applied sciences

MDPI

Article

Deep Learning Techniques Applied to Predict and Measure Finger Movement in Patients with Multiple Sclerosis

Dmitry Viatkin [1,2], Begonya Garcia-Zapirain [2] and Amaia Méndez Zorrilla [2,*]

[1] High School of Information Technologies and Automatic Systems, Northern Artic Federal University, 163002 Arkhangelsk, Russia; dmitrii.viatkin@opendeusto.es
[2] The Deustotech-LIFE (eVIDA) Research Group, Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain; mbgarciazapi@deusto.es
* Correspondence: amaia.mendez@deusto.es

Abstract: This research focuses on the development of a system for measuring finger joint angles based on camera image and is intended for work within the field of medicine to track the movement and limits of hand mobility in multiple sclerosis. Measuring changes in hand mobility allows the progress of the disease and its treatment process to be monitored. A static RGB camera without depth vision was used in the system developed, with the system receiving only the image from the camera and no other input data. The research focuses on the analysis of each image in the video stream independently of other images from that stream, and 12 measured hand parameters were chosen as follows: 3 joint angles for the index finger, 3 joint angles for the middle finger, 3 joint angles for the ring finger, and 3 joint angles for the pinky finger. Convolutional neural networks were used to analyze the information received from the camera, and the research considers neural networks based on different architectures and their combinations as follows: VGG16, MobileNet, MobileNetV2, InceptionV3, DenseNet, ResNet, and convolutional pose machine. The final neural network used for image analysis was a modernized neural network based on MobileNetV2, which obtained the best mean absolute error value of 4.757 degrees. Additionally, the mean square error was 67.279 and the root mean square error was 8.202 degrees. This neural network analyzed a single image from the camera without using other sensors. For its part, the input image had a resolution of 512 by 512 pixels, and was processed by the neural network in 7–15 ms by GPU Nvidia 2080ti. The resulting neural network developed can measure finger joint angle values for a hand with non-standard parameters and positions.

Keywords: convolutional neural network; layer; finger; joint angle; hands

Citation: Viatkin, D.; Garcia-Zapirain, B.; Méndez Zorrilla, A. Deep Learning Techniques Applied to Predict and Measure Finger Movement in Patients with Multiple Sclerosis. *Appl. Sci.* **2021**, *11*, 3137. https://doi.org/10.3390/app11073137

Academic Editors: Il Dong Yun and Soochahn Lee

Received: 28 February 2021
Accepted: 30 March 2021
Published: 1 April 2021

1. Introduction

Multiple sclerosis is a potentially disabling disease of the central nervous system, with the symptoms of this disease varying widely and depending on the amount of nerve damage [1]. Some people with multiple sclerosis may lose the ability to move, whereas others may experience long periods of remission without any new symptoms.

There is no cure for multiple sclerosis, although treatment can help speed up recovery from attacks, modify the course of the disease, and manage symptoms.

Symptoms of multiple sclerosis may differ greatly over the course of the disease depending on the location of the nerve fibers affected. Symptoms often affect movement, such as numbness or weakness in one or more limbs that typically occurs on one side of the body at a time, or the legs, hands or trunk.

Changes in human movement need to be detected to track the disease process and its treatment. People use their hands in all their daily activities, although the structure of the hand is complex, as it has many joints. In the case of multiple sclerosis of the hand, any limitations of each joint must be measured, and development of an automatic system capable of quickly measuring the position of the angles of the joints of the hand will

enable the development and treatment of multiple sclerosis to be monitored and treated. Nowadays, these measurements require an expensive set of calibrated medical sensors, and so using only one camera and computer vision algorithms for this task makes such a system cheap and generally available.

There are various methods for assessing the position and parameters of the hand. Many researchers measure this using computer vision techniques, such as gesture recognition, rather than measuring each angle of the finger independently. Many also focus on algorithms to measure other hand parameters besides finger hand position.

An approach based on an application involving computer vision for hand gesture recognition was described in the publications [2,3] in which separate hand gestures were able to be measured but each angle of the finger could not be measured independently. These approaches cannot be used independently for the angle of the finger.

There are studies that focus not only on hand gesture recognition, but also on describing the position of the hand by measuring individual angles of the finger joints. An approach based on strain sensors was described in the work [4]. In this research, sensors mounted on the hand measured its joint angle position, obtaining a 64 mean absolute error value of 1.63 degrees.

Publication [5] described an approach based on soft strain sensors. In this research, sensors mounted on the hand measure its joint angle position, obtaining an error value of 3.5 degrees.

A non-invasive automatic goniometer test device for joint angle measuring was developed in Reference [6], where measurement accuracy of the goniometer device proposed was less than 6.6 degrees.

In Reference [7], an approach to detect finger orientation using depth cameras is detailed. This approach does not focus on the medical field of activity and measurement of finger joints, but rather, aims to process the position of the human hand in real time. The research measures the position of the fingers of the hand based on data obtained from the entire 3D shape of the hand, with the Root Mean Square Error (RMSE) error for finger detection position being 8.74 degrees.

A system for estimating the natural postures of a user's fingers from the images was developed in Reference [8]. Images were captured by an omnidirectional video camera attached to the center of the user's palm in real time. The positional relationship between the camera and the user's fingers, the length between the finger joints, and interdependencies between the finger joints were provided for such purpose, and the error value of the system developed was five degrees.

A system for estimating the natural postures of a user's fingers from the images captured by a fisheye camera has been developed and described in Reference [9]. This research obtained average errors with an approximate value of 20 mm for fingertip tracking across the different device sizes. The contact finger and hand posture classifiers showed approximately 83% and 90% accuracy, respectively, across the device sizes.

In Reference [10], a system for recognizing trajectory-based dynamic hand gestures in real time was developed for human–computer interaction. This research obtained gesture detection and recognition performance to 73% accuracy in a stream of motion.

The work detailed in Reference [11] described a real-time on-device hand tracking pipeline that predicts a hand skeleton from a single RGB camera. The mean regression error value normalized by palm size obtained was 13.4%.

An electronic skin integrated with a deep neural network was developed by authors of [12]. The system developed captures dynamic motions from a distance without creating a sensor, and can be used in health-monitoring, motion tracking, and soft robotics.

A neural network-based algorithm with spatial pyramid pooling was proposed in Reference [13], which decodes gestures or sign language fingerspelling from videos. The algorithm described provided high benchmark accuracy with a fast-processing speed.

In Reference [14], a hand gesture recognition sensor was developed using ultra-wideband impulse signals. Convolutional neural network was used for reflected waveforms analysis for American sign language gesture classification, with average recognition accuracy also being above 90%.

Authors of [15] developed an approach for safe and object-independent human-to-robot handovers using real time robotic vision and manipulation based on human body part segmentation and hand/finger segmentation. The resulting robot was able to successfully interact with a human and take the object from the human in 81.9% of trials.

In the work detailed in Reference [16], data was measured using functional near-infrared spectroscopy during finger tapping tasks. The data augmentation method proposed was used to generate datasets, which in turn. were used to train an AlexNet neural network model. The acquired accuracy attained constituted an improvement compared to that obtained using original data.

