Deep Convolutional and LSTM Neural Network Architectures on Leap Motion Hand Tracking Data Sequences

Kosmas Kritsis[1,2], Maximos Kaliakatsos-Papakostas[1], Vassilis Katsouros[1] and Aggelos Pikrakis[1,2]

[1] *Institute for Language and Speech Processing, Athena Research Center, Greece*
E-mail: {kosmas.kritsis, maximos, vsk}@ilsp.gr
[2] *Department of Informatics, University of Piraeus, Greece*
E-mail: {kritsisk, pikrakis}@unipi.gr

***Abstract*—This paper focuses on the hand gesture recognition problem, in which input is a multidimensional time series signal acquired from a Leap Motion Sensor and output is a predefined set of gestures. In the present work, we propose the adoption of Convolutional Neural Networks (CNNs), either in combination with a Long Short-Term Memory (LSTM) neural network (i.e. CNN-LSTM), or standalone in a deep architecture (i.e. dCNN) to automate feature learning and classification from the raw input data. The learned features are considered as the higher level abstract representation of low level raw time series signals and are employed in a unified supervised learning and classification model. The proposed CNN-LSTM and deep CNN models demonstrate recognition rates of 94% on the Leap Motion Hand Gestures for Interaction with 3D Virtual Music Instruments dataset, which outperforms previously proposed models of handcrafted and automated learned features on LSTM networks.**

***Index Terms*—gesture recognition, 3D musical instrument interaction, CNN, LSTM, CNN-LSTM models**

I. Introduction

Gesture recognition is a critical task for designing robust interfaces that rely on non-haptic Human-Computer Interaction (HCI) through body motion and gestures. This paper expands on a previously presented method [1] for dynamic Hand Gesture Recognition (HGR) [2], applied in the context of musical gesture interaction. Such gestures are quite subtle [3] and this, in combination with the required effectiveness on instant visual and auditory feedback that should be taken into account when designing gesture-driven virtual music instruments [4], make this problem even more challenging.

Motion Capture (MoCap) sensors such as the Microsoft Kinect and Leap Motion, have been widely employed in studies related to music interaction [5]–[7], although focusing only on reactive mappings between the sensorial data and the control of musical parameters [8]. Recent advances in Deep Neural Networks (DNNs) outperformed the Gaussian Mixture Model/Hidden Markov Model (GMM/HMM) systems [9], while bringing tremendous improvements in temporal pattern recognition tasks. Furthermore, the prospects of deep Machine Learning (ML) architectures still need to be studied in expressive and real-time music interaction scenarios [10].

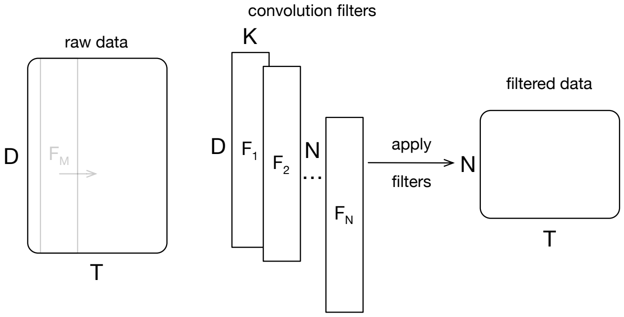


Fig. 1: Filtering process with 1D-CNN used for automated feature learning from the raw input data sequence.

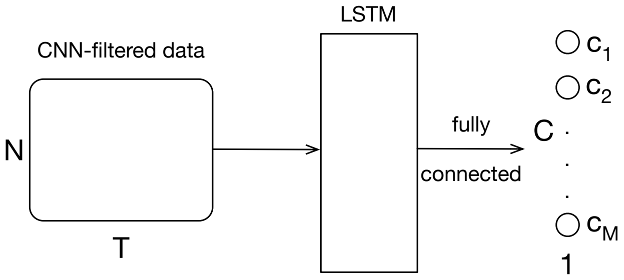


Fig. 2: CNN-LSTM Method: Feed the CNN-learned features (see Figure 1) to a LSTM neural network for sequence learning and with a fully connected layer for classification.

