engineering proceedings

MDPI

Proceeding Paper

Improving Hand Pose Recognition Using Localization and Zoom Normalizations over MediaPipe Landmarks [†]

Miguel Ángel Remiro, Manuel Gil-Martín * and Rubén San-Segundo

Speech Technology and Machine Learning Group (T.H.A.U. Group), Information Processing and Telecommunications Center, E.T.S.I. de Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain; email1@email.com (M.Á.R.); email2@email.com (R.S.-S.)

* Correspondence: manuel.gilmartin@upm.es; Tel.: +34-91-067-2500

[†] Presented at the 10th International Electronic Conference on Sensors and Applications (ECSA-10), 15–30 November 2023; Available online: https://ecsa-10.sciforum.net/.

Abstract: Hand Pose Recognition presents significant challenges that need to be addressed, such as varying lighting conditions or complex backgrounds, which can hinder accurate and robust hand pose estimation. This can be mitigated by employing MediaPipe to facilitate the efficient extraction of representative landmarks from static images combined with the use of Convolutional Neural Networks. Extracting these landmarks from the hands mitigates the impact of lighting variability or the presence of complex backgrounds. However, the variability of the location and size of the hands is still not addressed by this process. Therefore, the use of processing modules to normalize these points regarding the location of the wrist and the zoom of the hands can significantly mitigate the effects of these variabilities. In all the experiments performed in this work based on American Sign Language alphabet datasets of 870, 27,000, and 87,000 images, the application of the proposed normalizations has resulted in significant improvements in the model performance in a resource-limited scenario. Particularly, under conditions of high variability applying both normalizations resulted in a performance increment of 45.08%, increasing the accuracy from 43.94 ± 0.64% to 89.02 ± 0.40%.

Keywords: deep learning; computer vision; human activity recognition; hand pose recognition; landmarks; location normalization; zoom normalization

Citation: Remiro, M.Á.; Gil-Martín, M.; San-Segundo, R. Improving Hand Pose Recognition Using Localization and Zoom Normalizations over MediaPipe Landmarks. *Eng. Proc.* **2023**, *56*, x. https://doi.org/10.3390/xxxxx

Published: 15 November 2023

1. Introduction

Recent advances in deep learning and computer vision have been driving the development of Human Activity Recognition (HAR) 12, which consists of classifying the physical activities that people perform. One of the HAR research fields is Hand Pose Recognition, which has numerous applications and a great impact on individuals who are deaf or have limited speech and communicate using Sign Language.

In this context, many of the latest works focused on the use of MediaPipe to extract representative landmarks from hands combined with the use of neural networks. Using this approach, a previous work 3 obtained an accuracy of nearly 88% for the recognition of signs of the American Sign Language (ASL) alphabet using 87,000 images.

Even using MediaPipe, there are still aspects such as the variability of the location of the hand or its size that can negatively impact the performance of the model that previous works have not been focused on. This variability can significantly hinder the accurate recognition of hand poses, particularly when employing deep learning algorithms because they heavily rely on data for training. Most datasets contain standardized hand positions and sizes images so the ones with diverse locations and sizes could be misclassified and may hinder the modelts to generalize.

This paper aims to study the impact of hand location and zoom variability to propose efficient normalization techniques to mitigate these effects.

2. Materials and Methods

This section describes the datasets used in the experiments, the signal processing, the deep learning approach, and the evaluation methodology to assess the performance of the model.

