ACTA IMEKO
ISSN: 2221-870X
December 2021, Volume 10, Number 4, 97 - 102

Gesture recognition of sign language alphabet with a convolutional neural network using a magnetic positioning system

Emanuele Buchicchio[1], Francesco Santoni[1], Alessio De Angelis[1], Antonio Moschitta[1], Paolo Carbone[1]

[1] *Department of Engineering University of Perugia, Italy*

ABSTRACT
Gesture recognition is a fundamental step to enable efficient communication for the deaf through the automated translation of sign language. This work proposes the usage of a high-precision magnetic positioning system for 3D positioning and orientation tracking of the fingers and hands palm. The gesture is reconstructed by the MagIK (magnetic and inverse kinematics) method and then processed by a deep learning gesture classification model trained to recognize the gestures associated with the sign language alphabet. Results confirm the limits of vision-based systems and show that the proposed method based on hand skeleton reconstruction has good generalization properties. The proposed system, which combines sensor-based gesture acquisition and deep learning techniques for gesture recognition, provides a 100% classification accuracy, signer independent, after a few hours of training using transfer learning technique on well-known ResNet CNN architecture. The proposed classification model training method can be applied to other sensor-based gesture tracking systems and other applications, regardless of the specific data acquisition technology.

Section: RESEARCH PAPER

Keywords: gesture recognition; sign language; machine learning; CNN

Citation: Emanuele Buchicchio, Francesco Santoni, Alessio De Angelis, Antonio Moschitta, Paolo Carbone, Gesture recognition of sign language alphabet with a convolutional neural network using a magnetic positioning system, Acta IMEKO, vol. 10, no. 4, article 17, December 2021, identifier: IMEKO-ACTA-10 (2021)-04-17

Section Editors: Umberto Cesaro and Pasquale Arpaia, University of Naples Federico II, Italy

Received October 15, 2021; **In final form** December 4, 2021; **Published** December 2021

Corresponding author: Emanuele Buchicchio, e-mail: emanuele.buchicchio@studenti.unipg.it

1. INTRODUCTION

Sign language recognition (SLR) is a research area that involves gesture tracking, pattern matching, computer vision, natural language processing, linguistics, and machine learning [1]. The final goal of SLR is to develop methods and algorithms to build an SRL system (SLRS) capable of identifying signs, decoding their meaning, and producing some output that the intended receiver can understand (Figure 1).

The general SLR problem includes the following tasks:

1) letter/number sign gesture recognition,
2) word sign gesture recognition, and
3) sentence-level sign language translation

Available literature surveys [2]-[5] report that recent research achieved accuracy in the range of 80–100% for the first two tasks using vision-based and sensor-based approaches.

In this paper, we compare the performance of the two systems we developed: a vision-based system and a hybrid system with sensor-based data acquisition and vision-based classification stages.

1.1. SLRS Performance Assessment

In the instrumentation and measurement field, machine learning is used for processing indirect measurement results. An *indirect measurement* is defined in [6] as a “method of measurement in which the value of a quantity is obtained from measurements made by direct methods of measurement of other quantities linked to the measurand by a known relationship.” In the machine learning (ML) common jargon [7], the quantities that can be measured with a direct method are denoted as features x_1, x_2, …, x_n, and the measurand as y. The measurand y is linked to features by a functional relationship $y=f(x_1, x_2, …, x_n)$. The process of estimating f is known as “training.” In the training process, the ML model is trained with the given dataset to find the best possible approximation according to the selected optimality criterion. The trained model produces an estimation of y in response to the vector $x=(x_1, x_2, …, x_n)$.

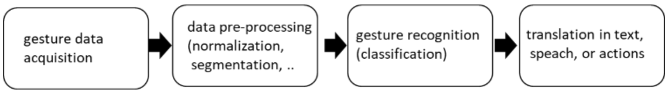

gesture data acquisition
data pre-processing (normalization, segmentation, ..
gesture recognition (classification)
translation in text, speach, or actions

Figure 1. Block diagram of a sign language recognition system (SLRS).

In the case of classification systems, the measurand y is the class to which an input vector x belongs. The most widely used performance metric for gesture SLRS is classification accuracy (defined as the ratio of correct predictions over the total predictions). In this work, accuracy was adopted both for model benchmark and as model optimality criterion.

1.2. Sign Language

Sign language (SL) is defined as "any means of communication through bodily movements, especially of the hands and arms, used when spoken communication is impossible or not desirable" [8]. Modern sign language originated in the 18th century when Charles-Michel de l'Épée developed a system for spelling out French words with a manual alphabet and expressing whole concepts with simple signs. Other national sign languages were developed from this system and became an essential means of communication among the hearing-impaired and deaf communities. According to the World Federation of the Deaf, today exist over 200 sign languages used by 70 million deaf [9].

