Towards extending real-time EMG-based gesture recognition system

Cristina Andronache*, Marian Negru*, Ana Neacsu,*,†, George Cioroiu *,‡, Anamaria Radoi * and Corneliu Burileanu*
*Center for Advanced Research on New Materials, Products and Innovative Processes, University Politehnica of Bucharest
†Centre de Vision Numérique, CentraleSupélec, Inria, Gif-sur-Yvette, France
‡Research Institute for Artificial Intelligence ”M. Draganescu”, Romanian Academy, Bucharest, Romania
Email: ana.neacsu@speed.pub.ro

***Abstract*—In biomedical applications, the Electromyographic (EMG) signal is used to record the electrical activity of the muscles during their contraction. EMG signal classification stands at the core of real time systems that aim to discriminate between user's movements without relying on other environmental conditions (as it is the case with gesture classification based on video). The classification of EMG signals is usually used to translate the sign language or to design computer interfaces based on gesture recognition. In this paper, we propose a continuation of our previous work, on a real-time automatic hand gesture recognition using fully connected neural networks and temporal features extracted from the EMG signals. This approach leads to a recognition of an increased number of gestures, starting from a system trained for discriminating between a small number of classes of recognized gestures. More precisely, using an innovative transfer learning method, we preserved the performance of the system (99.31% overall accuracy compared to previous 99.78%), while doubling the number of recognized gestures.**

***Keywords*—classification, EMG, machine learning, neural networks, transfer learning**

I. Introduction

Recently, a significant amount of research has been focused on human-computer interaction (HCI) [1]. Hand gestures represent a very natural and easy way to interact with devices around us, taking the Internet of Things (IoT) experience to a different, more organic level [2]. Sign Language Recognition (SLR) [3] and gesture-based control focus on recognizing a set of gestures that are interpreted to facilitate the interaction with other devices [4] or the development of intelligent prosthesis [5]. These type of intelligent prosthesis make use of the residual muscles in the forearm to detect the movement that the user is trying to perform even if the person lost their superior limbs [4]. The main advantage of this approach is that the movement is naturally executed, making the interaction easy. Moreover, in contrast to the alternative of gesture classification, which is based on classifying video recordings of movements [6], the classification is not dependent on external conditions (e.g., lighting, background patterns).

The results of EMG classification improved consistently in recent years. State-of-the-art classifiers are able to recognize a gesture from a limited number of possible outcomes with a precision of over 95% [1], [3]. However, almost all these high-performance classifiers are resource-consuming, having behind very complex learning systems based on DNNs (Deep Neural Networks) or CNNs (Convolutional Neural Networks). Because of their complexity, integrating these classifiers in real-time or embedded applications is very expensive and requires powerful resources.

Lately, different methods and approaches were employed in order to obtain best time-performance trade-off. For example, in [7], the authors propose a method to classify not only the gestures that a user performs, but also the force with which these gestures are performed; this is done by means of hyperdimensional computing (HD) and considering 3 effort levels. Another approach is based on hyperdimensional computing [8], which encodes the quantization of the EMG signal and also the spatial correlation between channels into a large hypervector. Another way of obtaining better classification results is to take into consideration the movement error rate [9]. A method based on training DNNs for each subject in the dataset has been presented in [10]; their method uses time domain power spectral descriptors (TDPSD) which are used as input for a feed-forward DNN model.

This paper proposes an extension of our previous work, previously detailed in [1]. In [1], we presented results evaluated on a publicly available dataset of an automatic gesture recognition system, based on EMG signals captured in real-time with the help of an armband. For classification, we designed a Fully Connected Network, capable of discriminating between 7 gestures (7-AGR). We reported a very good performance of the system, achieving an overall accuracy of over 99%. We further develop our work by creating an extended dataset that contains all the gestures from the original dataset and 6 additional ones. Furthermore, we present an original transfer learning method to ensure that the the new 13 gestures-based recognition system (13-AGR) inherits the representations learnt by the previous model. The additional gestures were chosen such that different muscles are activated during the movement.

The rest of the paper is organized as follows: in Section II the experimental setup is presented, including the details of the new dataset. The transfer learning method and the results illustrating are presented in Section III. Section IV is dedicated to concluding remarks.

II. Experimental setup

A. Initial Dataset

For the first part [1], we used a dataset that is available online[1]. This dataset consists of 7 basic gestures that are correlated to the basic movements of the wrist, namely four gestures for hand mobility (i.e., left / wrist flexion, right / wrist extension, down / ulnar deviation, up / radial deviation), two gestures for hand grip (hold / hand close, release / hand open) and one gesture for neutral position. All the gestures were performed using the Myo armband. Myo armband is a non-invasive gesture recognition device, manufactured by Thalmic Labs, equipped with 8 circularly arranged dry EMG sensors (i.e., each EMG signal contains information from 8 channels), placed on the forearm.