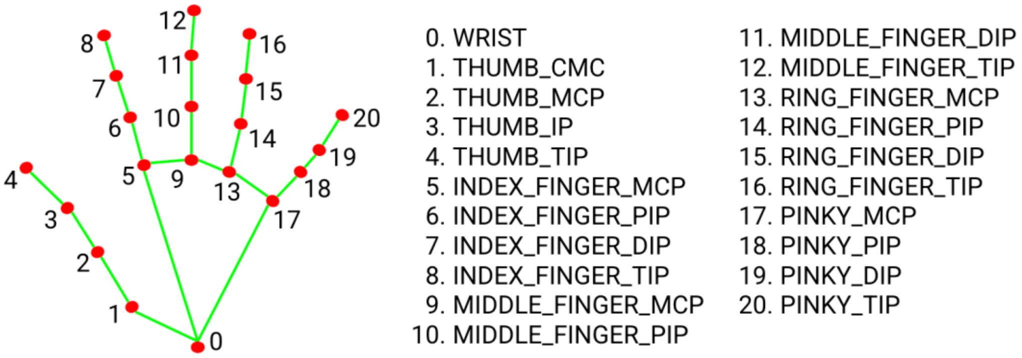
2.1. Datasets

In this work, we used three ASL Alphabet datasets. ASL Alphabet Test dataset 4 or so-called Dataset 1 has 870 images, 29 classes, 30 images per class and it is variable in terms of hand location and zoom. Synthetic ASL Alphabet 5 or so-called Dataset 2 has 27,000 images, 27 classes, 1000 images per class and it is static in terms of hand location and zoom. ASL Alphabet 6 or so-called Dataset 3 has 87,000 images, 29 classes, 3000 images per class and it is variable in terms of hand location and zoom.

2.2. Signal Processing

2.2.1. MediaPipe Hands

In this work, we used MediaPipe Hands 78, a specific module within the MediaPipe open source project capable of empowering real-time hand detection and tracking in images and videos, providing essential information regarding the precise position of 21 landmarks or key points on each hand. Each landmark is composed of the x and y coordinates and is related to a specific point in the hands, as shown in Figure 1.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 1. The specific location of the hand landmarks extracted by MediaPipe Hands [8].

Once these points are extracted, they serve as input to a neural network, enabling the model to discern patterns and effectively differentiate between various signs.

2.2.2. Modules to Include Location and Zoom Variability

To analyse the effect of normalizations under more extreme conditions of variability, two modules have been designed to include location and zoom variability. Their application on a dataset will generate another artificial dataset with the same number of images with a wider heterogeneity in terms of location or zoom.

To include location variability, the first module adds or subtracts equiprobably the same random value to the coordinates of the landmarks of each image. In this way, a new dataset is generated with the landmarks relocated at new random locations.

Similarly, the second module generates an artificial dataset by multiplying or dividing by the same coefficient all the coordinates of the landmarks. Thus, the size of the hands is randomly modified and the variability of the zoom is substantially increased.

Under these conditions of higher variability, the performance of the system may decrease but the potential of normalization algorithms can be tested.

2.2.3. Normalization Algorithms

To mitigate the location and zoom variability of the dataset, several algorithms have been developed. Norm_Loc algorithm used the landmark of the wrist (landmark 0) as the origin of coordinates and normalised all other points concerning it. In the Norm_Zoom algorithm, the maximum coordinate value is moved to the edge of the square of vertices (0,0), (0,1), (1,0), and (1,1), transforming the rest of the points proportionally so as not to lose the aspect ratio of the hand. The correct functioning of this normalization makes either the largest X-axis coordinate or the largest Y-axis coordinate become worth 1, while the rest of the points are multiplied by the same coefficient. The last algorithm is called Norm_Loc_Zoom and applies the two previous modules sequentially. These algorithms mitigate the location and zoom variability because they standardize these image characteristics considering the wrist location and the hand size. As these algorithms consist on simple mathematical operations, there is no increase in the overall computational cost.

2.3. Deep Learning

A deep learning structure with a feature learning subnet composed of a convolutional layer and a classification subnet composed of fully connected layers is used to recognize the different pose hands. This architecture is represented in Figure 2.