Sign language involves using facial expressions and different body parts, such as arms, fingers, hands, head, and body. One class of sign languages, also known as fingerspelling, is limited to a set of manual signs that represent the symbols of the letters of an alphabet performed with one hand [10]. The ASL signs of the alphabet letters are shown in Figure 2.

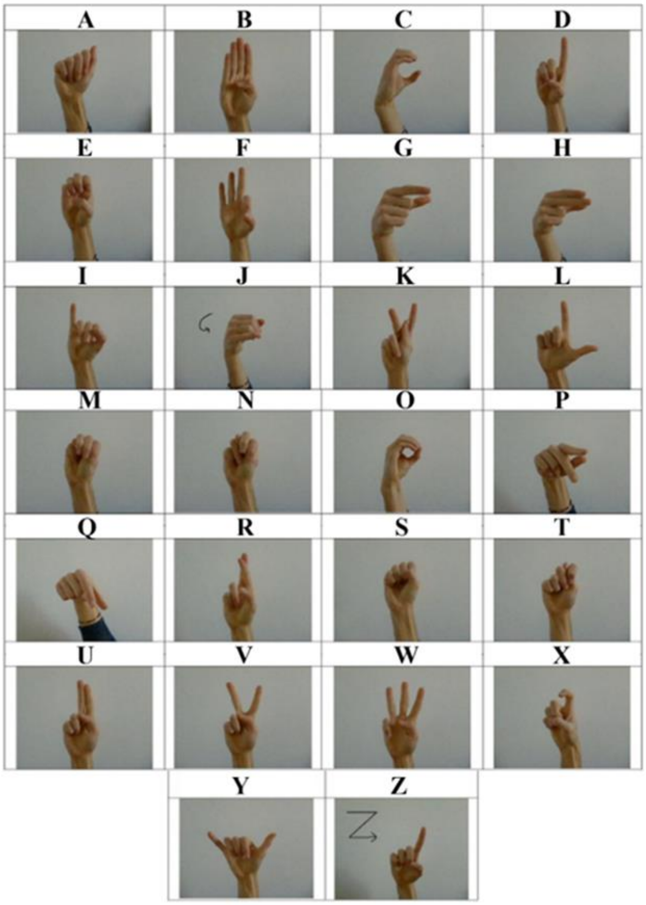


Figure 2. Letters of the American Sign Language (ASL) alphabet [11].

1.3. Vision-Based vs. Sensor-Based Approaches for Hands Tracking and Gesture Recognition

Many common devices and applications rely on tracking hands, fingers, or handheld objects. Specifically, smartphones and smartwatches track 2D finger position, a mouse tracks 2D hand position, and augmented reality devices like the Microsoft HoloLens 2 track the 3D pose of the finger. In addition to SLR, many other applications rely on hand gesture recognition such as augmented reality [12], assistive technology [13], [14], collaborative robotics [15], telerobotics [16], home automation [17], infotainment systems [18], [19], intelligence and espionage [20] and many others [21].

In this paper, we focused on recognizing static hand gestures associated with the letters of the alphabet for fingerspelling. Both computer-vision-based and sensor-based approaches were implemented for sign language alphabet recognition. Hand features extraction is a significant challenge for vision-based systems [11] because extraction is affected by many factors, such as lighting conditions, complex backgrounds in the image, occlusion, and skin color. Sensor-based gesture recognition systems are commonly implemented as gloves featuring various types of sensors. Sensor-based approaches have the advantage of simplifying the detection process and can help make the gesture recognition system less dependent on input devices. On the other hand, a disadvantage of sensor-based systems is that they can be expensive and too invasive for real-world deployment.

2. VISION-BASED SIGN LANGUAGE GESTURE RECOGNITION

Machine learning techniques are widely adopted for gesture classification tasks. Various public datasets are available for system performance assessment and benchmark. The American Sign Language MNIST Dataset [22], a flavor of the classic MNIST dataset [23], created for sign language gesture, is often used as a baseline. Other more complex datasets such as [24], [25] are also available.

2.1. Classic Machine Learning and Convolutional Neural Network on MNIST Dataset

The American Sign Language MNIST Dataset is in a tabular format similar to the original MNIST dataset. Each row in the CSV file has a label and 784 pixels values ranging from 0-255, representing a single 28 × 28 pixels greyscale image. In total, there are 27,455 training cases and 7,172 tests cases in this dataset. The classification accuracy was selected as the primary metric for models' performance assessment and benchmarking with other published comparable works.