The finger vein recognition problem was researched, in the course of which convolutional neural networks for feature extraction for the vein recognition task were used in publication [17]. Classification and correct identification accuracy in ranges upwards of 95% and equal error rates below 4% were obtained accordingly.

Many studies focus on specific gesture recognition and hand parameters rather than on determining the position of the hand. These studies also described the above work well when analyzing hands with standard parameters and positions.

The present work focuses on the development of a system for measuring finger joint angles based on camera image and is intended for work within the field of medicine to track the movement and limits of hand mobility in multiple sclerosis. In a healthy hand, the position of the fingers and its joints cannot change independently of each other. However, when the hand is injured, the degree of, finger mobility may change, as people with multiple sclerosis may have a hand with non-standard parameters and positions. As a result, conventional approaches to measuring arm position may not work properly in such cases.

The research also develops a system capable of measure hand finger joint angles for hands with standard and non-standard parameters and positions. This is possible because training and test datasets of images of the hand are used not only with normal, but also non-standard hand parameters.

Tracking the limitation of hand joint mobility in multiple sclerosis is a complex task, as there are many mobile joints in the hand, the position of which needs to be measured at a specific moment to a good level of accuracy. Measuring changes in hand mobility allows progress of the disease and its treatment process to be monitored, and the use of a single ordinary RGB camera in this research makes the system developed for finger angle measuring cheaper and easier to install and operate compared to sensor-based systems and specialist cameras. The system developed in the article will make it possible to take high-precision and high-speed measurements of the finger position at low cost.

Figure 1 shows the position of the joints on the hand selected.

Figure 1. The positions of joint angles measured.

The 12 measured hand parameters were chosen in this research as follows: 3 joint angles for the index finger, 3 joint angles for the middle finger, 3 joint angles for the ring finger, and 3 joint angles for the pinky finger.

The article is organized as follows: After providing the introduction in Section 1, Materials and Methods used in this paper are then described in Section 2. Section 2 describes the data collection process description and reasoning, the software and libraries used in the research, the neural network architectures used, and their metrics and hyperparameters. Section 3 describes the different approaches to neural network architecture development and training. This section describes different approaches to joint angle measurement based on a single image using convolution neural networks and the results obtained from these approaches. Results and Discussion are provided in Section 4, with the conclusions drawn from this research and future work being described at the end of the article.

2. Materials and Methods

2.1. Sample and Data

Computer vision and neural network approaches require big datasets. This research is intended to work within the field of medicine and requires a dataset with hand images and information about hand position in different positions and states.

It is necessary for the dataset to contain not only hand data and images in normal positions and states, but also hands with unusual parameters in unusual positions. Creating a large dataset containing hands with unusual parameters and in unusual positions with use of real data is both difficult and expensive.

A sensor-based hand position measurement approach was considered for the collection of hand position data, whereby a set of sensors is used to detect the position of each finger joint. The dataset obtained from this approach would contains only images of the hands covered by the sensors, and the computer vision system based on a neural network was trained using this dataset. As a result, this system would be unable to work with images without sensors, and so it was decided not to use a sensor-based approach to hand position measurement.

Since the use of sensors was not possible, a simulation approach was considered instead. A program using high polygonal 3D models of hands was developed to create a dataset, with the Unity engine 2019.1 (Unity Technologies, San Francisco, CA, USA) being

used to develop the hand simulation program. This program allows to simulate the real hand movement based on 3D model and capture the images of the hand. Program can change hand textures, colors, proportions, and position of elements of a hand, light and background to be simulated, and also enables the different parameters and positions of the hand to be measured, including the target 12 finger joint angles. Figure 2 shows a frame from the simulation process with additional markings of joint angles.

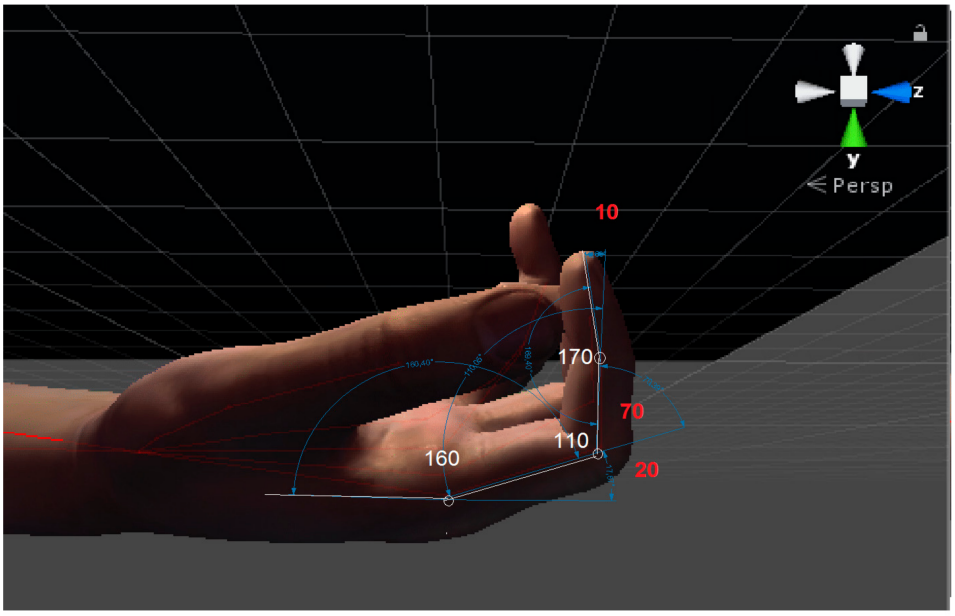

Figure 2. Frame from the simulation process with additional markings of joint angles.

The hand was rotated 90 degrees for the joint angles visualization. Camera for image capturing for dataset is directed to the front of the hand.

Data regarding the external and internal angles of the joints were collected during the simulation, with the external finger joint angles being used in the calculations by using values for external finger joint angles, it is also possible to calculate the values of the internal corners of the joints.

A dataset was created using this program that contained images of many different hands in different positions with joint angles being measured to a high degree of accuracy for every finger.

The angles of the finger joints on the three axes were changed: X axis values were varied from -10 degrees to 100 degrees, which simulated joint flexion; the Y-axis and Z-axis values were varied from -10 to 10 degrees, which simulated joint displacement. Changes in the values of the finger joint angles were limited so that the texture elements of the hand model did not overlap with each other.

The wrist position of the 3D model of the hand on three axes was also changed as follows: The X axis values were varied from -30 to 45 degrees, which simulated bending the arm at the wrist; on the Y-axis and Z-axis values varied from -15 to 15 degrees, which simulated wrist rotation.

The position of the joints in three-dimensional space relative to the arm model also shifted in the range of -5 to 5 percent relative to the parent model object size.

The size of the individual elements of the hand model was also changed, with the size of each phalanx in the three directions varying from -20 percent to 20 percent of the original size.

The lighting parameters of the 3D model were changed in terms of the intensity of the light, its color, the number of light sources, and their type. Standard Unity engine lighting sources were used for such purpose.

The structure of the 3D hand model was also changed, as was the material from which the hand is made. Each material had its own set of special parameters, although what they had in common were reflectivity values, colors, textures, and normal maps. Changing the parameters of the materials used made it possible to achieve an even greater level of variability, with five different types of materials with 10 different skin textures being used in the study.