B. Extended Dataset

We created an extended dataset, which is available online[2], to validate our method. For compatibility purposes, we used the same acquisition device and kept all the original gestures from the previous dataset, in the order they were defined, and added six more. We consider this dataset to be a starting point for the development of real-time Automatic Gesture recognition systems based on sEMG (surface EMG) signals.

We inspired from the American Sign Language (ASL) when we chose the new 6 categories of gestures. More precisely, we considered gestures that correspond to 5 letters ('I', 'D', 'V', 'F', 'L') and the "like" gesture. All 13 gestures are shown in Fig 1. The letters were chosen such that different groups of muscles are active while performing the given gesture and these 5 letters make use of all groups of forearm muscles. It is worth mentioning that the newly-included gestures use different groups of muscles (i.e., they do not overlap with the gestures included in the original dataset).

The extended dataset contains EMG signals acquired from 50 different subjects, from which 34 males and 16 females. All subjects were able-bodied students, aged between 21 and 25, who had no knowledge of the previous experiments and no prior experience with gesture execution (i.e., how to strain the muscles) or with Myo armband.

During the experimental setup, the position of the Myo armband was kept constant on the user's arm. The subjects were asked to perform all the gestures twice. For the first round of trials, the subjects received the name along with a fast visual representation of the gesture and were asked to execute the gestures in their own manner. The execution of the gestures was supervised by an external observer. During the second round of trials, the subjects received additional explanations on how to strain the muscle in order to correctly perform the gesture and they were asked to repeat the experiments.

A round consisted of executing each gesture (i.e., all the 13 gestures) by holding each gesture for about 5 seconds and then resting for another 5 seconds. Note that the rest gesture was recorded separately, not during every break. We started the recording while the user held a gesture and stopped it while the user was still holding the gesture. To further ensure that the acquired signal contains only useful information, we removed 0.25 seconds at the beginning and at the end of the 5 seconds period, thus resulting in 4.5 seconds of useful signal for each gesture.

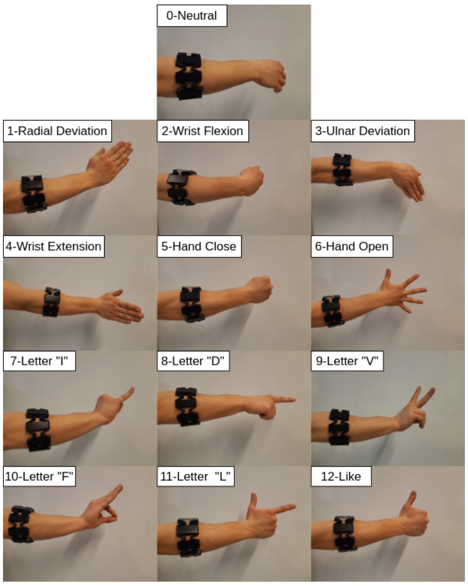

Fig. 1. The 13 hand gestures considered during the experiment

Finally, the extended dataset contains 13 categories of gestures collected from a total of 50 subjects, divided in two subsets. More precisely, we consider a gesture that was recorded with no prior knowledge to be called *Free*, while a gesture executed with additional information regarding the execution to be called *Assisted*.

III. Proposed method

A. Fine tuning strategy

For real-time control, latency is a very important factor to consider. An accurate classifier and a latency smaller than 250 ms ensures a natural, real-time feeling from the user's perspective [11]. Taking into consideration these time constraints, using frequency domain features is not recommended, since the processing time for these features is usually large (above 100 ms) [12]. In this regard, we consider only time-domain descriptors, already presented in [1].

In order to reduce the inference speed and obtain a real-time gesture recognition system, we propose a simple neural network architecture, comprised of only fully connected layers. We use leaky $ReLU$ for the activation part instead of other possible standard variants, as $ReLU$, $Sigmoid$, $Tanh$. Leaky $ReLu$ represents a very versatile activation, since it is unbounded (if compared to $Sigmoid$ and $Tanh$) and can also have negative values (if compared to $ReLU$). While this has the risk of being numerically unstable, this function can lead to better hidden representations [13]. The last layer uses Softmax to determine the highest class probability.

