2023 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Tomar, Portugal – April 26-27, 2023

Deep Convolutional Neural Networks applied to Hand Keypoints Estimation

Bruno M. Santos
SYSTEC (DIGI2), ARISE & ECE Dept.
Fac. de Engenharia, Univ. do Porto
Rua Dr. Roberto Frias,
4200-465 Porto, Portugal
up201806842@fe.up.pt

Pedro Pais
ECE Dept.
Fac. de Engenharia, Univ. do Porto
Rua Dr. Roberto Frias,
4200-465 Porto, Portugal
up201806644@fe.up.pt

Francisco M. Ribeiro
SYSTEC (DIGI2), ARISE & ECE Dept.
Fac. de Engenharia, Univ. do Porto
Rua Dr. Roberto Frias,
4200-465 Porto, Portugal
fmribeiro@fe.up.pt

José Lima
CeDRI
Instituto Politécnico de Braganca
Campus de Sta Apolónia, Portugal
jllima@ipb.pt

Gil Gonçalves
SYSTEC (DIGI2), ARISE & ECE Dept.
Fac. de Engenharia, Univ. do Porto
Rua Dr. Roberto Frias,
4200-465 Porto, Portugal
gil@fe.up.pt

Vítor H. Pinto
SYSTEC (DIGI2), ARISE & ECE Dept.
Fac. de Engenharia, Univ. do Porto
Rua Dr. Roberto Frias,
4200-465 Porto, Portugal
vitorpinto@fe.up.pt

***Abstract*—Accurate estimation of hand shape and position is an important task in various applications, such as human-computer interaction, human-robot interaction, and virtual and augmented reality. In this paper, it is proposed a method to estimate the hand keypoints from single and colored images utilizing the pre-trained deep convolutional neural networks VGG-16 and VGG-19. The method is evaluated on the FreiHAND dataset, and the performance of the two neural networks is compared. The best results were achieved by the VGG-19, with average estimation errors of 7.40 pixels and 11.36 millimeters for the best cases of two-dimensional and three-dimensional hand keypoints estimation, respectively.**

***Index Terms*—hand keypoints estimation, convolutional neural network, VGG, FreiHAND**

I. Introduction

In recent years, there has been a significant increase in the demand for human-computer interaction technology. One of the key aspects of this interaction is hand gesture recognition, which has numerous applications in areas such as sign language interpreting, human-robot interaction, human-car interaction, and others [1]. Hand keypoint estimation, which involves the detection of the keypoints of the hand, is an essential step in hand gesture recognition. In order to detect the keypoints, computer vision techniques are frequently applied; however, this is still a challenge due to problems such as hands' morphology, which requires a high number of keypoints to be reconstructed, changes in the environmental background, light conditions, occlusions, and other factors intrinsic to vision-based analysis methods.

In this work, it is proposed a method to estimate two-dimensional (2-D) and three-dimensional (3-D) hand keypoints positions from single Red-Green-Blue (RGB) images utilizing deep learning techniques. Specifically, slightly adapted Visual Geometry Group (VGG) networks [2], namely VGG-16 and VGG-19, two well-known convolutional neural networks (CNN) architectures, were used and compared. The proposed networks were trained and evaluated on the FreiHAND dataset [3], a large and diverse dataset with annotations of the hand keypoints positions.

In order to not start the training phase from scratch, pre-trained weights of VGG-16 and VGG-19 on the ImageNet dataset [4] were loaded, being the networks subsequently fine-tuned on the FreiHAND dataset to perform hand keypoint estimation.

The paper is structured in the following way: in Section II the related work is reviewed; in Sections III and IV the utilized networks and dataset, as well as the applied method are described; in Section V the evaluation process is explained; in Section VI are presented the experimental results and finally, in Section VII the conclusions of the work are given.

II. Related Work

The research field of hand keypoint detection/estimation has been highly explored on recent years. The use of algorithms for automatic detection of interest points of the human hand is of extreme relevance in several use cases. Some examples are sign language recognition, gesture interpretation for controlling systems, as well as human movement capture on extremely complex tasks. Thus, there are several scientific articles that address this issue.