II. Application Context and Two Novel Hand Gesture Recognition Methodologies

Among the most successful approaches in applying artificial neural networks principles for MoCap, incorporate the

We acknowledge support of this work by the *iMuSciCA* project, funded from the EU Horizon 2020 research and innovation program with grant agreement No 731861 and the project "*Computational Sciences and Technologies for Data, Content and Interaction*" (MIS 5002437), implemented under the Action "*Reinforcement of the Research and Innovation Infrastructure*", funded by the Operational Program "*Competitiveness, Entrepreneurship and Innovation*" (NSRF 2014-2020) and co-financed by Greece and EU (ERDF).

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

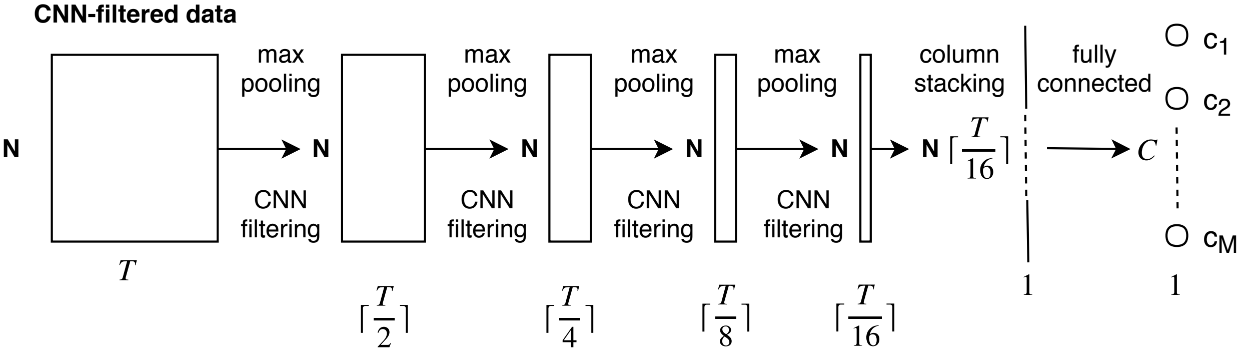


Fig. 3: dCNN Method: Consecutive application of convolution and max pooling operations followed by a fully connected layer for classification.

development of methods that rely on Convolutional Neural Networks (CNNs), which have proven effectiveness in filtering raw sensor data. For instance, Devineau et al. introduced a HGR system based on 1D–CNN architecture [11], that receive input sequences of raw hand-skeletal joint positions, achieving state-of-the-art performance on the DHG dataset from the SHREC 2017 3D Shape Retrieval Contest. Also, deep Convolutional Neural Networks (dCNNs) have been applied for Human Activity Recognition (HAR) from incoming skeletal tracking data provided by depth sensors [12]. In [13] Núñez et al. address HAR and HGR tasks by using a combination of CNN and Long Short-Term Memory (LSTM) recurrent networks on 3D data sequences obtained from full-body and hand skeleton tracking data, where the CNN training happens separately, adjusting the full CNN-LSTM architecture in a second stage. Another example of combining convolutional and recurrent architectures is [14], where an end-to-end 3DCNN-LSTM model is trained for recognizing gestures in videos, while achieving close to state-of-the-art accuracy on the ChaLearn dataset.