[1] https://github.com/Giguelingueling/MyoArmbandDataset

[2] https://speed.pub.ro/downloads/

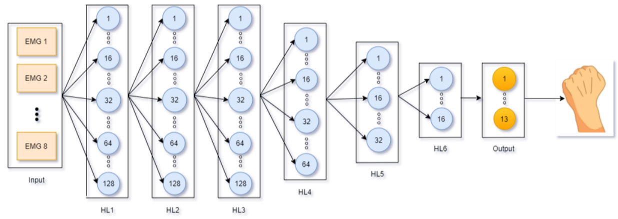

Fig. 2. Proposed Architecture for Automatic Gesture Recognition. Each EMG box represents a column vector containing 8 time-descriptors extracted from the signal. The last layer has 13 neurons representing the number of gestures being recognized.

A Batch Normalization step is inserted after each layer, to act as a regularizer and prevent overfitting [14]. The classical Stochastic Gradient Descent (SGD) is used as optimizer instead of more efficient techniques as ADAM [15]. Despite superior training outcomes, adaptive optimization methods (i.e., ADAM) have lower generalization capabilities [16]. The optimizer minimizes a *Categorical Cross Entropy* loss function. In order to compare the results obtained with the previous dataset presented in [1] and the extended dataset, we keep the same architecture of the proposed gesture recognition system, presented in Fig 2, changing only the output layer that represents the probability of each gesture class.

The architecture designed in [1] was able to recognize 7 gestures (7-AGR), and it was used as a baseline to infer the strategy for recognizing an increased number of gestures (i.e., 13 gestures). In order to avoid retraining the entire model, we consider a fine tuning strategy. As such, we adopt an idea that is frequently used in image classification, namely to change only the last layer of the neural network, presented in Fig. 3. Thus, we replace the output layer consisting of 7 neurons with another layer consisting of 13 neurons.

During the training process, the hidden layers of the 7-AGR model are frozen, allowing us to use the learning representation of the 7-AGR model. More precisely, starting from the previous 7-AGR model, we only alter the composition rule of the output layer, whereas the parameters of the hidden layers remain unmodified. We train the new model until the accuracy level stabilizes. Afterwards, we train the whole neural network, allowing the training process to modify both learning representations and the composition rule of the output layer.

Since our new 13-AGR system can be viewed as an extension of the previous one (7-AGR), we wish to make use of the results we have obtained so far and find a good starting point for the extended system. In other words, we want to preserve the same confidence of prediction for the feature vectors pertaining to the primary dataset (i.e., that contains EMG signals for 7 types of gestures).

In the following, we denote the 7-AGR and 13-AGR systems with T_7 and T_{13}, respectively. Let $\mathbf{x}$ be an input feature vector of one of the gestures that are common to both systems. Passing $\mathbf{x}$ through the networks, we have:

$$\begin{aligned} T_7(\mathbf{x}) &= \mathbf{y}_7 \\ T_{13}(\mathbf{x}) &= \mathbf{y}_{13} \end{aligned} \tag{1}$$

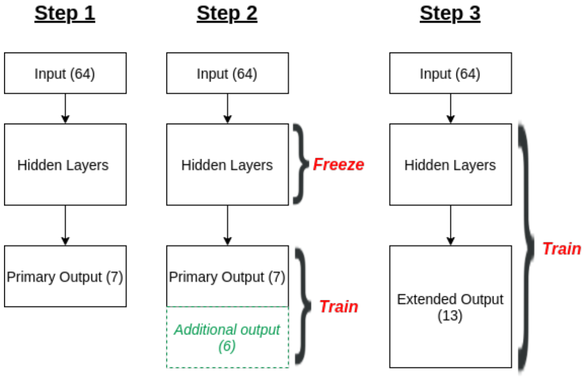

Fig. 3. Transfer learning strategy. The original architecture (i.e., 7-AGR) is shown in the left hand column. The steps of the training procedure of the new model (i.e., 13-AGR) are shown in the middle and the right hand columns.

where $\mathbf{y}_7$ and $\mathbf{y}_{13}$ represent the confidence vector for 7-AGR and 13-AGR respectively (i.e., a 7-dimension or 13-dimension vector consisting of the probabilities for each gesture class). Therefore, we impose the constraint that every input $\mathbf{x}$ is mapped to the same hidden representation $\mathbf{z}$, where $\mathbf{z}$ is the outcome of the network after the last hidden layer. The output of the systems will be given by:

$$\begin{aligned} \mathbf{y}_7 &= g(\mathbf{W}_7\mathbf{z} + \mathbf{b}_7) \\ \mathbf{y}_{13} &= g(\mathbf{W}_{13}\mathbf{z} + \mathbf{b}_{13}) \end{aligned} \tag{2}$$

where $\mathbf{W}_7$, $\mathbf{W}_{13}$, $\mathbf{b}_7$, $\mathbf{b}_{13}$ represent the weight matrices and the bias vectors of the last layer of 7-AGR and 13-AGR, respectively, while g denotes the *Softmax* activation function.

