applied sciences

MDPI

Article

Gesture Recognition of Sign Language Alphabet Using a Magnetic Positioning System

Matteo Rinalduzzi [1], Alessio De Angelis [1,*], Francesco Santoni [1], Emanuele Buchicchio [1], Antonio Moschitta [1], Paolo Carbone [1], Paolo Bellitti [2] and Mauro Serpelloni [2]

[1] Engineering Department, University of Perugia, 06125 Perugia, Italy; matteo.rinalduzzi@studenti.unipg.it (M.R.); francesco.santoni@unipg.it (F.S.); emanuele.buchicchio@studenti.unipg.it (E.B.); antonio.moschitta@unipg.it (A.M.); paolo.carbone@unipg.it (P.C.)

[2] Department of Information Engineering, University of Brescia, 25121 Brescia, Italy; paolo.bellitti@unibs.it (P.B.); mauro.serpelloni@unibs.it (M.S.)

* Correspondence: alessio.deangelis@unipg.it

Abstract: Hand gesture recognition is a crucial task for the automated translation of sign language, which enables communication for the deaf. This work proposes the usage of a magnetic positioning system for recognizing the static gestures associated with the sign language alphabet. In particular, a magnetic positioning system, which is comprised of several wearable transmitting nodes, measures the 3D position and orientation of the fingers within an operating volume of about 30 × 30 × 30 cm, where receiving nodes are placed at known positions. Measured position data are then processed by a machine learning classification algorithm. The proposed system and classification method are validated by experimental tests. Results show that the proposed approach has good generalization properties and provides a classification accuracy of approximately 97% on 24 alphabet letters. Thus, the feasibility of the proposed gesture recognition system for the task of automated translation of the sign language alphabet for fingerspelling is proven.

Keywords: gesture recognition; sign language recognition; fingerspelling; hand tracking; magnetic positioning system; machine learning; wearable electronics

Citation: Rinalduzzi, M.; De Angelis, A.; Santoni, F.; Buchicchio, E.; Moschitta, A.; Carbone, P.; Bellitti, P.; Serpelloni, M. Gesture Recognition of Sign Language Alphabet Using a Magnetic Positioning System. *Appl. Sci.* **2021**, *11*, 5594. https://doi.org/10.3390/app11125594

Academic Editor: Alfio Dario Grasso

Received: 19 May 2021
Accepted: 12 June 2021
Published: 17 June 2021

1. Introduction

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" [1]. The practice is probably older than speech and widespread over any place, epoch and culture. There are different sign languages with variability in hand shape, motion profile, and position of the hand, face, and body parts contributing to each sign.

Modern sign language originated in the mid-18th century, when Charles-Michel, abbé de l'Epée, developed a system for spelling out French words with a manual alphabet and expressing whole concepts with simple signs. From the l'Epée's system, the French Sign Language (FSL) was developed, which is still in use in France today and is the precursor of American Sign Language (ASL) and many other modern national sign languages [1]. Each country generally has its own native sign language, and some have more than one: the current edition of Ethnologue lists 150 sign languages [2]. Moreover, according to the World Federation of the Deaf, there are over 200 sign languages around the world and 70 million deaf people using them [3].

In the communicative hand/arm gesture taxonomies, modern sign language is considered as the most organized and structured form out of various gesture categories and is an important means of communication among the hearing-impaired and deaf community [4]. Sign language involves the usage of different parts of the body, such as fingers, hand, arm, head, body, and facial expression. One class of sign languages, known also

as fingerspelling, is limited to a collection of manual signs that represent the symbols of the letters of an alphabet using one hand [5]. The American Sign Language (ASL) signs corresponding to the letters of the alphabet are shown in Figure 1.

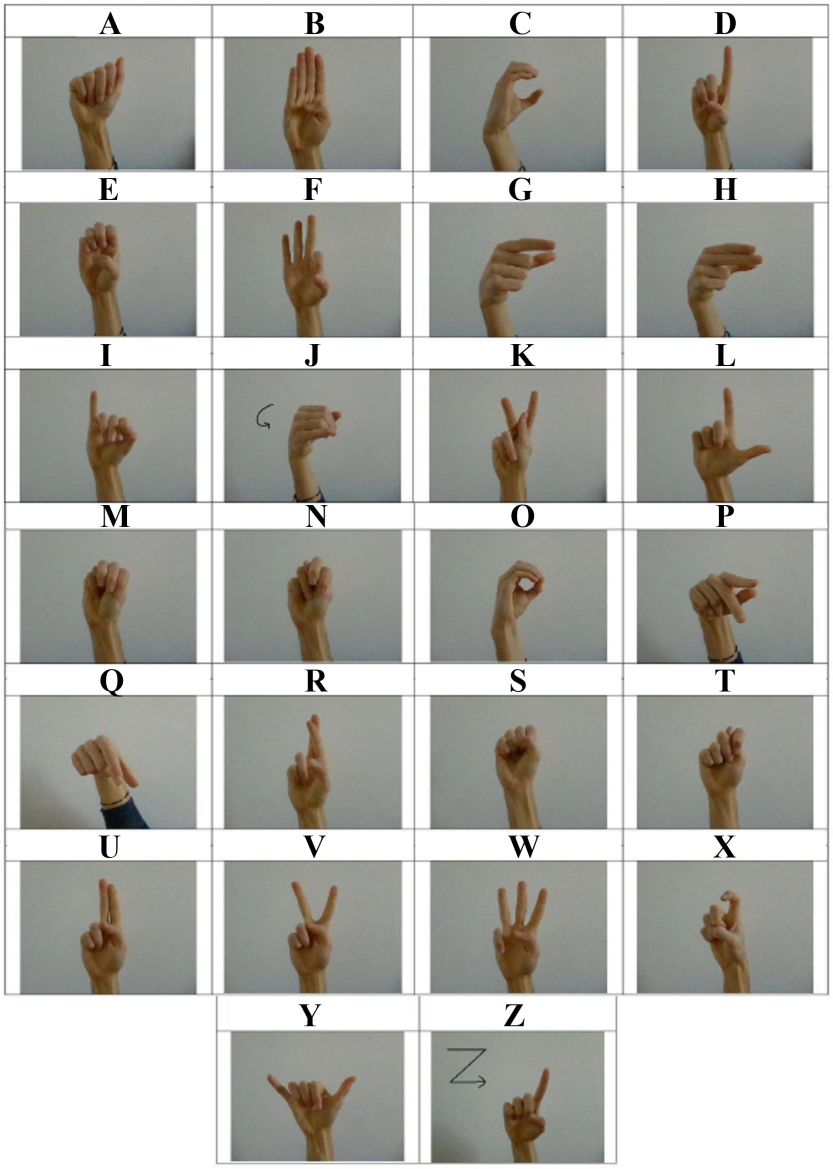


Figure 1. Letters of the American Sign Language (ASL) alphabet.

Sign language recognition (SLR) is a multidisciplinary research area that involves natural language processing, linguistics, pattern matching, computer vision and machine learning [6]. The final goal of sign language recognition is to develop methods and algorithms in order to build a sign language recognition system (SLRS) capable of identifying already produced signs, decoding their meaning and producing some sort of textual, audio or visual output in another language that the intended receiver can understand.

The problem of SLR by automatic systems can be considered as a special case of Natural Language Processing (NLP) or as a specific application of gesture recognition, which is included within the broad domain of Human Computer Interaction (HCI). In this sense, SLRS are considered as a special group of HCI devices, designed to enable effective interaction with an SL speaker. Therefore, research in sign language recognition is highly influenced by hand gesture recognition and HCI research. A search for topics like "gesture recognition", "hands tracking", "finger tracking", and "sign language recognition" on an

online scientific library may return over 100k results for each query, with a rising trend and a peak in a number of published papers in the period 2016–2020.

In the context of hand gesture recognition, many common devices and applications rely on the tracking of 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, other applications of hand gesture recognition include virtual reality, augmented reality [7], assistive technology [8,9], collaborative robotics [10], tele-robotics [11], home automation [12], infotainment systems [13,14], intelligence and espionage [15] and many others [16].

The general SLR problem includes the following three different tasks:

1. Static or continuous letter/number gesture recognition (classification problem);
2. Static or continuous single word recognition (classification problem);
3. Sentence-level sign language recognition (NLP problem).

Available literature surveys [4,16–18] report that recent research achieved an accuracy in the range of 80–100% for the first two tasks, when some given specific conditions are met.

In this paper, we focus on the recognition of static hand gestures associated with the letters of the alphabet for fingerspelling, which is part of Task 1. A similar task was also addressed by many other works, such as [19–35], using different approaches for realizing the software and hardware components of the proposed systems.

Both computer-vision-based and sensor-based approaches have been implemented for sign language alphabet recognition. The vision-based approaches [19–29] use RGB cameras or depth cameras (such as the Leap Motion controller or the Kinect) to capture the image or the video frames of the hand-performing signs. These frames are further processed to recognize the signs and produce some text or speech output. Hand features extraction is a major challenge for the vision-based systems: extraction is affected by many factors, such as lighting condition, complex backgrounds in the image, occlusion, and skin color.

Sensor-based gesture recognition systems are commonly implemented as data gloves featuring inertial measurement units and stretch sensors such as in [30–32], as surface electromyography sensors [33], or other wearable devices [34,35]. Sensor-based approaches have the key advantage of simplifying the detection process. On the other hand, a disadvantage of sensor-based systems is that they can be expensive, cumbersome, or too invasive for real-world deployment.

A fundamental observation that can be made by analyzing the state-of-the-art outlined above is that 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. In particular, in this paper, we demonstrate that training the classification model on data from a tracking system gives substantial advantage in terms of robustness to environmental condition and signer variability.