Two different models were trained to accomplish the letter/number gesture recognition task from static images using two different approaches: a classic ML model and a deep neural network (Figure 3).

The first model was selected among many model candidates obtained by applying different combinations of features engineering techniques, ML algorithms, and ensemble methods using the Automated ML (AutoML) service of Azure Machine Learning. Azure Machine Learning [26] is a cloud-based platform that provides tools for automation and orchestration of all training, scoring, and comparison operations. AutoML tests hundreds of models in a few hours with parallel job execution with no human interaction after the initial experiment and remote compute target cluster setup. The experiment generates many models that achieve 100% classification accuracy. Among

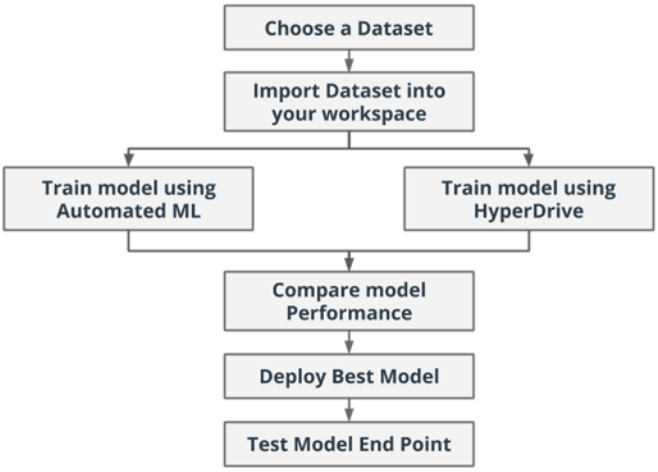
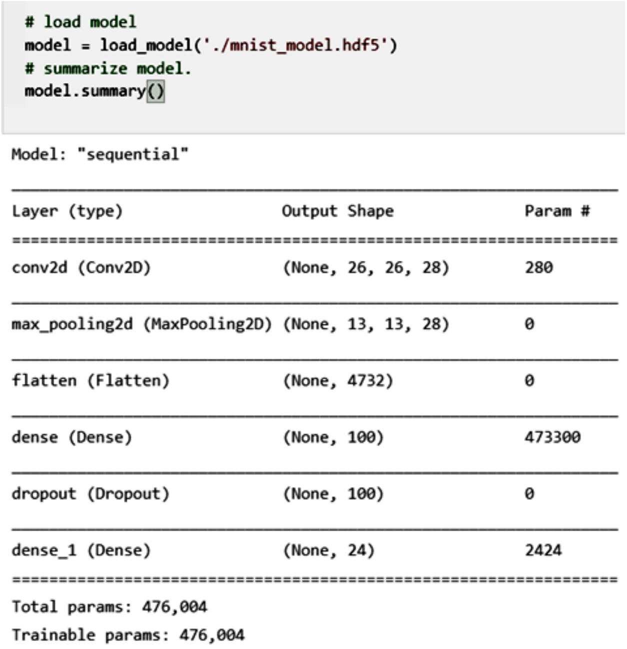

Choose a Dataset
Import Dataset into your workspace
Train model using Automated ML
Train model using HyperDrive
Compare model Performance
Deploy Best Model
Test Model End Point

Figure 3. Workflow for the comparison of various machine learning models for static gesture recognition using Azure SKD, AutoML and HyperDrive for operations automation.



```
# load model
model = load_model('./mnist_model.hdf5')
# summarize model.
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 28)        280
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 28)        0
_________________________________________________________________
flatten (Flatten)            (None, 4732)              0
_________________________________________________________________
dense (Dense)                (None, 100)               473300
_________________________________________________________________
dropout (Dropout)            (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 24)                2424
=================================================================
Total params: 476,004
Trainable params: 476,004
```

Figure 4. Deep CNN model architecture.

them, the "Logistic Regression" based model has a smaller memory footprint at runtime.

The second model was created with a minimal custom convolutional neural network (CNN) architecture (2D convolution, max pooling, flatten, dense layer, dropout, dense) commonly used for simple deep learning image recognition tasks (Figure 4). The model was built and trained with the Keras library. Model hyperparameters such as the number of neurons in layers, batch size, the number of training epochs, and dropout percentage were tuned using the HyperDrive service from Azure Machine Learning. The best scoring model achieves a classification accuracy score of 99.99%.

The best models from the two training pipelines were deployed as web services for production usage.

