Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
May 2020, vol. 25, no. 1, pp. 57–61
https://doi.org/10.2478/acss-2020-0007
https://content.sciendo.com

Hand Gesture Recognition in Video Sequences Using Deep Convolutional and Recurrent Neural Networks

Falah Obaid[1*], Amin Babadi[2], Ahmad Yoosofan[3]
[1, 3]*Electrical and Computer Engineering Department, University of Kashan, Kashan, Iran*
[2] *Department of Computer Science, Aalto University, Helsinki, Finland*

Abstract **– Deep learning is a new branch of machine learning, which is widely used by researchers in a lot of artificial intelligence applications, including signal processing and computer vision. The present research investigates the use of deep learning to solve the hand gesture recognition (HGR) problem and proposes two models using deep learning architecture. The first model comprises a convolutional neural network (CNN) and a recurrent neural network with a long short-term memory (RNN-LSTM). The accuracy of model achieves up to 82 % when fed by colour channel, and 89 % when fed by depth channel. The second model comprises two parallel convolutional neural networks, which are merged by a merge layer, and a recurrent neural network with a long short-term memory fed by RGB-D. The accuracy of the latest model achieves up to 93 %.**

Keywords **– Computer Vision (CV), Convolutional Neural Network (CNN), Deep Learning, Hand Gesture Recognition (HGR), Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM).**

I. Introduction

The main problem addressed in this article is to build a system for hand gesture recognition in videos by using deep learning architecture. The videos are recorded using a Kinect device in a vehicle cabinet. The gestures are performed under the challenging setting of clutter background and volatile illumination.

The gestures are done with both hands alternately at variant speeds. The hand gesture recognition systems aim at classifying and distinguishing hand gestures and using these classified gestures for controlling multiple devices based on computers, which are used in various fields such as industry and health.

Deep learning is a new branch of machine learning that is widely used in signal processing and computer vision. Unlike traditional machine learning techniques, deep learning does not depend on handcrafted features. Instead, it automatically extracts (learns) useful features from raw data.

We propose a model for HGR in videos, which comprises two stages. The first stage is pre-processing to overcome the variation of video lengths. The second stage is the deep learning stage to classify, label the frames and recognise the gestures.

II. Related Work

Hand gesture recognition of computer control system began with the invention of glove-based interfaces for human-computer interaction [1]. Then, the studies focused on removing any physically attached sensors and replacing them with computer vision methods. Over the past 30 years, computer interface technologies have improved revolutionarily and offered a totally wireless connection with less resistance to the wearer. Using cameras to recognise hand gestures began very early along with the development of the first wearable data gloves. There were many hurdles at the time in interpreting camera-based hand gesture detection. Traditional vision-based hand gesture recognition methods are still far from satisfactory for real-life applications [2]. Due to the limitations of the optical sensors, the quality of the captured images is sensitive to lighting conditions and cluttered backgrounds; thus, it is very difficult to detect and track hands robustly. Using a depth camera for capturing gestures eases a variety of computer vision problems such as background removal, light conditions, and blob detection.

Dynamic hand gesture recognition relies on learning temporal (i.e., trajectory, speed) and spatial (i.e., hand shape and location) features for a gesture. There are many techniques that have been used for classification, such as template-based approach [3], [4], statistical methods (Hidden Markov Analysis, Conditional Random Field, and causal analysis) [5]–[8].

In recent years, deep learning has won numerous contests in pattern recognition and machine learning. Deep learning has been decreasing reliance on engineered features to address increasingly complex recognition problems [9].

Molchanov *et al.* [10] developed an effective method of dynamic hand gesture recognition using 3D convolutional neural networks. They proposed a classifier that used a fused motion volume of normalized depth and gradient values of frames.

Camgoz *et al.* [11] proposed 3D convolution neural networks of large-scale user-independent continuous gesture recognition. They trained their deep network end-to-end (i.e., learning both the feature representation and the classifier). The architecture of their proposed network consists of eight 3D convolutional

[*] Corresponding authors e-mail: search2018k@gmail.com

layers, five 3D max-pooling layers, two fully connected layers and a softmax classification layer.

John *et al*. [12] used long-term recurrent convolution networks (LRCN) to classify video sequences of hand gestures. They improved the LRCN computational effciency and classification accuracy by extracting fewer representative frames of video sequence and inputting them to the LRCN framework.