We propose a sensor-based approach that employs a Magnetic Positioning System (MPS). The MPS is comprised of transmitting nodes and receiving nodes. The transmitting nodes are mounted on the fingers and hand to be tracked, whereas the receiving nodes are placed at known positions in the sides of the operational volume. An advantage of the proposed system, compared to the others mentioned above, is that it measures the absolute position of the fingers, thus enabling high accuracy and drift-free tracking. Another advantage is that the MPS is not sensitive to illumination conditions and the other factors affecting vision-based systems. Furthermore, it operates also 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 for the other applications of hand gesture recognition, such as human–machine interaction, virtual and augmented reality, robotic telemanipulation, and automation.

The remainder of this paper is organized as follows: the proposed magnetic positioning system, method for reconstructing the hand gesture, and machine learning classification models are described in Section 2. Then, experimental results are presented in Section 3.

A discussion of the results is then provided in Section 4, and conclusions are drawn in Section 5.

2. Materials and Methods

In this section, we describe the proposed system for recognizing hand gestures. First, we provide a description of the architecture of the magnetic positioning system used for tracking hand and fingers. Then, we present the kinematic hand model that is employed for reconstructing complete gestures from raw data. Finally, we illustrate the machine learning approach used to classify the gestures, along with the dataset, tools, and learning models.

2.1. Magnetic Positioning System (MPS)

We used an already existent MPS, presented with full details in [36,37]. The system, as presented in [36], was able to track the position and the orientation of six magnetic nodes. We already used that implementation to develop a hand tracking system that could track four fingers [38]. In order to track also the remaining finger, in this work the MPS has been upgraded with an additional magnetic node. The upgrade required some modifications that will be described in the following.

2.1.1. MPS System Architecture

The transmitting magnetic node is one of the main elements of the magnetic tracking system. Its task is to transmit in space a known signal to an array of receivers, so that it is possible to track its position in the working volume. The electromagnetic emission occurs by means of a coil appropriately excited with an oscillating voltage of known frequency. Each coil is coupled to a capacitor in parallel, thus forming a resonant LC circuit. Receivers are implemented as passive LC circuits. The oscillating magnetic field of each active node induces a measurable voltage on receivers, according to Faraday's law. The magnetic field of a single coil is modeled as an elementary magnetic dipole [39]. The position and orientation of each magnetic node (the orientation being defined as the direction of the coil central axis) are estimated by measuring the induced voltages on each receiver, then inverting the mathematical model by means of numerical optimization.

A photo of the transmitting nodes mounted directly on the fingers is shown in Figure 2. The benefit of this configuration is that it is modular and one or more nodes can be easily removed if there is no need to track all fingers. Furthermore, a photo of the tracking nodes mounted on a glove is shown in Figure 3. The advantage of this alternative configuration is that it is quick and convenient to wear. Furthermore, the glove is comfortable and suitable for usage over long periods of time without causing distress to the user. Coils on the fingers are supplied with a square wave generated by a custom device described in Section 2.1.2, while the two coils on the back of the hand (Figure 3) are supplied by a Cypress PSoC 5LP microcontroller, as described in [36,38]. Supplied signal frequencies are chosen to fall on the resonance band of the receivers. The chosen frequencies are 175,138, 177,075, 179,021, 180,961, 183,031, 184,971 and 187,043 Hz.

The working volume of the MPS is shown in Figures 3 and 4; the reference frame is indicated. There are overall 24 receivers inserted into wooden panels. Voltages are measured by means of analog-to-digital converters (ADCs) integrated on Texas Instruments Delfino TMS320F28379D microcontroller units (MCUs). Each MCU reads 4 receivers. Another MCU is used in a master-slave configuration to control all the others, and to transmit data to a PC via serial connection. A custom software calculates the position and orientation of each node. The measurement rate of the MPS with seven nodes is 31 Sa/s, therefore enabling robust tracking even in the presence of fast motion or shaking of the hand. The Euclidean and angular error are very similar to the six-nodes version of the MPS [36], with mean values being, respectively, 3.5 mm and 2.6°.

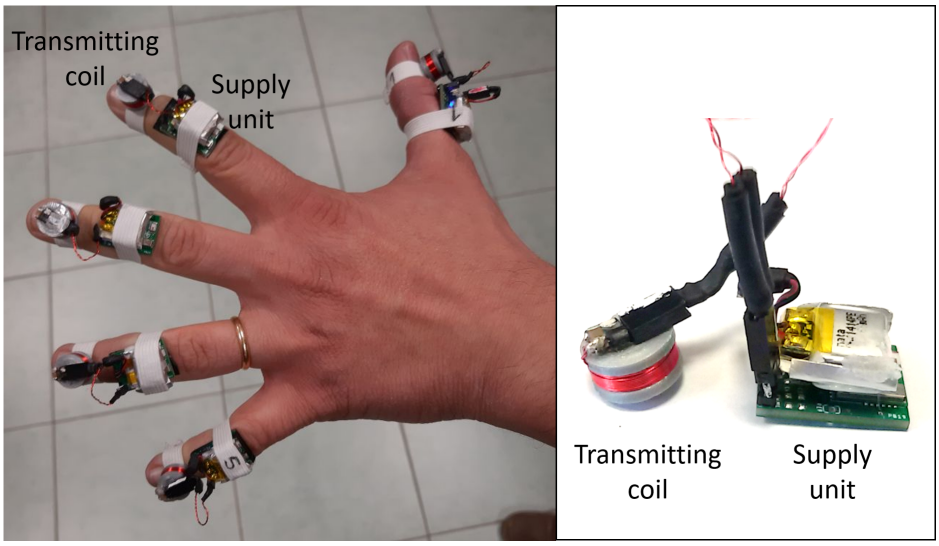


Figure 2. (**Left**) Tracking nodes of the MPS mounted directly on the fingers. (**Right**) A single magnetic node with its supply unit described in Section 2.1.2.

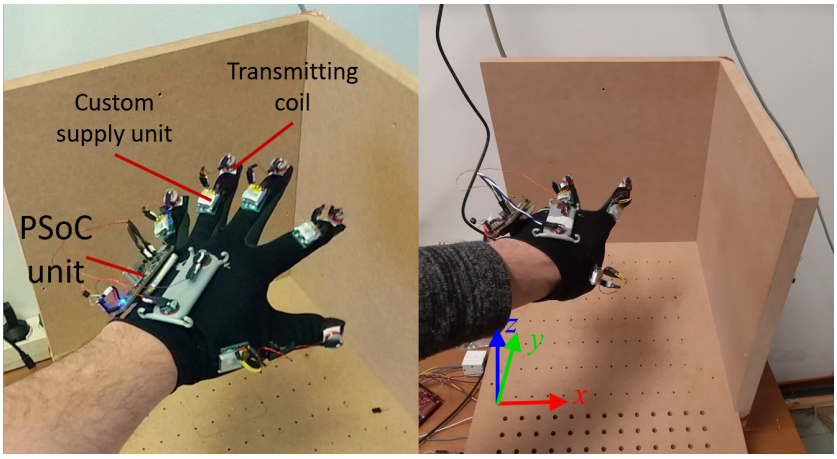


Figure 3. Tracking nodes of the MPS mounted on a glove, shown inside the MPS operational volume. Letters have been articulated as if an interlocutor were watching from the *xy* plane (e.g., letter V is shown on the right).

In Reference [36], the ADCs were configured to work in 12-bit single-ended mode, while the ADC voltage range was 3 V. Transmitting nodes were supplied with a square wave whose amplitude was 2.5 V. A constant 1.5 V signal offset was added to the input of each ADC in order to make the received signal oscillate between 0 and 3 V. For the present work, the addition of the seventh node (the configuration being the same) caused saturation issues on the ADCs.

In order to avoid saturation and to raise the signal-to-quantization-noise ratio (SQNR), we reconfigured the ADCs to work in 16-bit differential mode and removed the signal offset. The differential mode allows to measure signals in the range $(-3\ ,\ 3)$ V; the SQNR is thus increased by a factor of 2^3. With the new configuration, it has been possible to use seven transmitting nodes supplied with a 3.3 V square wave, avoiding saturation issues.

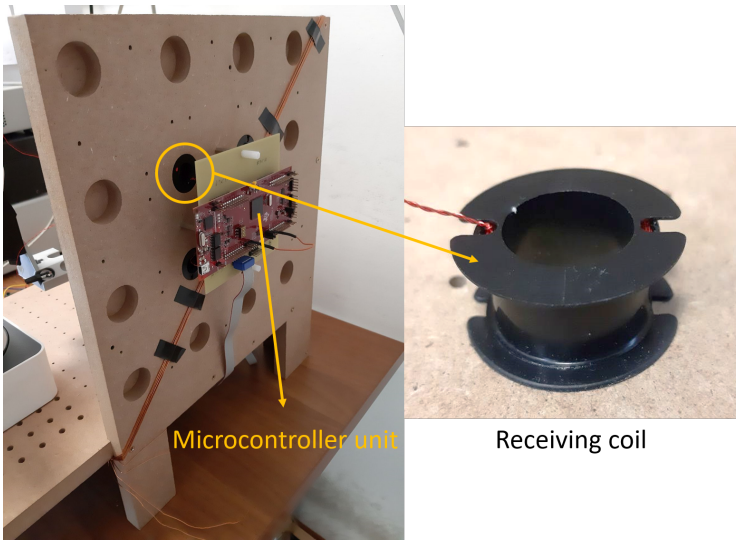


Figure 4. Rear view of the *yz* plane of the magnetic positioning system. Receiving coils inserted into carved slots are visible, together with the microcontroller unit acquiring the signal on four receivers.

