A Real-time Hand Gesture Recognition System for Human-Computer and Human-Robot Interaction

Valerio Ponzi[1], Emanuele Iacobelli[2], Christian Napoli[1,3] and Janusz Starczewski[4]

[1]Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Roma, 00185, Italy
[2]Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Roma, 00185, Italy
[3]Institute for Systems Analysis and Computer Science, Italian National Research Council, Via dei Taurini 19, Roma, 00185, Italy
[4]Department of Computational Intelligence, Czestochowa University of Technology, al. Armii Krajowej 36, Czestochowa, 42-200, Poland

Abstract

The proposed hand gesture recognition (HGR) system is designed to enhance human-computer interaction (HCI) and human-robot interaction (HRI), which are crucial areas of research aimed at improving the way humans interact with computer or robot systems. With the growing need for intelligent computers and robots in a range of applications, including healthcare, manufacturing, and education, both HCI and HRI have gained significant importance. In this context, the HGR system plays a vital role by enabling natural and intuitive communication between humans and technology through hand gestures. The presented system uses a single camera and efficient image processing techniques that enable real-time gesture detection. Unlike other methods, our approach employs a basic video camera, which is widely available on most computers, eliminating the need for expensive and specialized hardware.

Keywords

Hand Gesture Recognition, Machine Learning, Deep Learning, Convolutional Neural Network

1. Introduction

Hand gesture recognition (HGR) is a technology that enables the identification and interpretation of hand and finger movements in order to understand and respond to user actions. This technology analyzes the visual signals produced by hand gestures and finds the characteristic patterns connected to particular commands or actions using computer vision algorithms and machine learning techniques. With numerous applications ranging from virtual reality to industrial automation, HGR is a growing area of research and development.

Hand gesture detection can be divided into two main categories: *static* and *dynamic*. Static HGR is the ability to detect the static position of the hands at a given moment. For example, it can be used to detect a hand pointing in a direction or to detect an open or closed hand. On the other hand, dynamic HGR refers to the ability to detect hand movements in real time. This technology can be used to detect gestures such as waving or finger movements. One of the main applications of hand gesture recognition is in human-computer interaction. Users can interact with devices in a more intuitive and natural way by employing hand gestures. For instance, without needing a real mouse or keyboard, hand gestures can be used to operate video games, move around virtual worlds, or carry out tasks on a computer screen. Controlling a computer mouse in this way offers a more flexible, intuitive, and natural way of interacting with the computer than traditional input devices, making it one of the most promising and practical applications. This technology can also benefit users with disabilities, injuries, or ergonomic issues that make it difficult or uncomfortable to use a conventional mouse, as well as those who prefer a more immersive and engaging way of navigating and manipulating digital content. Additionally, there are other potential uses for HGR in industries also as the manufacturing field. HGR can be used to control machines and processes in the environment. For instance, workers can use hand gestures to activate machinery or control robotic arms, allowing for more efficient and safer manufacturing processes. In conclusion, HGR is a rapidly developing field that presents many chances to enhance how people interact with technology. The application-specific requirements and the trade-off between accuracy and user comfort determine the best hand gesture detection method.

The paper proposes a real-time and computationally efficient hand gesture recognition system with four steps: Frame Recording, Hand Recognition, Hand Segmentation, and Gesture Recognition. It uses a simple algorithm to detect and segment hands and predict executed gestures. In contrast to current approaches, the suggested hand gesture recognition system stands out for being less expensive and eco-friendly. Instead of the complex hardware and sensors needed by traditional systems, it accomplishes this by capturing hand gestures using only

ICYRIME 2022: International Conference of Yearly Reports on Informatics, Mathematics, and Engineering. Catania, August 26-29, 2022

ponzi@diag.uniroma1.it (V. Ponzi); iacobelli@diag.uniroma1.it (E. Iacobelli); cnapoli@diag.uniroma1.it (C. Napoli); janusz.starczewski@pcz.pl (J. Starczewski)

a camera. This significantly lessens the requirement for additional resources, increasing the system's sustainability and long-term cost-effectiveness.