Different variants of the background structures and fifty different textures were used, and the structure of the background objects on which these textures were applied was changed. These objects were rotated and moved on three planes, and the hand model was placed at different distances from the background objects.

These changes made it possible to create different hand models at different positions in different environments. In a healthy hand, the position of the fingers and their joints cannot change independently of each other, although the degree of finger mobility may change when the hand is injured. Obtaining hand positions that cannot be achieved by a healthy person makes it possible to expand the applicability of the system described in this study for medical purposes, while other hand parameter recognition algorithms focus on processing hand images without any significant anomalies.

Images from the dataset were used to train the neural network, and the machine learning open source library Tensorflow 1.15.0 (Google, Mountain View, CA, USA) and the programming language python 3 were used in the neural network development process. For its part, the programming language OpenCV 3.4.0 (Intel, Santa Clara, CA, USA) library was used for additional image processing, such as image noises, filters, blur, and color transformations. This additional image processing allows the influence of random factors on the neural network training, increasing the model robustness.

2.2. Models, Data Analysis and Computational Methods

Details about the neural networks used in this research are described in this subsection.

The architecture of a convolutional neural network was proposed by Yann LeCun [18], with a. convolutional neural network referring to a special type of architecture comprising artificial neural networks aimed at effective image recognition.

The work presented in this paper is based on two neural network architectures: MobileNetV2 [19] and convolutional pose machine [20].

Authors of [21] describe the first version of MobileNet, a neural network architecture for mobile and embedded video surveillance systems. This network is based on upgraded architecture that uses depth-separated convolutions to build light and deep neural networks. MobileNet has two simple, global hyperparameters that effectively compensate for delay and accuracy. These hyperparameters allow you to choose the right size model for your application based on the constraints of the task.

MobileNet is based on convolutional blocks, which have 2 layers in this architecture. The first layer is called a depthwise convolution and performs lightweight filtering by applying a single convolutional filter per input channel. The second layer is a 1 by 1 convolution layer with an ReLU activation function, which is responsible for building new features by computing linear combinations of the input channels.

A second version of MobileNet called MobileNetV2 is described in [20].

The MobileNetV2 architecture is based on an inverted residual structure, with the residual block input and output providing narrow places. MobileNetV2 uses lighter packages to filter objects at the middle extension level.

MobileNetV2 is based on convolution blocks consisting of two types of blocks. One is a residual block with step 1 and the other is a block with step 2 for size reduction. There are 3 layers for both types of blocks as follows: The first layer is a 1 by 1 convolution with ReLU activation function, whereas the second layer is a depth convolution, and the third layer is a 1 by 1 convolution without any nonlinearity activation function.

Pose Machines provide a sequential prediction framework for learning rich implicit spatial models.

The convolutional pose machine (CPM), is a combination of pose machines and provides a convolutional approach to the neural network. This machine contains framework convolutional layers for learning image features and image-dependent spatial models for pose estimation [21]. It implicitly builds model long-range dependencies between variables in structured prediction tasks such as articulated pose estimation.

The convolutional pose machine achieves this by designing sequential architecture comprising convolutional networks that directly operate on belief maps from previous stages and by producing increasingly refined estimates for part locations, without the need for explicit graphical model-style inference.

2.3. Measurement of Variables

Different methods are used to assess the quality of training and neural network performance in the training process. Prediction of the finger joint angle is a regression problem, for which purpose metrics are used as follows: Mean absolute error, mean squared error, and root mean square error.

The mean absolute error (MAE) is used to measure the error between paired observations expressing the same phenomenon. A special feature of MAE is its resistance to emissions in data, and MAE is calculated as follows:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |xi - yi|}{n} \quad (1)$$

where xi is the actual value, yi the predicted value, and n refers to the number of samples.

The mean squared error (MSE) is calculated as being the mean square difference between predicted and actual values. The result is always positive regardless of predicted or actual values, and the ideal value is 0. A square value means that larger errors result in more errors than smaller errors, which in turn means that the model is penalized for larger errors. MSE is calculated as follows:

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (xi - yi)^2 \quad (2)$$

RMSE is used to measure the difference between the values predicted by the model or evaluator and the values observed. It is calculated as the square root of MSE and large error values have a disproportionate impact on RMSE. RMSE is calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (xi - yi)^2}{n}} \quad (3)$$

3. Neural Network Development and Training

3.1. Adoption of Images without Preprocessing and Marking for Convolutional Neural Network (CNN) Training

The task of measuring joints angles in the image was divided into two subtasks: Measuring the set of parameters of the hand in the image and measuring the finger joint angles.

The subtask of measuring parameters of the hand in the image is algorithmically similar to the task of classifying images. Algorithms of convolutional neural networks work properly with image analysis and can also work properly with this task, insofar as they are able to create a set of features for images and then send it to analyze and measure the finger joint angles.

The subtask of measuring the finger joint angles by hand parameters is a regression task. In this step, fully connected layers are used to analyze a set of features and measure the finger joint angles.

To measure the parameters of the hand in the image, it was decided to use pretrained convolutional neural networks for the classification of images, and classification layers in

these networks were replaced by fully connected layers to work with the regression task. This solution was selected because measuring the finger joint angles by hand parameters is a regression task. A set of features for the hand is created after image processing by the convolutional layers, and fully connected layers allow this set of features to be analyzed and finger joint angles to be calculated.

The following neural network architectures were analyzed: MobileNet, MobileNetV2, InceptionV3, DenseNet121, DenseNet169, ResNet152, ResNet101, and VGG16.

The first approach to training the neural network was taken without preliminary image processing, with images having the following dynamic parameters: Different backgrounds, different lights, different state of hand, different colors, and other different parameters. The mean absolute error, or MAE, for finger joint angle prediction based on this approach was 19–30 degrees, with the best results being obtained by MobileNet, MobileNetV2, and VGG16. Adam optimizer with a learning rate of 0.001 was used in the training process. Batch size comprised 5 images, the number of epochs was 100, and loss function was the mean absolute error.

The best neural network architecture turned out to be MobileNetV2, the mean absolute error value of which was 19.897 degrees. The training process for MobileNet, MobileNetV2, and VGG16 architectures are shown in Figure 3. It was assumed that in the training process, the neural network cannot identify a good set of features that allow it to measure finger joint angles. For their part, weights of neural networks focus on local features and the training process was stopped at the local minimum.