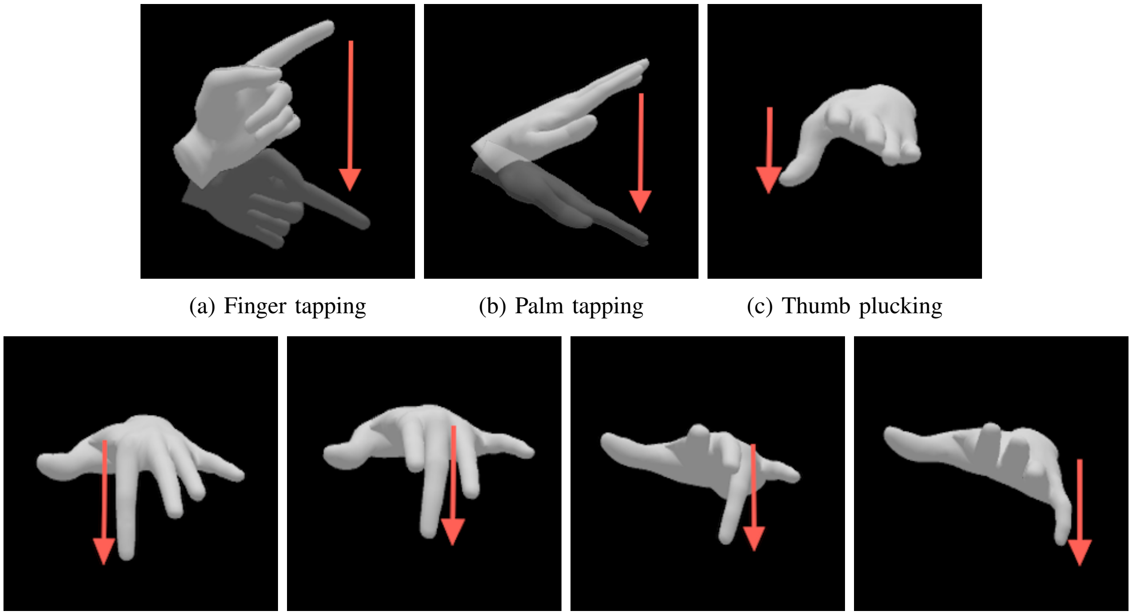
The purpose of the presented paper is to utilize the filtering effectiveness of CNNs in modeling multidimensional data sequences – in this case sensorial data. To this end, two novel methods that incorporate CNNs are presented: one uses CNN for feature learning and LSTM for sequence learning, while the other method applies consecutive convolutions and max pooling operations in a deep network. The proposed architectures are compared with a method presented recently [1] on a dataset of musical hand gestures. Statistically significant improvements are reported in the experimental results, reaching an average accuracy of 94.32% and 94.44% respectively. The remainder of the paper is organized as follows: Section II describes the context of the application and the architecture of the proposed gesture recognition methods; Section III describes the experimental setup and presents the evaluation results; and Section IV concludes the paper with reference to future research.

The iMuSciCA[1] project develops a STEAM (Science, Technology, Arts and Mathematics) platform that allows students study scientific/engineering principles for constructing 3D virtual musical instruments and afterwards interact with them and generate music. Enabling intuitive and efficient music performance with 3D instruments using the Leap Motion Sensor is a vital aspect for engaging students in music creative activities. Additionally, it is important to incorporate gestures that are inspired by those used for interacting with real-world instruments (e.g. plucking a string or tapping a drum or a piano key), towards offering a realistic and physical-related performance experience. The developed methodologies focus on identifying quick gestures that rapidly trigger events on the 3D virtual musical instruments. These methods could also be used for generic-purpose gesture recognition tasks as well.

The methods investigated in this paper leverages on the ability of the CNNs, in their 1-D form, to capture relations in sequences of events and extract meaningful features that increase classification accuracy. In particular, we adopt CNNs in two different architectures; the first deploys a convolution layer for feature extraction which subsequently feed a LSTM network for sequence learning and a fully connected layer for classification in the gesture classes; the second is a deep CNN architecture with several 1D convolutional layers in the features dimension, followed by max pooling in the time dimension. In both methods, the input corresponds to time series of raw data coming from the Leap Motion Sensor within an examined time window.

Figure 1 shows the 1D-CNN filtering process used in both presented methodologies. The Leap Motion Sensor produces time sequences of D measurements, corresponding to raw data of positions, velocities and directions, among others, of the fingers and the palm skeletons. Let us denote with T the time window, expressed in number of frames, of the input sequence of raw data. On each D-dimensional sequence of T length, we employ N 1D convolutional filters of kernel size K and sliding across frames for extracting one feature for each frame using the ReLU activation function; we use zero padding for preserving the frame count. Thereby, each filter produces a single 1D time series, which are stacked in an $N \times T$ matrix.

[1] http://www.imuscica.eu/



(a) Finger tapping (b) Palm tapping (c) Thumb plucking

(d) Index plucking (e) Middle plucking (f) Ring plucking (g) Pinky plucking

Fig. 4: Illustration of the considered instrumental gesture classes corresponding to the right hand as they were performed in the (LMHGIf3DVMI) dataset [15]. The temporal evolution of the fingers' motion trajectories follows the direction of the arrow.