2. Related Works

There are two main approaches to hand gesture recognition: *Contact-based* and *Vision-based* [1, 2, 3, 4].
Contact-based methods involve the use of sensors on a glove to extract information about hand rotations, acceleration projections, and finger bending angles [5]. This approach can achieve high accuracy, especially after a calibration process to adapt the sensors to the user's hand. However, it can be costly and may not lead to a natural interaction [6]. On the other hand, Vision-based methods use visual devices such as stereo cameras, time of flight cameras, or Kinect sensors to extract depth information and create a 3D representation of the scene. Monocular systems with a single RGB camera have also been used in recent periods. These methods are generally cheaper and more adaptable than contact-based methods. Moreover a relevant number of studies are tackling the problem from the point of view of behavioural analysis and theory of mind[7, 8, 9]. Over the years, various methods have been proposed for hand gesture recognition. These range from the simplest method of wearing a colored glove [10] that is recognized by a video camera, to methods that use skin color recognition [11] followed by hand shape recognition. More advanced methods involve the use of machine learning, such as Skeleton-Based Recognition [12] and Deep-Learning Based Recognition [13]. Both contact-based and vision-based methods have their advantages and disadvantages, and the choice of which method to use depends on the specific application and environment. Vision-based methods are typically used in human-computer interaction and human-robot interaction applications, while contact-based methods are more commonly used in wearable devices for control purposes. Hand gesture technology has two primary areas of application, which are sign language recognition and video gaming. Sign language is a means of communication for individuals who are unable to speak, and it involves a sequence of hand gestures that represent letters, numbers, and expressions. Researchers have proposed several approaches for sign language recognition, including the use of gloves or uncovered hand interaction with a camera using computer vision techniques to identify the gestures [14] [15]. In contrast, video gaming utilizes hand and body movements to interact with the game. The Microsoft Kinect Xbox is an excellent example of gesture interaction for gaming purposes, as it employs a camera placed over the screen that connects with the Xbox device through the cable port to track the user's hand and body movements [16].

3. Proposed Method

For the proposed system, a simple and efficient algorithm capable of working in real-time and with a small computational effort is proposed. The system pipeline comprises four main steps: Frame Recording, Hand Recognition, Hand Segmentation, and Gesture Recognition. Specifically, for each image captured by the camera, a hand detection process is performed to identify the portions of the image where hands are present. Subsequently, a hand segmentation step is conducted to generate a mask that represents the shape of the detected hands. The resulting mask is used as input for the Gesture Recognition step, which predicts the executed gesture.

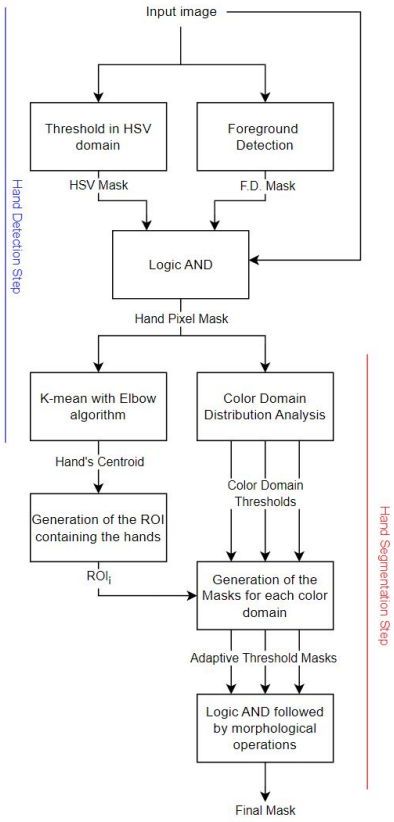

Figure 1: Pipeline scheme for the Hand Detection and Hand Segmentation steps.

3.1. Hand Detection Step

The Hand Detection step is implemented with the aim of generating a mask that represents the pixels corresponding to a hand in an RGB image, along with a set of points that indicate the centroids of the detected hand regions.