Gatupalli *et al.* [5] conducted a state-of-the-art study on the use of deep learning techniques applied to the detection of keypoints in fast sequential motion, using RGB images. They compared the two methods considered most widely used

in the literature, namely those proposed by Zimmermann *et al.* [6] and Simon *et al.* [7]. The accuracy of both methods, considering a threshold of 11 pixels, stays under 60%.

Santavas *et al.* [8] have presented a novel and lightweight convolutional neural network architecture for estimating 2D hand pose from single RGB images. The results showed a decrease of average pixel distance error up to 72,90% when compared to Zimmermann *et al.* [6] and Bouk. *et al.* [9] methods. Noteworthy is the 4 pixels error obtained in the FreiHAND dataset.

Chen *et al.* [10] have proposed an approach to recover hand mesh from single RGB images based on the known semantic relations among hand joints. Applying this novel approach, the authors achieve better results in comparison to other methods, such as [9] [3] and [11]. Namely in the FreiHAND dataset where a PA-MPJPE[1] of 6.9 millimeters was obtained.

Amaliya *et al.* [12] have compared three widely used real-time systems that detect human hand articulations to recognize sign language - OpenPose [13], TensorFlowLite [14] and MediaPipe Hands [15]. Although quantitative metrics are not used to evaluate each tool, the authors evaluate multiple gestures captured with one camera (Kinect 2.0). Although MediaPipe has a difficulty dealing with the collected images, it is able to be more accurate and faster than the other platforms for hand identification. As for OpenPose and TensorFlowLite, they are very equivalent, the latter being slightly faster in processing.

III. Background

In this section, it will be introduced the convolutional neural networks to be utilized in the work, as well as the dataset for training and evaluation procedures.

A. VGG-16

The VGG-16 is a deep CNN that was introduced by Karen Simonyan and Andrew Zisserman in 2014 [2]. The network is a well-established deep CNN that has been widely used for various computer vision tasks. Its ability to effectively capture fine-grained details in images, combined with its relatively small size, makes it a popular choice for a wide range of applications. Furthermore, the utilization of a known network model, such as VGG-16, has the advantage of permitting the start of the training phase with pre-trained parameters, which usually accelerates the process and increases the final accuracy.

B. VGG-16 Architecture

The VGG-16 architecture consists of a set of layers, including 13 convolution layers, 5 max-pooling layers, and 3 fully connected layers. The "16" in the name of the network stands for the number of convolution and fully connected layers. Convolution layers are the core layers of a CNN. In these layers, a kernel is applied to the input image to produce feature maps that capture patterns or other relevant features from an image. Max pooling layers are responsible for downsampling the input images while preserving the most important information. The fully connected layers are located at the end of the network and utilize the information from the feature maps to perform the classification task. In the original version of VGG-16, the output of the last fully connected layer consists of 1000 nodes, each one representing a class.

[1]explained in Section V-A

C. VGG-19

The concept of the VGG-19 network is the same as the VGG-16. The only difference is the number of convolution layers, as the name implies. Considering this, all that has been described in the previous sections regarding the VGG-16 network applies to VGG-19.

D. FreiHAND Dataset

The FreiHAND is a dataset for training and evaluate deep neural networks for the task of hand pose and shape estimation from RGB images and it was created by Christian Zimmermann *et al.* [3]. The creation of this dataset was motivated by the poor generalization verified in the previous existing datasets, i.e., the models trained with a specific dataset offered poor results when applied to a different one.

The dataset is composed of a set of 32560 unique training samples in addition to 3960 evaluation samples. The training images were acquired from a total of 24 subjects, thus ensuring good data diversification and variance. The referred 32560 samples were captured in green screen, this way allowing for data augmentation by the change of the background scene. Applying this method, the final size of the training dataset quadrupled, resulting in a total of about 132k samples. The resolution of the provided images is 224x224 pixels, which fits perfectly with the input of VGG networks. It should be noted that all samples are annotated with 3-D keypoints positions and hand shapes. In this paper, only the keypoints positions were considered.

IV. Method

This section describes the utilized methods to estimate the 2-D and 3-D hand keypoints, having as input a single RGB image. To achieve this, an adapted version of a VGG-16 and a VGG-19 were considered and then compared in terms of performance.