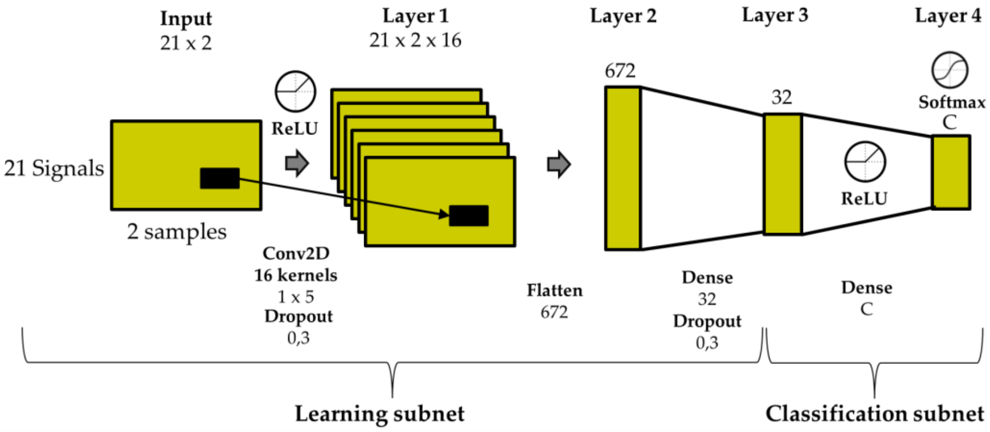


Figure 2. Deep learning architecture used in this work to classify the hand poses.

First, a two-dimensional convolutional layer is added following the input layer. This layer uses the Conv2D function to apply 16 filters to the input of the network, performing convolution operations to facilitate feature extraction. This layers learn feature engineering by itself via filters optimization. The ReLU activation function is applied to the output of this layer to introduce non-linearity and enable the neural network to learn more complex representations of the data. Next, the Dropout layer is added to regularize the network and prevent overfitting. Specifically, a Dropout rate of 0.3 is applied, meaning that 30% of the outputs from the previous layer are randomly deactivated during training. This helps to prevent the network from becoming too dependent on specific neurons and thus avoids overfitting. The Flatten layer converts the output of the previous layer into a one-dimensional vector. The data is then processed by a Dense layer. In this case, this layer consists of 32 neurons directly connected to all neurons from the previous layer. Thus, the neural network performs a linear and non-linear transformation of the input data, allowing the network to learn more complex relationships between the features extracted by the previous layers. Another Dropout layer with the same rate of 0.3 is applied afterward. The ReLU activation function is also employed.

Finally, the output layer consists of a dense layer with a number of neurons corresponding to the number of classes. The softmax activation function is used to calculate the probability of belonging to each of the possible classes. The output layer produces the final outputs of the model, representing the probability distribution over the different classes.

2.4. Evaluation Methodology

In this work, k-fold cross-validation is used to assess model performance more accurately and robustly. The data set is divided into k subsets (or folds) and the system is trained on k-1 sets, while the remaining set is tested. This process is repeated as many times as there are folds, obtaining a result for each one. In this way, a weighted average of the test results achieved can be calculated, obtaining a much more accurate and robust evaluation of the system, reaching to test all the available data.

Regarding the evaluation metrics, the model performance is measured with accuracy, which is the most common metric in classification problems. It calculates the proportion of correctly classified examples out of the total number of examples. As seen in Equation (1), it is obtained by dividing the sum of true positives and true negatives by the total number of instances. *This way, an increment of accuracy implies that the overall system better recognize the classes.* In this work, we used a confidence interval with a 95% significance level attached to the accuracy values.

$$Accuracy\ (\%) = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} \quad (1)$$

3. Results

This section provides the results for the different datasets, using the original ones and the ones obtained after including the artificial hand and zoom variability. This way, the system was tested with four variants for each dataset: the original dataset (when no normalization algorithm was applied to the data) and the three versions with artificial variability in location, zoom, or both.

In this work, we have been focused on improving the performance in a resource-limited scenario, so we used 10 epochs and batch size value for each dataset according to this aspect and the number of frames in each dataset.

3.1. Results for Dataset 1

Table 1 below shows the accuracy rates obtained with Dataset 1 using a batch size of 15. Under these conditions, the system is not able to learn enough from the data in the given number of epochs, giving very low accuracy rates when no normalization is applied. When applied, significant improvements in rates are observed, raising the accuracy from 46.18 ± 3.49 to 88.17 ± 2.26 in conditions of high localization and zoom variability when applying the Norm_Zoom normalization, which offers better results than the application of both normalizations (Norm_Loc_Zoom).