2.1.2. Transmitting Nodes

Each transmitting node can be programmed to generate a unique coil excitation frequency in a predetermined range. A ready-to-use integrated solution based on the PRoC (Programmable Radio-on-Chip) CYBLE-222014 by Cypress Semiconductor [40] was selected to minimize the overall node size. The device is based on an ARM Cortex-M0 processor and integrates into a small form factor of 10 × 10 mm a number of components needed for its operations like crystal oscillators, antenna chip and passive elements. CYBLE-222014-01 supports several peripheral functions, such as analog-to-digital converters, timers, counters, and Pulse Width Modulators (PWM), as well as serial communication protocols (I2C, UART, and SPI) through its programmable architecture. CYBLE-222014-01 includes a royalty-free BLE (Bluetooth low Energy) stack compatible with Bluetooth 4.2 and provides up to 16 general purpose input/outputs (GPIOs) and has 256 KB of memory. The peripherals used for the expected functionality of the transmitting node are: BLE stack, PWM, timers, clock dividers and GPIO.

Figure 5 shows the block diagram of the transmitting node. The first block is the power supply section. The wearable device has been designed to be battery powered; however, multiple power sources are supported due to the presence of a fixed 3.3 V voltage regulator (TSP71533 by Texas Instruments) that permits obtaining a stable supply for a wide range of input voltage: 3.75–24 V. The second block is the core of the device represented by CYBLE-222014 with its internal used peripherals. The PWM peripheral is configured to produce an output voltage signal with a frequency in the range 149.47–210.53 kHz with 500 Hz steps, the duty-cycle being fixed at 50%. The output of the PWM device is connected directly to the coil to set the square wave, at the desired frequency. The printed circuit board (PCB) has been designed using EAGLE Autodesk software for schematic, routing and Gerber extraction (Figure 6a). The result is a 2-layer PCB whose dimensions are 25 × 15 mm (Figure 6b).

In Figure 7, the PCB implementation of the transmitting node is represented with the elements described above. In addition, the wearable device is equipped with both a signaling LED to assess its status and a programmable SMD (Surface Mounting Device) low-profile tactile switch (SW) as a simple user interface. To further reduce the dimensions, the programming interface is obtained through a series of pads to whom the programming probes will be connected.

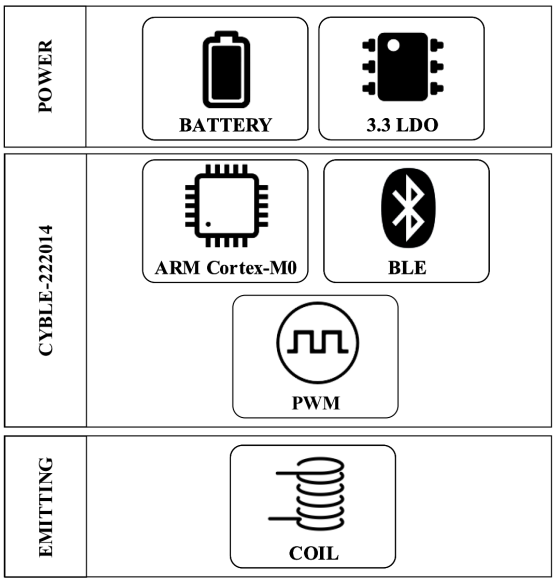


Figure 5. Block diagram of the transmitting node.

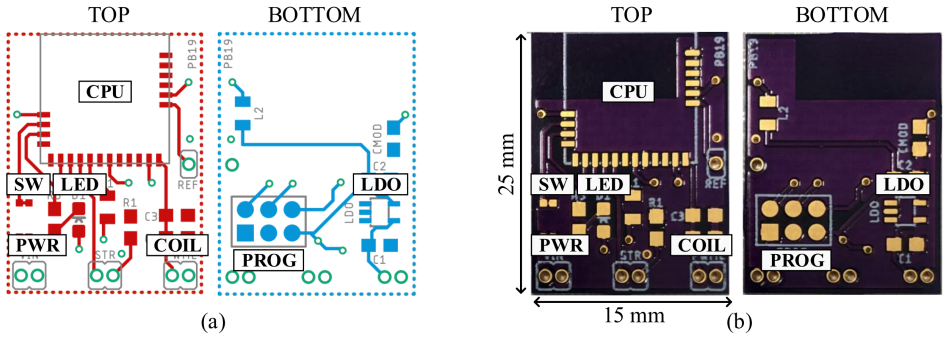


Figure 6. Printed circuit board design of the transmitting node. Representation of the routing result (**a**) and fabricated PCB without components (**b**).

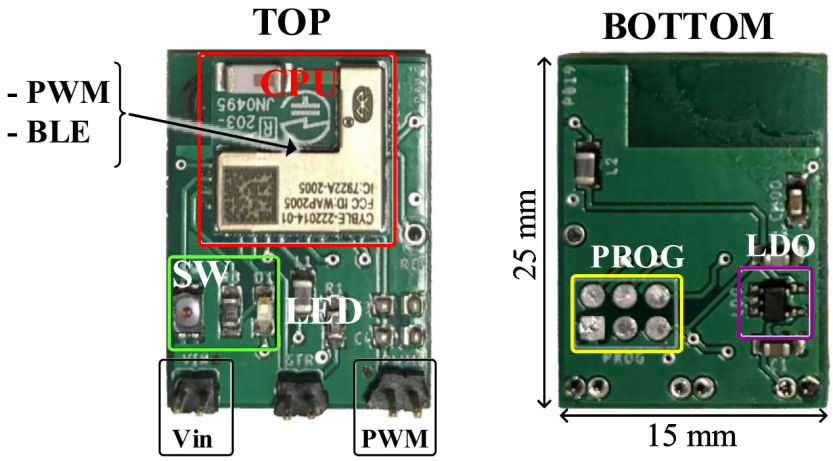


Figure 7. Photo of the realized PCB implementation of the transmitting node with parts populated.

The wearable nature of the device requires no wired connections for data exchange to increase its portability. In this case, the BLE wireless interface was chosen as it is a widely used standard. Therefore, data exchange does not require ad hoc programs but can be

managed through the many generic apps available for both PC and mobile devices. Using the wireless interface, it is possible to set the frequency of the square wave generated by the PWM peripheral.

The wireless communication channel is provided by the on-board BLE facility, which guarantees low-power communication. The GATT profile (Generic Attribute Profile) is used to exchange data according to the BLE protocol [41], and the transmitting node acts as a Bluetooth server. One client-writable custom characteristic is defined to store the desired PWM frequency. The transmitting node, after the power up sequence, starts advertising its presence and accepts client connections. A connected client can set a new PWM frequency by writing a new value in the specific characteristic. The output frequency of the PWM peripheral is obtained through a fractional division of the internal main clock. To easily set the right divider ratio through the BLE connection, a lookup table is implemented in the CYBLE-222014 memory. The value sent from the BLE client represents the index of this table. The index range goes from 0 to 122, mapping the available frequency interval (149.5 to 210.5 kHz, 500 Hz-steps). When a new index is received by the transmitting node, the on-board firmware checks its validity, updates the clock divider ratio and the PWM output frequency changes.

The fractional hardware clock divider permits a good degree of flexibility but does not allow an arbitrary choice of the output frequency; the maximum deviation between the real and the desired frequency is, however, always less than 50.39 Hz and known a priori for each selected frequency. Hence, under such conditions, the receiving section works properly. To improve the usability of the overall system, the last received index is also saved in the ROM memory and restored at each power-on cycle.

Five transmitting nodes have been realized and are placed on each finger of the hand. Since every transmitting node must be addressed univocally, every board has been programmed with a unique Bluetooth name to facilitate its individuation in the client discovery process.

Each PCB is supplied by a 3.7 V, 85 mAh Li-Po battery. The current absorbed when the resonant coil is disconnected is 13 mA; when the coil is connected, the current is 27 mA. Battery duration in working conditions can thus be estimated as $\approx$3 h. The size of the battery is 15 $\times$ 15 $\times$ 5 mm, and it has been chosen in order to not exceed the size of the PCB. The battery can thus be attached directly on the PCB, obtaining a compact, light and easy-to-handle device. Such a small device can be comfortably mounted on any body part, or on other objects, in different positioning scenarios.

2.2. Reconstruction of the Hand Gesture

In this section, we provide details on the reconstruction method that allows obtaining complete gesture information from raw data provided by the magnetic positioning system. The method is based on a kinematic model of the hand.

In order to capture the gesture of the hand, all the magnetic nodes have been mounted on a glove for easy and rapid wearing. The gesture is reconstructed by means of the technique illustrated in [38], which we called MagIK (magnetic and inverse kinematics). The method allows to reconstruct the movement of the hand with 24 degrees-of-freedom (DOF). Positions and orientations of all the magnetic nodes, measured by means of the MPS, are fed to a kinematic model of the hand, allowing 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 shown in Figure 3. Here we review the principal features of MagIK and illustrate some empirical modifications that we introduced in the model in order to optimize the reconstruction of the gesture among different test subjects.

Transmitting nodes are mounted on the hand, as shown in Figure 8. Nodes 1 and 2 are used to determine a Cartesian reference frame rigidly attached to the palm. Nodes 3–7 are used to measure the position and orientation of each finger tip with respect to the MPS frame, then, by a coordinate transformation, position $\mathbf{r}_0^i$ and orientation $\hat{\mathbf{n}}_0^i$ with respect to the hand reference frame are obtained (we use the same notation as in [38]; i

identifies the fingers, numbering them from I to V starting from the thumb; for position vectors, homogeneous coordinates are used, i.e., in general $\mathbf{r} = [x, y, z, 1]^T$). We also define positions $\mathbf{R}^i$ and orientations $\hat{\mathbf{N}}^i$ of the finger tips, with respect to a reference frame attached to each IP/DIP joint, as $\mathbf{R}^i = \left[L_4^i, 0, 0, 1\right]^T$ and $\hat{\mathbf{N}}^i = [0, 1, 0]^T$, where L_4^i is the length of the terminal phalanx of each finger (L_3^{I} for the thumb).