A. Networks Adaptation

Firstly, in order to adapt the mentioned neural networks to the specific application case, adjustments were done in the last set of three fully connected layers. The activation function applied in these layers was Rectified Linear Activation Function (ReLU). The input and output sizes of these layers were adjusted, being the number of nodes gradually decreasing until the desired size of output nodes. For the 2-D case, the output sizes were defined as 2048, 256, and lastly 42, which corresponds to the "x" and "y" positions for each one of the 21 hand keypoints. In the other case, the output sizes were defined as 2048, 512, and lastly 63 since now each keypoint has a new depth coordinate.

B. Dataset Split

The FreiHAND dataset was divided into different ways in order to compare the obtained results.

Firstly, were only considered the green screen images from the dataset, thus allowing to test the performance of the VGG

networks in images where the hand is easily distinguishable from the background. This way, 80% of samples of this subset were utilized for the training procedure, 10% samples for validation, and the remaining 10% samples for evaluation.

Secondly, the whole dataset was considered. This time, 90% of the training images were utilized for the training phase, the rest 10% for the validation process, and the evaluation samples were utilized for evaluation. Given the fact that the training samples are repeated every 32560 images, to avoid validating the model with positions that the network has already trained with, the 10% of validation samples were chosen as the last 10% samples of each group of 32560 images.

Lastly, an experience was done without the validation step, i.e., utilizing all training samples for training and all the evaluation samples for evaluation.

C. Training

For the training process, an Adam optimizer with a learning rate of 1e-3 was applied. Adam is a simple and computationally efficient algorithm for gradient-based optimization, furthermore, it has shown robustness in a wide range of machine-learning optimization problems [16].

As mentioned at the beginning of this section, the parameters' weights were not initialized randomly, i.e., the networks were not trained from scratch. At the beginning of the training phase, the weights of both networks were loaded from "IMAGENET1K_V1" a pre-trained model that contains weights and parameters learned from training on a large-scale image recognition task called ImageNet [4].

At the end of each epoch, the generated model was validated in the respective validation dataset and an error, explained in Section V-A, was calculated and saved. In case of a smaller error of the current iteration relative to the previous one, the model was saved, otherwise, the model was disregarded. The application of this technique prevents the model from overfitting the training data.

The total number of epochs was empirically defined as 15, except for the case where the validation step was not performed, in this specific case, the number of epochs was defined as 10 since more epochs would eventually lead to overfitting of the model to the training data.

The network training was done in a computer with the following specifications:

- Central Processing Unit (CPU): 11th Gen Intel® Core™ i9-11900KF @ 3.50GHz
- Random Access Memory (RAM): 64.0 GB
- Graphics Processing Unit (GPU): 2 x NVIDIA GeForce GTX 2060 6GB

V. Evaluation

The performance of both VGG-16 and VGG-19 for the multi-dimensional hand keypoints detection was evaluated in the FreiHAND dataset, a large and very generalized dataset.

A. Evaluation Metrics

For the 2-D hand keypoints detection, the evaluation metric utilized was the average euclidean distance in pixels from the ground truth keypoints to the predicted ones. Additionally, a graphic was created in order to give some visual feedback on the results. This graph consisted of the 2-D distribution of the estimated keypoints relative to the ground truth. From this graphic, the mean value and the standard deviation in each coordinate were extracted.

For the 3-D hand keypoints case, the evaluation metric utilized was the average euclidean distance in millimeters (mm) from the ground truth keypoint to the predicted one, as known as Mean Per Joint Position Error (MPJPE) in literature. As in the 2-D case, a graphic was also created giving the relative position of predicted keypoints, but this time in 3-D. Additionally, it also included a euclidean distance considering relative depth for each hand, instead of absolute depth. Finally, the last metric utilized was also a euclidean distance, but this time applying the Generalized Procrustes Analysis (GPA) [17] technique. This metric is known as Procrustes-Aligned MPJPE (PA-MPJPE). The GPA method standardizes both real and predicted keypoints by performing isomorphic translation, rotation, and scaling to achieve the best fit between two hand shapes. GPA is a rigid shape analysis, i.e., this technique does not change the intrinsic shape of the keypoints.

VI. Results

In this section, the results from the work will be presented and then discussed.

