Gesture Recognition in RGB Videos Using Human Body Keypoints and Dynamic Time Warping

Pascal Schneider, Raphael Memmesheimer, Ivanna Kramer and Dietrich Paulus

Active Vision Group, Institute for Computational Visualistics, University of Koblenz-Landau, 56070 Koblenz, Germany
{pschneider, raphael, ivannamyckhal, paulus}@uni-koblenz.de,
http://homer.uni-koblenz.de, http://agas.uni-koblenz.de

Abstract. Gesture recognition opens up new ways for humans to intuitively interact with machines. Especially for service robots, gestures can be a valuable addition to the means of communication to, for example, draw the robot's attention to someone or something. Extracting a gesture from video data and classifying it is a challenging task and a variety of approaches have been proposed throughout the years. This paper presents a method for gesture recognition in RGB videos using *OpenPose* to extract the pose of a person and *Dynamic Time Warping* (DTW) in conjunction with *One-Nearest-Neighbor* (1NN) for time-series classification. The main features of this approach are the independence of any specific hardware and high flexibility, because new gestures can be added to the classifier by adding only a few examples of it. We utilize the robustness of the Deep Learning-based OpenPose framework while avoiding the data-intensive task of training a neural network ourselves. We demonstrate the classification performance of our method using a public dataset.

1 Introduction

Gesture recognition is an active field of research with applications such as automatic recognition of sign language, interaction of humans and robots or for new ways of controlling video games. The main application we have in mind is an accessible way to use gestures for interacting with service robots.

Deep learning-based approaches have set new records in classification tasks in terms of their performance throughout the last few years. Consequently, they have also been applied to the problem of gesture recognition, where they could also provide good results. However, this usually comes at the cost of being very data-intensive. As with many deep learning techniques, good performance can usually only be reached with large amounts of labeled training samples. Our goal is therefore to present an approach which allows adding new gestures to the classifier with minimal effort. The training process we employ significantly reduces the overhead. Moreover, removing a gesture from the model does not

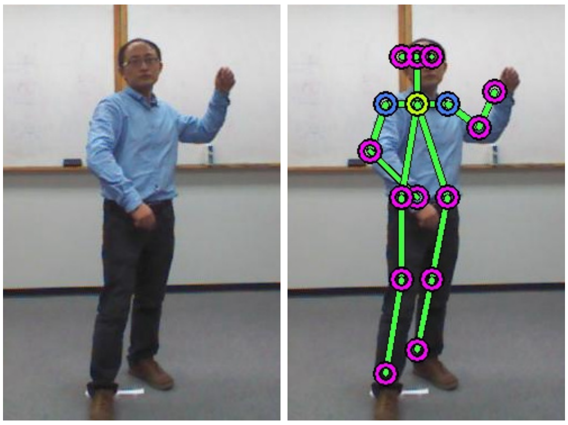


Fig. 1: Example for an extracted pose using OpenPose from the UTD-MHAD dataset [1].

involve any further cost, whereas for many other machine learning algorithms this would entail re-training the entire model [2].

Moreover, we want to avoid the need of any specific hardware. For example, the *Microsoft Kinect* is a popular platform for training models and collecting data for gesture recognition [3] [4] [5] [6], since it provides not only an RGB video but also depth data. The task of gesture recognition can be simplified by placing special markers on the person's body [7] or special gloves for hand gestures [8]. Since the main application we have in mind is human-robot interaction for service robots, relying on installing hardware on humans or manipulating the environment beforehand are impractical.

We present a method that completely avoids proprietary platforms and the need of specific hardware and instead only relies on RGB video that can be recorded using any camera with reasonable video quality in attempt to make gesture recognition more accessible. The key idea is to combine the capability of the deep learning-based OpenPose framework for extracting poses from color images and DTW, a well-established method for time-series classification.

The paper is structured as follows: in Section 2 we give an overview of some recently proposed methods for gesture recognition as well as a selection of relevant papers for both OpenPose and Dynamic Time Warping. In Section 3 we describe our approach, which is summarized in Fig. 2. In Section 4 we present the results of our experiments. We conclude our findings in Section 5 and motivate possible future research in Section 6.