Lai and Yanushkevich [13] proposed combining two deep learning techniques – the convolutional neural networks (CNN) and the recurrent neural networks (RNN) – for automated hand gesture recognition using both depth and skeleton data. An overall accuracy of 85.46 % was achieved on the dynamic hand gesture – 14/28 dataset.

Ma *et al*. [14] proposed a recognition method based on stacked denoising autoencoder (SDAE). They suggested an effective SDAE network for ASL dataset. Experiment results showed that, compared with stacked autoencoder (AE), deep belief network (DBN) and convolutional neural network (CNN) etc., the designed SDAE demonstrated better performance, the accuracy in ASL dataset was up to 98.07 %, while the training time was reduced to 1 h.

III. Method

Within the framework of the research, we propose a method for hand gesture recognition.

The method consists of the following main task:

- pre-processing;
- extracting spatial feature for each frame and giving a label for each class of frames using CNN;
- classifying sequence by means of RNN.

Figure 1 illustrates the framework of our method.

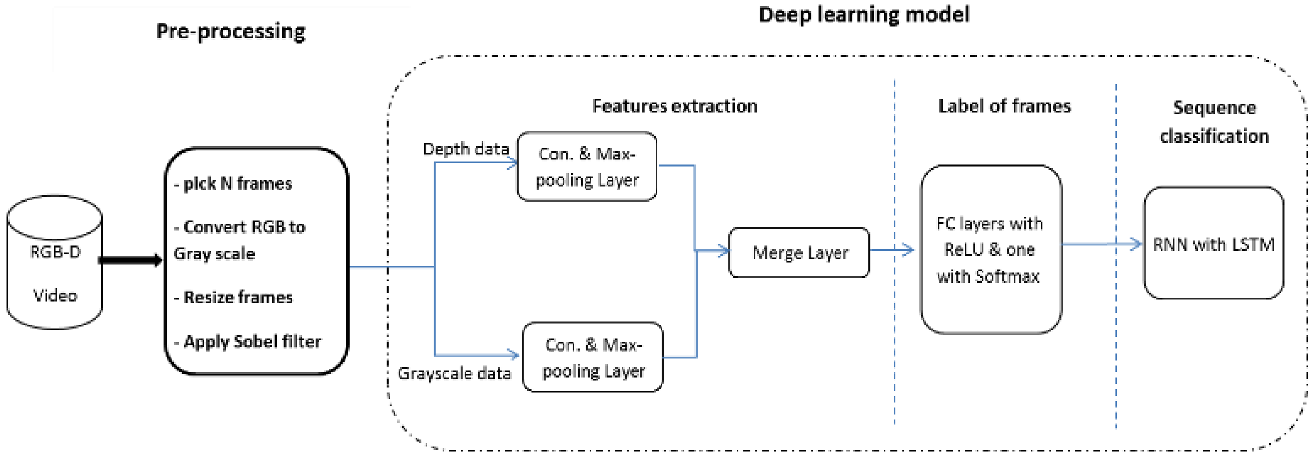


Fig. 1. The framework of the method.

A. Pre-Processing

A deep learning system is a fully trainable system beginning from raw input data, for example, image pixels, to the final output of recognised objects. Although the deep learning algorithms require little pre-processing in order to work properly, a special care is needed for the input data format to build an effective deep learning model.

In the research, a very short subsequence is used instead of entire video relying on the results mentioned in reference. The study shows that a few frames (1–7 frames) are sufficient for basic action recognition, with rapidly diminishing returns, as more frames are added. Consequently, 8 frames are picked instead of an entire video to use in our work [15].

For each video, more than one sample (8 frames) is chosen by shifting the time periods and taking into consideration that each sample includes the beginning and end of the video. The Grayscale frames are used, so the CNN does not depend on the background colour of the frames. The frames are resized from (115 × 250) pixels to (58 × 128) pixels. Sobel filter is applied to minimise the illumination variation effect on performance.

Data normalization is applied to ensure that each input pixel has a similar data distribution, which leads to faster convergence while training the model.

B. Data Augmentation

We augment the data by rotating the frames of videos in clockwise and counter-clockwise directions at different angles (15, 30, and 45 degrees). This also gives more robustness in the neural network against rotations in images, which commonly happen in hand gesture datasets.