This mask is obtained by combining two different masks obtained from color analysis in the HSV color domain and foreground detection. The color analysis approach involves static thresholding of the image, using pre-defined skin limit values in the HSV domain that may be adjusted based on variations in skin tone or lighting conditions within the image. The threshold values for Saturation or Value properties may vary from 0 to 255. However, for the Hue property, which represents the dominant color family, the range is limited from 6 to 28. Foreground detection is a well-established computer vision technique that is used to distinguish between dynamic and static pixels in image sequences by detecting moving objects. To accomplish this, adjacent frames are analyzed to establish a model of the image's background and identify changes that occur. The generated mask, up to this point from the system, is then applied to the original camera frame to generate the Hand Pixel Mask, which contains the pixels representing the possible detected hands. The hands' centroids are now determined using a clustering algorithm, specifically a k-means algorithm [**?** 17, 18], applied to the Hand Pixel Mask. However, tuning the parameter k is crucial to obtaining accurate results, and this parameter is trained autonomously using the Elbow algorithm. The Elbow algorithm determines the minimum total intra-cluster distance in order to identify the optimal value of k. The Sum of Squared Distances (SSD), which in this particular case is computed as the squared sum of distances between the pixels and their corresponding centroids for each cluster, is used in order to determine the best value for k. This process involves adding another cluster and assessing whether the total SSD significantly improves over the previous k value. Moreover, the distance of the hand from the camera can influence this measure, since the closer the hand is to the camera, the higher the pixel density on the image. Therefore, the SSD is normalized based on the number of pixels present on the Hand Pixel Mask. Furthermore, it is important to note that the Elbow method relies on the slope of the resultant function, which represents the normalized SSD values obtained over the iterations on k. As a result, it is essential to establish a slope threshold to act as a significant metric for stopping the K increment process when the function starts to become flat. In the event that this occurs, the algorithm must be interrupted, and the previous stored K value must be returned. The threshold values for the slope and the normalized SSD play a critical role in the sensitivity of the system in detecting new clusters. To minimize the occurrence of false-positive clusters, the proposed system includes an additional algorithm that matches the centroids computed in the current frame with those computed in the previous frame, using a distance metric such as the Euclidean distance. Since the value of K is recomputed in each frame and can vary over time, the mapping is not absolute, and it is possible for new centroid clusters to be missed in situations where the hands are in close proximity or overlapping. To make the system more robust to noisy effects, a new position of the centroids are computed in the following way:

$$\text{newCentroidPos}_t = \text{centroidPos}_{t-1} + \text{step} \cdot \Delta_t,$$

$$\Delta_t = \text{detectedCentroidPos}_t - \text{centroidPos}_{t-1}$$

This approach enables the system to track the trajectory of each hand accurately in the image, even if a completely wrong new observation is detected for the hand in some sporadic time steps. For that reason, this error would not significantly affect the results if enough frames per second are captured. We have put a lot of effort on computing the right hands' centroids since they are critical in eliminating any potential artifacts present in the Hand Pixel Mask that represent other parts of the person's skin.

3.2. Hand Segmentation Step

The Hand Segmentation step is implemented with the aim of refining the output of the previous phase by generating an Adaptive Skin Mask, by leveraging the outputs of the Hand Detection step. This Adaptive Skin Mask is built by using a more flexible threshold for selecting the skin pixels that can adjust to varying lighting conditions that may affect the hands over time. This approach aims to provide greater flexibility compared to the fixed threshold used in the Hand Detection step. The Hand Pixel Mask is used in order to analyze the pixel distribution across various color domains, such as RGB, HSV, and YCBCR, through histogram analysis. Each domain produces a unique threshold based on the mean and variance of the found distributions. Specifically:

$$\text{upperBound}_{D_i} = \text{mean}_{D_i} + 2 \cdot \text{variance}$$

$$\text{lowerBound}_{D_i} = \text{mean}_{D_i} - 2 \cdot \text{variance}$$

and only the pixels that remain inside these bounds are considered skin pixels. By converting the original RGB image into different domains and focusing on the region of interest (ROI) generated by using the Hand's Centroids, multiple masks can be generated. These masks are then combined using a logical AND along with morphological operations to improve the accuracy of the Adaptive Skin Mask. It is important to note that in the case of multiple hand detections in the image, the pixel distribution analysis is performed on each ROI. This enables the system to adapt to different lighting effects that may affect the hands.