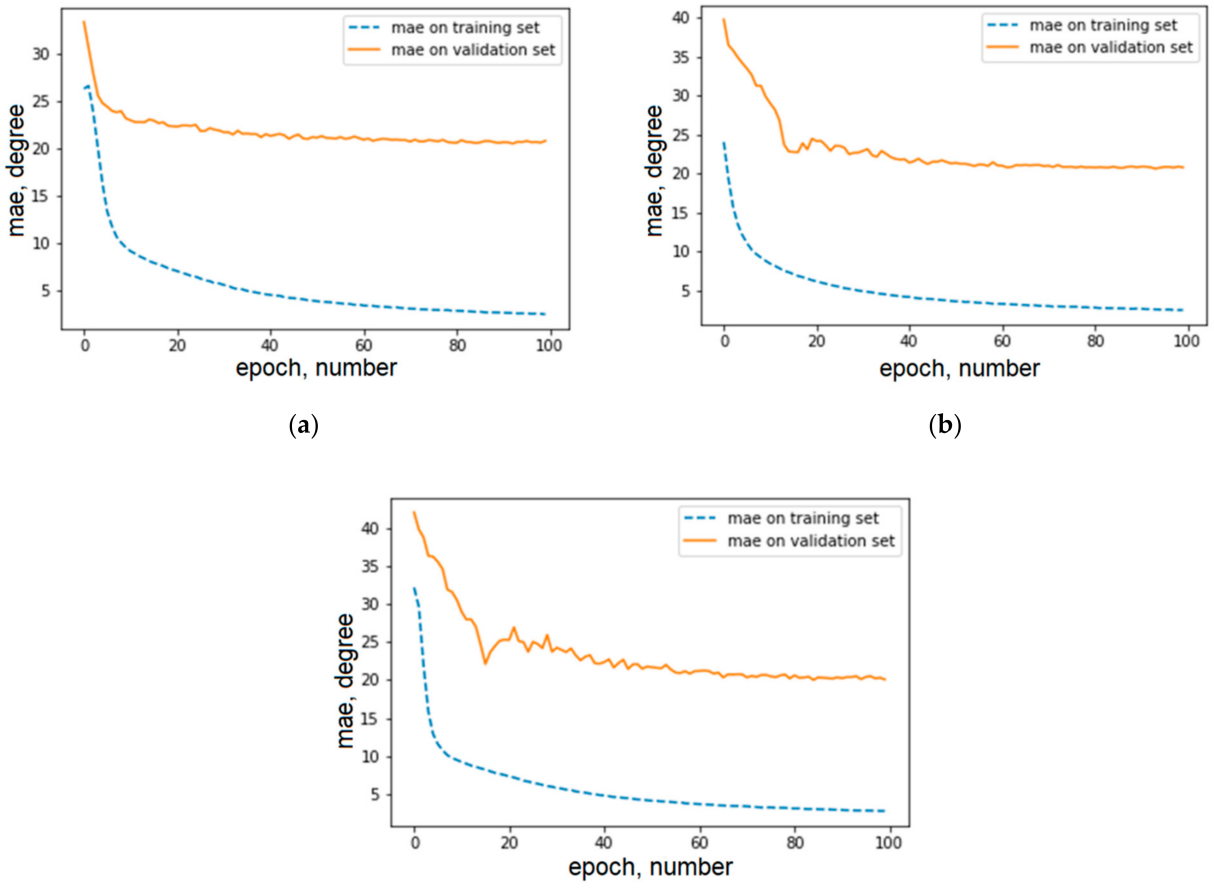

Figure 3. Mean absolute error values in the training process for neural networks, showing best results: (**a**) VGG16, (**b**) MobileNet, and (**c**) MobileNetV2.

MobileNetV2 produced the best results and was selected for the next stages of research, although at this stage the approach based on modifying and re-training neural networks for image classification evidenced poor accuracy and major errors.

The approach based on the convolutional pose machine algorithm was considered at the next stage.

3.2. Use of Convolutional Pose Machine for Measuring Joint Angles

This approach is based on the use of the open-source project described in [22]. The image of the hand is processed by a neural network from this project, with the position of the characteristic points of the hand being calculated in this image, and hand position in 2D image space described accordingly.

Use of coordinates of characteristic hand points obtained after image processing by a convolutional pose machine was considered for the purpose of measuring the joint angles. The 2D coordinates of the points were processed using different methods and were sent for processing by a fully connected neural network, this approach providing a result with a MAE between 11–22 degrees. The best result was obtained by transforming the coordinates of characteristic points into information about the line distances between each two points and the angles between each two lines.

This approach also produced results with low accuracy and major errors. It is assumed that in using only the information obtained from coordinates of characteristic points, a lot of information about the hand would be lost. The convolutional pose machine also has its own level of accuracy and error, which also affects the value of the final error. To obtain accurate coordinates of the characteristic points of the hand, the image must be of high quality and resolution, although in this case, performance decreased in geometric progression, as did the growth of the image area. The finger joint angles were measured with a mean absolute error value of 11 degrees by image with a resolution of 1024 by 1024 pixels, and this image was processed using a convolutional pose machine in 0.4–0.5 s by a GPU Nvidia 2080ti. Despite further increase in image resolution, the quality of recognition accuracy of characteristic points of the hand failed to increase accordingly.

The first approach was based on modification and re-training of neural networks for image classification and evidenced a poor level of accuracy and major errors, which was also the case when the convolutional pose machine was used. However, this approach may work with images with dynamic parameters.

An approach based on combination of first and second approaches was also considered.

3.3. Adoption of Images with Marking Applied for CNN Training

This approach is based on the idea of applying color markers to the hand image. Figure 4 shows the image marking process.

Figure 4. Image marking process.

The source image on the left side of Figure 4 is analyzed by convolutional pose machine from previous paragraph and this neural network creates an array with information about

characteristic points. The marked image is created from this array, as shown in the center of Figure 4. In this image, the characterizing points of the hands are marked by different colors and connected to each other by lines, and the combined image on the right side of Figure 4 is also created. This combined image was used for training purposes.

The convolution kernel of convolutional neural networks reacts most sharply in the presence of sharp color gradients in the image. The markup applied to the image will help in the neural network training process by showing the most important points of the hand, while the markup on the image shows the joint positions for the neural network.

Previously, the convolutional neural network was described as unable to ensure good accuracy for images with dynamical parameters. For optimization of the training process, the marked dataset was based on one constant background and one hand with the moving fingers and static proportions and other parameters. This transformation of the dataset reduces the impact of any factors that may have a negative impact on the training process involved in the neural network. Marked images with one background and static parameters were replaced by original unmarked images and hand images with dynamic parameters at the following stages.

The modified version of the MobileNetV2 neural network produced the best results in the first approach proposed, and so this is the version that was selected. In this case, the last 4 layers of the MobileNetV2 neural network were removed, and the outputs of the "block_16_project_BN" layer were sent to the flattened layer and then analyzed by the fully connected layer of 12 output neurons. Adam optimizer with a learning rate of 0.001 was used in the training process. Batch size comprised 5 images, the number of epochs was 100, and loss function was the mean absolute error.

Different versions of the MobileNetV2 for different image resolutions were tested, with the MobileNetV2 training process for different image resolutions on the marked dataset being shown in Table 1.

Table 1. Results of modified MobileNetV2 training for different image resolutions.

Width (px)	Height (px)	Mean Absolute Error (Degrees)	Root Mean Square Error (Degrees)	Number of Model Parameters
256	256	4.572	8.571	2,089,036
368	368	3.187	7.817	2,396,236
440	440	2.687	6.258	2,595,916
512	512	2.408	6.029	2,826,316
640	640	2.401	5.911	3,379,276
768	768	2.398	5.574	4,055,116

For high accuracy of the neural network with high-resolution images to be ensured, it is necessary to increase the number of network parameters, which slows down the neural network. The network accuracy increases by using high resolution images, however after reaching a resolution of 512 by 512 pixels the accuracy increase slows down sharply. The architecture of the modified MobileNetV2 neural network is shown in Figure 5.