C. Network Architecture

We propose two network architectures. The first architecture called Model I comprises two types of the neural network, i.e., CNN for spatial feature extraction and RNN for temporal feature extraction. Two CNNs are used for feature extraction:

1. Extracting features from the grayscale data;
2. Extracting features from the depth data.

The merge layer merges feature vector output of both CNNs and feeds them to the fully connected layer. Then softmax layer gets on a label for each frame.

Each CNN consists of four Convolution layers followed by Max-pooling layer. The size of all convolution kernels is 3 × 3 with stride 1. The number of convolution kernels in each layer is 32, 32, 64 and 128, respectively. The size of all pooling kernels is 2 × 2, with stride 2 × 2.

The flatten layer flattens the output of the last Max-pooling layers, then the merge layer merges them. The output of merge layer feeds to a fully connected neural network layer with 128 neurons followed by another layer with 64 neurons that is followed by 32-neuron layer. Finally, there is a softmax layer with 19 neurons corresponding to the number of gestures in the dataset, as shown in Fig. 2.

The second network architecture, i.e., Model II, consists of a single CNN fed from grayscale or depth data.

After training, the CNNs are fed by training and testing data. The output of individual frames of each gesture converts into a sequence and is used as a dataset for training and testing RNN.

The RNN with long short-time memory (LSTM) is used for temporal feature extraction and classification of the sequence of frames. The RNN comprises two LSTM layers and one softmax layer. Table I shows the structure of RNN.

TABLE I

THE STRUCTURE OF LSTM

Layer	Batch input	Activation function	No. of units
LSTM	(N, 8, 19)	Tanh	100
LSTM		Tanh	64
FCL		Softmax	19

D. Training Method

The training is performed using the Adam method, with a learning rate = 0.01, decay = 0.0002, beta-1 = 0.9, beta-2 = 0.999 and mini-batch size of 32 for CNN. For the RNN with LSTM, we use the Adam method with a learning rate = 0.001, decay = 0.0002, beta-1 = 0.9, beta-2 = 0.999 and batch size = 16. We split the dataset into training (70 %) and testing (30 %).