In order to obtain the same confidence level between 7-AGR and 13-AGR systems, then we consider same outputs for all the gestures included in the primary dataset, whilst, for the new gestures, we force the output of 13-AGR to be 0. Mathematically, this translates to:

$$\begin{aligned} \mathbf{y}_{13}(i) &\simeq \mathbf{y}_7(i), \quad \forall i \leq 6 \\ \mathbf{y}_{13}(i) &\simeq 0, \quad \forall i > 6 \end{aligned} \tag{3}$$

where i represents the index of the element in the confidence vector. This leads to having the i^{th} element of $\mathbf{W}_{13}\mathbf{z} + \mathbf{b}_{13}$ going to $-\infty$ for $i > 6$. The simplest way to achieve these requirements is to carefully design the last layer weights as:

$$\mathbf{W}_{13} = \begin{bmatrix} \mathbf{W}_7 \\ \mathbf{0}_{6\times16} \end{bmatrix}$$

$$\mathbf{b}_{13} = \begin{bmatrix} \mathbf{b}_7 \\ -1000 \times \mathbf{1}_6 \end{bmatrix}$$

where we chose $-1e^{03}$ as a value that ensures $e^{-1000} \simeq 0$ in *32-Bit Floating Point* precision. $\mathbf{0}_{6\times16}$ is a zero matrix of 6×16 values, whereas $\mathbf{1}_6$ is a vector with all elements equal to 1.

By replacing this in the last equation of system (2), we get:

$$\begin{aligned} \mathbf{y}_{13} &= g\left(\begin{bmatrix} \mathbf{W}_7 \\ \mathbf{0}_{6\times16} \end{bmatrix} \cdot \mathbf{z} + \begin{bmatrix} \mathbf{b}_7 \\ -1000 \times \mathbf{1}_6 \end{bmatrix}\right) \\ &= g\left(\begin{bmatrix} \mathbf{W}_7 \cdot \mathbf{z} + \mathbf{b}_7 \\ -1000 \times \mathbf{1}_6 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{y}_7 \\ \mathbf{0}_6 \end{bmatrix} \end{aligned} \tag{4}$$

TABLE I. Per class accuracy comparison of 13-AGR and 7-AGR

	Neutral	Rad. Dev.	Wrist Flex.	Ulnar Dev.	Wrist Ext.	Hand Close	Hand Open	”I”	”D”	”V”	”F”	”L”	Like
13-AGR-Free	88.52%	81.96%	88.08%	88.67%	88.44%	93.02%	85.76%	87.86%	77.62%	83.77%	82.53%	80.06%	88.58%
13-AGR-Assisted	99.88%	99.38%	99.54%	99.86%	99.52%	99.66%	98.77%	98.55%	99.00%	98.66%	98.77%	99.33%	98.33%
7-AGR	99.96%	99.75%	99.86%	99.72%	99.72%	99.86%	99.58%	-	-	-	-	-	-

which satisfies equations (3). Thus, the proposed fine tuning strategy results in a convenient initialization of weights as shown in Equation (4).

B. Results

We run experiments on the aforementioned, starting from the previous model (7-AGR). Furthermore, we made a clear distinction between the *Assisted* gestures and the *Free* ones, since we expect the latter to be a harder problem because every person executes the gestures in his/her own (natural) way.

Just like in our previous experiment, in order to have a quasi-stationary signal, we split the signal using a sliding window consisting of 250 ms (50 frames), this time with a step of 125 ms (25 frames), thus having an overlap of 50%. We obtaining two signal pools: one for *Assisted* gestures and one for *Free* ones. Next, we processed those pools of signal and we extracted the same time features as in our original work. Randomly, we separated each pool of features into two disjoint sets, one for fine-tuning and one for testing. Note that we did not use this time a set for validation. The ratio of the fine-tuning set and test set is 4:1.

By following the algorithm described before, we tested the new model (13-AGR) using the previous 7-AGR model and we obtained an accuracy of 51% for the *Assisted* gesture and 31% for the *Free* gesture. Since the system was not trained for 13 gestures, we can clearly observe that our proposed algorithm works. As mentioned before, we froze all the layers except the output one and we train both the *Assisted* and *Free* models until the network has converged. For the *Assisted* version, we reach a test accuracy of 85%, while the *Free* version only reached 35%. We believe this happens because the frozen part of the model excels at detecting certain patterns in muscles that translates to features and can easily distinguish between gestures. In the *Free* version, each subject executes the gestures in his/her own unique style, while in the *Assisted* version we taught the subjects how to execute each gesture, thus making sure to achieve those muscle patterns.

