Italian sign language alphabet recognition from surface EMG and IMU sensors with a deep neural network

Paolo Sernani, Iacopo Pacifici, Nicola Falcionelli, Selene Tomassini and Aldo Franco Dragoni

Information Engineering Department, Università Politecnica delle Marche, Via Brecce Bianche 12, 60131 Ancona, Italy

Abstract

The use of surface electromyography (EMG) and Inertial Measurement Unit (IMU) data emerged as a possible alternative to computer vision-based gesture recognition. As a consequence, the convenience of using such data in the automatic recognition of sign languages, a natural application of gesture recognition, has been investigated in scientific literature. Most of the methodologies and evaluations are based on traditional machine learning techniques, such as SVMs, relying on selected handcrafted features. Instead, leveraging on the findings about deep Long Short Term Memory (LSTM) architectures to process time series, we propose a deep LSTM-based neural network for the recognition of the Italian Sign Language alphabet with surface EMG and IMU data. To preliminary validate our methodology, we collected a dataset recording gesture samples with the Myo Gesture Control Armband. We obtained a 97% accuracy on the proposed dataset.

Keywords

Sign Language Recognition, Bidirectional LSTM, Long Short Term Memory, Deep Learning, Surface Electromyography, EMG, Inertial Measurement Unit, IMU, Italian Sign Language, LIS

1. Introduction

In the last three decades, automatic gesture recognition has been investigated in many applications domains. In fact, hand gestures are recognized as a natural, ubiquitous and meaningful part of communicating [1]. Therefore, extensive research has been devoted to making hand gestures a natural and effective mode of non-verbal communication with computer interfaces [2]. The possible applications are countless, including touchless interaction with smart objects [3], rehabilitation and personal health systems [4, 5], human-robot collaboration [6], interaction with smart home reasoning systems [7, 8], and many others.

Obviously, the automatic recognition of sign language gestures is an eminent application field for the advancements in gesture recognition. To this end, the earliest researches in computer vision [9] evolved with the use of depth sensors, such as those of the Microsoft Kinect [10] and Leap Motion [11]. An alternative methodology is emerging in recent years: the use of wearable devices with surface electromyography (EMG) and Inertial Measurement Unit (IMU) sensors [12]. Using EMG and IMU sensors has the disadvantage of forcing a user to wear the device (on both hands,

RTA-CSIT 2021: 4th International Conference Recent Trends and Applications In Computer Science And Information Technology, May 21–22, 2021, Tirana, Albania

p.sernani@univpm.it (P. Sernani);
S1091039@studenti.univpm.it (I. Pacifici);
n.falcionelli@pm.univpm.it (N. Falcionelli);
s.tomassini@pm.univpm.it (S. Tomassini);
a.f.dragoni@univpm.it (A.F. Dragoni)

CEUR Workshop Proceedings (CEUR-WS.org)

for complex gestures). However, it does not require a fixed camera which might be vulnerable to varying lighting conditions, in addition to having a limited range of vision and causing privacy issues.

In this regard, we present a deep learning methodology for the recognition of the Italian Sign Language (LIS) alphabet using EMG and IMU data. Specifically, this paper adds the following contributions to the state of the art about sign language gesture recognition:

- we propose a deep neural network architecture to classify the EMG and IMU data corresponding to the 26 letters of the LIS alphabet. We based our network on the bidirectional Long Short Term Memory (LSTM) architecture, as it has been already proven useful to process time series, e.g. in speech [13] and gesture recognition [14];
- we propose a dataset with 30 gesture samples for each letter of the LIS alphabet, collected to preliminary evaluate our approach. Each sample includes the data from the 8 EMG sensors and the IMU of the Myo Gesture Control Armband, a commercial wearable device designed to collect EMG signals and IMU data when moving the hand and the arm.

To guarantee the reproducibility of our approach, as well as encourage further developments of the research in this field, the experiments and the dataset are publicly available in two dedicated GitHub repositories.

The rest of the paper is structured as follows. Section 2 lists some studies related to the presented research. Section 3 explains the proposed approach, with the necessary background about the LSTM architecture, and describes the dataset collected to evaluate our method. Section 4 discusses a preliminary experimental evaluation of our neural network, explaining the setup of the experiments and presenting the results. Finally, Section 5 draws the conclusions of this research work.