A. 2-D Hand Keypoints Estimation

Applying the adapted version of the VGG-16 to the FreiHAND dataset, and considering the 3 different dataset splits, the results shown in Figure 1 and in Table I were obtained. In comparison, the results obtained utilizing the VGG-19 network are presented in Figure 2 and in Table II.

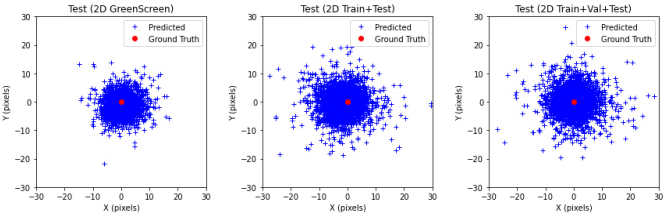

Fig. 1. 2-D VGG-16 Graphical Results

TABLE I
2-D VGG-16 Quantitative Results

	GreenScreen	Train+Val+Test	Train+Test
Average Error (pixels)	8.25	9.63	9.42
Mean (x,y) (pixels)	(0.25, -1.05)	(-0.08, 0.17)	(-1.46, -0.15)
Std (x,y) (pixels)	(3.27, 3.05)	(4.48, 3.98)	(4.44, 3.87)

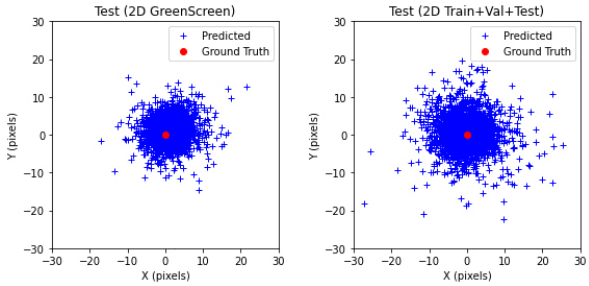

Fig. 2. 2-D VGG-19 Graphical Results

TABLE II
2-D VGG-19 QUANTITATIVE RESULTS

	GreenScreen	Train+Val+Test
Average Error (pixels)	7.40	8.36
Mean (x,y) (pixels)	(1.01, 1.18)	(-0.32, 0.33)
Std (x,y) (pixels)	(3.22, 2.99)	(4.17, 3.72)

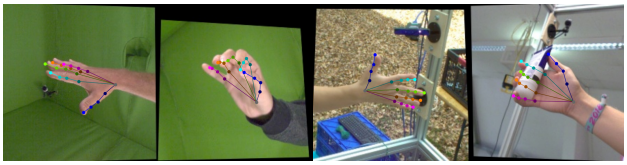

Fig. 3. 2-D VGG-19 Visual Results

B. 3-D Hand Keypoints Estimation

For the 3-D keypoints estimation, the results presented in Figure 4 and in Table III were achieved from the utilization of the VGG-16 network. The consideration of relative depth for each hand instead of absolute depth, leads to the results in Figure 5 and in Table IV. Lastly, the results shown in Figure 6 and in Table V were achieved considering the application of the GPA method.

In order to compare the performance of VGG-19 relative to the VGG-16, the methods that proven superior, namely the application of PCA to the green screen images dataset and the dataset divided into training, validation, and testing data, were applied to the VGG-19. The results are presented in Figure 7 and in Table VI.

Just note that all graphics presented have the same boundaries in all coordinates to simplify the comparison of the results.

C. Discussion of Results

First, the influence of the dataset split, as explained in Subsection IV-B will be analysed.

The idea behind splitting the dataset into only two sets of data (training and evaluation) was to train the network with more images that in this case correspond to "teaching" the network a larger number of hand positions to get better