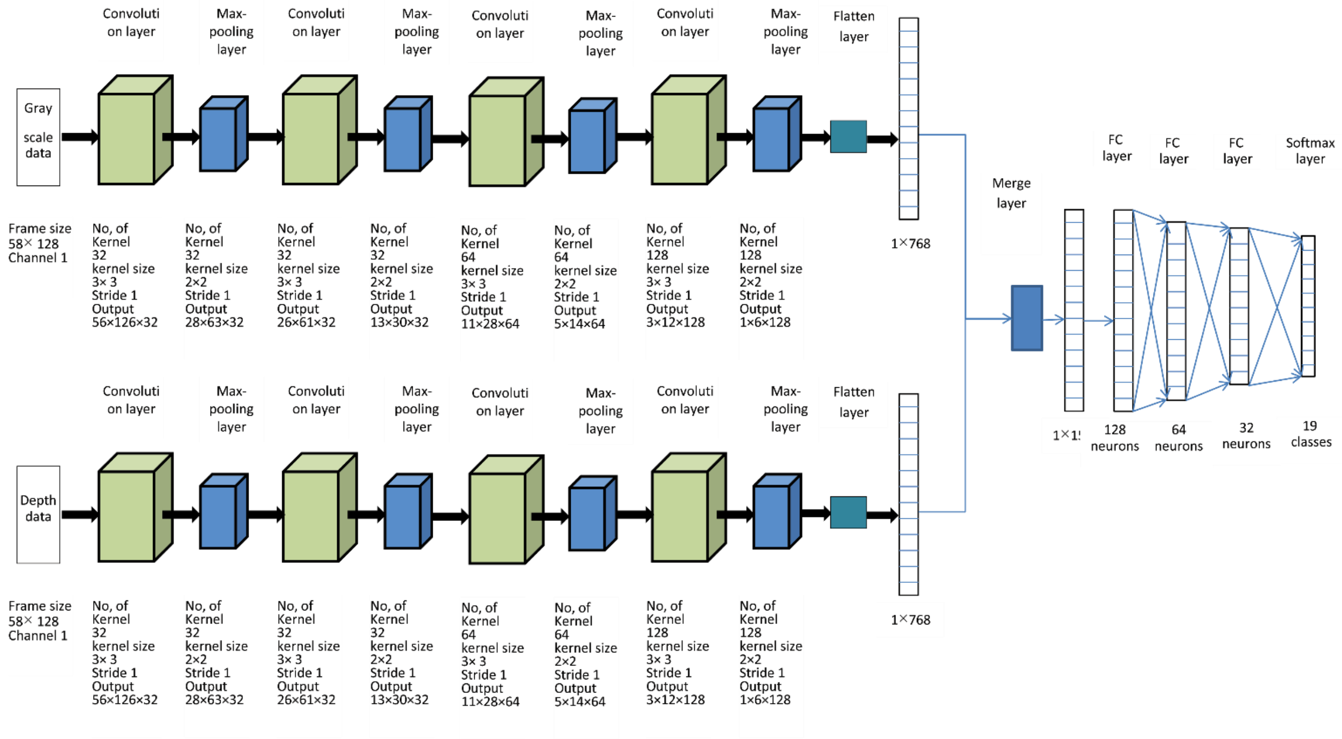

Convolution layer
Max-pooling layer
Convolution layer
Max-pooling layer
Convolution layer
Max-pooling layer
Convolution layer
Max-pooling layer
Flatten layer
Gray scale data
Frame size 58× 128 Channel 1
No, of Kernel 32 kernel size 3× 3 Stride 1 Output 56×126×32
No, of Kernel 32 kernel size 2×2 Stride 1 Output 28×63×32
No, of Kernel 32 kernel size 3× 3 Stride 1 Output 26×61×32
No, of Kernel 32 kernel size 2×2 Stride 1 Output 13×30×32
No, of Kernel 64 kernel size 3× 3 Stride 1 Output 11×28×64
No, of Kernel 64 kernel size 2×2 Stride 1 Output 5×14×64
No, of Kernel 128 kernel size 3× 3 Stride 1 Output 3×12×128
No, of Kernel 128 kernel size 2×2 Stride 1 Output 1×6×128
1×768
Merge layer
FC layer
FC layer
FC layer
Softmax layer
1×1!
128 neurons
64 neurons
32 neurons
19 classes
Convolution layer
Max-pooling layer
Convolution layer
Max-pooling layer
Convolution layer
Max-pooling layer
Convolution layer
Max-pooling layer
Flatten layer
Depth data
Frame size 58× 128 Channel 1
No, of Kernel 32 kernel size 3× 3 Stride 1 Output 56×126×32
No, of Kernel 32 kernel size 2×2 Stride 1 Output 28×63×32
No, of Kernel 32 kernel size 3× 3 Stride 1 Output 26×61×32
No, of Kernel 32 kernel size 2×2 Stride 1 Output 13×30×32
No, of Kernel 64 kernel size 3× 3 Stride 1 Output 11×28×64
No, of Kernel 64 kernel size 2×2 Stride 1 Output 5×14×64
No, of Kernel 128 kernel size 3× 3 Stride 1 Output 3×12×128
No, of Kernel 128 kernel size 2×2 Stride 1 Output 1×6×128
1×768

Fig. 2. The architecture of CNN used for feature extraction and labelling of frames.

IV. RESULTS

The metrics was used to measure and compare model performance. The classification accuracy and logarithmic loss were used as metrics to evaluate the model. We conducted many experiments to evaluate the performance of the proposed system. The effect of the number of picked frames and the channel (RGB, Depth, or RGB-Depth) were used to fed data to the system. In experiment 1, Model II was used with the RGB channel and eight picked frames. The validation accuracy of the CNN was 82.8 %, and the validation loss was 0.56. In experiment 2, Model II was used with the depth channel and eight picked frames. The validation accuracy of the CNN was 89 %, and the validation loss was 0.39. In experiment 3, Model I was used with the RGB-D and eight picked frames. The validation accuracy of the CNN was 93 %, and the validation loss was 0.25.

As noted, the results obtained from using Model I outperformed the results of Model II because of the advantage of feature extraction from the depth and colour data. Table II shows the validation accuracy and loss validation of Models I and II.