2. Related Works

The use of EMG and IMU data for the recognition of sign language gestures has been validated by several studies. For example, Savur and Sahin [15] got 91% accuracy on the American Sign Language (ASL) alphabet, using a Support Vector Machine (SVM) classifier. Wu et al. [12] proposed the design of a wearable device and a feature selection method to collect EMG and IMU data for the recognition of gestures. They validated their proposal on the ASL gestures, getting a top accuracy of 96% with a comparison of traditional machine learning approaches (Nearest Neighbor, Naive Bayes, Decision Tree, and SVM). In [16] Abreu et al. evaluated the use of the Myo Armband for the Brazilian Sign Language alphabet by defining 20 SVM binary classifiers to recognize 20 letters, in a one-vs-all strategy. Similarly to these works, we use EMG and IMU data (from the Myo Armband) to recognize the letters of the LIS alphabet. However, instead of relying on traditional machine learning methods and feature selection, we propose a deep neural network, leveraging on a deep architecture to learn the gesture representation which allows the classification.

Recurrent Neural Networks, in particular those based on the LSTM and bidirectional LSTM architectures, have been validated for representing and classifying complex sequential data simultaneously, such as in modeling human gesture structure and temporal dynamics [14]. Some research works are presenting LSTM-based architectures for sign language recognition. For example, Liu et al. [17] propose to use the LSTM architecture to perform recognition by analyzing the trajectory of skeleton joints provided by the

Microsoft Kinect; Guo et al. [18] combine a 3D Convolutional Neural Network with the LSTM to classify gestures from videos, in a transfer-learning approach; Mittal et al. design a LSTM-based architecture to recognize words and sentences of the Indian Sign Language from Leap Motion data [19]. Similarly to these works, we also based our system on the LSTM architecture, but we rely on EMG and IMU data, instead of visual data. In the need of data to train our method, we synthetically augmented our dataset to preliminary validate our method, using data augmentation also to add intra-class variation in our samples and prevent overfitting.

3. Materials and Methods

Recurrent Neural Networks (RNN) use recurrent connections to model the flow of time in a sequence of data [20], and are therefore particularly suited to work with time series. LSTM are a type of RNN which are capable of learning long-time dependencies in the data. As we want to recognize gestures from a sequence of time-ordered EMG and IMU data, our system is based on the LSTM architecture. Moreover, we also collected a dataset to test the accuracy of the proposed system in the recognition of the LIS gestures.

3.1. LSTM and Bidirectional LSTM

LSTM is a well-known RNN architecture, proposed by Hochreiter and Schmidhuber [21]. As showed in Figure 1, the basic hidden unit of a LSTM network is composed of a self-recurrent cell, called memory cell, whose input/output is regulated by three multiplicative gates, i.e. the input gate, the output gate, and the forget gate. A LSTM layer is composed by a series of such units and the network interacts with the memory cells only by using the

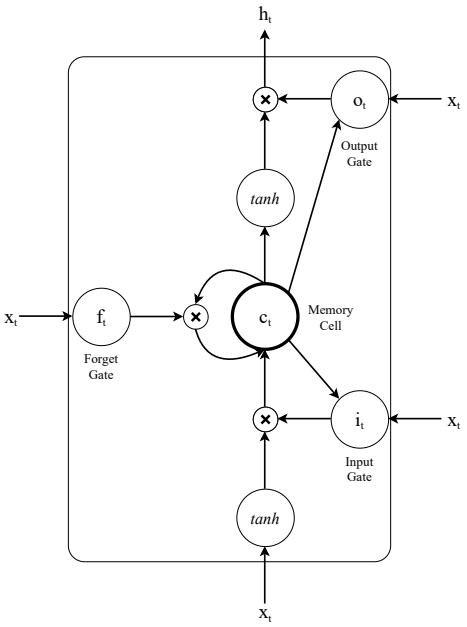

Figure 1: An LSTM unit, with the input/output of the memory cell regulated by the input, output, and forget gates.

gates.

As pointed out in [13], the output h_t at time point t of an LSTM hidden unit is regulated by the following equations:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

where i_t, f_t, o_t, and c_t are the activation vectors of the input gate, forget gate, output gate, and memory cell at time point t, σ is the sigmoid function, b denotes the bias of each gate/cell, and W are diagonal weight matrixes. The output vector y_t at time point t of an hidden layer is therefore given by:

$$y_t = W_{hy}h_t + b_y$$

where W_{hy} is the weight matrix and b_y the bias vector.