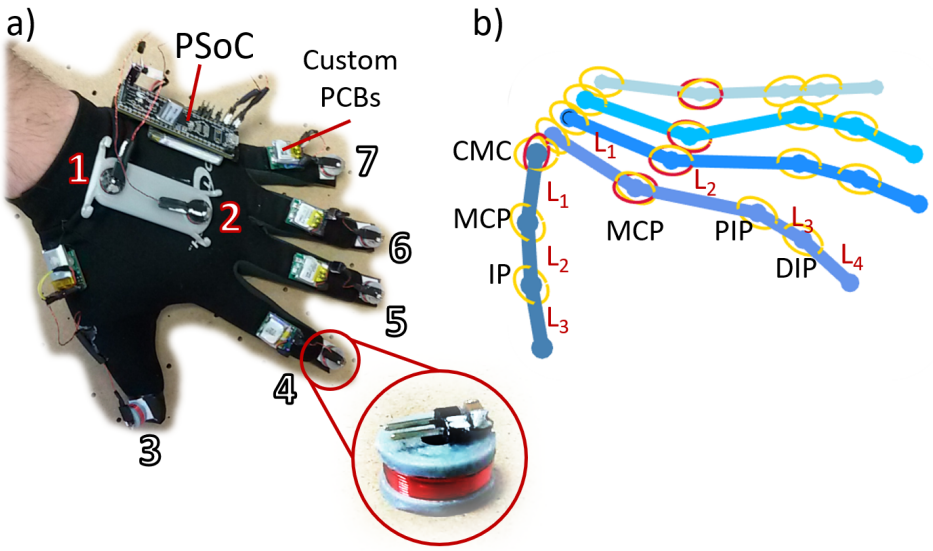


Figure 8. (**a**) A picture of the magnetic glove. Transmitting nodes are numbered. The PSoC used to supply nodes 1 and 2, together with its own battery, is attached to the glove on the back of the hand. Custom PCBs are attached to the phalanges. (**b**) The complete hand reconstruction resulting from the kinematic model. CMC–carpometacarpal joints; MCP—metacarpophalangeal joints; PIP—proximal interphalangeal joints; DIP—distal interphalangeal joints; and IP—interphalangeal joint. Lengths of the phalanges are indicated with L.

The following relations hold:

$$\boldsymbol{r}_0^i = T^i\left(\boldsymbol{d}^i, \boldsymbol{\theta}^i, \boldsymbol{a}^i, \boldsymbol{\alpha}^i\right)\boldsymbol{R}^i, \tag{1}$$

$$\hat{\boldsymbol{n}}_0^i = T_{\mathrm{rot}}^i\left(\boldsymbol{\theta}^i, \boldsymbol{\alpha}^i\right)\hat{\boldsymbol{N}}^i, \tag{2}$$

where T^i is a 4×4 roto-translation matrix defined according to the widely used modified Denavit–Hartenberg (MDH) formalism introduced by Craig [42] to describe mechanical links. In this formalism, a reference frame is attached to each link (all the details can be found in [38]); x-axes are oriented along link directions; z-axes are attached to the rotation axis of each link; y-axes are thus determined by the right-hand rule. The MDH parameters are defined as follows: angle θ measure rotations around z-axes; angle α measure link twists around x-axes; length a measure translations along x-axes; length d measure link shifts along z-axes. T_{rot}^i is the 3×3 rotation submatrix of T^i, obtained from the latter by removing the fourth row and the fourth column.

For the kinematic model of the hand, only angles θ and $\alpha_{\mathrm{MCP}}^{\mathrm{I}}$ are mobile, while all the other MDH parameters are fixed. MDH parameters for the hand are given in Tables 1 and 2, where L_j^i are the lengths of the phalanges, determined through the calibration procedure described in [38]. Tabulated parameters are the same as in [38], except for some empirical modifications for the thumb that have been introduced for a more realistic gesture reconstruction. Angle $\alpha_{CMC,a}^{\mathrm{I}}$ is $\frac{11}{18}\pi$ (110°), while in [38], the value was $\frac{\pi}{2}$. For the angle $\alpha_{\mathrm{MCP}}^{\mathrm{I}}$, the following relation holds (a modification of an analogous relation in [38]):

$$\alpha^{\mathrm{I}}_{\mathrm{MCP}} \equiv \alpha\left(\theta^{\mathrm{I}}_{\mathrm{CMC,a}}\right) = \alpha_0 + \frac{\pi}{18}\frac{\theta^{\mathrm{I}}_{\mathrm{CMC,a}} - \theta^{\mathrm{I}}_{\mathrm{CMC,a,\,min}}}{\theta^{\mathrm{I}}_{\mathrm{CMC,a,\,max}} - \theta^{\mathrm{I}}_{\mathrm{CMC,a\,min}}}, \tag{3}$$

where α_0 is the twist angle of the thumb measured when it is fully extended during the calibration phase. Minimum and maximum values $\theta_{\mathrm{CMC,a,\,min}}$ and $\theta_{\mathrm{CMC,a,\,max}}$, are reported in Table 3.

Table 1. MDH parameters for fingers II–V.

	Joint	a	d	θ	α
5	DIP	L^i_3	0	$\theta^i_{\mathrm{DIP,f}}$	0
4	PIP	L^i_2	0	$\theta^i_{\mathrm{PIP,f}}$	0
3	MCP,f	0	0	$\theta^i_{\mathrm{MCP,f}}$	$\pi/2$
2	MCP,a	L^i_1	0	$\theta^i_{\mathrm{MCP,a}}$	$-\pi/2$
1	CMC,f	0	0	$\theta^i_{\mathrm{CMC,f}}$	$\pi/2$

Table 2. MDH parameters for the thumb.

	Joint	a	d	θ	α
d	IP	L^{I}_2	0	$\theta^{\mathrm{I}}_{\mathrm{IP,f}}$	0
c	MCP	L^{I}_1	0	$\theta^{\mathrm{I}}_{\mathrm{MCP,f}}$	$-\alpha^{\mathrm{I}}_{\mathrm{MCP}}$
b	CMC,f	0	0	$\theta^{\mathrm{I}}_{\mathrm{MCP,f}}$	0
a	CMC,a	0	0	$\theta^{\mathrm{I}}_{\mathrm{CMC,a}}$	$11\pi/18$

In order to determine the length of the phalanges for each finger, during the calibration procedure, the distances between nodes 3–7 and node 1 are measured, then all the lengths L^i_j are obtained through a set of proportionality coefficients determined from average anthropometric data [43]. The detailed procedure is illustrated in [38], where nodes 3–7 were considered to be mounted on the middle of the distal phalanx of each finger. In this work, node 3 is positioned approximately at 2/3 of the distal phalanx of the thumb, while nodes 4–7 are positioned at 3/4 of the distal phalanges. Hence, the calibration described in [38] has been performed using an updated set of proportionality coefficients reported in Table 4.

Table 3. Lower and upper bound of angles θ.

	Thumb	Index	Middle	Ring	Pinkie
$\theta_{\mathrm{CMC,a}}$	$[0°, 125°]$	N.D.	N.D.	N.D.	N.D.
$\theta_{\mathrm{CMC,f}}$	$[0°, 70°]$	$[0°, 5°]$	0	$[0°, 10°]$	$[0°, 15°]$
$\theta_{\mathrm{MCP,a}}$	N.D.	$[-30°, 30°]$	$[-22°, 22°]$	$[-22°, 22°]$	$[-25°, 25°]$
$\theta_{\mathrm{MCP,f}}$	$[0°, 60°]$	$[-40°, 90°]$	$[-40°, 90°]$	$[-40°, 95°]$	$[-40°, 95°]$
$\theta_{\mathrm{PIP,f}}$	N.D.	$[0°, 120°]$	$[0°, 110°]$	$[0°, 120°]$	$[0°, 135°]$
$\theta_{\mathrm{DIP,f}}$	$[-10°, 90°]$	$[-5°, 80°]$	$[-5°, 80°]$	$[-5°, 80°]$	$[-5°, 90°]$

Table 4. Proportionality coefficients for finger segment lengths.

c	i = I	II	III	IV	V
j = 1	0.332	0.316	0.328	0.326	0.364
2	0.279	0.387	0.360	0.359	0.332
3	0.273	0.153	0.173	0.165	0.140
4	N.D	0.192	0.186	0.200	0.218

The reconstructed structure of the hand, resulting from the kinematic model, is shown in Figure 8b. Since $\boldsymbol{R}^i$ and $\hat{\boldsymbol{N}}^i$ are fixed, while $\boldsymbol{r}_0^i$ and $\hat{\boldsymbol{n}}_0^i$ are measured by means of the MPS, the values of all the mobile MDH parameters, i.e., the gesture, are obtained by inverting the model in Equations (1) and (2), solving a nonlinear optimization problem. While MPS measurements are performed in real-time, reconstruction of the gesture through the kinematic model is computed off-line. We developed a MATLAB script by solving the optimization problem on stored MPS data. We tested it on a computer equipped with a six-core Intel i7-8750H CPU at 2.20 GHz, running Windows 10, obtaining ≈55 hand reconstructions per second.