It should be noted that the ReLU output for each time frame is locally normalized across a neighborhood of n time frames by following the 1D version of the normalization proposed in the development of the ImageNet [16], i.e.

$$b_x^i = a_x^i / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(T-1,i+n/2)} (a_x^j)^2 \right)^{\beta} ,$$

where b_x^i is the normalized activation value and a_x^i is the non-normalized one.

The first method examined, hereby referred to as CNN-LSTM, is illustrated in Figure 2 and is similar to the one presented in [1]. The similarity has to do with the fact that a LSTM network is employed for learning sequential data; the difference is that the method proposed herein employs a CNN layer in place of the embedding layer for automated feature representation, i.e. the CNN filtering depicted in Figure 1. The output of the LSTM is used as input to a fully connected layer with linear activation for classification among the targeted classes. During training, the system learns the parameters of the CNN filters, the parameters of the LSTMs as well as the weights of the fully connected output layer.

The second method is illustrated in Figure 3 and it is referred to as dCNN. This method is based on a deep CNN architecture, with 4-layers applying consecutive 1D convolution and max pooling operations; this process begins with the CNN-generated feature matrix and consecutively "bisects" it four times. The outputs of the fourth layer are stacked to form a vector of size $N \times T/16$ which are feed to a fully connected layer with linear activation for classification. During training, the system learns the CNN filters in Figure 1 and the weights at the fully connected output layer.

III. Experimental Results

The evaluation experiments have been conducted on the Leap Motion Hand Gestures for Interaction with 3D Virtual Music Instruments (LMHGIf3DVMI) dataset which is available on-line [15]. The LMHGIf3DVMI dataset includes gesture sequences of the right hand from 10 participants (5 female and 5 male) for eight gesture classes using the Leap Motion Sensor. The dataset includes in total 1019 samples, with 10-15 samples for each gesture per participant. The gesture classes are related to the fingers plucking and the palm and index finger tapping, as depicted in Figure 4, in addition to an 'unknown' class, representing arbitrary hand and finger movements. Each gesture is a time series with maximum length of $T = 75$ frames, where each frame contains $D = 186$ measurements as they are provided from the Leap Motion SDK.

In all experiments we use $N = 64$ 1D convolutional filters with kernel size $K = 2$. The training parameters that produce the reported results, include a learning rate of 0.001 using the Adam optimisation algorithm [17] for the minimization of the cross entropy cost function, with L2 regularisation of weights set to 0.015, and gradient clipping in the range of $[-1, 1]$ during back propagation. Furthermore, we employ a dropout rate of 0.5 on the LSTM cells as well as on the stacked features of the dCNN. In addition, for the local normalization after ReLU activation, we have set $k = 1$, $\alpha = 0.0002$, $\beta = 0.75$ and the neighborhood "radius" $n = 5$. The LSTM network in the CNN-LSTM methodology includes 128 LSTM cells.

TABLE I: Recognition accuracy and run time of the CNN-LSTM and dCNN methods and comparison with the LSTM method.

	Accuracy			**Run time (ms)**		
Fold No	**LSTM**	**CNN-LSTM**	**dCNN**	**LSTM**	**CNN-LSTM**	**dCNN**
0	90.91%	92.93%	95.96%	3.253	2.047	0.695
1	89.29%	95.54%	97.32%	2.547	2.475	0.614
2	94.50%	94.50%	94.50%	2.846	2.525	0.741
3	91.07%	93.75%	92.86%	3.179	2.761	0.427
4	94.59%	94.59%	98.20%	2.731	1.797	0.623
5	86.36%	99.09%	91.82%	2.611	1.886	0.735
6	93.69%	93.69%	93.69%	2.803	3.455	0.440
7	92.73%	90.00%	91.82%	2.602	1.852	0.916
8	92.59%	94.44%	95.37%	2.743	4.558	0.453
9	91.96%	94.64%	92.86%	2.547	3.155	0.695
Average	**91.77%**	**94.32%**	**94.44%**	**2.786**	**2.651**	**0.633**
Unbiased STD	**2.40%**	**2.14%**	**2.12%**	**0.237**	**0.831**	**0.149**

TABLE II: Confusion matrix resulted by the proposed dCNN method.