Traditional LSTMs, as RNNs in general, process input data in ascending temporal order. Therefore, their outputs is mostly based on previous context. However, when data is processed at once, as it might happen with the classification of gestures, the recognition of a pattern might be more effective with the use of future context as well. To this end, Bidirectional RNNs [22] and, specifically, Bidirectional LSTMs [20] have been proposed. The basic idea of such models is to present the training sequences both forwards and backwards, using two separate recurrent nets, which are connected to the same output layer.

Therefore, we based our deep neural network on the Bidirectional LSTM architecture, as the gesture are processed once done, taking advantage of both previous and future context.

3.2. Proposed Dataset

To evaluate the proposed architecture, we developed a dataset including all the 26 gestures of the LIS alphabet. Most of the letters of the alphabet is represented with static gestures, while the "G", "H", and "Z" are performed by moving the hand as well. We recorded 30 samples for each letter, building a dataset composed of 780 samples. The dataset is publicly available as a GitHub repository[1].

All the collected gesture were performed by the same person (male, 24 years old) wearing a Myo Gesture Control Armband[2] on his right arm, always in the same position. In fact, each sample of the dataset is composed of the raw data produced by the 8 EMG sensors and IMU of the Myo Armband. The time window for the acquisition of each sample was 2 seconds, sampling both the EMG and IMU data at 200 Hz. The subject was required to self-collect the samples with a desktop application that we developed specifically for the gesture acquisition.

Each data sample for each gesture representing a letter is included in a json file containing both the EMG and the IMU data. The EMG data is organized into an *emg* object including the following fields:

- *frequency*, i.e. the sampling frequency (in Hz) of the values from the EMG sensors. This value is 200 for all the samples;
- *data*, a 400 x 8 integer matrix. Each row is then an 8-dimensional array including the values from the 8 EMG sensors of the Myo Armband. Therefore, data is the time series of the values from the EMG sensors during the acquisition of the gesture.

Similarly, the IMU data of the sample is organized into an *imu* object with the following fields:

- *frequency*, i.e. the sampling frequency (in Hz) of the values from the IMU. This value is 200 for all the samples;
- *data*, a 400 elements length object array. Each object has three fields, namely gyroscope (an array composed by 3 floating point values), acceleration (an array composed by 3 floating point values), and rotation (an array composed by 4 floating point values).

In addition, each json file includes a timestamp, representing the date and time of the gesture acquisition, and the duration of each acquisition, which is 2000 for all the samples. The information about the acquisition duration and the sampling frequency are redundant in the current version of the dataset, as

[1] https://github.com/airtlab/An-EMG-and-IMU-Dataset-for-the-Italian-Sign-Language-Alphabet

[2] https://web.archive.org/web/20200528111822/https://support.getmyo.com/hc/en-us/articles/202648103-Myo-Gesture-Control-Armband-tech-specs

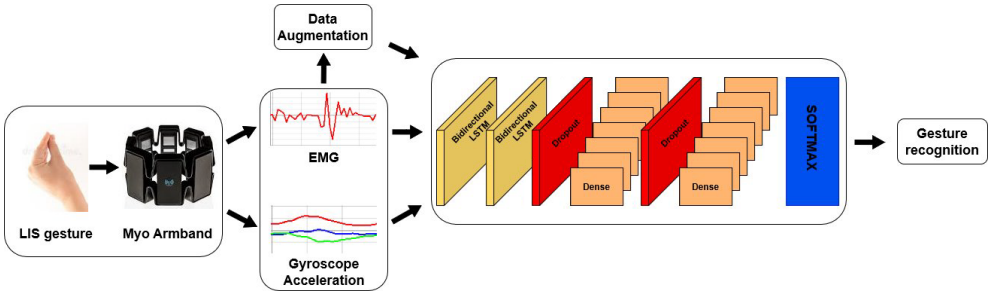

Figure 2: The proposed gesture recognition pipeline: the time series of EMG and IMU data are fed into a deep neural network for classification. To train the neural networks and prevent overfitting, both EMG and IMU data were synthetically augmented.

they are the same for all the gestures. However, this information might be useful in the future, when we might add samples varying the acquisition time window or the sampling frequency. The complete dataset specification is available in a dedicated open-access data paper [23].

