the remaining classes, classification results are shown in Tab. 3. The model misclassified other classes as "Doing

Other Things" class, therefore precision for this specific class is low but recall is very high. The prediction results in Tab. 3 show that the model achieved 97% average test accuracy on the unseen data which is close to the validation accuracy of 97.5%. The confusion matrix obtained after making predictions on the new data using the proposed model is shown in Fig. 10.

The confusion matrix shows that most of the predictions are accurate. For the first gesture, "Swiping Left", the model predicted 98% of the video clips as true positive while only 2% were predicted as true negative. Similarly, for the "Swiping Right" gesture, 97% of the video clips were predicted as true positive while only 3% were predicted as true negative. The simplest gesture among all classes is the "thumb up" gesture, which was predicted 100% correctly. Results for the remaining gestures are also shown in Fig. 10 with true positive, true negative and false negative information.

5 Conclusion

A new deep-learning model is proposed that learns spatial-temporal features of dynamic hand gesture sequences in a video-stream. The architecture consists of a 3D-CNN followed by an LSTM network which learns both spatial and temporal features of all video frames under complex background and lighting conditions. The proposed model was trained on a subset of 20BN-jester dataset that contained 15 classes with unique hand gestures. The Combination of 3D-CNN with LSTM gives better results as compared to MobileNetv2 + LSTM. To avoid overfitting, batch normalization and L2 regularization has been used. The proposed model achieved 99% training, 97.5% validation and 97% predictive accuracy during real-time testing. In the future, more advanced deep learning [27–31] techniques will be applied for human gesture recognition.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] V. I. Pavlovic, R. Sharma and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, 1997.

[2] Y. Zhang, W. Zhou, Y. Wang and L. Xu, "A real-time recognition method of static gesture based on DSSD," *Multimedia Tools and Applications*, vol. 79, pp. 17445–17461, 2020.

[3] Y. Peng, H. Tao, W. Li, H. Yuan and T. Li, "Dynamic gesture recognition based on feature fusion network and variant convlstm," *IET Image Processing*, vol. 14, no. 11, pp. 2480–2486, 2020.

[4] R. Han, F. Zhiquan, X. Tao, A. Changsheng, X. Wei *et al.,* "Multi-sensors based 3D gesture recognition and interaction in virtual block game," in *Int. Conf. on Virtual Reality and Visualization*, Zhengzhou, China, IEEE, pp. 391–392, 2017.

[5] X. Liang, H. Li, W. Wang, Y. Liu, R. Ghannam *et al.,* "Fusion of wearable and contactless sensors for intelligent gesture recognition," *Advanced Intelligent Systems*, vol. 1, no. 7, pp. 1900088, 2019.

[6] O. K. Oyedotun, S. N. Tackie, E. O. Olaniyi and A. Khashman, "Data mining of students' performance: Turkish students as a case study," *International Journal of Intelligent Systems and Applications*, vol. 7, no. 9, pp. 20, 2015.

[7] W. Wang, J. Yang, J. Xiao, S. Li and D. Zhou, "Face recognition based on deep learning," in *Int. Conf. on Human Centered Computing, HCC 2014*, Switzerland, Cham, Springer, vol. 8944, pp. 812–820, 2014.

[8] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.

[9] J. Materzynska, G. Berger, I. Bax and R. Memisevic, "The jester dataset: A large-scale video dataset of human gestures," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision Workshops*, Seol, Korea, 2019.

[10] S. Herath, M. Harandi and F. Porikli, "Going deeper into action recognition: A survey," *Image and Vision Computing*, vol. 60, pp. 4–21, 2017.

[11] Y. Zhao, W. Wang and Y. Wang, "A real-time hand gesture recognition method," in *2011 Int. Conf. on Electronics, Communications and Control (ICECC)*, Ningbo, China, pp. 2475–2478, 2011.

[12] H.-Y. Chung, Y.-L. Chung and W.-F. Tsai, "An efficient hand gesture recognition system based on deep CNN," in *IEEE Int. Conf. on Industrial Technology*, Melbourne, VIC, Australia, pp. 853–858, 2019.