The (zipped) size of the CNN model is about 17 MB when the logistic regression model size is only 0.8 MB. Simple and lightweight models should be preferred if there is no performance penalty.

2.2. Vision-based Classification Accuracy

The 100% accuracy was confirmed after deployment with test cases from the American Sign Language MNIST.

Simple classic ML models could not recognize gestures in realistic images with variable backgrounds and light conditions. The CNN model scores over 90% accuracy on a subset of the "ASL Alphabet" [24] image dataset that includes more "realistic" light and background conditions. However, while deployed as a web service, the performance on image stream from a live camera was not satisfactory for production usage in challenging conditions such as partial line of sight obstruction, presence of shadows in the image, and confusing backgrounds like in the test case of ASL Alphabet Test dataset [25].

3. SENSOR-BASED GESTURE RECOGNITION WITH DEEP CNN ON VISUAL GESTURE REPRESENTATION

Our experiment with a vision-based approach confirms both performance and limitation described in other works. Given the result of our experiments and other works, in this paper, we propose an SLRS system that combines a sensor-based approach in the acquisition stage and computer vision techniques in the gesture recognition stage (Figure 5).

3.1. Hand Tracking with Magnetic Position System (MPS)

The magnetic Positioning System (MPS) described in [27] is immune from many problems that affect computer vision techniques such as occlusion, light condition, shadows, skin colors.

The MPS is composed of transmitting nodes and receiving nodes. The transmitting nodes are mounted on the fingers and hand to be tracked (Figure 6), whereas the receiving nodes are placed at known positions on the sides of the operational volume. An advantage of the sensor-based systems is that they are not sensitive to illumination conditions and the other factors affecting vision-based systems. Furthermore, MPS can also operate in the presence of obstructions caused by objects or body parts. Therefore, the proposed approach enables robust and reliable tracking of the hand and fingers. It is thus suitable for SLR and the other applications of hand gesture recognition, such as human-machine interaction, virtual and augmented reality, robotic telemanipulation, and automation.

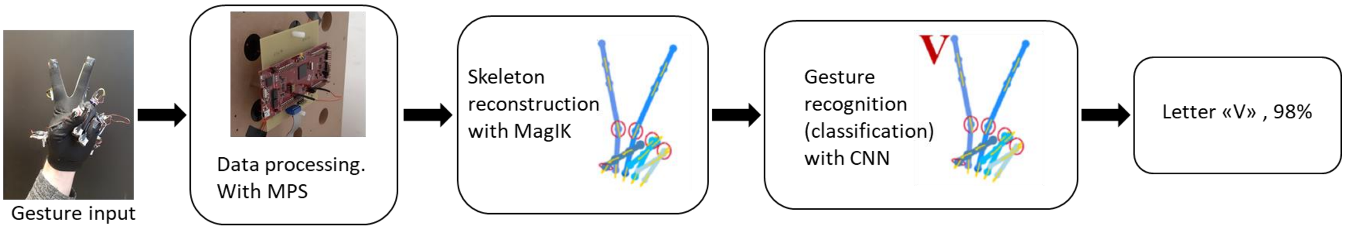

Gesture input
Data processing. With MPS
Skeleton reconstruction with MagIK
Gesture recognition (classification) with CNN
V
Letter «V» , 98%

Figure 5. Proposed SLRS with sensor-based data acquisition and vision-based gesture recognition.

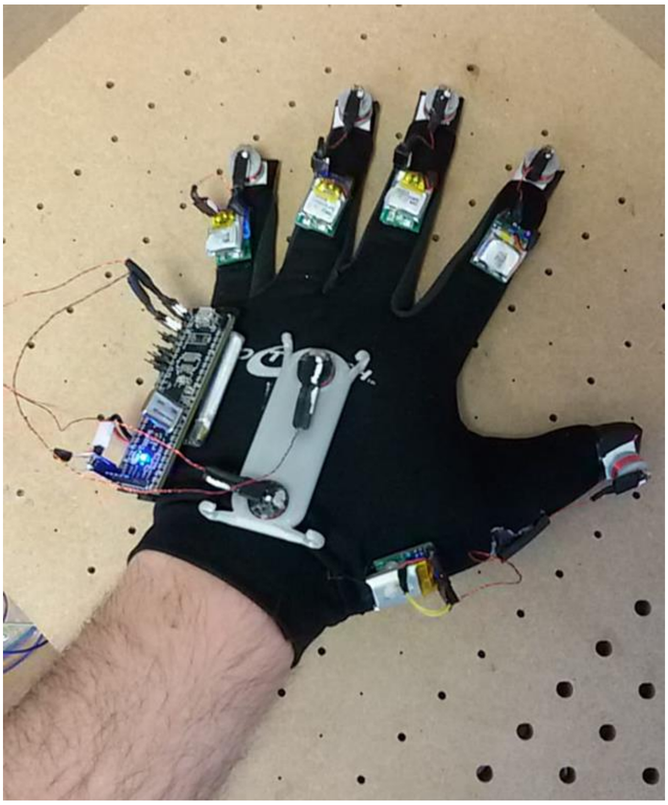


Figure 6. MPS transmitting coils mounted on a wearable glove.