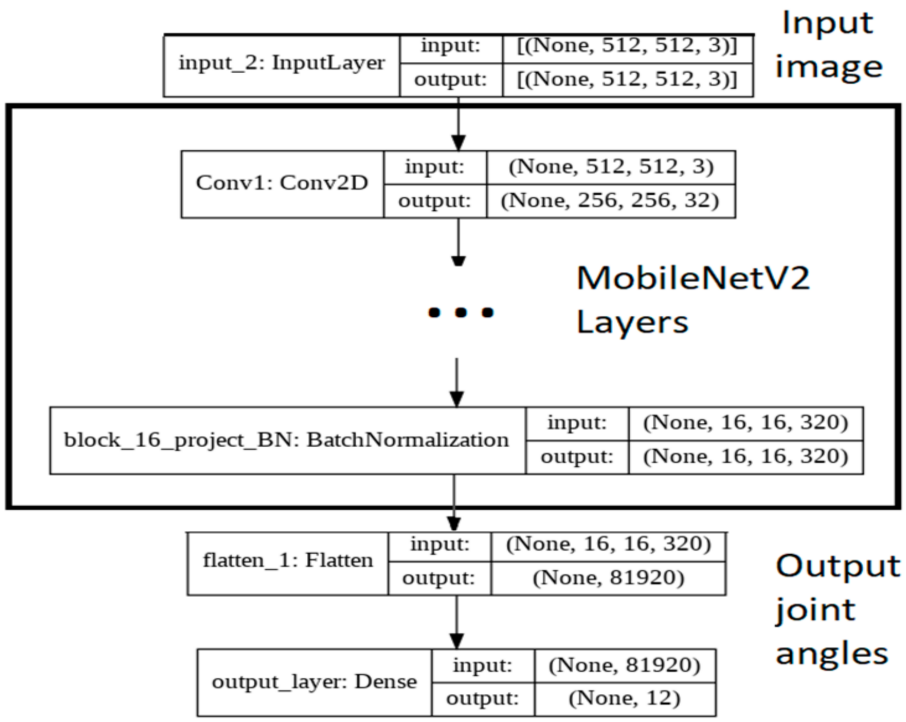

Figure 5. Architecture of modified MobileNetV2.

Increasing the resolution of the input image above 512 by 512 pixels does not provide any significant increase in the accuracy of the neural network, although it increases the size of the neural network and decreases its speed. Based on the data analysis from Table 1, a resolution of 512 by 512 pixels was selected as the suitable size of the input image.

The training process of modified MobileNetV2 for images of 512 by 512 pixels in size on the marked images is shown in Figure 6. No data augmentation or dynamic parameters were used at this stage, although marked images of the hand in a fixed position with a constant background were employed. Only the position of the fingers of the hand was changed in the images.

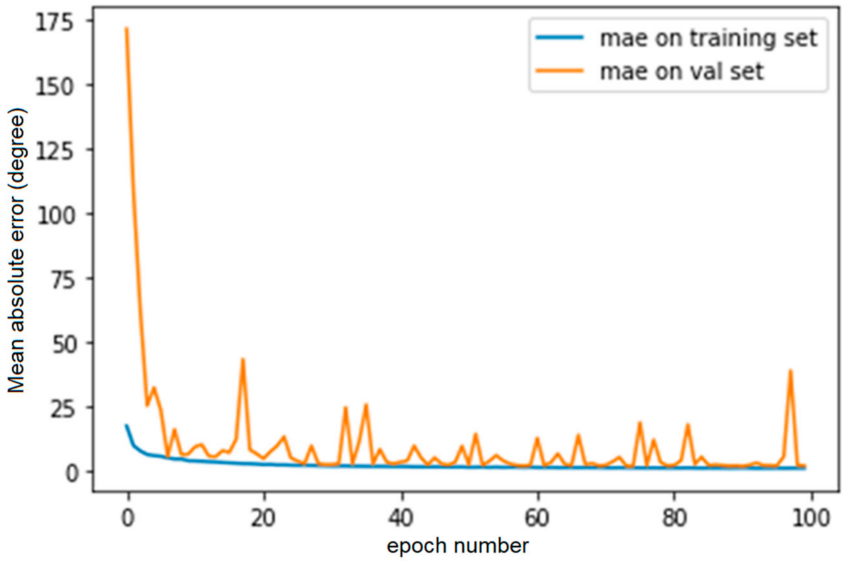

Figure 6. Mean absolute error values in the neural network training process for marked images of 512 by 512 pixels in size.

As a result of training using the marked images, the best mean absolute error value of 2.408 degrees was obtained, and neural network parameters for this mean absolute error value were saved.

After training using the marked dataset, the marked images in the dataset were then replaced by original images without marking. These images still have no dynamic background hand parameters, but they are not processed by convolutional pose machine.

Pretrained on marked images, the neural network then trains on the unmarked images. Training is provided in 30 epochs with a learning rate of 0.0001, with the loss function being the mean absolute error. The training process is shown in Figure 7.

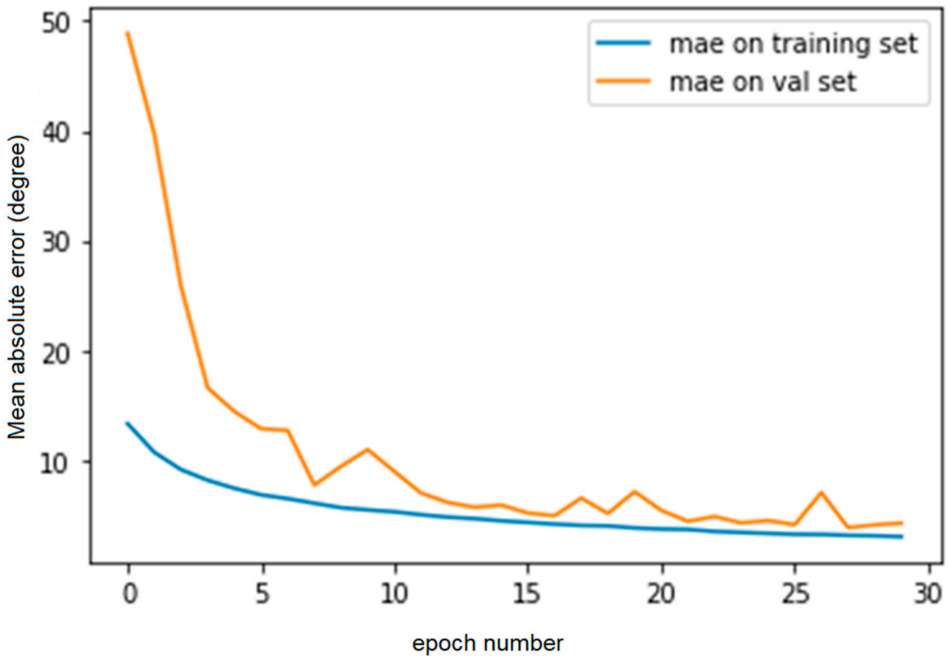

Figure 7. Mean absolute error values in the neural network training process for unmarked images of 512 by 512 pixels in size.