2 Related Work

Using Dynamic Time Warping for gesture recognition is an established approach [9][5][10]. For time-series classification (TSC) in general, DTW in combination with a *One-Nearest-Neighbor* (1NN) classifier has shown to provide very strong performance [11][12]. DTW has been prominently used in the field of speech recognition since the 1970's. A lot of research has been focused on reducing the computational complexity, e.g. by introducing global constraints such as the *Sakoe-Chiba-Band* [13], the *Itakura-Parallelogram* [14] or the *Ratanamahatana-Koegh-Band* [15]. Notable work in improving the performance of DTW has also been done by Salvador and Chan, who proposed an approximation of DTW with linear time and space complexity [16].

A detailed description of Dynamic Time Warping and the constraints is beyond the scope of this paper and hence omitted here. Introductions to the DTW algorithm and some extensions can be found in [17] and [18].

The growing popularity of deep learning has also influenced research in the field of gesture recognition. The method presented in this paper only uses deep learning for extracting the pose of people, not for the classification. Others have presented neural network architectures to address the problem of gesture recognition directly as a whole. Examples include the *Two Streams Recurrent Neural Network* proposed by Chai et al. [19] or the *Recurrent 3D Convolutional Neural Network* (R3DCNN) by Molchanov et al. [20].

The problem of recognizing gestures in a video or any other sequence of data can be split into two sub-problems: segmentation and recognition. A sequence of data might contain any number of gestures, therefore the individual gestures have to be segmented first. If both segmentation and recognition are performed, it is commonly referred to as *continuous* gesture recognition. Whereas if only recognition is done, this is called *isolated* gesture recognition. Our method only addresses the latter. An approach to extending a DTW-based gesture recognition to the continuous case is given in [9].

A general survey on different gesture recognition techniques can be found in [21], also including DTW as an approach.

There has also been research on performing gesture recognition on single RGB images using *Convolutional Pose Machines* and different supervised learning techniques [22]. The authors concluded that an extension towards using sequences rather than single images could presumably lead to significant improvements.

Our work is focused on how the human poses extracted by OpenPose can be processed and used as input signals for Dynamic Time Warping in such a way that these two components form a processing pipeline which ultimately yields a classification of human gestures using only RGB images. What sets this apart from proposed methods based on the *Microsoft Kinect* [23][3][5] is that we do not use depth data and extract the pose key points ourselves instead of relying on ones provided by the Kinect framework. This makes our approach independent of any special sensor hardware.

Rwigema et al. proposed an approach to optimize weights for gesture recognition when using weighted DTW [24]. They also used the *UTD-MHAD* dataset to verify the performance of their method and achieved an accuracy of 99.40%. The key difference compared to our method is their choice of data to perform the recognition on. While we restrict ourselves to only the RGB video, Rwigema et al. aimed at a multi-sensor setup using skeleton joint frames and data from a depth sensor and inertial sensor.

3 Approach

Fig. 2 shows the basic processing pipeline of the proposed method approach. Its individual steps will be detailed in the following.

3.1 Recording RGB Videos

Avoiding the need for special hardware is one of the key aspects of the method we want to present. We therefore only use RGB videos. The image quality and resolution have to be sufficiently high to enable OpenPose to reliably extract the pose key points. Moreover, the video frame rate has to be high enough to provide adequate spatial resolution of the signals. Most customary web cams will nowadays meet this requirement which we hope will make this method very accessible.

3.2 Pose Estimation

To extract the pose, we use the OpenPose framework, which is based on Convolutional Pose Machines [25]. It features different pose models such as *MPI*, *COCO* and *BODY_25*. We chose the *COCO* model, because we consider its 18 key points to provide a good trade-off between a detailed representation of the human pose and complexity. OpenPose also supports extracting key points for the face, hands [26] and feet [27], but for our application aimed at full-body gestures these key points add hardly any useful information while greatly increasing the computational complexity.

3.3 Normalization

The pose key points from OpenPose are given in image coordinates. We normalize the key points first before passing them on to the DTW classifier to achieve scale invariance and translational invariance. This is necessary, because otherwise the key points' coordinates are dependent on the position of the person was standing relative to the camera. We ignore rotational invariance, since we consider this to be much less relevant, because humans can be expected to be in a mostly upright position under normal circumstances. However, adding rotational invariance might be necessary if tilt of the camera has to be corrected for.