We conducted two experiments to study the effect of the number of picked frames. In experiment 4, Model I was used with the RGB-D and four picked frames. The validation accuracy of the CNN was 85 %, and the validation loss was 0.58. In experiment 5, Model II was used with the RGB-D and sixteen picked frames. The validation accuracy of the CNN was 86 %, and the validation loss was 0.54. The optimal number of picked frames was 8 frames. Table III shows the validation accuracy and loss of Model II for different numbers of picked frames. The results of our models were compared with the baseline method proposed by Molchanov *et al*. [10], who employed 3DCNN and other method that used the same data set for evaluation. The comparison showed that both Model I and Model II outperformed those methods. Table IV illustrates the comparison between our models and the models that used the same dataset for test. Fig. 3 presents the confusion matrix on the test data and precision for all classes.

TABLE II

THE VALIDATION ACCURACY AND LOSS VALIDATION OF MODEL I AND MODEL II

Model	Model II (colour)	Model II (depth)	Model I (colour + depth)
Validation accuracy	82 %	89 %	93 %
Validation loss	0.56	0.39	0.25

TABLE III

THE STRUCTURE OF LSTM THE VALIDATION ACCURACY AND LOSS OF MODEL I WITH DIFFERENT NUMBER OF PICKED FRAMES

No. of frames	Four frames	Eight frames	Sixteen frames
Validation accuracy	85 %	93 %	86 %
Validation loss	0.58	0.25	0.54