The neural network was trained on the unmarked images, from which the best mean absolute error value of 2.563 degrees was obtained. The neural network mean absolute error value was increased after training by unmarked images, although performance of the computer vision system had increased significantly. This occurred because images no longer require additional markup obtained using a convolutional pose machine based neural network. The convolutional pose machine was no longer used in the system after this stage.

3.4. Use of Unmarked Images to Train the Modified MobileNetV2 Neural Network

After training on the unmarked dataset, some parts of the images in it were replaced by images with dynamic parameters. New hand images have different backgrounds and different hand parameters, and this approach allows the neural network to recognize joint angles in the training process without there being any additional elements or hints in the image. Replaced images were also processed by different filters at this stage as follows: Blur, noise, color filtering, and others. This stage increases the mean absolute error for the dataset, although such processing reduces errors and increases reliability of the neural network in real life work.

After pretraining on unmarked images without dynamic parameters, the neural network then trains on the unmarked images with dynamic parameters. Table 2 shows the accuracy of the neural network after completion of the full training process, depending on the number of images that are replaced at each stage. The minimum mean absolute error

value was 4.945, this value being obtained after all unmarked images without dynamic parameters were replaced by images with dynamic hand and background parameters 10 times by parts of 10% size. For each stage we used a learning rate value of 0.001, 40 epochs, and a batch size comprising 5 images. Images were mixed in the training process so as to ensure their uniform distribution in batches used in the neural network training process.

Table 2. Neural network training results regarding the number of images that are replaced at each stage.

Replaced Image Part Per Stage, as a%	Stages	Mean Absolute Error for Final Model, in Degrees	Root Mean Square Error for Final Model, in Degrees
50	2	8.757	10.021
40	3	8.851	10.153
30	4	8.179	10.072
20	5	6.673	9.113
10	10	4.945	8.641
5	20	5.257	8.875

Table 3 shows the results of the training process with 10% replaced image parts per stage, and also the mean absolute error value at the end of each stage of training. In Table 3 the mean absolute error value increased in the first three stages and then there was a sharp decrease at the following stages. A similar pattern was observed in experiments with a different stage size. The growth of the error at the first stages was because the neural network was originally trained using simplified data and was unable to fully learn to work with unmarked images with dynamic parameters. The neural network recognizes complicated elements of images as noise or error values at the first stages, although all variations of neural networks in the experiment started to reduce the error value on reaching a certain percentage of the image's replacement. Experimentally, it was noticed that error value reduction occurred when 20–30% of images were replaced by versions with dynamical parameters.

Table 3. Results of training process with 10% replaced image part stage.

Step	Replaced Images, as a%	Mean Absolute Error for Final Model, in Degrees	Root Mean Square Error for Final Model, in Degrees
1	10	6.878	9.517
2	20	7.282	9.621
3	30	8.272	10.027
4	40	8.178	10.001
5	50	7.421	9.892
6	60	7.012	9.492
7	70	6.511	9.187
8	80	5.578	8.988
9	90	5.087	8.702
10	100	4.945	8.504

Training process for the last stage of the training process with 10% replaced image part stage shown in Figure 8.

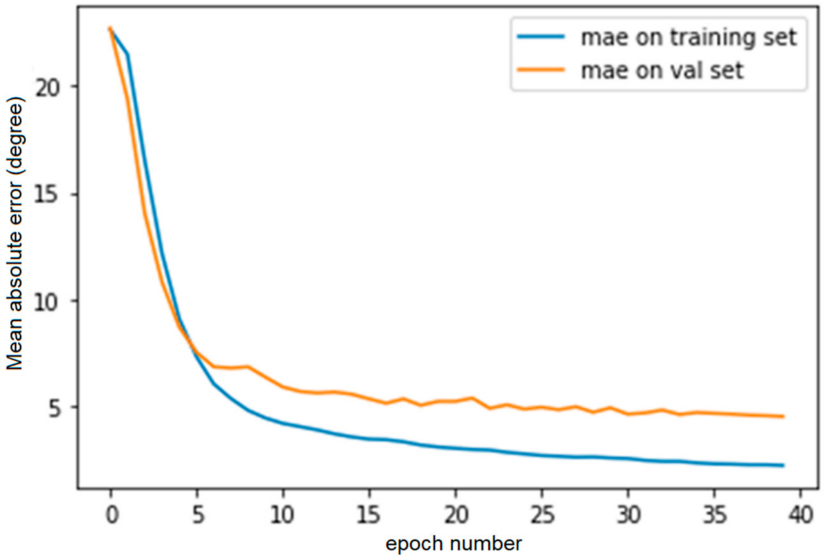

Figure 8. The last stage of the training process with 10% replaced image part stage.

This neural network in the learning process using undetected images was trained to correctly work with different hand parameters, such as different finger sizes, deformations of the hand, skin color, type of light, variants in the background, image noises, and filters.

3.5. Use of the Background Subtraction Function to Improve Modified MobileNetV2 Neural Network Results

During training it was noticed that the background of the image had the greatest influence on neural network accuracy. Since the camera is considered stationary, it was decided to use an algorithm to subtract the background, which was implemented using the OpenCV library. Figure 9 shows the result of background subtraction for the image.

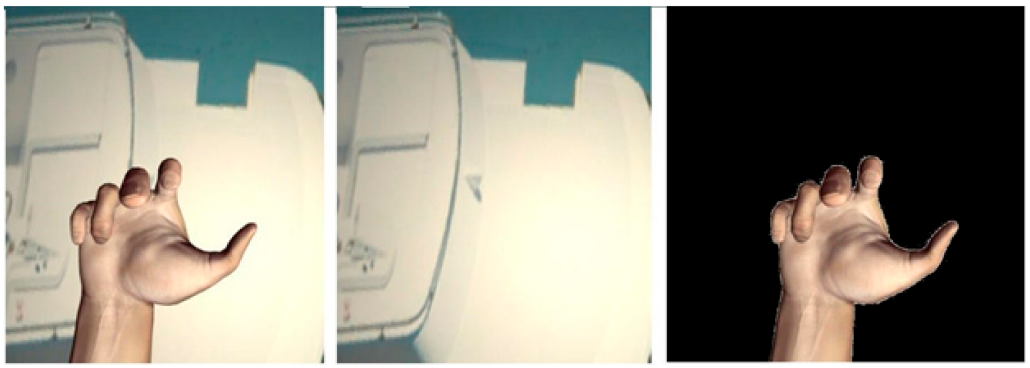

Figure 9. Result of background subtraction for the image.

After pretraining on images with dynamic parameters, the neural network then trains on the images following background subtraction. These images have different dynamic parameters, resolution, and quality, and are subject to transformation by different filters. Training is provided in 40 epochs with a learning rate of 0.0001, with the loss function being the mean absolute error. The training process is shown in Figure 10.