Static bounds on angles θ have to be set in the optimization algorithm, in order to obtain a realistic hand pose. In this work, according to the modifications illustrated above, we had to set different bounds for the thumb as compared to [38]. The new set of bounds is reported in Table 3. Another important physiological constraint is the limitation of the MCP abduction angle as fingers II–V are flexed. In [38], this dynamic constraint was implemented introducing a dependence on $\theta_{\mathrm{MCP,f}}$ for the lower and upper bounds of $\theta_{\mathrm{MCP,a}}$ as given in [44]. In this work, we substituted that relation with a more realistic one that we determined empirically. For fingers II–IV:

$$\theta_{\mathrm{MCP,a,dmax}} = \left(1 - \frac{\theta_{\mathrm{MCP,f}}}{\theta_{\mathrm{MCP,f,smax}}}\right)\theta_{\mathrm{MCP,a,smax}}, \tag{4}$$

We write *smin* and *smax* to indicate static minima and maxima as reported in Table 3, while *dmin* and *dmax* indicate dynamic minima and maxima. For finger V:

$$\theta_{\mathrm{MCP,a,dmax}} = \theta_{\mathrm{MCP,a,smin}} + \frac{\pi}{36} + 2\left(1 - \frac{\theta_{\mathrm{MCP,f}}}{\theta_{\mathrm{MCP,f,smax}}}\right)\left(\theta_{\mathrm{MCP,a,smax}} - \frac{\pi}{72}\right) \tag{5}$$

while for all fingers II–V:

$$\theta_{\mathrm{MCP,a,dmin}} = \theta_{\mathrm{MCP,a,smin}}. \tag{6}$$

The resulting reconstruction of the hand pose for the articulation of some letters is shown in Figure 9.

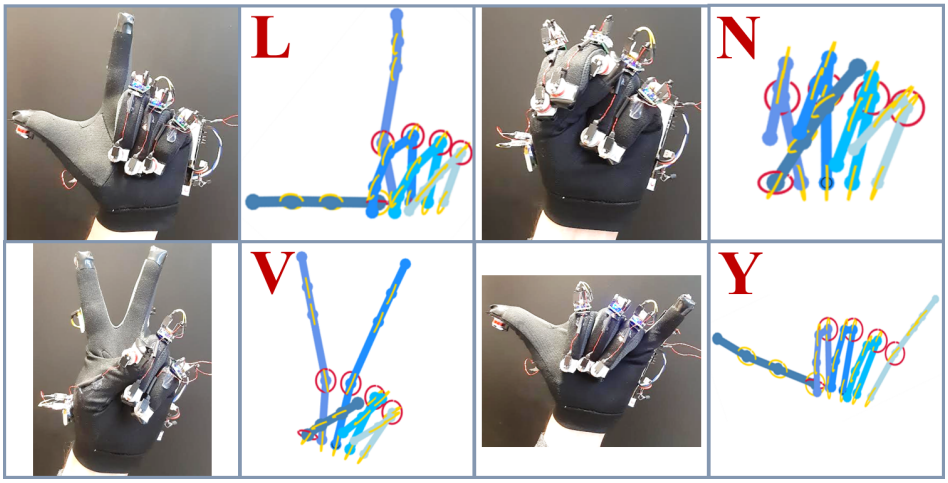


Figure 9. Some examples of letters articulated while wearing the glove, and their respective reconstructions obtained through the kinematic model.

2.3. Machine Learning Algorithm

In this section, we describe the method, based on support vector machines, that we developed to classify the hand gestures. Furthermore, we describe the tools used to implement the machine learning pipeline and we define the considered models.

The fundamental idea behind the proposed classification method is to train the model on a large publicly-available database of images, from which we extract tracking data using a prebuilt hand detection and tracking solution. Then, we apply this trained model to data acquired by our magnetic positioning system.

2.3.1. Support Vector Machine

Our goal was to develop a machine learning model capable of recognizing the 24 static letters from the ASL alphabet. The letters J and Z are not considered, since they are associated with dynamic gestures. The model takes as input the coordinates of finger joints obtained by using the magnetic positioning system, as described in detail in Sections 2.1 and 2.2. The output of the model is the class that best represents the input vector, and each letter is assigned to one class. Among all the machine learning algorithms suitable for classification purposes, we decided to use Support Vector Machine (SVM). Before choosing SVM, we considered a set of classification algorithms including k-nearest neighbors, logistic regression and others, but the one that gave us the best results was SVM with Gaussian kernel.

SVM is a supervised machine learning binary classifier that aims to find the parameters of the hyperplane that best separates the data into two classes [45]. To train the SVM, an optimization problem is solved. Specifically, given a training set of n samples $(\boldsymbol{x}_i, y_i)$ with $i = 1, \ldots, n$, where $\boldsymbol{x}_i$ is in general a multi-dimensional input vector and y_i is a scalar, which can be either -1 or 1, the optimization problem can be formulated as follows [46]

$$\underset{(\boldsymbol{\beta},\beta_0)}{\arg\min}\left(\frac{1}{n}\sum_{i=1}^{n}\max\{0, 1 - y_i(\boldsymbol{\beta}^T\varphi(\boldsymbol{x}_i) + \beta_0)\} + \lambda\|\boldsymbol{\beta}\|_2^2\right), \tag{7}$$

where $(\boldsymbol{\beta}, \beta_0)$ is the set of parameters that identifies the hyperplane: in a k-dimensional space, $\boldsymbol{\beta}$ is a k-dimensional vector and β_0 is a scalar that represents the intercept. The symbol $\|\cdot\|_2$ denotes the $L2$-norm operator and λ is a tuning parameter. The regularization term $\lambda\|\boldsymbol{\beta}\|_2^2$ forces the solution to be unique and avoids overfitting on training data. The nonlinear mapping function $\varphi(\cdot)$ is defined by using the Gaussian kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq \varphi^T(\boldsymbol{x}_i)\varphi(\boldsymbol{x}_j) = e^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\delta^2}$, where δ is a tuning parameter [47–49]. The optimization problem Equation (7) is convex; therefore, it can be efficiently solved using a numerical solver.

The SVM is not intended for binary classification only. Even if the initial formulation was intended for binary classification, every classifier that performs binary classification can be extended to multiclass classification by using one of the approaches between One versus One (OvO) or One versus All (OvA). If c is the number of classes, with OvA, the multiclass problem is reduced to c binary problems, one for every class: each class is compared to all the others put together. Classification of a new vector is performed by applying the vector to the c classifiers and by assigning it to the most likely class. The problem is that there are cases where multiclass linearly separable points are not linearly separable with an OvA approach. In these cases, the OvO approach can be used. This approach consists of building a collection of $c(c-1)/2$ binary classifiers, one for each non-ordered pair of classes. Classification of a new sample is performed with a majority mechanism: decisions of all the binary classifiers are evaluated and the sample is assigned to the class that wins the most. In case of a tie, the class is chosen at random. If a set of multiclass data are linearly separable, the OvO approach can always be used. As far as computational cost is concerned, OvO requires a significantly higher number of classifiers, but each classifier is trained with samples that come only from two classes. In this paper, we used an OvO approach.

2.3.2. Description of Tools Used

Our workflow started with the selection of a high-quality dataset for the ASL alphabet letters from Kaggle. Kaggle is a subsidiary of Google LLC that provides a platform as a service for data science purposes. It allows users to write, store and execute code written

in Python and R. Due to its extensive collection of machine learning libraries and packages, we decided to use Python as the programming language. We used the implementation of SVM provided by the Scikit-learn library and, instead of approaching the classification problem as a computer vision problem, we used the ready-to-use and efficient Python solutions offered by MediaPipe as prebuilt Python packages.

MediaPipe is an open source framework designed by Google, which allows building pipelines to perform inference over arbitrary sensory data [50]. It has a selection of high accuracy machine learning models for human body parts detection and tracking, able to track key points on these body parts, which are called landmarks. In particular, landmarks are three-dimensional coordinates points normalized within a [0, 1] interval. Google trained these custom-built models on its largest and most diversified datasets. Developers can use these models by modifying the pipelines of information flow or simply by implementing a system on top of the custom-built models.

In particular, MediaPipe comes with a hand tracking solution [51] named MediaPipe Hands that, by accessing a 2D camera, detects a hand and tracks its movement inside the field of view of the camera. It utilizes a pipeline that consists of two models:

- A palm detection model that takes as input a 2D image and returns an oriented bounding box that contains the detected hand;
- A hand landmark model that takes as input the output box from the previous model and returns a prediction of the hand skeleton with landmark points, a number indicating the probability of hand presence in the cropped image and a binary classification of handedness, i.e., if the detected hand is a left or a right hand.

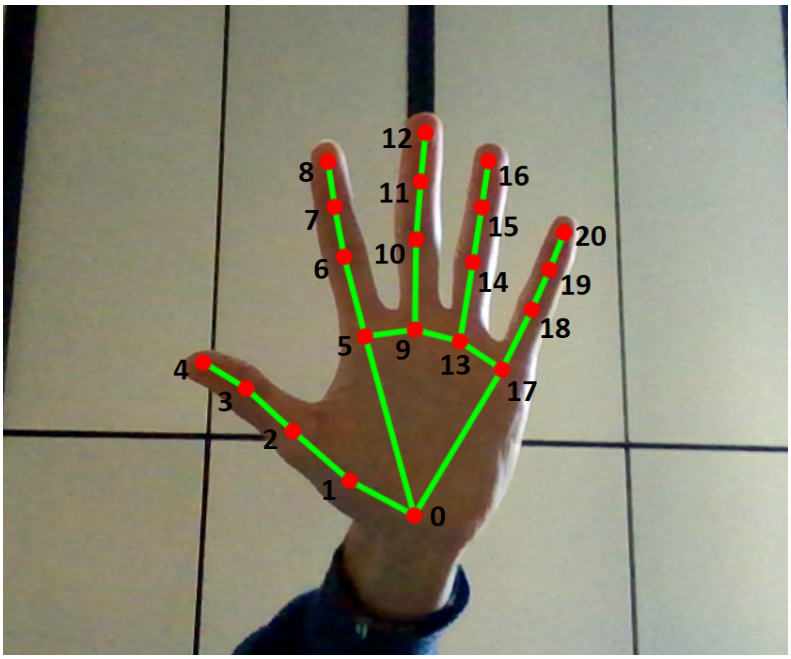
The pipeline is built in a way that palm detection is performed only when the hand landmark model is no longer able to detect hand presence. The resulting model is lightweight enough to run in real-time also on mobile devices [52].