3.2. Gesture Recognition Using Skeleton Reconstruction

Classic machine learning models can achieve 100% accuracy on static sign language recognition tasks on laboratory datasets like [24]. CNN deep learning models score high accuracy (over 90%) on realistic images. Classic machine learning models can achieve 100% accuracy on static sign language recognition tasks on laboratory datasets. CNN deep learning models score high accuracy (over 90%) on realistic images with variable light. However, these high performances are not robust and cannot be easily replicated in real-world operating conditions.

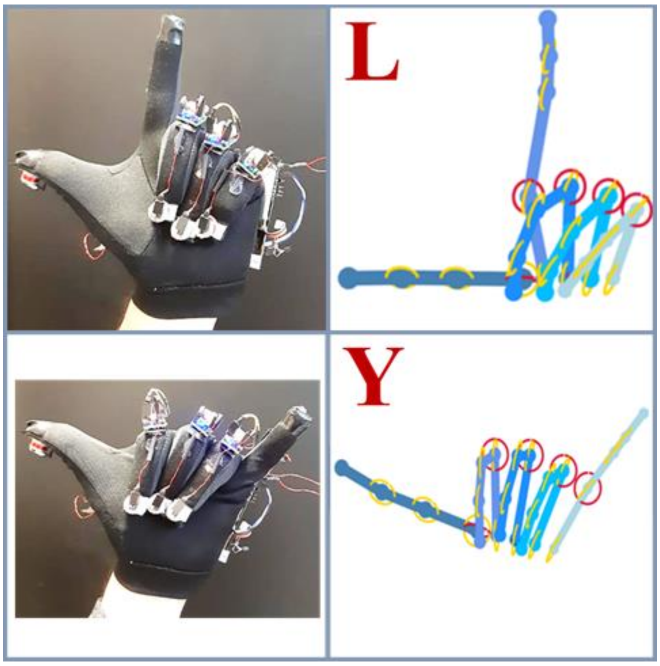


Figure 7. Examples of ASL letters (Y and L) articulated while wearing the glove, and their respective reconstructions obtained through the kinematic model and MagIK technique.

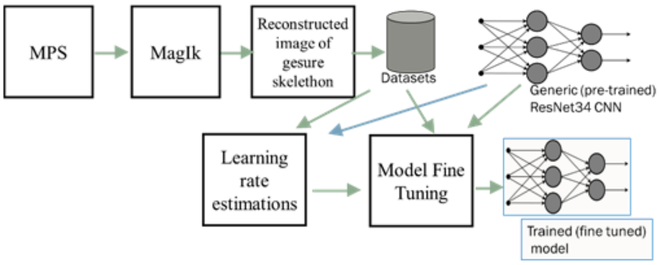

MPS
MagIk
Reconstructed image of gesure skelethon
Datasets
Generic (pre-trained) ResNet34 CNN
Learning rate estimations
Model Fine Tuning
Trained (fine tuned) model

Figure 8. Training pipeline for ResNet43 CNN with transfer learning.

In our paper [11], we demonstrated that training the classification model on data from a tracking system gives substantial advantages in terms of robustness to environmental conditions and signer variability.

The hand gesture is reconstructed using the technique illustrated in [28], with the improvements added in [11], which we called MagIK (magnetic and inverse kinematics). The method, with some empirical modification introduced in the model to optimize the reconstruction of the gesture among different test subjects, allows reconstructing the movement of the hand with 24 degrees of freedom (DOF). Positions and orientations of all the magnetic nodes estimated by the MPS are sent to a kinematic model of the hand, to obtain the position and flexion of each joint and the position and orientation of the whole hand with respect to the MPS reference frame. As the last step, MagIK produces a visual representation, such as the examples shown in Figure 7. We call this technique “skeleton reconstruction”.

3.3. Efficient Deep CNN Training for Sign Language Recognition

Many pre-trained deep learning models are proven to be adequate for image/video classification tasks. We chose the ResNet34 CNN because the ResNet (residual network) architecture achieves good results in image classification tasks and is relatively fast to train [29].