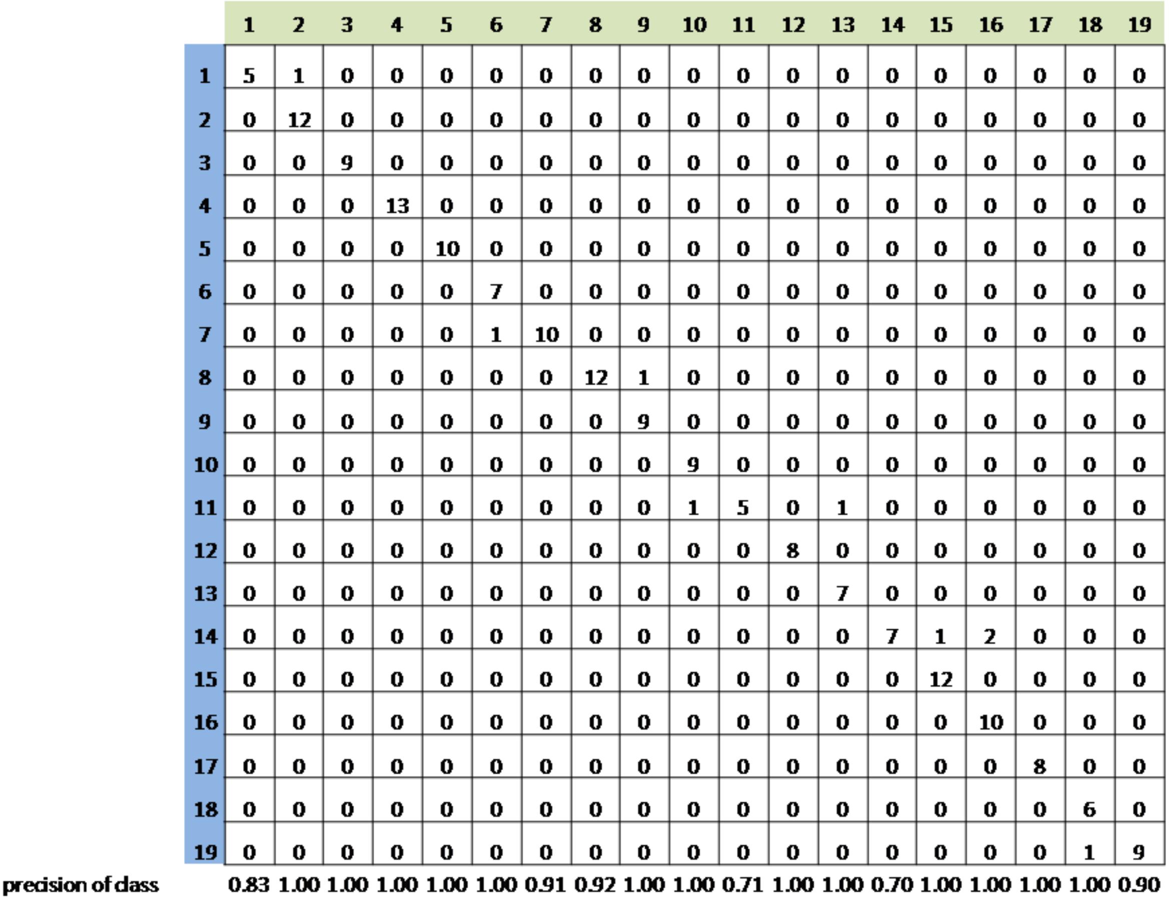
TABLE IV

THE COMPARISON BETWEEN OUR MODELS AND OTHER MODELS USING THE SAME DATASET

Researcher	Method	Modality	Validation accuracy
Our Model I	CNN + LSTM	RGBD	93 %
Our Model II	CNN + LSTM	D	89 %
Our Model II	CNN + LSTM	RGB	82 %
Molchanov *et al.*	3DCNN	RGBD	77.5 %
Ohn-Bar and Trivedi [16]	HOG + HOG2	RGBD	64.5 %
Oreifej O., Liu Z. [17]	HON4D	D	64.5 %

V. CONCLUSIONS

In the research, the objectives achieved were mainly based on recognising the hand gesture in sequence video. The research finding was related to determine the optimal architecture of CNN and LSTM that achieved significant accuracy. The model used some picked frames of the video sequence and CNN was used for feature extraction. These frames were classified and labelled. LSTM was used for classifying the gestures depending on the sequence of label frames.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	10	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	12	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	5	0	1	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	7	1	2	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9
precision of class	0.83	1.00	1.00	1.00	1.00	1.00	0.91	0.92	1.00	1.00	0.71	1.00	1.00	0.70	1.00	1.00	1.00	1.00	0.90

Fig. 3. The confusion matrix and precision of classes for test data.

The VIVA dataset was used to evaluate the model. A variety of experiments have been undertaken on the dataset using only colour data, depth data, and colour and depth together, also using different number of frames. The highest outcome for accuracy was 93 % when colour data, depth data and eight picked frames were used together.

VI. Future Research

The suggestion for future research could be along the following points.

1. Preparing the model to generalise it on another dataset as possible.
2. Binarizing the network to enable the model for implementation using the embedded hardware systems.
3. Using only CNN for HGR in the video sequence, by implementing a binary image that represents the superposition of all the skeletons of the hand region for all images (hand postures) within the gesture image sequence.

References

[1] P. Premaratne, "Historical development of hand gesture recognition", in Human Computer Interaction Using Hand Gestures. Cognitive Science and Technology. Singapore: Springer, 2014, pp. 5–29. https://doi.org/10.1007/978-981-4585-69-9_2

[2] C. S. Chua, H. Guan, Y. K. Ho, "Model-based 3D hand posture estimation from a single 2D image", *Image and Vision computing*, vol. 20, no. 3, 2002, pp. 191–202. https://doi.org/10.1016/S0262-8856(01)00094-4

[3] Z. Lai, Z. Yao, C. Wang, H. Liang, H. Chen, W. Xia, "Fingertips detection and hand gesture recognition based on discrete curve evolution with a kinect sensor", *2016 Visual Communications and Image Processing* (VCIP), IEEE, pp. 1–4, 2016. https://doi.org/10.1109/VCIP.2016.7805464

[4] C. Wang, Z. Liu, M. Zhu, J. Zhao, S. C. Chan, "A hand gesture recognition system based on canonical superpixel-graph", *Signal Processing: Image Communication*, vol. 58, pp. 87–98, 2017. https://doi.org/10.1016/j.image.2017.06.015

[5] A. Joshi, C. Monnier, M. Betke, S. Sclaro, "A random forest approach to segmenting and classifying gestures", *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition* (FG), IEEE. pp. 1–7, 2015. https://doi.org/10.1109/FG.2015.7163126

[6] A. Ghotkar, P. Vidap, K. Deo, "Dynamic hand gesture recognition using hidden Markov Model by Microsoft Kinect Sensor", *International Journal of Computer Applications*, vol. 150, no. 5, pp. 5–9, 2016. https://doi.org/10.5120/ijca2016911498