3.3. System Architecture

Figure 2 depicts the architecture of the proposed gesture recognition system, used to identify the gestures of the LIS alphabet. The user performs the gesture wearing the Myo Armband; the data from the EMG sensors and the IMU are the input for our deep neural network, based on the Bidirectional LSTM architecture. The system labels the input data with one of the 26 letters of the alphabet, identifying the gesture made by the user. As explained in Section 4, to evaluate our system, we synthetically augmented the data in our dataset during the training process, trying to use more samples and reduce overfitting.

Table 1 lists all the layers included in our deep neural network. Among the available data, we used the 8 series with the values from the 8 EMG sensors of the Myo Armband. Concerning the IMU, we took the two 3-dimensional vectors with values from the accelerometer and the gyroscope. Therefore, each sample is fed into the network as a 400 x 14 matrix, i.e. there are 400 14-dimensional vectors for each samples. The first network layer is a bidirectional LSTM. It processes the input with 64 hidden units, returning in output 128 hidden state values (64 for the forward sequence, 64 for the backward sequence) for each of the 400 vectors in a sample. In fact, each hidden unit is configured to output a value for each vector in the sample matrix, as proposed by Graves et al [24] to stack multiple LSTM layers. Thus, the second layer is a also a bidirectional LSTM. However, being the last recurrent layer, each of the 32 hidden units returns a single value for the entire sample. Therefore, the output of the second layer is composed of 64 values (32 for the forward sequence, 32 for the backward one). A 50% dropout performs the diluition of the LSTM output, to prevent overfitting.

Table 1
The deep neural network model used for the gesture recognition. The total number of trainable parameters is 87,514.

Layer	Output Shape	Param #
Bi-LSTM	(400, 128)	40448
Bi-LSTM	(64)	41216
Dropout (0.5)	(64)	0
Fc1 (ReLU)	(64)	4160
Dropout (0.5)	(64)	0
Fc2 (Softmax)	(26)	1690

The sample classification is performed by a sequence of two fully connected layer. The first includes 64 hidden units, using the rectifier as the activation function. After another 50% dropout for regularization, the output is processed by the 26 units of the second fully connected layer. The softmax activation function of each unit computes the probability distribution over the 26 classes, i.e. the letters of the LIS alphabet.

4. Experimental Evaluation

We evaluate our model by collecting preliminary results on the proposed dataset. We actually want understand to which extent our deep neural network is a viable solution to recognize the LIS gestures based on EMG and IMU sensor data. As the collected dataset includes only 780 samples, which might be too few for a deep learning approach, we also applied data augmentation, with the twofold objective of testing with more data and prevent overfitting. Even if data augmentation has been proven useful to get general results [25], the experiments should be considered as early stage, and therefore inevitably suffer from some threats to validity.

4.1. Data Augmentation

To augment the proposed dataset, we apply the technique presented in [26]. Ohashi et al. point out that, during the gesture recognition with a wearable device such as the Myo Armband, the user is supposed to wear the device always with the same configuration (i.e. identical placement and rotation). In this way, the sensors would be attached to the user's arm in the same positions every time the device is used. However, a displacement is very likely to happen when detaching and attaching the device again. Therefore, samples with various rotation angles are desirable in the training data of a gesture recognition model. As the data from the accelerometer and the gyroscopse are 3D vectors, they can be easily rotated by multiplying with a rotation matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

Such transformation rotates the coordinate system of θ degrees, counterclockwise, around the x-axis. Ohashi et al also propose the following formulation to apply the same rotation to the data of 8 EMG sensors:

$$emg_i^{(\theta)} = \frac{f(d)emg_{i-\eta}^{(ori)} + f(1-d)emg_{i-\eta-1}^{(ori)}}{f(d) + f(1-d)}$$

$$\eta = \lfloor \theta/\phi \rfloor$$

$$\phi = 360/N$$

$$d = \theta/\phi - \eta$$

Here, $emg_i^{(\theta)}$ is the reading of the ith EMG sensor when rotating the armband of θ degrees; $emg_i^{(ori)}$ is the reading of the i-th sensor in the original data; N is the number of available EMG sensors; $f(d)$ is the polynomial function $f(d) = d^2$. Intuitively, if the rotation places the i-th sensor between the original positions of the j-th and $(j+1)$-th sensors, the reading of the i-th sensor in the rotated data is computed as the interpolation of the readings of the j-th and $(j+1)$-th sensors and the distance from those sensors.