3.3. Gesture Recognition Step

In the final phase of the pipeline, the Gesture Recognition step involves the use of a Deep Convolutional Neural Network (DCNN) that has been trained to accurately classify

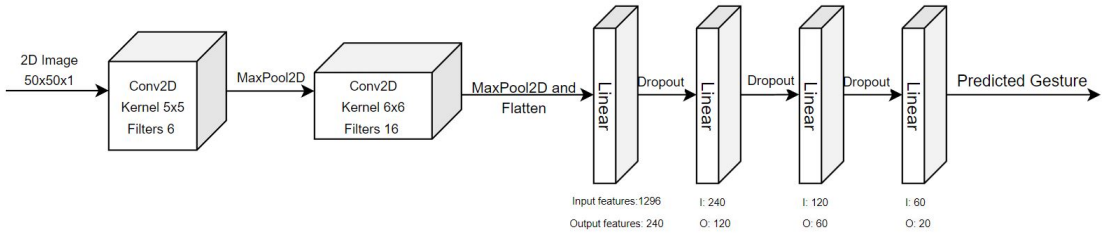

Figure 2: The architecture of the proposed DCNN for the Gesture Recognition.

and recognize the specific gestures performed by the user. Through the use of data augmentation techniques and the training on a large dataset of labeled gesture samples, the DCNN can effectively identify and classify the different gestures executed by the user with a high degree of accuracy and reliability. In particular, the structure of the model is presented in Fig. 2. It is composed of two 2D convolutional layers (activation function ReLU and kernel size 6x6 and 16x16, respectively) each of them followed by a single 2D MaxPool layer (kernel 2x2). After that, four fully connected layers are used in order to produce the final prediction of the gesture. In detail, the input and output features of these layers can be found in the Fig. 2. In order to train the model a SGD optimizer is used with a learning rate equal to 0.004 and momentum equal to 0.9. In addition, a scheduler with an exponential decay with gamma equal to 0.9 is used to decrease at each epoch the learning rate.

3.3.1. Dataset

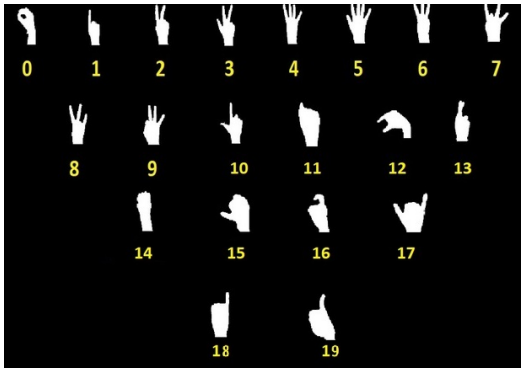

Figure 3: The 20 classes included in the Dataset.

To train the Gesture Recognition model, a comprehensive dataset [19] consisting of a total of 24,000 images and 20 distinct static hand gestures (Fig. 3) has been used. Specifically, the training dataset consists of 18,000 images, with 900 images corresponding to each gesture, while the remaining 6,000 images (300 for each gesture) are divided between the validation and test datasets. In addition, various data augmentation techniques such as random rotation (within the range of +15° to -15°), padding, random cropping, flipping, etc. have been applied to increase the robustness of the trained model. However, as the images in this dataset are segmented by humans, they do not account for the potential noise that may be present in general images obtained through unsupervised algorithms. To address this limitation, Salt and Pepper noise with p=0.2 was introduced to better simulate real-world images and to increase the generalization power of the network.

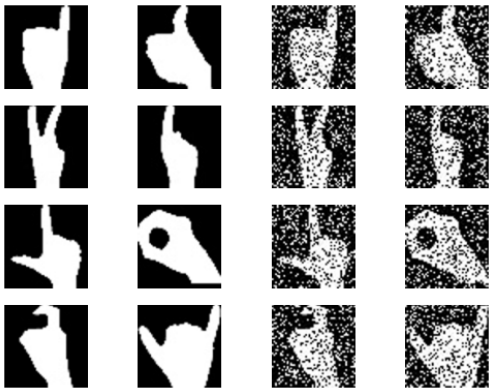

Figure 4: Original images contained in the dataset and their corresponding augmented versions.

4. Results

Regarding the obtained results, a test accuracy of 93.8% was achieved after training the model for 15 epochs. The accuracy and loss plots during the training and validation phases are shown in Fig. 5, 6, respectively. These plots indicate that the model was not overfitting the training

dataset. The Confusion Matrix (Fig. 7) demonstrates that the model is highly capable of accurately predicting all the different classes. The worst predicted class is the class 5, which is sometimes confused with the class 2 due to their similarities, even under perfect conditions without introducing noise (as shown in Fig. 3). This behavior is also reflected in the F1 score shown in Table 1.