[7] H. D. Yang, "Sign language recognition with the kinect sensor based on conditional random fields", *Sensors*, vol. 15, no. 1, pp. 135–147, 2015. https://doi.org/10.3390/s150100135

[8] A. Joshi, S. Ghosh, M. Betke, S. Sclaro, H. Pfister, "Personalizing gesture recognition using hierarchical bayesian neural networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6513–6522. https://doi.org/10.1109/CVPR.2017.56

[9] F. J. Ordó˜nez, D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition", *Sensors*, vol. 16, no. 1, p. 115, 2016. https://doi.org/10.3390/s16010115

[10] P. Molchanov, S. Gupta, K. Kim, J. Kautz, "Hand gesture recognition with 3D convolutional neural networks", in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition workshops*, 2015, pp. 1–7. https://doi.org/10.1109/CVPRW.2015.7301342

[11] N. C. Camgoz, S. Hadfield, O. Koller, R. Bowden, "Using convolutional 3D neural networks for user-independent continuous gesture recognition", in *23rd International Conference on Pattern Recognition* (ICPR), IEEE, 2016, pp. 49–54. https://doi.org/10.1109/ICPR.2016.7899606

[12] V. John, A. Boyali, S. Mita, M. Imanishi, N. Sanma, "Deep learning based fast hand gesture recognition using representative frames", in *International Conference on Digital Image Computing: Techniques and Applications* (DICTA), 2016, IEEE, 2016, pp. 1–8. https://doi.org/10.1109/DICTA.2016.7797030

[13] K. Lai, S. N. Yanushkevich, "CNN+RNN depth and skeleton based dynamic hand gesture recognition", in *24th International Conference on Pattern Recognition* (ICPR), IEEE, 2018, pp. 3451–3456. https://doi.org/10.1109/ICPR.2018.8545718

[14] M. Ma, Z. Gao, J. Wu, Y. Chen, Q. Zhu, "A recognition method of hand gesture based on stacked denoising autoencoder", *Proceedings of the Fifth Euro-China Conference on Intelligent Data Analysis and Applications*, *Advances in Intelligent Systems and Computing*, Springer, Cham, vol. 891, 2018, pp. 736–744. https://doi.org/10.1007/978-3-030-03766-6_83

[15] K. Schindler, L. Van Gool, "Action snippets: How many frames does human action recognition require?", in *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2008, IEEE, 2008, pp. 1–8. https://doi.org/10.1109/CVPR.2008.4587730

[16] E. Ohn-Bar, M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations", *IEEE Transactions on Intelligent Transportation Systems* vol. 15, 2014, pp. 2368–2377. https://doi.org/10.1109/TITS.2014.2337331

[17] O. Oreifej, Z. Liu, "Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 716–723. https://doi.org/10.1109/CVPR.2013.98

Falah Obaid Director of the Information Technology Center in the Babylon government. He received M. S., Computer Engineering from Kashan university, Kashan, Iran. and now he is a Ph. D. candidate in University of Technology, Baghdad, Iraq.
E-mail: search2018k@gmail.com
ORCID iD: https://orcid.org/0000-0001-6662-7816

Amin Babadi is a Ph. D. candidate at Department of Computer Science, Aalto University, Finland. He works under supervision of Prof. Perttu Hämäläinen. He was also a visiting researcher at Imager lab, University of British Columbia, Canada, where he worked with Prof. Michiel van de Panne. His current research focuses on developing efficient, creative movement artificial intelligence (AI) for physically-simulated characters in multi-agent settings.
ORCID iD: https://orcid.org/0000-0003-4930-9917

Ahmad Yoosofan is an instructor in the Department of Computer Engineering at University of Kashan, where he has been a faculty member since 2004. From 2009 to 2016 he served as head of department. He received Computer Engineering B. S. from Shiraz University in 1999, and Artificial Intelligence M. S. from Shiraz University in 2003. His research interests span in variety Computer field including Text Processing, Blockchain Technology, Machine Vision and Information Technology. He is co-author of two books. For additional information search his name online. Website: http://yoosofan.github.io/en/
ORCID iD: http://orcid.org/0000-0002-3165-087X