Therefore, we apply such rotation technique to our data, given that, with this approach, Oshahi et al. got better performance than augmenting data with gaussian noise, with rotating data around all the three axis, and with linear interpolation. As in their work, we rotate the data with the angles in the following set:

$$\{-30^\circ, -22.5^\circ, -15^\circ, -7.5^\circ, 7.5^\circ, 15^\circ, 22.5^\circ, 30^\circ\}$$

By rotating the data, we get 780 samples for

each angle, adding the 6,240 synthetic samples to the 780 originally collected with the Myo Armband.

4.2. Experimental Setup

We tested the proposed deep neural network on the original dataset, as well as on the augmented dataset. We applied a stratified shuffle split cross-validation scheme to validate the accuracy of our model. To this end, we firstly repeated a randomized 80-20 split 5 times, using the 80% of the data as the training set, and the 20% as the test set, preserving the percentage of samples from each class, in each split. The 12.5% of the training data, i.e. the 10% of the entire dataset, was used as validation data for the training of the neural network. Then, we repeated the same randomized split 30 times on each dataset, to collect more general results.

We used the Root Mean Square Propagation (RMSProp) optimizer to minimize the Categorical Cross-Entropy loss function during the training of the neural network. The number of training epochs varied for each split, as we early stopped the training after 5 epochs without an improvement on the minimum validation loss, restoring the weights corresponding to the best validation loss. Table 2 shows the number of training epochs in each split, in the 5 split experiments. For the 30 split experiments the mean number of training epochs was 42.77 (± 9.01) for the original dataset, and 37.67 (± 7.80) on the augmented dataset. The batch size was 32 samples in each split of each experiment.

A Jupyter notebook with the described experiments is available in a GitHub public repository[3], in order to guarantee the reproducibility of the tests. The tests ran on Google Colab with the GPU runtime, using Keras 2.4.3, TensorFlow 2.4.1, and scikit-learn 0.22.2.post1.

[3] https://github.com/airtlab/italian-sign-language-recognition/

Table 2
Number of training epochs in each split (s1-s5) of the 5 split experiments, with and without Data Augmentation (DA).

	s1	**s2**	**s3**	**s4**	**s5**
without DA	59	34	41	42	54
with DA	34	54	36	45	34

4.3. Results

Table 3 shows the prediction accuracy on the test set obtained by repeating 5 times the stratified shuffle split of the dataset. With the 780 samples of the original dataset, the mean accuracy is 57.44% with a standard deviation of 5.46% over the 5 splits of the experiment. In other words, around half of the test samples gets misclassified. In fact, using only 576 samples for the network training (with 78 samples used as validation data) results in a poor performance of our model.

Instead, with the 7,020 samples of the augmented dataset, the mean accuracy increases to 97.36%, and the standard deviation decreases to 0.62% over the 5 splits. Using 4,914 samples for training (with 702 samples used for validation) significantly improves the performance of our model. The lower standard deviation shows that the model trained on the augmented dataset exhibits a better generalization. Intuitively, most of the misclassification errors occurs with gestures which look similar. For example, in the first split, the "V" is erroneously identified as the "U" 9 times and as the "F" one time, while other 44 samples are correctly identified. Similary, 3 "U" samples are wrongly identified as "V". In the same split, the "W" is misclassified only one time, being identified as the "V".

The results are similar when repeating the tests on 30 random stratified shuffle splits of the dataset, as showed in Table 4. The mean value of accuracy is 58.69% (± 4.37%) for the original dataset and 97.07% (± 1.32%) on the

Table 3
Accuracy results with and without Data Augmentation (DA). The table includes the accuracy value in each random split of the dataset, obtained with the stratified shuffle split cross-validation scheme.

	Split 1	Split 2	Split 3	Split 4	Split 5	Mean
without DA	62.18%	51.92%	51.92%	55.77%	65.38%	**57.44 ± 5.46%**
with DA	97.22%	98.36%	96.94%	97.72%	96.58%	**97.36 ± 0.62%**

Table 4
Mean number of training epochs and mean accuracy on 30 random stratified shuffle splits, with and without Data Augmentation (DA).

	Epoch #	Accuracy
without DA	42.77 ± 9.01	58.69 ± 4.37%
with DA	37.67 ± 7.80	97.07 ± 1.32%

augmented dataset. Therefore, both in the experiments with 5 splits and 30 splits, the training on augmented data is more stable than with the original data, resulting in a lower standard deviation on the test accuracy. Moreover, the tests did not highlight any significant difference in the recognition of static gestures (most of the letters) with respect to the dynamic ones ("G", "H", and "Z"), scoring similar class-wise precision and recall values.