		Predicted							
		RHIP	**RHIT**	**RHMP**	**RHPP**	**RHPT**	**RHRP**	**RHTP**	**RHUK**
Ground Truth	**RHIP**	95.77%	0.85%	0.83%	0.00%	0.00%	0.85%	0.00%	1.70%
	RHIT	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	RHMP	2.34%	0.00%	86.80%	0.00%	0.00%	4.66%	0.80%	5.41%
	RHPP	0.00%	0.00%	0.00%	96.19%	0.63%	1.27%	0.00%	1.91%
	RHPT	0.00%	0.83%	0.81%	2.46%	94.15%	0.00%	0.00%	1.75%
	RHRP	0.78%	0.00%	0.00%	1.57%	0.79%	95.26%	0.00%	1.61%
	RHTP	0.00%	0.00%	0.00%	0.00%	0.00%	0.69%	97.93%	1.38%
	RHUK	1.19%	1.88%	1.80%	0.59%	0.59%	3.08%	1.27%	89.60%

The experiments run on a computer with Intel Core i7 at 2.80 GHz processor and 16 GB RAM. For the implementation of the methods we have used the TensorFlow$^{\text{TM}}$ framework. In the experiments we have employed the 10-fold cross validation of the LMHGIf3DVMI dataset. The training of all models has been performed using batch sizes of 50 samples with 1000 training epochs. In the table I we present the recognition rates and the run times on the testing sets of the 10-folds using the proposed architectures of CNN-LSTM and dCNN, and compare them with our previously developed method based on a LSTM architecture [1].

One can observe in Table I that the average accuracy of the proposed CNN-LSTM and dCNN methods, increases by 2.5% approximately, in comparison with the previous LSTM method. This increase is statistically significant at the 5% significance level across all folds, as indicated by a two-sided Wilcoxon [18] rank sum test on the given accuracy distributions, rejecting the hypothesis that the accuracy distributions between the LSTM results and the results from the CNN-LSTM (p-value$= 0.0232$) and dCNN (p-value$= 0.0373$) come from distributions from the same median. The differences between the CNN-LSTM and dCNN methodologies are not statistically significant (p-value$= 0.8499$). Furthermore, in the same table one can see the average run time for recognizing one gesture sample of 75 frames. The LSTM and CNN-LSTM methods require approximately the same amount of time, around 2.7 msecs. The dCNN method requires around 0.6 msecs, which is more than 4 times faster than the LSTM and CNN-LSTM methods.

In Table II we present the confusion matrix resulted by the experiments on the testing sets of the 10-folds of the LMHGIf3DVMI dataset. One can observe that for the most gesture classes there is little confusion with the other classes, with the gesture classes of "unknown" (RHUK) and middle finger plucking (RHMP) demonstrating the most misclassifications. In particular, the gesture of the middle finger plucking (RHMP) is confused with the gestures of index finger plucking (RHIP), ring finger plucking (RHRP) and "unknown" (RHUK). This is somehow expected as most people tend to move downwards the index and ring fingers when perform the middle finger plucking gesture. In addition, the "unknown" (RHUK) is also expected to be confused with the other gesture classes as it contains all kind of gestures including the finger plucking gestures.

IV. Conclusions and Future Work

This paper has presented two novel methods based on deep neural networks for hand gesture recognition on data obtained from the Leap Motion Sensor. Experimental results were extracted from performing gesture recognition on a dataset of eight hand gestures classes. The proposed methods outperformed an LSTM-based method that has been presented in previous work. The CNN-LSTM method is similar to the previously developed LSTM method, with the difference of using a 1D convolutional layer instead of a fully connected one for feature embedding, resulting to a significant increase of the recognition accuracy. Similarly, the dCNN architecture has also exhibited statistically significant improvement of the gesture recognition results in comparison with the LSTM method, but no statistically significant difference in comparison with the CNN-LSTM method. However, the computation time of the dCNN method is significantly smaller compared to the LSTM and CNN-LSTM methods.