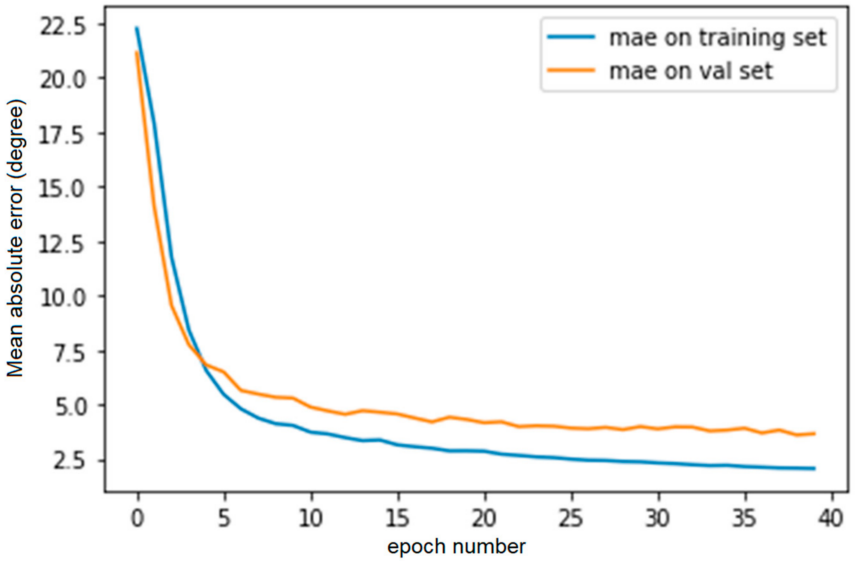

Figure 10. Mean absolute error values in the training process of neural network for images following background subtraction.

For a neural network trained on images without background, the minimum mean absolute error value was 4.757 degrees, the mean square error was 67.279, and the root mean square error was 8.202. This neural network analyzes a single image from the camera, and the input image had a resolution of 512 by 512 pixels. This image was processed by the neural network in 7–15 ms using a GPU Nvidia 2080ti.

4. Results

This research considered an approach to measuring the finger joint angles in fields of medicine based on computer vision and neural networks. The convolution neural network for analysis of static RGB images from the camera for 12 finger joint angle measuring was duly developed for such a purpose. This neural network receives only the images from the camera and does not use any other input data. The research focused the analysis of each image on the video stream independently of other images from that stream.

Reconstructing a full 3D model of the hand is not necessary in order to achieve the main aim of the research. Rather, the aim was to measure the finger joint angles of the hand. This is a usual regression task, hence metrics such as the mean absolute error and mean square error.

A comparison of neural network accuracy with the results obtained from previous research is shown in Table 4.

Table 4. Comparison of research results.

Algorithm Article	Main Device	Feature	Mean Absolute Error, in Degrees	Root Mean Square Error, in Degrees
[4]	Fiber Bragg grating strain sensor	Only sensors are used	1.63	-
[5]	Soft strain sensors	Only sensors are used	3.5	-
[6]	Goniometer device	Use of non-invasive automatic goniometer test device	6.6	-
[7]	Depth cameras	Not aimed at medical use. Use of depth cameras	-	8.74
[8]	Omnidirectional camera	Use of hand preset information. Use of omnidirectional camera	5	-
Author's proposal	RGB camera	Use of single image from RGB camera. No hand preset information used. Hand image must have resolution of at least 512 by 512 pixels	4.945	8.504
Author's proposal	RGB camera	Use of single image from RGB camera. No hand preset information used. Hand image must have resolution of at least 512 by 512 pixels. Subtraction of hand image background applied	4.757	8.202

The articles [9–17] were a source of inspiration, although a comparison of their results with those obtained in this article is not possible.

The results obtained from the neural network cannot be properly compared to those from previous research [9,11]. Research [9,11] focused on creation of a 3D hand model, measured by other metrics, and it should be noted that such research does not work in the case of a hand with abnormal parameters.

The results obtained from the neural network cannot be properly compared to those from previous research [10,13,14], as this research focused on hand gesture recognition. Gesture recognition is a classification task, and one gesture may represent different positions of the finger joints of the hand, and the task considered in this paper is a regression task for predicting finger joint positions. It is thus not possible to properly compare the classification task and the regression task.

The results obtained from the neural network cannot be properly compared to those from previous research [12]. The research [12] describes a sensor-based system, albeit one focused on describing the parameters of the physical principles of operation by developing a neural network to work with the sensor, although the research [12] does not provide results for measuring hand position accuracy. Thus, it is not possible to properly compare properly [12] and the system developed in this article.

The results obtained from the neural network cannot be properly compared to those from previous research [15–17].

The research [15] focuses on finding an object in the image, segmenting it, and analyzing its shape to plan further interactions, but does not determine the position of the hand in the familiar human form. The hand is described by the neural network in terms of special features regarding the human-interaction task, and so a comparison of results is not possible.

The research [16] focused on finger tapping analyze, but does not describe other hand parameters besides finger tapping. Thus, a comparison of results is not possible.

The research [17] focused on vein recognition analyze, but does not describe other hand parameters or position besides vein recognition. Thus, a comparison of results is not possible.

The best results were obtained after applying a model based on processing images with subtracted background of the hand. These images had different dynamic parameters, resolution and quality, and were subject to transformation by different filters. Specifically, this neural network had a mean absolute error value of 4.757 degrees, with a mean square error of 67.279, and analyzed a single image from the camera. The image had a resolution of 512 by 512 pixels, with the input image being processed by the neural network in 7–15 ms using a GPU Nvidia 2080ti.

This result was obtained via training using a changing dataset, and this training of the neural network was carried out at several stages with consecutive complication of data for training purposes as follows: Training on images with markup and invariable hand parameters in the image, training on images without markup with invariable hand parameters in the image, training on images without markup with dynamic hand parameters in the image, and training on images without markup with dynamic hand parameters in the image after subtracting the background of the image.

The neural network developed was trained on simulated data subjected to strong processing. The use of simulated data made it possible to collect a dataset of images of people's hands with various parameters and in various positions, which is an extremely difficult task in the real world. The application of additive data processing will allow testing with patients in the future since the neural network developed was designed for medical tasks for patients with multiple sclerosis. People may have hands with non-standard parameters and in non-standard positions, and a feature of the neural network is the ability to measure finger joint angle values for hands with non-standard parameters and positions.

5. Conclusions

This research focused on the development of a system for measuring finger joint angles based on camera image and was intended to work within the field of medicine to track the movement and limits of hand mobility in multiple sclerosis.

The convolution neural network for analysis of static RGB images from the camera for 12 finger joint angle measuring was developed accordingly, with the best results being obtained after applying a neural network model based on processing images with subtracted background of the hand.

The neural network developed allows non-contact measurement of finger joint angles to be undertaken using a single RGB camera without the use of additional sensors. The use of a single ordinary RGB camera in this research makes the system developed for finger angle measuring cheaper, and easier to install and operate compared to sensor-based systems and specialist cameras. A camera must support at least 30 frames/sec and the resolution 720p for the correct neural network work. A camera with better parameters will give better results.

Image analysis approaches based on different neural network architectures were considered to achieve a lightweight and fast neural network for real-time work.

The option of building a neural network based on convolutional pose machine architecture was considered. However, convolutional pose machine-based neural network architecture is a slow algorithm and not suitable for real-time operation on low computing power devices. It also does not work correctly for hands with unusual parameters. The convolutional pose machine-based neural network architecture was removed from the final version of the neural network to improve performance. However, the results of convolutional pose machine-based image analysis and markup were used to train a light MobileNetV2 based neural network.