These preliminary results encourage the use of wearable devices equipped with EMG and IMU sensors to execute the recognition of the LIS with deep neural networks. Most of the samples gets correctly identified by our LSTM-based model. As expected, the data augmentation improves the performance, and our model gets better results with more data, highlighting the need of expanding the collected dataset.

4.4. Threats to validity

Being in early stage, the presented research inevitably suffers from some threats to validity. Concerning the collected dataset, all the gesture samples were performed by the same subject. Samples from more subjects are necessary to get more general conclusions. Moreover, we arbitrary fixed the time window for the gesture acquisition to 2 seconds. Such time window is worth of further research, as this time might vary from person to person and also for more complex gestures.

Concerning the presented results, we built our model on the results of existing literature about LSTMs to process time series, especially in speech and gesture recognition. However, a systematic study on alternative models as well as a comparison on more datasets should be performed to get more results, and therefore validate our method.

5. Conclusions

We presented a deep learning approach for the recognition of the LIS alphabet, based on surface EMG and IMU data. Specifically, we developed a deep neural network based on the bidirectional LSTM architecture. To validate our method, we built a dataset including 30 gesture samples for each letter of the alphabet. The gestures were recorded from the 8 EMG sensors and the IMU of the Myo Armband. To ensure the proper training of our model, with enough samples, we used data augmentation, simulating the rotation of the armband. The results are preliminary, but promising: on the augmented dataset, our model got 97% accuracy, showing few classification errors on very similar gestures. The source code of the experiments and the dataset are available as public GitHub repositories, to guarantee the reproducibility of the tests. Moreover, the public dataset is available for further tests.

The presented research is in early stage, since a systematic study of alternative deep

neural network configurations and architectures, as well as comparison on other datasets are necessary to fully validate our approach. Also the proposed dataset can be improved. In the current versions, all the samples were performed by the same subject, using a fixed time window. More subjects and different acquisition setups should be included to expand the dataset. Finally, the evaluation was based on one hand gestures. Tests on two hands gestures, combining data from two devices, are necessary to understand the viability of the proposed method in the real world.

References

[1] S. D. Kelly, S. M. Manning, S. Rodak, Gesture gives a hand to language and learning: Perspectives from cognitive neuroscience, developmental psychology and education, Language and Linguistics Compass 2 (2008) 569–588. doi:10.1111/j.1749-818X.2008.00067.x.

[2] J. S. Sonkusare, N. B. Chopade, R. Sor, S. L. Tade, A review on hand gesture recognition system, in: 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 790–794. doi:10.1109/ICCUBEA.2015.158.

[3] L. Montanaro, P. Sernani, A. F. Dragoni, D. Calvaresi, A touchless human-machine interface for the control of an elevator, in: Proceedings of the 2nd International Conference on Recent Trends and Applications in Computer Science and Information Technology, volume 1746 of *CEUR Workshop Proceedings*, 2016, pp. 58–65. URL: http://ceur-ws.org/Vol-1746/paper-10.pdf.

[4] A. E. F. Da Gama, T. M. Chaves, L. S. Figueiredo, A. Baltar, M. Meng, N. Navab, V. Teichrieb, P. Fallavollita, MirrARbilitation: A clinically-related gesture recognition interactive tool for an AR rehabilitation system, Computer Methods and Programs in Biomedicine 135 (2016) 105–114. doi:10.1016/j.cmpb.2016.07.014.

[5] N. Falcionelli, P. Sernani, A. Brugués, D. N. Mekuria, D. Calvaresi, M. Schumacher, A. F. Dragoni, S. Bromuri, Indexing the event calculus: Towards practical human-readable personal health systems, Artificial Intelligence in Medicine 96 (2019) 154–166. doi:10.1016/j.artmed.2018.10.003.

[6] D. Calvaresi, A. Vincentini, A. Di Guardo, D. Cesarini, P. Sernani, A. F. Dragoni, Exploiting a touchless interaction to drive a wireless mobile robot powered by a realtime operating system, in: Proceedings of the 2nd International Conference on Recent Trends and Applications in Computer Science and Information Technology, volume 1746 of *CEUR Workshop Proceedings*, 2016, pp. 1–10. URL: http://ceur-ws.org/Vol-1746/paper-01.pdf.