An example of the hand model reconstructed by MediaPipe Hands is shown in Figure 10, which also lists the 21 hand landmarks for the detected hand. Each of these landmarks represents a three-dimensional coordinate: x and y assume values in the [0, 1] range and they correspond to the position of the pixel in the image, normalized with respect to image width and height, while z represents the distance from the palm. Smaller values of z are linked to closer points to the camera. The scaling factor chosen for the z coordinate is roughly the same as the one used for the x coordinate.

2.3.3. Description of the Dataset

Many datasets that contain images of alphabet letters from the American Sign Language are publicly available. In this paper, we use a dataset created by a group of students at UC Santa Barbara, available from Kaggle [53]. The dataset is divided into 24 folders, each folder contains hundreds of pictures taken from five different subjects. The dataset is highly diversified, since each picture is slightly different from the others, such as the angles, positions of the hand, illumination and background change. Most of the pictures were taken using the left hand.

Prior to using this public dataset for training the machine learning model, we performed a data cleaning operation. Such an operation was necessary because some images were referring to an erroneous gesture and others could have been misinterpreted by MediaPipe due to the presence of more than one hand inside the picture or due to bad light conditions. We removed such images after visual inspection, thus obtaining a filtered dataset, which contained almost 7000 images, 80% of all images. Then, we applied the MediaPipe framework to the whole dataset and saved the resulting landmark data to local. This allowed us to discard the images not well recognized by MediaPipe and to obtain a training set made of quality data.



X	Y	Z
0.5232	0.9016	0
0.3951	0.8343	- 0.0488
0.2967	0.7415	- 0.0917
0.2132	0.6652	- 0.1347
0.1393	0.6288	- 0.1805
0.4076	0.5240	- 0.0844
0.3640	0.3575	- 0.1315
0.3386	0.2527	- 0.1708
0.3186	0.1652	- 0.1944
0.4881	0.5096	- 0.0826
0.4695	0.3106	- 0.1212
0.4590	0.1866	- 0.1680
0.4516	0.7980	- 0.2103
0.5619	0.5309	- 0.0777
0.5743	0.3544	- 0.1129
0.5813	0.2370	- 0.1491
0.5849	0.1331	- 0.1781
0.6297	0.5818	- 0.0748
0.6649	0.4524	- 0.1095
0.6902	0.3699	- 0.1381
0.7043	0.2881	- 0.1629

Figure 10. Hand landmarks (red points) extracted by MediaPipe Hands and corresponding coordinates.

2.3.4. Models

We considered three different models that use different input data but share the same logic, which is illustrated in Figure 11. Each model is a SVM classifier with a Gaussian kernel, as described in Section 2.3.1, trained on 90% of the filtered dataset and tested on the remaining 10%. The models are all trained on landmark data extracted by MediaPipe from the images of the dataset described in Section 2.3.3. The models are then validated using data from the magnetic positioning system.

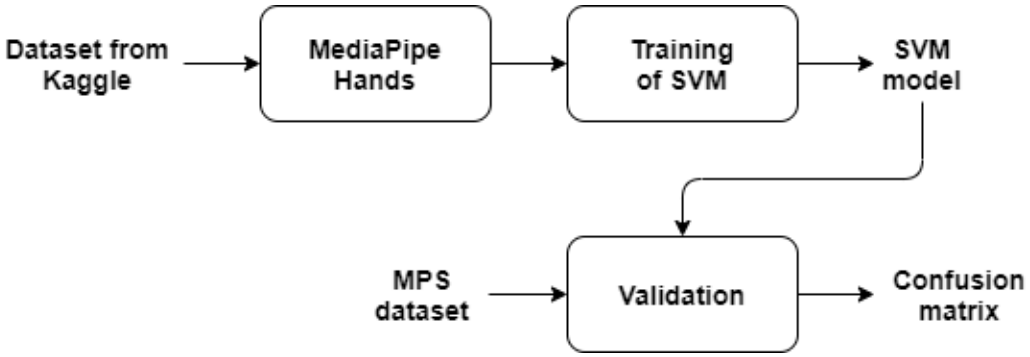


Figure 11. Block diagram of the proposed method.

The models differ in the reference system used and in the number of points considered for each frame:

Model A. Only the fingertips, i.e., the points corresponding to labels 4, 8, 12, 16, 20 of Figure 10, are used as input. The reference system used is the one chosen by MediaPipe: the origin of the xy plane is placed at the top-left corner, while the origin of the z axis is placed at the wrist. The x and y are normalized to [0, 1] by dividing by image width and height, respectively, while z uses approximately the same scale as x [51]. The model is validated by using raw data extracted from six nodes of the magnetic positioning system. The hand kinematic model is not used. The additional node is the one placed at the wrist as it was necessary to determine the origin of the z axis.

Model B. The coordinates of the 21 hand landmarks given by MediaPipe are used as the input. The reference system used is the same as Model A, but this time, the model is validated by using data from the magnetic positioning system that have been processed using the kinematic hand model described in Section 2.2. This allowed us to reconstruct the position of 21 points within the hand, while using only seven transmitting nodes in the MPS.

Model C. The same features chosen for Model B are used, but a reference coordinate system aligned to the hand is considered. Specifically, the x versor has the same direction of the vector that connect the endpoints of themiddle metacarpal bone, the z versor is orthogonal to the plane in which metacarpal bones of the index finger and middle finger lie, and the y versor is the cross product between the x and z versor. The main advantage is that this model is also capable of recognizing gestures when the hand is rotated at an arbitrary orientation.

3. Results

Experimental data were obtained from three subjects, denoted as signers, which recorded the 24 static letters of the ASL alphabet using the MPS. For each letter, about 400 samples were recorded by each signer, keeping the hand at approximately the same orientation within the operational volume of the MPS. The orientation of the hand is defined assuming that an interlocutor is viewing from the xy plane of the MPS.

Additional samples were recorded by rotating and moving the hand in different orientations while performing each letter. These additional samples were used to characterize the performance of Model C to show the feasibility of recognizing gestures even from different viewpoints and orientations. In Model C, the letters that differ only by the rotation of the hand have been aggregated into the same class. This is because it is not possible to discriminate them using a reference coordinate system that is aligned with the hand. Specifically, the letters that are aggregated in the same class are the couples D-P and H-U.

The results obtained by Model A, Model B, and Model C are shown by the confusion matrices in Figures 12–14, respectively. Furthermore, a summary of the classification accuracy achieved by the three considered models on data acquired by the signers individually

and on the global MPS dataset is provided by Table 5. The global dataset is built by combining data of all three signers.

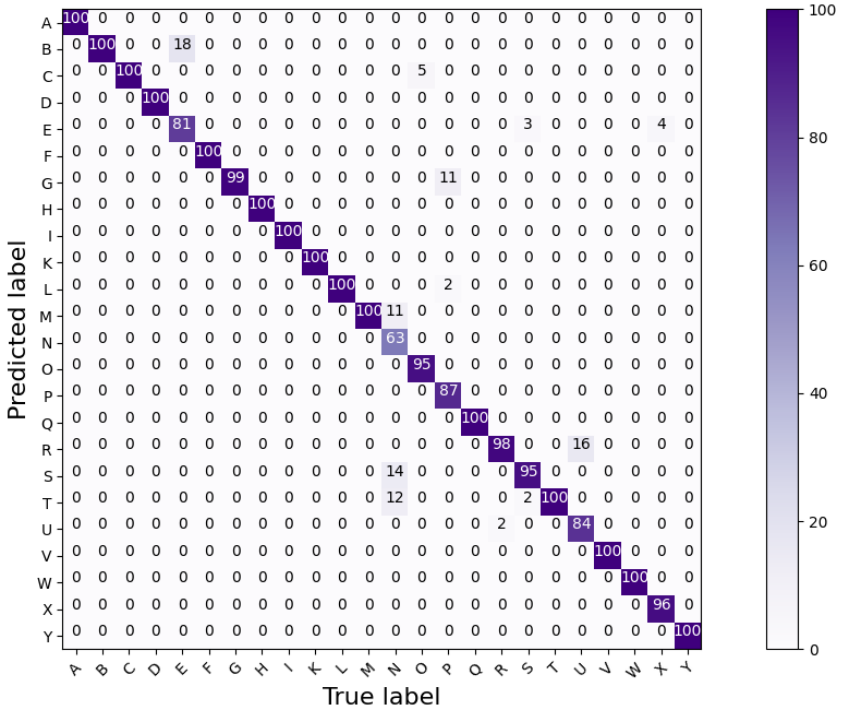


Figure 12. Confusion matrix showing the classification performance of the proposed method using Model A, where only fingertip positions are used as inputs to the SVM classifier model.