Afterwards, we train the network, obtaining the two versions of 13-AGR. The accuracy for *Assisted* one is 99.31%, while the accuracy for *Free* one is 85.73%, showing that the system is capable of good recognition for the natural gestures. Since the architecture is almost the same as the original one, the prediction time remains the same as before. Per class performance is detailed in Table I. This is a very good performance level, considering that the number of gesture classes is almost double and it shows that the proposed architecture is suited for learning good representations of the EMG signals.

IV. Conclusion

In this paper, we continued our previous work on EMG classification [1]. We advanced our prior experiments by making our own dataset with 13 gestures (the 7 gestures from the previous database and 6 new gestures). Moreover, this paper proposes a method to double the number of classifiable movements. This is done by employing a modified transfer learning algorithm that is usually used in image classification. The results show that the proposed method is capable of making use of the knowledge learnt with 7 gestures and with a relative small amount of extra training can reliably classify a total of 13 gestures, obtaining an overall accuracy of 99.31% for 13 gestures compared to 99.78% for 7 gestures.

References

[1] A. A. Neacsu, G. Cioroiu, A. Radoi, and C. Burileanu, “Automatic EMG-based hand gesture recognition system using time-domain descriptors and fully-connected neural networks,” in *Int. Conf. Telecommunications Signal Process.*, Budapest, Hungary, 1–3 Jul 2019, pp. 232–235.

[2] O. Fukuda, Y. Takahashi, N. Bu, H. Okumura, and K. Arai, “Development of an iot-based prosthetic control system,” *Journal Robot. Mechatronics*, vol. 29, no. 6, pp. 1049–1056, 2017.

[3] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, “A framework for hand gesture recognition based on accelerometer and EMG sensors,” *IEEE Trans. IEEE Trans. Syst. Man. Cybern. A Syst. Hum.*, vol. 41, no. 6, pp. 1064–1076, 2011.

[4] S. Bitzer and P. Van Der Smagt, “Learning EMG control of a robotic hand: towards active prostheses,” in *Proc. Int. Conf. Robot. Autom.*, 2006, pp. 2819–2823.

[5] H.-P. Huang, Y.-H. Liu, and C.-S. Wong, “Automatic EMG feature evaluation for controlling a prosthetic hand using supervised feature mining method: an intelligent approach,” in *Proc. Int. Conf. Robot. Autom.*, vol. 1. IEEE, 2003, pp. 220–225.

[6] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, “Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video,” *Int. Journal Comp. Vision*, vol. 126, no. 2-4, pp. 430–439, 2018.

[7] A. Moin, A. Zhou, S. Benatti, A. Rahimi, L. Benini, and J. M. Rabaey, “Analysis of contraction effort level in EMG-based gesture recognition using hyperdimensional computing,” in *IEEE Biomed. Circ. Syst. Conf.*, 2019, pp. 1–4.

[8] A. Rahimi, S. Benatti, P. Kanerva, and J. M. Rabaey, “Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition,” in *Proc. Int. Conf. Reboot. Comp.*, 2016, pp. 1–8.

[9] A. Gijsberts, M. Atzori, C. Castellini, H. Müller, and B. Caputo, “Movement error rate for evaluation of machine learning methods for sEMG-based hand movement classification,” *IEEE trans. on Neural Syst. Rehan. Eng.*, vol. 22, no. 4, pp. 735–744, 2014.

[10] A. K. Mukhopadhyay and S. Samui, “An experimental study on upper limb position invariant EMG signal classification based on deep neural network,” *Biomed. Sig. Process. Control*, vol. 55, p. 101669, 2020.

[11] N. V. Thakor, “Ieee transactions on neural systems and rehabilitation engineering,” *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 14, no. 1, pp. 1–4, 2006.

[12] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, “Deep learning for electromyographic hand gesture signal classification using transfer learning,” *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 27, no. 4, pp. 760–771, 2019.

[13] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2014, pp. 1891–1898.

[14] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *Proc. Int. Conf. Mach. Learn.*, pp. 448–456, 2015.

[15] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *Int. Conf. Learning Representations*, 2014.

[16] N. S. Keskar and R. Socher, “Improving generalization performance by switching from adam to sgd,” *arXiv preprint arXiv:1712.07628*, 2017.