Figure 8 illustrates the training pipeline implemented with PyTorch and FastAI [30] library. Transfer learning approaches allow fast training of the deep CNN (ResNet34) model.

The optimal learning rate for training was estimated with the Cyclical Learning Rates method [31] to avoid time-consuming multiple runs to perform hyperparameters sweeps.

The rules of thumb for the selection of learning rate value from [31] are:

1) one order of magnitude less than where the minimum loss was achieved; and
2) the last point where the loss was clearly decreasing.

The Loss estimation plot (Figure 9) produced by the algorithm implementation in the FastAI library suggested a learning rate in the range $10^{-2} – 10^{-3}$.

Model fine-tuning was performed using FastAI API with a sequence of freeze, fit-one-cycle, unfreeze, and fit-one-cycle operations using the «discriminative learning rate» method. The training continued until error rate, validation loss, and training loss converged to zero after four epochs (Figure 10).

3.4. Gesture Classification Inference with MPS

The trained model, after the fine-tuning process, was developed in an inference pipeline (Figure 11) that takes the

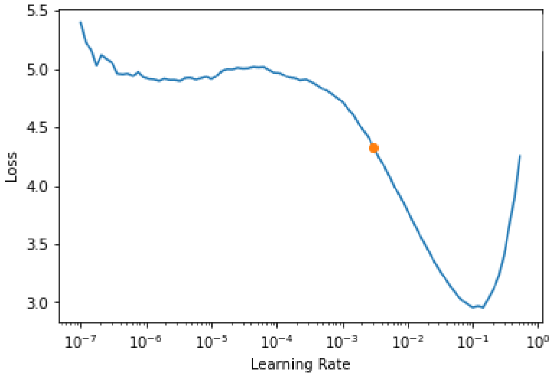


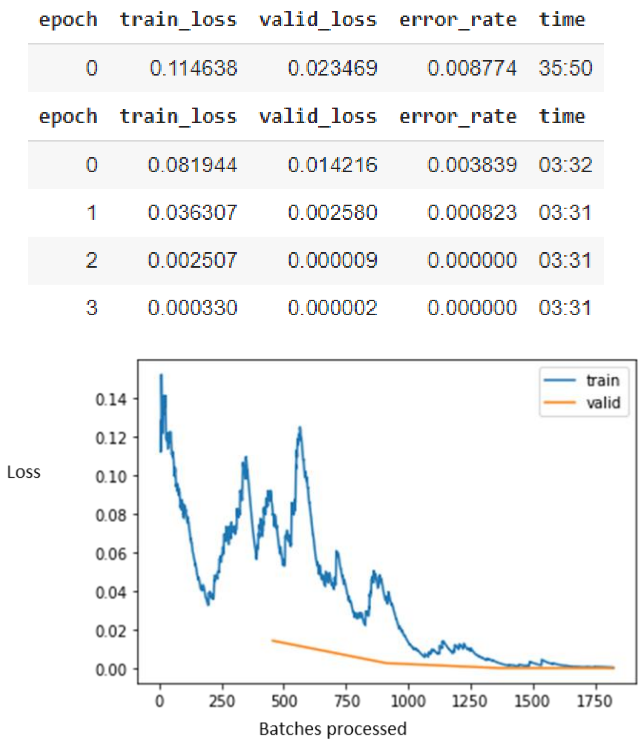
Figure 9. Loss estimation plot against learning rate values for optimal learning rate selection. The optimal value for training is in range 10^{-2} – 10^{-3}.

output generated by MPS control software and, for each acquired frame:

1) Reconstructs the gesture using MagIK model kinematic model,
2) Exports the visual representation as a bitmap image,
3) Feeds the CNN model with the generated gesture image and get the array of confidence values associated with each class in the training dataset, and
4) Printouts the label of the sign class with the highest confidence value.

4. CONCLUSIONS

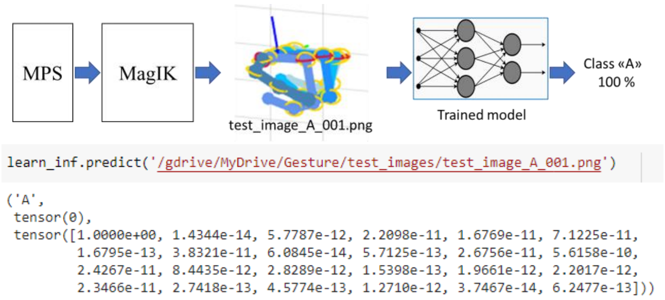
Classic machine learning models can only achieve 100% accuracy on static sign language recognition tasks on laboratory datasets [22]. Deep CNN models can accomplish the task with over 90% accuracy also on more realistic images [24]. However, these high performances are not robust and cannot be replicated in real-world operating conditions. Combining sensor-based acquisition, visual reconstruction of the skeleton, and a deep CNN classification model, the proposed system achieves 100% inference accuracy on gestures performed by different people after a few epochs of training. We cannot achieve 100% accuracy with classic machine learning in comparable experimental conditions.