Future research will incorporate testing all three methods in a real-time setting, where sensorial data frames will be buffered into an integrated system for real-time gesture interaction. This study will not only reveal the actual performances in a real-time framework, but it will also indicate whether the computation time superiority of the dCNN method is indeed crucial. Another aspect that will be studied in future work is to exclude the "no gesture" class from the workflow and allow classification decisions on the basis of the classification probabilities exceeding a certain threshold so that the system is "confident enough" of its decision.

ACKNOWLEDGMENTS

We would like to thank all the colleagues from the *Athena Research Center* and the students from the *University of Piraeus* for participating in the data collection process, as well as the partners from the *iMuSciCA* and "*Computational Sciences and Technologies for Data, Content and Interaction*" projects for their constant support and the provided motivation.

REFERENCES

[1] K. Kritsis, A. Gkiokas, M. Kaliakatsos-Papakostas, V. Katsouros, and A. Pikrakis, "Deployment of lstms for real-time hand gesture interaction of 3d virtual music instruments with a leap motion sensor," in *Proceeding of the 15th Sound and Music Computing Conference (SMC2018) , Limassol, Cyprus*, Jul. 2018, pp. 331–338.

[2] S. Y. Boulahia, E. Anquetil, F. Multon, and R. Kulpa, "Dynamic hand gesture recognition based on 3d pattern assembled trajectories," in *7th IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA 2017), Montreal, QC, Canada*, Dec. 2017.

[3] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA*, Jun. 2016, pp. 4207–4215.

[4] A. Bouënard, M. M. Wanderley, and S. Gibet, "Gesture control of sound synthesis: Analysis and classification of percussion gestures," *Acta Acustica united with Acustica*, vol. 96, no. 4, pp. 668–677, 2010.

[5] J. Han and N. Gold, "Lessons learned in exploring the leap motion(tm) sensor for gesture-based instrument design," in *Proceedings of the International Conference on New Interfaces for Musical Expression*. London, United Kingdom: Goldsmiths, University of London, June 2014, pp. 371–374.

[6] M. Mandanici, A. Rodà, and S. Canazza, "A conceptual framework for motion based music applications," in *2nd IEEE VR Workshop on Sonic Interactions for Virtual Environments (SIVE@VR 2015), Arles, France*, Mar. 2015, pp. 9–13.

[7] Y. Zhang, S. Liu, L. Tao, C. Yu, Y. Shi, and Y. Xu, "Chinar: Facilitating chinese guqin learning through interactive projected augmentation," in *Proceedings of the Third International Symposium of Chinese CHI (Chinese CHI '15), Seoul, Republic of Korea*, Apr. 2015, pp. 23–31.

[8] C. P. Martin, K. Olav Ellefsen, and J. Torresen, "Deep Predictive Models in Interactive Music," *ArXiv e-prints*, Jan. 2018.

[9] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia*, Apr. 2015, pp. 4580–4584.

[10] B. Caramiaux and A. Tanaka, "Machine learning of musical gestures," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Daejeon, Republic of Korea, May 2013, pp. 513–518.

[11] G. Devineau, F. Moutarde, W. Xi, and J. Yang, "Deep learning for hand gesture recognition on skeletal data," in *13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, Xi'an, China*, May 2018, pp. 106–113.

[12] H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "Exploiting deep residual networks for human action recognition from skeletal data," *Computer Vision and Image Understanding*, vol. 170, pp. 51–66, 2018. [Online]. Available: https://doi.org/10.1016/j.cviu.2018.03.003

[13] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognition*, vol. 76, pp. 80–94, 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.10.033

[14] K. Mullick and A. Namboodiri, "Learning deep and compact models for gesture recognition," *CoRR*, vol. abs/1712.10136, 2017. [Online]. Available: http://arxiv.org/abs/1712.10136

[15] K. Kritsis, A. Gkiokas, M. Kaliakatsos-Papakostas, V. Katsouros, and A. Pikrakis, "Leap Motion Hand Gestures for Interaction with 3D Virtual Music Instruments (LMHGIf3DVMI)," June 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1260336

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.