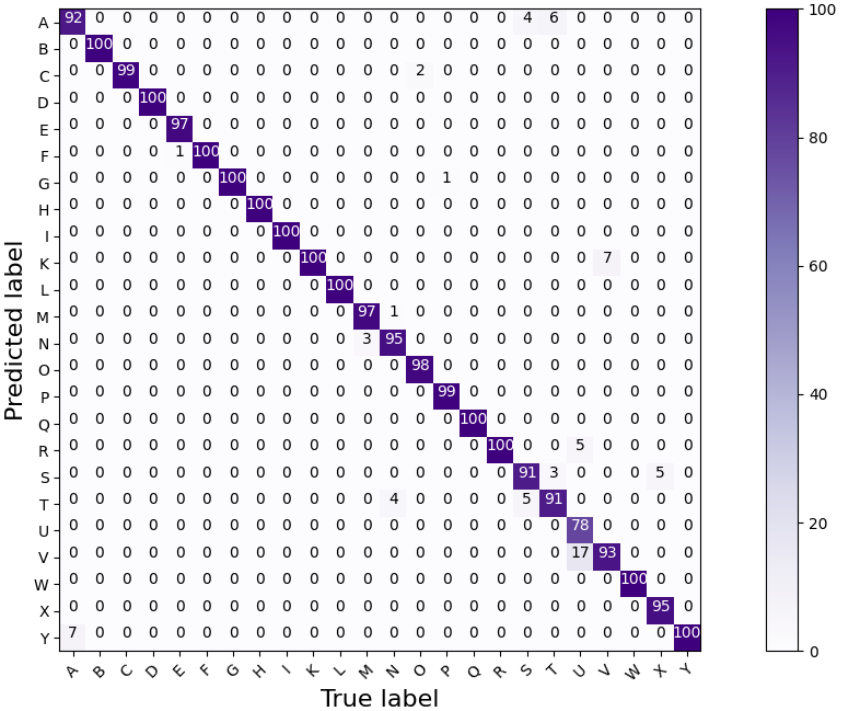


Figure 13. Confusion matrix showing the classification performance of the proposed method using Model B, where 21 hand landmarks, reconstructed by the kinematic model in Section 2.2, are used as inputs to the SVM classifier model.

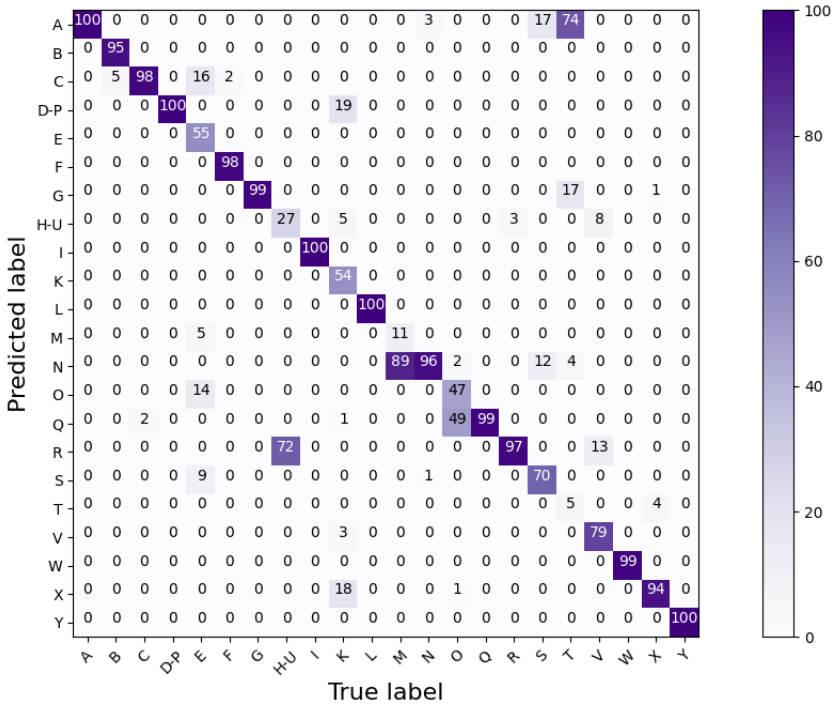


Figure 14. Confusion matrix showing classification performance of the proposed method using Model C, where the SVM input landmarks are given in a coordinate system aligned with the hand.

Table 5. Classification accuracy of the three considered models (%).

	Model A	Model B	Model C
Signer 1	97.01	97.92	76.77
Signer 2	93.52	97.51	74.37
Signer 3	94.84	94.68	81.47
Global	94.95	96.79	77.19

For benchmark purposes, an additional experiment is conducted by performing both training and testing of the SVM on the data from the MPS exclusively. Thus, the data from the image dataset processed with the MediaPipe library are not used. Specifically, training is performed on MPS data from Signer 1 and Signer 2, whereas testing is performed on MPS data from Signer 3. The goal of this additional experiment is to study the best possible performance of the proposed method, since in this case, we operate on data acquired by the same system. Such additional benchmark experiment resulted in an accuracy of 95.66% for Model A, 97.71% for Model B, and 98.17% for Model C.

4. Discussion

From the experimental results presented in Figures 12–14, it is possible to notice that the best performance is provided by Model B, among the considered models. As shown in Table 5, Model B results in an accuracy of approximately 97%, which is comparable with the best results from the literature presented in Section 1. However, the proposed approach is more robust than the methods from the literature since it uses a tracking system to extract hand features, which makes it less sensitive to environmental conditions, changes in the input device, and signer variability.

On the other hand, the less complex Model A results in a slightly lower accuracy of approximately 95%. This proves that the kinematic hand model used in Model B for reconstructing the hand position, as explained in Section 2.2, provides richer information with respect to the raw data used by Model A.

From the confusion matrix shown in Figure 12, it is possible to notice some non-zero off-diagonal elements, which correspond to misclassified letters. In particular, when using Model A, the letter N is incorrectly classified as M in 11% of cases, as S in 14% of cases, and as T in 12% of cases. Such misclassifications are due to the fact that the positions of the fingertips in the letters M, N, S, and T are similar. Since it uses only the fingertip positions, Model A is not robust for capturing the differences between these letters. On the other hand, Model B, which employs a reconstruction of the entire hand by the kinematic model, allows for a more robust classification. In fact, from Figure 13, it can be seen that the letter N is misclassified in only 5% of the cases when using Model B.

Overall, the results demonstrate the feasibility of the proposed machine learning approach. In particular, it is shown that an SVM classification model can be trained on data from a large publicly-available dataset of images, which are processed by a hand tracking algorithm (MediaPipe Hands) and then successfully tested on a completely different system, the MPS. Thus, the proposed method demonstrated a high generalization capability, which results in a fundamental robustness to environmental factors.

Furthermore, results from Model C show that it is possible to recognize hand gestures from arbitrary viewpoints and orientations, using a 3D tracking system such as the MPS. Therefore, the recognition system is more flexible when using Model C. The results are worse than Model A and Model B in terms of accuracy, but the additional flexibility could be exploited also in other hand gesture recognition applications.

Finally, by comparing the results in Table 5 with those of the benchmark experiment that uses MPS data exclusively, it may be observed that the benchmark results in only a slight increase in accuracy for Models A and B. Therefore, the good generalization property of the proposed method is further demonstrated. Instead, the benchmark results in a considerable accuracy increase for Model C, from 77% to 98%, thanks to the training performed on data from the magnetic positioning system. This increase proves that it is possible to implement a gesture recognition system that is both accurate and flexible, by using the developed MPS exclusively.

5. Conclusions

In this paper, we proposed a machine learning method in conjunction with a magnetic positioning system for recognizing the static gestures associated with the sign language alphabet. The architecture and features of the magnetic positioning system were described. The proposed machine learning method is based on a support vector machine trained on a large publicly available image dataset, which is processed by the MediaPipe hand tracking library.

The proposed method is experimentally tested on data obtained from the magnetic positioning system, utilizing a kinematic model of the hand. Results validate the proposed approach, which demonstrated good generalization properties and resulted in a classification accuracy of approximately 97%.

Therefore, the proposed gesture recognition method may enable the development of efficient automated translation systems for sign language. Such systems have the potential to support efficient communication and human–machine interaction for individuals that are unable to speak and hearing-impaired.

Future developments of the research activity we presented in this paper involve the application of the developed method and experimental setup to other hand-gesture recognition problems. Such problems include, among others, augmented reality scenarios, robotics, and automation applications. Finally, future analysis will be focused on extending the proposed method to dynamic gesture recognition.