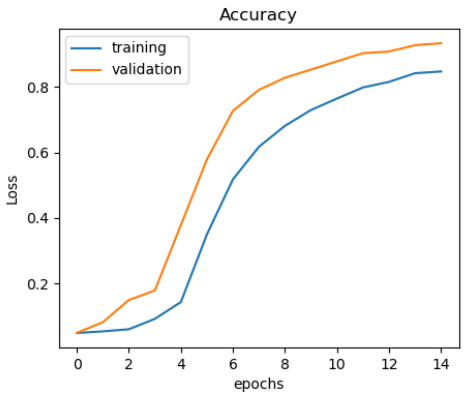

Figure 5: Accuracy over 15 epochs for the training and validation of the model.

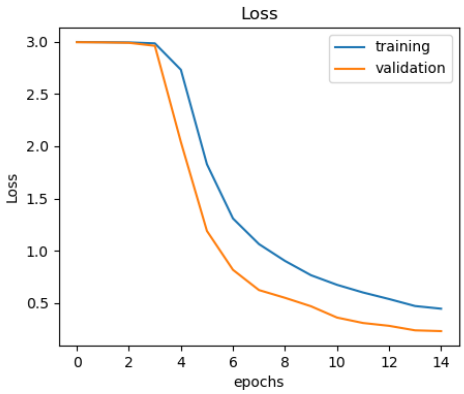

Figure 6: Loss over 15 epochs for the training and validation of the model.

Classes	F1-score
Class-1	0.979
Class-2	0.881
Class-3	0.995
Class-4	0.980
Class-5	0.990
Class-6	0.826
Class-7	0.964
Class-8	0.978
Class-9	0.905
Class-10	0.981
Class-11	0.937
Class-12	0.988
Class-13	0.915
Class-14	0.934
Class-15	0.795
Class-16	0.921
Class-17	0.893
Class-18	0.965
Class-19	0.909
Class-20	0.837

Table 1
F1 score obtained by the model on the test dataset.

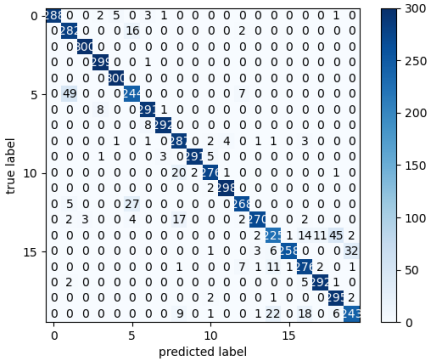

Figure 7: Confusion Matrix.

5. Conclusions

Our paper presented a potential solution for developing accurate hand gesture recognition (HGR) system. Based on the results, it can be said that the proposed method has shown high accuracy and real-time functionality. The test has indeed achieved an accuracy of 93.8% after training the model for 15 epochs. Despite the accurate detection and segmentation of hands, the research also focuses on testing the effectiveness of a robust convolutional neural network (CNN) capable of extracting features even in the presence of imprecise masks. By defining various scenarios based on accuracy, it can be concluded that the proposed CNN model can still produce satisfactory results in all classes.

The proposed system could therefore have great potential for various applications from the most known such as human-computer interaction, virtual reality, and sign language recognition to new ones. For example, during the ongoing Covid-19 pandemic, a possible application is the use of gesture recognition and mouse tracking in

hospitals, which can help reduce the spread of the virus by minimizing contact with shared surfaces. With the aid of this technology, hospital staff and patients can interact with computer systems and medical equipment without physically touching them. This can support a more hygienic and effective hospital environment while also assisting in the prevention of the virus and other infectious diseases. Furthermore, individuals with physical limitations or disabilities may benefit particularly from the use of gesture-based interfaces because it makes it possible for them to interact with technology in a more organic and intuitive way. Therefore, hand gesture recognition technology holds the promise of revolutionizing healthcare and enhancing patient care.

References

[1] S. Pepe, S. Tedeschi, N. Brandizzi, S. Russo, L. Iocchi, C. Napoli, Human attention assessment using a machine learning approach with gan-based data augmentation technique trained using a custom dataset, OBM Neurobiology 6 (2022). doi:10.21926/obm.neurobiol.2204139.