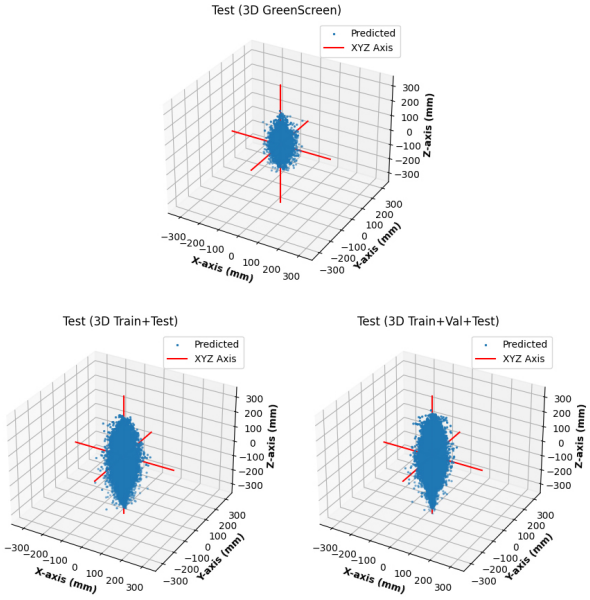

Fig. 4. 3-D VGG-16 Graphical Results

TABLE III
3-D VGG-16 QUANTITATIVE RESULTS

	GreenScreen	
MPJPE	33.73	
Mean (x,y,z) (mm)	(3.82, 3.23, 13.3)	
Std (x,y,z) (mm)	(10.38, 10.48, 34.75)	
	Train+Val+Tes	**Train+Test**
MPJPE	74.27	79.71
Mean (x,y,z) (mm)	(0.71, -0.53, -15.23)	(-1.40, -1.05, -36.93)
Std (x,y,z) (mm)	(14.12, 13.56, 89.2)	(14.82, 14.29, 88.32)

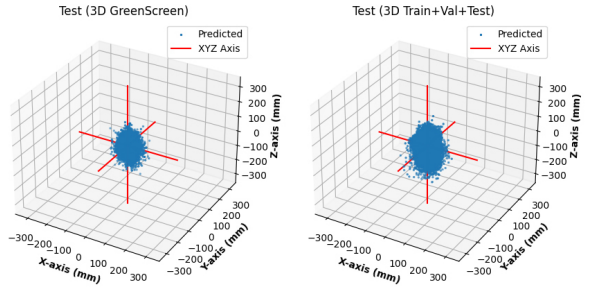

Fig. 5. 3-D VGG-16 Graphical Results (relative depth)

TABLE IV
3-D VGG-16 QUANTITATIVE RESULTS (RELATIVE DEPTH)

	GreenScreen	Train+Val+Tes
MPJPE (relative depth)	24.98	30.75
Mean (x,y,z) (mm)	(3.82, 3.23, -6.59)	(0.71, -0.53, -0.37)
Std (x,y,z) (mm)	(10.38, 10.48, 25.11)	(14.12, 13.56, 33.83)

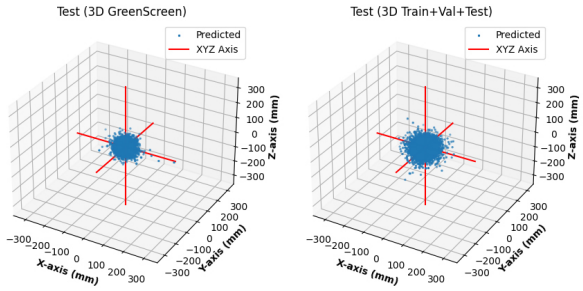

Fig. 6. 3-D VGG-16 Graphical Results (GPA)

TABLE V
3-D VGG-16 QUANTITATIVE RESULTS (GPA)

	GreenScreen	Train+Val+Tes
PA-MPJPE	12.45	13.57
Mean (x,y,z) (mm)	(0.00, 0.00, 0.00)	(0.00, 0.00, 0.00)
Std (x,y,z) (mm)	(8.81, 9.11, 11.17)	(11.12, 11.63, 12.45)

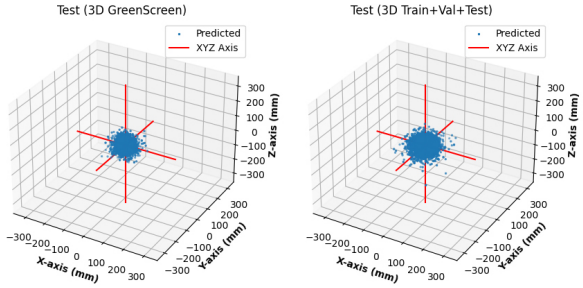

Fig. 7. 3-D VGG-19 Graphical Results (GPA)

TABLE VI
3-D VGG-19 QUANTITATIVE RESULTS (GPA)