Author Contributions: Conceptualization, A.D.A., F.S., E.B., P.C.; methodology, M.R., A.D.A., F.S., E.B., A.M., P.C., P.B., M.S.; software, M.R., F.S., P.B., M.S.; validation, M.R., A.D.A., F.S.; investigation, M.R., A.D.A., F.S., E.B.; data curation, M.R., A.D.A., F.S.; writing—original draft preparation, M.R., A.D.A., F.S., E.B., A.M., P.C., P.B., M.S.; writing—review and editing, M.R., A.D.A., F.S., E.B., A.M., P.C., P.B., M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Encyclopedia Britannica, "Sign Language". 12 November 2020. Available online: https://www.britannica.com/topic/sign-language (accessed on 12 May 2021).
2. Eberhard, D.M.; Simons, G.F.; Fennig, C.D. (Eds.) *Ethnologue: Languages of the World*, 24th ed.; SIL International: Dallas, TX, USA, 2021. Available online: https://www.ethnologue.com/subgroups/sign-language (accessed on 12 May 2021).
3. World Federation of the Deaf. Available online: http://wfdeaf.org/our-work/ (accessed on 12 May 2021).
4. Wadhawan, A.; Kumar, P. Sign Language Recognition Systems: A Decade Systematic Literature Review. *Arch. Comput. Methods Eng.* **2019**, *28*, 785–813. [CrossRef]
5. Fingerspelling. Wikipedia. Available online: https://en.wikipedia.org/wiki/Fingerspelling (accessed on 12 May 2021).
6. Cooper, H.; Holt, B.; Bowden, R. Sign Language Recognition. In *Visual Analysis of Humans*; Moeslund, T., Hilton, A., Krüger, V., Sigal, L., Eds.; Springer: London, UK, 2011.
7. Dong, J.; Tang, Z.; Zhao, Q. Gesture recognition in augmented reality assisted assembly training. *J. Phys. Conf. Ser.* **2019**, *1176*, 032030. [CrossRef]
8. Ascari Schultz, R.E.O.; Silva, L.; Pereira, R. 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. [CrossRef]
9. Kakkoth, S.S.; Gharge, S. 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. [CrossRef]
10. Simão, M.A.; Gibaru, O.; Neto, P. Online Recognition of Incomplete Gesture Data to Interface Collaborative Robots. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9372–9382. [CrossRef]
11. Ding, I.; Chang, C.; He, C. 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. [CrossRef]
12. Yang, S.; Lee, S.; Byun, Y. Gesture Recognition for Home Automation Using Transfer Learning. In Proceedings of the 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Bangkok, Thailand, 21–24 October 2018; pp. 136–138. [CrossRef]
13. Ye, Q.; Yang, L.; Xue, G. Hand-free Gesture Recognition for Vehicle Infotainment System Control. In Proceedings of the 2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan, 5–7 December 2018; pp. 1–2. [CrossRef]
14. Akhtar, Z.U.A.; Wang, H. WiFi-Based Gesture Recognition for Vehicular Infotainment System—An Integrated Approach. *Appl. Sci.* **2019**, *9*, 5268. [CrossRef]
15. Meng, Y.; Li, J.; Zhu, H.; Liang, X.; Liu, Y.; Ruan, N. Revealing your mobile password via WiFi signals: Attacks and countermeasures. *IEEE Trans. Mob. Comput.* **2019**, *19*, 432–449. [CrossRef]
16. Cheok, M.J.; Omar, Z.; Jaward, M.H. A review of hand gesture and sign language recognition techniques. *Int. J. Mach. Learn. Cyber.* **2019**, *10*, 131–153. [CrossRef]
17. Elakkiya, R. Machine learning based sign language recognition: A review and its research frontier. *J. Ambient. Intell. Hum. Comput.* **2020**. [CrossRef]
18. Rastgoo, R.; Kiani, K.; Escalera, S. Sign Language Recognition: A Deep Survey. *Expert Syst. Appl.* **2021**, *164*. [CrossRef]
19. Sharma, S.; Kumar, K. ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. *Multimed. Tools Appl.* **2021**. [CrossRef]
20. Luqman, H.; El-Alfy, E.S.; BinMakhashen, G.M. Joint space representation and recognition of sign language fingerspelling using Gabor filter and convolutional neural network. *Multimed. Tools Appl.* **2021**, *80*, 10213–10234. [CrossRef]
21. Shi, B.; Del Rio, A.M.; Keane, J.; Michaux, J.; Brentari, D.; Shakhnarovich, G.; Livescu, K. American sign language fingerspelling recognition in the wild. In Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 145–152. [CrossRef]
22. Jiang, X.; Zhang, Y.D. Chinese sign language fingerspelling via six-layer convolutional neural network with leaky rectified linear units for therapy and rehabilitation. *J. Med. Imaging Health Inform.* **2019**, *9*, 2031–2090. [CrossRef]
23. Tao, W.; Leu, M.C.; Yin, Z. American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. *Eng. Appl. Artif. Intell.* **2018**, *76*, 202–213. [CrossRef]
24. Bird, J.J.; Ekárt, A.; Faria, D.R. British Sign Language Recognition via Late Fusion of Computer Vision and Leap Motion with Transfer Learning to American Sign Language. *Sensors* **2020**, *20*, 5151. [CrossRef] [PubMed]
25. Barbhuiya, A.A.; Karsh, R.K.; Jain, R. CNN based feature extraction and classification for sign language. *Multimed. Tools Appl.* **2021**, *80*, 3051–3069. [CrossRef]
26. Warchoł, D.; Kapuściński, T.; Wysocki, M. Recognition of Fingerspelling Sequences in Polish Sign Language Using Point Clouds Obtained from Depth Images. *Sensors* **2019**, *19*, 1078. [CrossRef] [PubMed]
27. Lee, C.K.M.; Ng, K.K.H.; Chen, C.-H.; Lau, H.C.W.; Chung, S.Y.; Tsoi, T. American sign language recognition and training method with recurrent neural network. *Expert Syst. Appl.* **2021**, *167*, 114403. [CrossRef]

28. Rastgoo, R.; Kiani, K.; Escalera, S. Multi-Modal Deep Hand Sign Language Recognition in Still Images Using Restricted Boltzmann Machine. *Entropy* **2018**, *20*, 809. [CrossRef] [PubMed]
29. Chong, T.-W.; Lee, B.-G. American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach. *Sensors* **2018**, *18*, 3554. [CrossRef]
30. Pezzuoli, F.; Corona, D.; Corradini, M.L.; Cristofaro, A. Development of a Wearable Device for Sign Language Translation. In *Human Friendly Robotics*; Ficuciello, F., Ruggiero, F., Finzi, A., Eds.; Springer: Cham, Switzerland, 2019. [CrossRef]
31. Yuan, G.; Liu, X.; Yan, Q.; Qiao, S.; Wang, Z.; Yuan, L. Hand Gesture Recognition Using Deep Feature Fusion Network Based on Wearable Sensors. *IEEE Sensors J.* **2020**, *21*, 539–547. [CrossRef]
32. Ahmed, M.A.; Zaidan, B.B.; Zaidan, A.A.; Salih, M.M.; Al-qaysi, Z.T.; Alamoodi, A.H. Based on wearable sensory device in 3D-printed humanoid: A new real-time sign language recognition system. *Measurement* **2021**, 108431. [CrossRef]
33. Khomami, S.A.; Shamekhi, S. Persian sign language recognition using IMU and surface EMG sensors. *Measurement* **2021**, 108471. [CrossRef]
34. Siddiqui, N.; Chan, R.H.M. Hand Gesture Recognition Using Multiple Acoustic Measurements at Wrist. *IEEE Trans. Hum. Mach. Syst.* **2021**, *51*, 56–62. [CrossRef]
35. Zhao, T.; Liu, J.; Wang, Y.; Liu, H.; Chen, Y. Towards Low-Cost Sign Language Gesture Recognition Leveraging Wearables. *IEEE Trans. Mob. Comput.* **2021**, *20*, 1685–1701. [CrossRef]
36. Santoni, F.; De Angelis, A.; Moschitta, A.; Carbone, P. A Multi-Node Magnetic Positioning System with a Distributed Data Acquisition Architecture. *Sensors* **2020**, *20*, 6210. [CrossRef]. [CrossRef]
37. Santoni, F.; De Angelis, A.; Skog, I.; Moschitta, A.; Carbone, P. Calibration and Characterization of a Magnetic Positioning System Using a Robotic Arm. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 1494–1502. [CrossRef]. [CrossRef]
38. Santoni, F.; De Angelis, A.; Moschitta, A.; Carbone, P. MagIK: A Hand-Tracking Magnetic Positioning System Based on a Kinematic Model of the Hand. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]. [CrossRef]
39. Moschitta, A.; De Angelis, A.; Santoni, F.; Dionigi, M.; Carbone P.; De Angelis, G. Estimation of the Magnetic Dipole Moment of a Coil Using AC Voltage Measurements. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2495–2503. [CrossRef]. [CrossRef]
40. Cypress Semiconductor, "CYBLE-222014-01 EZ-BLE™ Creator Module". Available online: https://www.cypress.com/file/230691/download (accessed on 20 April 2021).
41. Heydon, R. *Bluetooth Low Energy: The Developer's Handbook*; Prentice Hall: Indianapolis, IN, USA, 2013.
42. Craig, J.J. *Introduction to Robotics, Mechanics and Control*, 3rd ed.; Pearson: London, UK, 1986.
43. Greiner, T.M. *Hand Anthropometry of U.S. Army Personnel*; Technical Report Natick/TR-92/011; U.S. Army Natick Research, Development and Engineering Center Natick: Natick, MA, USA, 1991.
44. Lee, J.; Kunii, T.L. Constraint-Based Hand Animation. In *Models and Techniques in Computer Animation*; Thalmann, N.M., Thalmann, D., Eds.; Springer: Tokyo, Japan, 1993; pp. 110–127.
45. Rosenblatt, F. *The Perceptron—A Perceiving and Recognizing Automaton*; Technical Report 85-460-1; Cornell Aeronautical Laboratory: 1957. Available online: https://blogs.umass.edu/brain-wars/files/2016/03/rosenblatt-1957.pdf (accessed on 19 May 2021).
46. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer Science and Business Media: Berlin/Heidelberg, Germany, 2009.
47. Gunn, S.R. *Support Vector Machines for Classification and Regression*; Technical Report; University of Southampton: Southampton, UK, 1998.
48. Zanaty, E.; Afifi, A. Support Vector Machines (SVMs) with Universal Kernels. *Appl. Artif. Intell.* **2011**, *25*, 575–589. [CrossRef]
49. Jordan, M.; Thibaux, R. The Kernel Trick, CS281B/Stat241B: Advanced Topics in Learning and Decision Making. Available online: https://people.eecs.berkeley.edu/~jordan/courses/281B-spring04/lectures/lec3.pdf (accessed on 16 June 2021)
50. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.; Yong, M.; Lee, J.; et al. MediaPipe: A Framework for Building Perception Pipelines. 2019. Available online: https://arxiv.org/abs/1906.08172 (accessed on 19 May 2021).
51. MediaPipe Hands. Available online: https://google.github.io/mediapipe/solutions/hands.html (accessed on 25 April 2021).
52. Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.; Grundmann, M. MediaPipe Hands: On-Device Real-Time Hand Tracking. 2020. Available online: https://arxiv.org/abs/2006.10214 (accessed on 19 May 2021).
53. ASL Sign Language Alphabet Pictures [Minus J, Z]. Available online: https://www.kaggle.com/signnteam/asl-sign-language-pictures-minus-j-z (accessed on 25 April 2021).