[2] V. Ponzi, S. Russo, V. Bianco, C. Napoli, A. Wajda, Psychoeducative social robots for an healthier lifestyle using artificial intelligence: a case-study, in: CEUR Workshop Proceedings, volume 3118, 2021, pp. 26–33.

[3] G. De Magistris, R. Caprari, G. Castro, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Vision-based holistic scene understanding for context-aware human-robot interaction, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 13196 LNAI (2022) 310–325. doi:10.1007/978-3-031-08421-8_21.

[4] N. Brandizzi, A. Fanti, R. Gallotta, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Unsupervised pose estimation by means of an innovative vision transformer, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 13589 LNAI (2023) 3–20. doi:10.1007/978-3-031-23480-4_1.

[5] L. Dipietro, A. M. Sabatini, P. Dario, A survey of glove-based systems and their applications, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38 (2008) 461–482. doi:10.1109/TSMCC.2008.923862.

[6] H. Cheng, L. Yang, Z. Liu, Survey on 3d hand gesture recognition, IEEE transactions on circuits and systems for video technology 26 (2015) 1659–1673.

[7] N. Brandizzi, S. Russo, G. Galati, C. Napoli, Addressing vehicle sharing through behavioral analysis: A solution to user clustering using recency-frequency-monetary and vehicle relocation based on neighborhood splits, Information (Switzerland) 13 (2022). doi:10.3390/info13110511.

[8] V. Marcotrigiano, G. Stingi, S. Fregnan, P. Magarelli, P. Pasquale, S. Russo, G. Orsi, M. Montagna, C. Napoli, C. Napoli, An integrated control plan in primary schools: Results of a field investigation on nutritional and hygienic features in the apulia region (southern italy), Nutrients 13 (2021). doi:10.3390/nu13093006.

[9] N. Brandizzi, S. Russo, R. Brociek, A. Wajda, First studies to apply the theory of mind theory to green and smart mobility by using gaussian area clustering, in: CEUR Workshop Proceedings, volume 3118, 2021, pp. 71–76.

[10] L. Lamberti, F. Camastra, Real-time hand gesture recognition using a color glove, in: Image Analysis and Processing–ICIAP 2011: 16th International Conference, Ravenna, Italy, September 14-16, 2011, Proceedings, Part I 16, Springer, 2011, pp. 365–373.

[11] K. B. Shaik, P. Ganesan, V. Kalist, B. Sathish, J. M. M. Jenitha, Comparative study of skin color detection and segmentation in hsv and ycbcr color space, Procedia Computer Science 57 (2015) 41–48.

[12] C. Xi, J. Chen, C. Zhao, Q. Pei, L. Liu, Real-time hand tracking using kinect, in: Proceedings of the 2nd International Conference on Digital Signal Processing, ICDSP 2018, Association for Computing Machinery, New York, NY, USA, 2018, p. 37–42. URL: https://doi.org/10.1145/3193025.3193056. doi:10.1145/3193025.3193056.

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

[14] M. R. Islam, U. K. Mitu, R. A. Bhuiyan, J. Shin, Hand gesture feature extraction using deep convolutional neural network for recognizing american sign language, in: 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), IEEE, 2018, pp. 115–119.

[15] R.-H. Liang, M. Ouhyoung, A real-time continuous gesture recognition system for sign language, in: Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 558–567. doi:10.1109/AFGR.1998.671007.

[16] G. Marin, F. Dominio, P. Zanuttigh, Hand gesture recognition with leap motion and kinect devices, in: 2014 IEEE International conference on image processing (ICIP), IEEE, 2014, pp. 1565–1569.

[17] G. Magistris, C. Rametta, G. Capizzi, C. Napoli, Fpga implementation of a parallel dds for wide-band applications, in: CEUR Workshop Proceedings, vol-

ume 3092, 2021, pp. 12–16.
[18] C. Ciancarelli, G. De Magistris, S. Cognetta, D. Appetito, C. Napoli, D. Nardi, A gan approach for anomaly detection in spacecraft telemetries, Lecture Notes in Networks and Systems 531 LNNS (2023) 393–402. doi:10.1007/978-3-031-18050-7_38.
[19] R. Arya, Hand gesture recognition dataset, 2020. URL: https://www.kaggle.com/cihan063/autism-image-data.