	GreenScreen	Train+Val+Tes
PA-MPJPE	11.36	12.8
Mean (x,y,z) (mm)	(0.00, 0.00, 0.00)	(0.00, 0.00, 0.00)
Std (x,y,z) (mm)	(8.19, 8.03, 10.38)	(10.2, 10.59, 11.91)

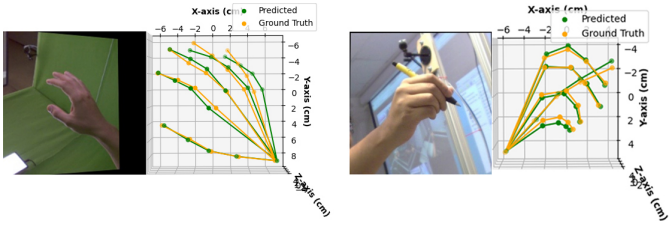

Fig. 8. 3-D VGG-19 Visual Results (GPA)

results from the network. However, despite the larger number of training images, the results, both in 2-D (Figure 1 and Table I) and 3-D (Figure 4 and Table III), didn't show a significant improvement in the network performance and even aggravated the performance for the three-dimensional case. This is believed to have happened because, during the training process the validation step was skipped which may have resulted in overfitting. Therefore, this way of dataset split was ignored for the next tests.

Concerning the green screen images dataset, despite having less than 25% of training images compared to the whole dataset, the results presented in Subsection VI lead to believe that the network actually performs better in this dataset. From this result, it is possible to conclude that it is extremely important for the network to distinguish the hand from the background and a possible way to improve the results would be to use an additional CNN that gives the hand shape as output.

Now, comparing the performance of the VGG-16 and the VGG-19 in the two-dimensional keypoints estimation, it is possible to infer that, as expected due to the extra 3 convolution layers, the VGG-19 presents better results. Quantitatively, the utilization of VGG-19 resulted in a reduction of the average error of 10,30% (0.85 pixels) for the green screen dataset and 13.19% (1.27 pixels) for the whole dataset.

Considering the three-dimensional hand keypoints estimation, the MPJPE metric showed high values (Table III). Analyzing the respective 3-D graphic (Figure 4) it is possible to conclude that the problem resides in the estimation of depth, which was already expected considering that the input to the network consisted of single RGB images. As an attempt to reduce these values, an approach was followed considering not absolute depth, but relative depth, i.e., the wrist keypoint was considered as the depth referential origin and all other keypoints were adjusted taking into account this new referential. The results of this approach can be seen in Figure 5 and in Table IV. It is visually perceptible the decrease of variance in the z coordinate, which is confirmed by the quantitative results. The MPJPE decreased 25.94% (8.75 mm) and 58.60% (43.52 mm) for the green screen and the whole dataset, respectively. As an alternative approach, the scale factor given by the dataset for each image was utilized to resize all the images, this way, the size of the hands would only depend on the distance to the camera and thus a better estimation of the depth value of each image was predictable. However, this approach led to a slight increase in the calculated errors and was therefore disregarded. Lastly, it was also calculated a well-known metric in the field of keypoints estimation, the PA-MPJPE. Here the results were notably better compared to the other cases, achieving an error of 12.45 mm and 13.57 mm for the green screen dataset and the whole dataset, respectively. Additionally, analysing the Figure 6 it is possible to observe that the distribution of the predicted keypoints is nearly uniform for all the coordinates, forming a shape close to a sphere.

Once again comparing the performance of VGG-16 and VGG-19, particularly for this last case, improvements can also

be seen, although inferior when compared to 2-D keypoint estimation. Quantitatively, there was an improvement of 8.76% (1.09 mm) relative to the green screen dataset and an improvement of 5.67% (0.77 mm) relative to the whole dataset.

Despite the progress of the results, they do not reach the results obtained by the state-of-the-art methods described in Section II, namely the 4 pixels error achieved by Santavas *et al.* and the PA-MPJPE of 6.9mm achieved by Chen *et al.* for 2-D and 3-D hand pose estimation, respectively.

VII. Conclusions

In this work, the performance of an adjusted pre-trained deep neural network was evaluated in the task of estimating the position of hand keypoints. The networks tested were VGG-16 and VGG-19 and the utilized dataset was FreiHAND.