[7] D. N. Mekuria, P. Sernani, N. Falcionelli, A. F. Dragoni, Reasoning in multi-agent based smart homes: A systematic literature review, in: Ambient Assisted Living, Springer International Publishing, Cham, 2019, pp. 161–179. doi:10.1007/978-3-030-05921-7_13.

[8] D. N. Mekuria, P. Sernani, N. Falcionelli, A. F. Dragoni, Smart home reasoning systems: a systematic literature review, Journal of Ambient Intelligence and Humanized Computing (2019) 1–18. doi:10.1007/s12652-019-01572-z.

[9] T. Starner, J. Weaver, A. Pentland, Real-time american sign language recognition using desk and wearable computer based video, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 1371–1375. doi:10.1109/34.735811.

[10] E. Gani, A. Kika, Albanian sign language (AlbSL) number recognition from both hand's gestures acquired by kinect sensors, CoRR abs/1608.02991 (2016). URL: http://arxiv.org/abs/1608.02991.

[11] D. Naglot, M. Kulkarni, Real time sign language recognition using the

leap motion controller, in: 2016 International Conference on Inventive Computation Technologies (ICICT), volume 3, 2016, pp. 1–5. doi:10.1109/INVENTIVE.2016.7830097.

[12] J. Wu, L. Sun, R. Jafari, A wearable system for recognizing american sign language in real-time using IMU and surface EMG sensors, IEEE Journal of Biomedical and Health Informatics 20 (2016) 1281–1290. doi:10.1109/JBHI.2016.2598302.

[13] A. Graves, N. Jaitly, A. Mohamed, Hybrid speech recognition with deep bidirectional LSTM, in: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, 2013, pp. 273–278. doi:10.1109/ASRU.2013.6707742.

[14] C. Li, C. Xie, B. Zhang, C. Chen, J. Han, Deep fisher discriminant learning for mobile hand gesture recognition, Pattern Recognition 77 (2018) 276–288. doi:10.1016/j.patcog.2017.12.023.

[15] C. Savur, F. Sahin, Real-time american sign language recognition system using surface EMG signal, in: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015, pp. 497–502. doi:10.1109/ICMLA.2015.212.

[16] J. G. Abreu, J. M. Teixeira, L. S. Figueiredo, V. Teichrieb, Evaluating sign language recognition using the Myo Armband, in: 2016 XVIII Symposium on Virtual and Augmented Reality (SVR), 2016, pp. 64–70. doi:10.1109/SVR.2016.21.

[17] T. Liu, W. Zhou, H. Li, Sign language recognition with long short-term memory, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 2871–2875. doi:10.1109/ICIP.2016.7532884.

[18] D. Guo, W. Zhou, H. Li, M. Wang, Hierarchical LSTM for sign language translation, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 6845–6852. URL: https://ojs.aaai.org/index.php/AAAI/article/view/12235.

[19] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, B. B. Chaudhuri, A modified LSTM model for continuous sign language recognition using leap motion, IEEE Sensors Journal 19 (2019) 7056–7063. doi:10.1109/JSEN.2019.2909837.

[20] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Networks 18 (2005) 602–610. doi:10.1016/j.neunet.2005.06.042.

[21] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[22] M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (1997) 2673–2681. doi:10.1109/78.650093.

[23] I. Pacifici, P. Sernani, N. Falcionelli, S. Tomassini, A. F. Dragoni, A surface electromyography and inertial measurement unit dataset for the italian sign language alphabet, Data in Brief 33 (2020) 106455. doi:10.1016/j.dib.2020.106455.

[24] A. Graves, A. Mohamed, G. E. Hinton, Speech recognition with deep recurrent neural networks, CoRR abs/1303.5778 (2013). URL: http://arxiv.org/abs/1303.5778.

[25] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of Big Data 6 (2019) 1–48. doi:10.1186/s40537-019-0197-0.

[26] H. Ohashi, M. O. A. Al-Naser, S. Ahmed, T. Akiyama, T. Sato, P. Nguyen, K. Nakamura, A. Dengel, Augmenting wearable sensor data with physical constraint for DNN-based human-action recognition, in: Time Series Workshop @ ICML, 2017, pp. 1–5. URL: https://www.dfki.de/fileadmin/user_upload/import/9676_TSW2017_paper_9.pdf.