epoch	train_loss	valid_loss	error_rate	time
0	0.114638	0.023469	0.008774	35:50

epoch	train_loss	valid_loss	error_rate	time
0	0.081944	0.014216	0.003839	03:32
1	0.036307	0.002580	0.000823	03:31
2	0.002507	0.000009	0.000000	03:31
3	0.000330	0.000002	0.000000	03:31

Figure 10. Loss and error rate values recorded during the training process.



```
learn_inf.predict('/gdrive/MyDrive/Gesture/test_images/test_image_A_001.png')
```

```
('A',
 tensor(0),
 tensor([1.0000e+00, 1.4344e-14, 5.7787e-12, 2.2098e-11, 1.6769e-11, 7.1225e-11,
         1.6795e-13, 3.8321e-11, 6.0845e-14, 5.7125e-13, 2.6756e-11, 5.6158e-10,
         2.4267e-11, 8.4435e-12, 2.8289e-12, 1.5398e-13, 1.9661e-12, 2.2017e-12,
         2.3466e-11, 2.7418e-13, 4.5774e-13, 1.2710e-12, 3.7467e-14, 6.2477e-13]))
```

Figure 11. Inference pipeline with MPS and skeleton reconstruction and an example of execution from Jupyter Notebook python environment.

The sensor-based approach is immune from many problems that affect computer vision techniques such as occlusion, light condition, shadows, skin colors. Building a gesture recognizer on top of a tracking system, instead of direct classification from a sensor stream, can help make the gesture recognition system less dependent on input devices. Skeleton tracking allows for good generalization: system performances are robust across different sign performers and classifications do not rely on specific hand characteristics.

The classification method implemented in this work can be applied to almost any sensor-based dataset: the only requirement is to provide a convenient visual representation of input data to be used both in training and inference. After replacing the MagIK with another method suitable for the specific application, other stages of the training pipeline and inference pipeline do not need any change and can be directly used for many other applications.

REFERENCES

[1] H. Cooper, B. Holt, R. Bowden, Sign language recognition. In Visual analysis of humans; moeslund, T., Hilton, A., Krüger, V., Sigal, L., Eds.; Springer 2011.
DOI: 10.1007/978-0-85729-997-0_27

[2] A. Wadhawan, P. Kumar, Sign language recognition systems: A decade systematic literature review. Arch. Comput. Methods Eng. 28 (2019) pp. 785–813.
DOI:10.1007/s11831-019-09384-2

[3] M. J. Cheok, Z. Omar, M. H. Jaward, A review of hand gesture and sign language recognition techniques. Int. J. Mach. Learn. Cyber 10 (2019) pp. 131–153.
DOI: 10.1007/s13042-017-0705-5

[4] R. Elakkiya, Machine learning based sign language recognition: A review and its research frontier. J. Ambient. Intell. Hum. Comput. 2020.
DOI: 10.1007/s12652-020-02396-y

[5] R. Rastgoo, K. Kiani, S. Escalera, Sign language recognition: A deep survey. Expert Syst. Appl. 164 (2021).
DOI: 10.1016/j.eswa.2020.113794

[6] "IEC standard 60050–300", International Electrotechnical Vocabulary (IEV) - Part 300: Electrical and Electronic Measurements and Measuring Instruments, International Electrotechnical Commission, Jul. 2001.

[7] S. Shirmohammadi, H. Al Osman, Machine learning in measurement Part 1: error contribution and terminology

confusion, IEEE Instrumentation & Measurement Magazine, 24(2) (2021) pp. 84-92.
DOI: 10.1109/MIM.2021.9400955

[8] Encyclopedia Britannica, Sign Language. Online [Accessed December 05 2021]
https://www.britannica.com/topic/sign/language

[9] World Federation of the Deaf. Online [Accessed December 05 2021].
http://wfdeaf.org/our-work

[10] fingerspelling. Wikipedia. Online [Accessed December 05 2021] https://en.wikipedia.org/wiki/Fingerspelling