As expected, the overall performance of the VGG-19 surpassed the VGG-16 by an average of 9.48%, achieving errors of 7.40 pixels and 11.36 millimeters for two-dimensional and three-dimensional hand keypoints estimation, respectively, on the green screen subset and errors of 8.36 pixels and 12.8 millimeters on the whole dataset.

The results achieved show that the utilization of a pre-trained network adapted accordingly to the desired finality is an option to consider, at least in a first approach to the problem. This way, instead of starting the training phase from scratch, the weights are already initialized, which decreases the number of epochs and consequently training time, furthermore it also helps the model to converge faster.

Additionally, the intuitive way in which results are presented throughout this article are valuable to better understand the correct or incorrect network operation and allows for easy interpretation and inference regarding adjustments to be made.

Acknowledgment

The authors acknowledge the support of R&D Unit SYSTEC Base (UIDB/00147/2020) and Programmatic (UIDP/00147/2020) and the ARISE Associated Laboratory (LA/P/0112/2020), as well as the support of projects: Continental FoF, with reference POCI-01-0247-FEDER-047512, co-funded by FEDER, through COMPETE 2020, *Digitalização da Arte Humana (Cibertoque)*, with reference POCI-01-0247-FEDER-072627, co-funded by FEDER, through COMPETE 2020 and Next-Gen Quality Control IoRT System with reference POCI-01-0247-FEDER-072616, co-funded by FEDER, through COMPETE 2020.

References

[1] M. Yasen and S. Jusoh, "A systematic review on hand gesture recognition techniques, challenges and applications," *PeerJ Computer Science*, vol. 2019, no. 9, 2019, all Open Access, Gold Open Access, Green Open Access. [Online]. Available: https://doi.org/10.7717/peerj-cs.218

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, Conference paper. [Online]. Available: https://doi.org/10.48550/arXiv.1409.1556

[3] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. J. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," vol. 2019-October, 2019, Conference paper, pp. 813 – 822, all Open Access, Green Open Access. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00090

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. [Online]. Available: https://doi.org/10.1109/CVPR.2009.5206848

[5] S. Gattupalli, A. R. Babu, J. R. Brady, F. Makedon, and V. Athitsos, "Towards deep learning based hand keypoints detection for rapid sequential movements from rgb images," in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, ser. PETRA '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 31–37. [Online]. Available: https://doi.org/10.1145/3197768.3201538

[6] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," 2017. [Online]. Available: https://arxiv.org/abs/1705.01389

[7] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," 2017. [Online]. Available: https://arxiv.org/abs/1704.07809

[8] N. Santavas, I. Kansizoglou, L. Bampis, E. Karakasis, and A. Gasteratos, "Attention! a lightweight 2d hand pose estimation approach," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11488 – 11496, 2021, all Open Access, Green Open Access. [Online]. Available: https://doi.org/10.1109/JSEN.2020.3018172

[9] A. Boukhayma, R. De Bem, and P. H. Torr, "3d hand shape and pose from images in the wild," vol. 2019-June, 2019, Conference paper, pp. 10835 – 10844, all Open Access, Green Open Access. [Online]. Available: https://doi.org/10.1109/CVPR.2019.01110

[10] X. Chen, Y. Liu, C. Ma, J. Chang, H. Wang, T. Chen, X. Guo, P. Wan, and W. Zheng, "Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration," 2021, Conference paper, pp. 13269 – 13278, all Open Access, Green Open Access. [Online]. Available: https://doi.org/10.1109/CVPR46437.2021.01307

[11] G. Moon and K. M. Lee, "I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12352 LNCS, pp. 752 – 768, 2020, all Open Access, Green Open Access. [Online]. Available: https://doi.org/10.1007/978-3-030-58571-6_44

[12] S. Amaliya, A. N. Handayani, M. I. Akbar, H. W. Herwanto, O. Fukuda, and W. C. Kurniawan, "Study on hand keypoint framework for sign language recognition," in *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, 2021, pp. 446–451.

[13] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[14] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow, Large-scale machine learning on heterogeneous systems," 11 2015.

[15] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[16] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015, Conference paper. [Online]. Available: https://doi.org/10.48550/arXiv.1412.6980

[17] J. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33 – 51, 1975. [Online]. Available: https://doi.org/10.1007/BF02291478