Table 1. Results for Dataset 1.

Dataset 1	Normalization Used	Accuracy (%)	Upgrade (%)
Original *	None	43.77 ± 3.47	
	Norm_Loc	67.94 ± 3.26	24.17
	Norm_Zoom	**78.63 ± 2.87**	34.86
	Norm_Loc_Zoom	**79.64 ± 2.81**	35.87
Artificial location	None	34.22 ± 3.32	
	Norm_Loc	**61.45 ± 3.40**	27.23
Artificial zoom	None	33.84 ± 3.31	
	Norm_Zoom	**76.84 ± 2.95**	43
Artificial location and zoom	None	46.18 ± 3.49	
	Norm_Loc	55.09 ± 3.48	8.91
	Norm_Zoom	**88.17 ± 2.26**	41.99
	Norm_Loc_Zoom	77.35 ± 2.93	31.17

* The modules that include artificial hand location and zoom variability have been not applied.

3.2. Results for Dataset 2

For this dataset, we used a batch size of 1000. As can be seen in Table 2, the application of the different proposed normalizations supposes significant improvements in the different variability conditions.

The improvement of the accuracy was 45.08% in conditions of high variability in location and zoom when both normalizations are applied. Moreover, the model achieves higher accuracy rates with zoom normalization than with location normalization.

With the original dataset, the model achieves an accuracy of 87.61 ± 0.42 with Norm_Zoom against 83.61 ± 0.48 achieved with Norm_Loc. This difference becomes even wider under conditions of high variability of location and zoom: 87.96 ± 0.42 in contrast to 74.42 ± 0.56.

Table 2. Results for Dataset 2**.**

Dataset 2	Normalization Used	Accuracy (%)	Upgrade (%)
Original *	None	79.95 ± 0.51	
	Norm_Loc	83.61 ± 0.48	3.66
	Norm_Zoom	87.61 ± 0.42	7.66
	Norm_Loc_Zoom	**94.64 ± 0.29**	14.69
Artificial location	None	48.15 ± 0.64	
	Norm_Loc	**87.22 ± 0.43**	39.07
Artificial zoom	None	67.53 ± 0.60	
	Norm_Zoom	**91.91 ± 0.35**	24.38
Artificial location and zoom	None	43.94 ± 0.64	
	Norm_Loc	74.42 ± 0.56	30.48
	Norm_Zoom	87.96 ± 0.42	44.02
	Norm_Loc_Zoom	**89.02 ± 0.40**	45.08

* The modules that include artificial hand location and zoom variability have been not applied.

3.3. Results for Dataset 3

With this dataset, we used a batch size of 5000. The proposed normalizations continue to result in significant improvements in scenarios specified in Table 3.

By applying zoom normalization, not only much higher accuracy rates are obtained than when applying localization normalization, but superior results are obtained than those obtained by applying both normalizations.

Table 3. Results for Dataset 3.

Dataset 3	Normalization Used	Accuracy (%)	Upgrade (%)
Original *	None	44.99 ± 0.39	
	Norm_Loc	67.97 ± 0.36	22.98
	Norm_Zoom	**85.03 ± 0.28**	40.04
	Norm_Loc_Zoom	82.13 ± 0.30	37.14
Artificial location	None	58.68 ± 0.38	
	Norm_Loc	**61.18 ± 0.38**	2.50
Artificial zoom	None	57.90 ± 0.38	
	Norm_Zoom	**88.56 ± 0.25**	30.66
Artificial location and zoom	None	47.09 ± 0.39	
	Norm_Loc	69.86 ± 0.36	22.77
	Norm_Zoom	**86.43 ± 0.27**	39.34
	Norm_Loc_Zoom	82.85 ± 0.29	35.76

* The modules that include artificial hand location and zoom variability have been not applied.