[11] M. Rinalduzzi, A. De Angelis, F. Santoni, E. Buchicchio, A. Moschitta, P. Carbone, P. Bellitti, M. Serpelloni, Gesture recognition of sign language alphabet using a magnetic positioning System. Appl. Sci. 11 (2021), 5594.
DOI:10.3390/app11125594

[12] J. Dong, Z. Tang, Q. Zhao, Gesture recognition in augmented reality assisted assembly training. J. Phys. Conf. Ser. 1176(3) (2019), art. 032030.
DOI: 10.1088/1742-6596/1176/3/032030

[13] R. E. O. Ascari Schultz, L. Silva, R. Pereira, Personalized interactive gesture recognition assistive technology. In Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems, Vitória, Brazil, 22–25 October 2019.
DOI: 10.1145/3357155.3358442

[14] S. S: Kakkoth, S. Gharge, Real Time Hand Gesture Recognition and its Applications in Assistive Technologies for Disabled. In Proceedings of the Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018.
DOI: 10.1109/ICCUBEA.2018.8697363

[15] M. A. Simão, O. Gibaru, P. Neto, Online recognition of incomplete gesture data to interface collaborative robots, IEEE Trans. Ind. Electron. 66 (2019) pp. 9372–9382.
DOI: 10.1109/TIE.2019.2891449

[16] I. Ding, C. Chang, C. He, A kinect-based gesture command control method for human action imitations of humanoid robots. In Proceedings of the 2014 International Conference on Fuzzy Theory and Its Applications (iFUZZY2014), Kaohsiung, Taiwan, 26–28 November 2014; pp. 208–211.
DOI: 10.1109/iFUZZY.2014.7091261

[17] S. Yang, S. Lee, Y. Byun, Gesture recognition for home automation using transfer learning, 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Bangkok, Thailand, 21–24 Oct. 2018, pp. 136–138.
DOI: 10.1109/ICIIBMS.2018.8549921

[18] Q. Ye, L. Yang, G. Xue, Hand-free gesture recognition for vehicle infotainment system control, 2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan, 5–7 December 2018; pp. 1–2.
DOI: 10.1109/VNC.2018.8628409

[19] Z. U. A. Akhtar, H. Wang, WiFi-based gesture recognition for vehicular infotainment system—An integrated approach, Appl. Sci. 9 (2019), art. 5268.
DOI: 10.3390/app9245268

[20] Y. Meng, J. Li, H. Zhu, X. Liang, Y. Liu, N. Ruan, Revealing your mobile password via WiFi signals: Attacks and countermeasures, IEEE Trans. Mob. Comput. 19(2) (2019) pp. 432–449.
DOI: TMC.2019.2893338

[21] M. J. Cheok, Z. Omar, M. H. Jaward, A review of hand gesture and sign language recognition techniques, Int. J. Mach. Learn. Cyber. 10 (2019) pp. 131–153.
DOI: 10.1007/s13042-017-0705-5

[22] The American Sign Language MNIST Dataset. Online [Accessed December 05 2021]
https://www.kaggle.com/datamunge/sign-language-mnist

[23] LeCun, Y., & Cortes, C. (2010). MNIST handwritten digit database. AT&T Labs. Online [Accessed December 05 2021]
http://yann.lecun.com/exdb/mnist

[24] ASL Alphabet. Online [Accessed December 05 2021]
https://www.kaggle.com/grassknoted/asl-alphabet

[25] ASL Alphabet Test, online [Accessed December 05 2021]
https://www.kaggle.com/danrasband/asl-alphabet-test

[26] Azure Machine Learning Product Overview. Online [Accessed December 05 2021]
https://azure.microsoft.com/it-it/services/machine-learning/#product-overview

[27] F. Santoni, A. De Angelis, A. Moschitta, P. Carbone, A multi-node magnetic positioning system with a distributed data acquisition architecture, Sensors 20(21) (2020), art. 6210, pp. 1-23.
DOI: 10.3390/s20216210

[28] F. Santoni, A. De Angelis, A. Moschitta, P. Carbone, MagIK: A hand-tracking magnetic positioning system based on a kinematic model of the hand, IEEE Transactions on Instrumentation and Measurement 70 (2021), art. 9376979
DOI: 10.1109/TIM.2021.3065761

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27-30 June 2016, pp. 770-778.
DOI: 10.1109/CVPR.2016.90

[30] J. Howard, S. Gugger, Fastai: A layered API for deep learning, Information 11(2) (2020), art. 108.
DOI: 10.3390/info110201081

[31] L. N. Smith, Cyclical learning rates for training neural networks. Online [Accessed December 05 2021]
https://arxiv.org/abs/1506.01186