The use of MobileNetV2 architecture, originally developed for image neural network processing on mobile devices, in turn makes it possible to run the system on systems with low computing power. The system developed in the article will thus make it possible to take high-precision and high-speed measurements of the finger position at low cost.

Accuracy of the neural network developed can, probably, be further improved by analyzing the camera image sequences and neural network hand joint angles measured. Current neural network analysis of each image is undertaken in the video stream independently and does not analyze time sequences. In future research, it is planned that this approach will be used to improve recognition accuracy of the corners of the finger joints.

Although the neural network can work without applying the background subtraction algorithm, the best results are nonetheless obtained after doing so. At this point, for the developed neural network to work, the camera must be stationary during measurement. This algorithm will work even if camera will be moved. However, model quality scientifically decreases because image will have artifacts and deformations. This model created for medical aims and must have good quality, but this model cannot give good quality if camera is not static.

In the future the plan is to replace the background subtraction algorithm used now with a neural network based on the MaskRCNN algorithm, which will be much less susceptible to loss of accuracy as a result of variations in brightness or bias. For now, the MaskRCNN algorithm is expected to be based on the MobileNetV2 architecture neural network developed in this study with minimal pre-training.

To obtain high performance and reduce the volume of the neural network, the optimization method based on the Monte Carlo method will be applied, and the unused weights in the network will be removed without loss of accuracy.

The quantization algorithm will also be applied to the neural network in the future, which should also significantly increase the speed of the neural network by transferring the weights of the neural network from the format float32 to the format float16.

Author Contributions: Funding acquisition, B.G.-Z. and A.M.Z.; Investigation, D.V., B.G.-Z. and A.M.Z.; Methodology, B.G.-Z.; Project administration, B.G.-Z. and A.M.Z.; Resources, B.G.-Z.; Software, D.V.; Supervision, A.M.Z.; Writing—original draft, D.V.; Writing—review & editing, B.G.-Z. and A.M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the eVida group (research group recognized by Basque Government).

Data Availability Statement: Raw data were generated in the EVIDA LAB, University of Deusto. Derived data supporting the findings of this study are available from the corresponding author D. Viatkin on request.

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

References

1. Cosh, A.; Carslaw, H. Multiple sclerosis: Symptoms and diagnosis. *InnovAiT* **2014**, *7*, 651–657. [CrossRef]
2. Chen, F.; Deng, J.; Pang, Z.; Nejad, M.B.; Yang, H.; Yang, G. Finger Angle-Based Hand Gesture Recognition for Smart Infrastructure Using Wearable Wrist-Worn Camera. *Appl. Sci.* **2018**, *8*, 369. [CrossRef]
3. Zhou, Y.; Jiang, G.; Lin, Y. A novel finger and hand pose estimation technique for real-time hand gesture recognition. *Pattern Recognit.* **2016**, *49*, 102–114. [CrossRef]
4. Kim, J.S.; Kim, B.K.; Jang, M.; Kang, K.; Kim, D.E.; Ju, B.-K.; Kim, J. Wearable Hand Module and Real-Time Tracking Algorithms for Measuring Finger Joint Angles of Different Hand Sizes with High Accuracy Using FBG Strain Sensor. *Sensors* **2020**, *20*, 1921. [CrossRef]
5. Lu, S.; Chen, D.; Liu, C.; Jiang, Y.; Wang, M. A 3-D finger motion measurement system via soft strain sensors for hand rehabilitation. *Sens. Actuators A Phys.* **2018**, *285*, 700–711. [CrossRef]
6. Tran, D.P.; Morita, D.; Sato, N.; Morita, Y.; Takekawa, M. Improvement of non-invasive semi-automatic test device for measurement of finger joints range of motion: Reduction in burden on therapist. In Proceedings of the 2016 16th International Conference on Control, Automation and Systems (ICCAS), Gyeongju, Korea, 16–19 October 2016; pp. 423–427.
7. Mayer, S.; Mayer, M.; Henze, N. Feasibility analysis of detecting the finger orientation with depth cameras. In Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services—MobileHCI '17, Vienna, Austria, 4–7 September 2017. [CrossRef]
8. Maruyama, Y.; Kono, Y. Estimating Finger Postures by Attaching an Omnidirectional Camera to the Center of a User's Palm. In Proceedings of the 2018 International Conference on Advanced Visual Interfaces, Riva del Sole, Castiglione della Pescaia, Grosseto, Italy, 29 May–1 June 2018.
9. Park, K.; Kim, S.; Yoon, Y.; Kim, T.-K.; Lee, G. DeepFisheye: Near-surface multi-finger tracking technology using fisheye camera. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, Minneapolis, MN, USA, 20–23 October 2020; Volume 3415818, pp. 1132–1146.
10. Kılıboz, N.Ç.; Güdükbay, U. A hand gesture recognition technique for human–computer interaction. *J. Vis. Commun. Image Represent.* **2015**, *28*, 97–104. [CrossRef]
11. Fan, Z.; Valentin, B.; Andrey, V.; Andrei, T.; George, S.; Chuo-Ling, C.; Matthias, G. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv* **2020**, arXiv:2006.10214.
12. Kim, K.K.; Ha, I.; Kim, M.; Choi, J.; Won, P.; Jo, S.; Ko, S.H. A deep-learned skin sensor decoding the epicentral human motions. *Nat. Commun.* **2020**, *11*, 2149. [CrossRef] [PubMed]
13. Ashiquzzaman, A.; Lee, H.; Kim, K.; Kim, H.Y.; Park, J.; Kim, J. Compact spatial pyramid pooling deep convolutional neural network based hand gestures decoder. *Appl. Sci.* **2020**, *10*, 7898. [CrossRef]
14. Kim, S.Y.; Han, H.G.; Kim, J.W.; Lee, S.; Kim, T.W. A Hand Gesture Recognition Sensor Using Reflected Impulses. *IEEE Sens. J.* **2017**, *17*, 2975–2976. [CrossRef]
15. Rosenberger, P.; Cosgun, A.; Newbury, R.; Kwan, J.; Ortenzi, V.; Corke, P.; Grafinger, M. Object-Independent Human-to-Robot Handovers Using Real Time Robotic Vision. *IEEE Robot. Autom. Lett.* **2021**, *6*, 17–23. [CrossRef]
16. Woo, S.-W.; Kang, M.-K.; Hong, K.-S. Classification of Finger Tapping Tasks using Convolutional Neural Network Based on Augmented Data with Deep Convolutional Generative Adversarial Network. In Proceedings of the 2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob), New York, NY, USA, 29 November–1 December 2020; Volume 9224386, pp. 328–333.
17. Chawla, B.; Tyagi, S.; Jain, R.; Talegaonkar, A.; Srivastava, S. Finger Vein Recognition Using Deep Learning. *Adv. Intell. Syst. Comput.* **2021**, *1164*, 69–78.
18. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
19. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

20. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Hartwig, A. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
21. Convolutional Pose Machines Tensorflow. 2020. Available online: https://github.com/timctho/convolutional-pose-machines-tensorflow (accessed on 18 February 2021).
22. Wei, S.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional Pose Machines. *arXiv* **2016**, arXiv:1602.00134.