4. Discussion and Conclusions

When a limited time of training is used, the performance of a hand pose recognizer model can decrease due to the variability of location and zoom in the instances used to train the neural network. The application of location and zoom normalizations results in significant accuracy improvements in this situation. These techniques are more impactful when the variability is higher. For example, the performance of the system has raised from 43.94 ± 0.64% to 89.02 ± 0.40% (45.08%) applying both normalizations.

Comparing both normalizations, the zoom normalization results in a better performance of the model compared to the location normalization, reaching higher rates in all the studied scenarios. In addition, the application of zoom normalization has resulted in better results compared to applying both normalizations sequentially in some situations. From this, it can be deduced that this algorithm not only mitigates the effects of size variability, but it also mitigates those of location variability.

For future work, it could be interesting to apply the proposed techniques in other datasets related to hand pose recognition with a wide variety of classes, such as thumb up, thumb down, open hand, or okay.

Author Contributions: Conceptualization, M.A.R., M.G.-M., and R.S.-S.; Methodology, M.A.R., M.G.-M., and R.S.-S.; Software, M.A.R. and M.G.-M.; Validation, M.G.-M. and R.S.-S.; Formal analysis, M.G.-M.; Investigation, M.A.R. and M.G.-M.; Resources, R.S.-S.; Data curation, M.A.R. and M.G.-M.; Writing—original draft preparation, M.A.R. and M.G.-M.; Writing—review and editing, R.S.-S.; Visualization, M.G.-M.; Supervision, R.S.-S.; Project administration, R.S.-S.; Funding acquisition, R.S.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in Kaggle repository: ASL alphabet test: https://www.kaggle.com/datasets/danrasband/asl-alphabet-test; Synthetic ASL alphabet: https://www.kaggle.com/datasets/lexset/synthetic-asl-alphabet and Asl alphabet: https://www.kaggle.com/datasets/grassknoted/asl-alphabet.

Acknowledgments: The work leading to these results was supported by the Spanish Ministry of Science and Innovation through the projects AMIC-PoC, BeWord, GOMINOLA, and TremorDetect: Detección de la enfermedad de Parkinson a través de señales inerciales (PDC2021-120846-C42, PID2021-126061OB-C43, PID2020-118112RB-C21, and PID2020-118112RB-C22 funded by MCIN/AEI/10.13039/501100011033, by E.T.S.I. de Telecomunicación, and by the European Union “NextGenerationEU/PRTR”), and ASTOUND (101071191 HORIZON-EIC-2021-

PATHFINDERCHALLENGES-01) funded by the European Commission. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

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

References

1. Gil-Martín, M.; San-Segundo, R.; de Córdoba, R.; Pardo, J.M. Robust Biometrics from Motion Wearable Sensors Using a D-vector Approach. *Neural Process. Lett.* **2020**, *52*, 2109–2125. https://doi.org/10.1007/s11063-020-10339-z.
2. Gil-Martín, M.; López-Iniesta, J.; Fernández-Martínez, F.; San-Segundo, R. Reducing the Impact of Sensor Orientation Variability in Human Activity Recognition Using a Consistent Reference System. *Sensors* **2023**, *23*, 5845.
3. Shin, J.; Matsuoka, A.; Hasan, M.A.; Srizon, A.Y. American Sign Language Alphabet Recognition by Extracting Feature from Hand Pose Estimation. *Sensors* **2021**, *21*, 5856. https://doi.org/10.3390/s21175856.
4. Rasband, D. ASL Alphabet Test. Available online: https://www.kaggle.com/datasets/danrasband/asl-alphabet-test (accessed on).
5. Lexset. Synthetic ASL Alphabet. Available online: https://www.kaggle.com/datasets/lexset/synthetic-asl-alphabet (accessed on).
6. Akash. Asl Alphabet. Available online: https://www.kaggle.com/datasets/grassknoted/asl-alphabet (accessed on).
7. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.L.; Yong, M.G.; Lee, J.; et al. MediaPipe: A Framework for Building Perception Pipelines. *arXiv* **2019**, arXiv:1906.08172.
8. Google. Hand Landmarks Detection Guide. Available online: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker (accessed on).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.