[13] P. Bao, A. I. Maqueda, C. R. del Blanco and N. García, "Tiny hand gesture recognition without localization via a deep convolutional network," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 3, pp. 251–257, 2017.

[14] P. Neethu, R. Suguna and D. Sathish, "An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks," *Soft Computing*, vol. 24, pp. 1–10, 2020.

[15] J. Li, X. Liu, M. Zhang and D. Wang, "Spatio-temporal deformable 3D convnets with attention for action recognition," *Pattern Recognition*, vol. 98, pp. 107037, 2020.

[16] P. Molchanov, S. Gupta, K. Kim and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Boston, MA, USA, pp. 1–7, 2015.

[17] F. Obaid, A. Babadi and A. Yoosofan, "Hand gesture recognition in video sequences using deep convolutional and recurrent neural networks," *Applied Computer Systems*, vol. 25, no. 1, pp. 57–61, 2020.

[18] N. L. Hakim, T. K. Shih, S. P. Kasthuri Arachchi, W. Aditya, Y.-C. Chen *et al.,* "Dynamic hand gesture recognition using 3DCNN and LSTM with FSM context-aware model," *Sensors*, vol. 19, no. 24, pp. 5429, 2019.

[19] P. Nguyen and T. N. Luong, "Two-stream convolutional network for dynamic hand gesture recognition using convolutional long short-term memory networks," *Vietnam Journal of Science and Technology*, vol. 58, no. 4, pp. 514, 2020.

[20] O. Köpüklü, A. Gunduz, N. Kose and G. Rigoll, "Real-time hand gesture detection and classification using convolutional neural networks," in *14th IEEE Int. Conf. on Automatic Face & Gesture Recognition*, Lille, France, pp. 1–8, 2019.

[21] Y. Zhang, L. Shi, Y. Wu, K. Cheng, J. Cheng *et al.,* "Gesture recognition based on deep deformable 3D convolutional neural networks," *Pattern Recognition*, vol. 107, pp. 107416, 2020.

[22] L. Pigou, M. Van Herreweghe and J. Dambre, "Gesture and sign language recognition with temporal residual networks," in *Proc. of the IEEE Int. Conf. on Computer Vision Workshops*, Venice, Italy, pp. 3086–3093, 2017.

[23] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera *et al.,* "Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Las Vegas, NV, USA, pp. 56–64, 2016.

[24] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 4489–4497, 2015.

[25] R. Zhang, Y. Liu and H. Sun, "Physics-informed multi-LSTM networks for metamodeling of nonlinear structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 369, pp. 113226, 2020.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh *et al.,* "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.

[27] M. Rashid, M. Alhaisoni, S.-H. Wang, S. R. Naqvi, A. Rehman *et al.,* "A sustainable deep learning framework for object recognition using multi-layers deep features fusion and selection," *Sustainability*, vol. 12, pp. 5037, 2020.

[28] M. S. Sarfraz, M. Alhaisoni, A. A. Albesher, S. Wang and I. Ashraf, “StomachNet: Optimal deep learning features fusion for stomach abnormalities classification,” *IEEE Access*, vol. 8, pp. 197969–197981, 2020.

[29] A. Majid, M. Yasmin, A. Rehman, A. Yousafzai and U. Tariq, “Classification of stomach infections: A paradigm of convolutional neural network along with classical features fusion and selection,” *Microscopy Research and Technique*, vol. 83, pp. 562–576, 2020.

[30] N. Hussain, M. Sharif, S. A. Khan, A. A. Albesher, T. Saba *et al.,* “A deep neural network and classical features based scheme for objects recognition: An application for machine inspection,” *Multimedia Tools and Applications*, vol. 1, pp. 1–23, 2020.

[31] M. Sharif, S. Kadry, G. Manogaran and T. Saba, “A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection,” *Image and Vision Computing*, vol. 106, pp. 104090, 2021.