The normalization is a simple coordinate transformation done in two steps:

1. **Translation:** All the key points are translated such that the neck key point becomes the origin of the coordinate system. This is achieved by subtracting the neck key points coordinates from all other key points.

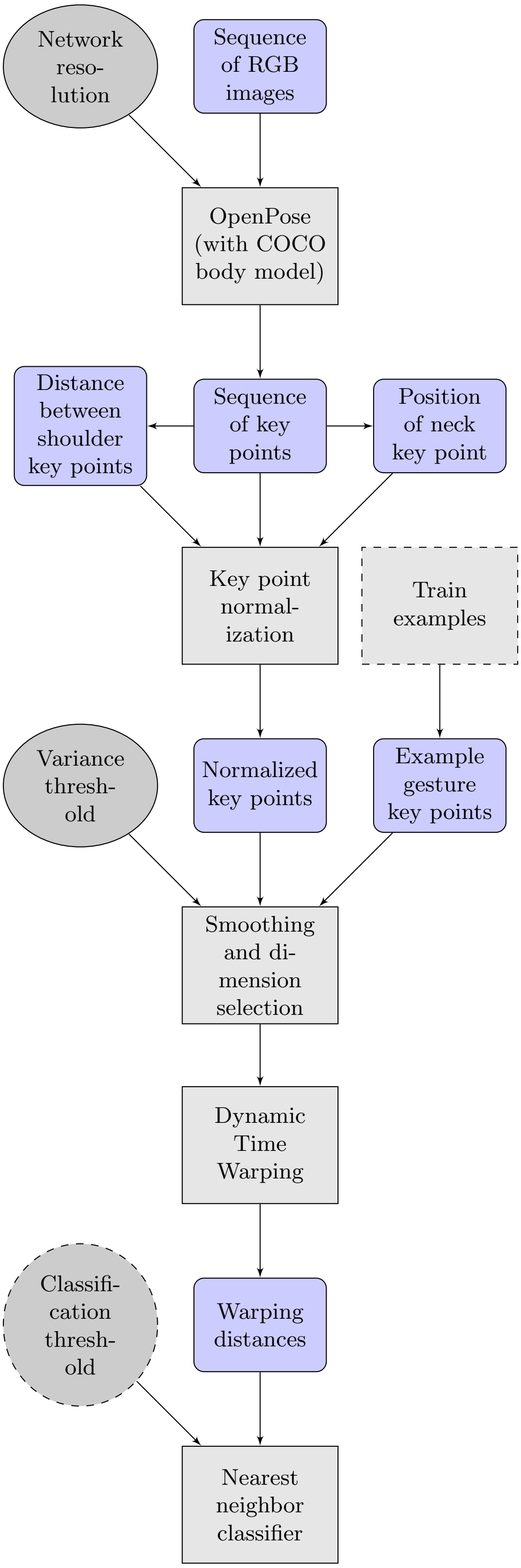

Network reso- lution
Sequence of RGB images
OpenPose (with COCO body model)
Distance between shoulder key points
Sequence of key points
Position of neck key point
Key point normal- ization
Train examples
Variance thresh- old
Normalized key points
Example gesture key points
Smoothing and di- mension selection
Dynamic Time Warping
Classifi- cation thresh- old
Warping distances
Nearest neighbor classifier

Fig. 2: Overview of the processing pipeline of our method. (Grey rectangles represent processing steps, blue rectangles represent data, ellipses represent parameters.)

2. **Scaling:** The key points are scaled such that the distance between the left shoulder and right shoulder key point becomes 1. This is done by dividing all key points coordinates by the distance between the left and right shoulder key point.

The scale normalization is inspired by Celebi et al. [23]. It can be easily seen that this way of normalizing the scale can fail when the person is not oriented frontal to the camera since the shoulder-to-shoulder distance we consider here is not the actual distance in the world but instead its 2D projection onto the image plane. This leads to an important assumption of our approach: the person performing the gesture has to be oriented (roughly) frontal to the camera.

Fig. 1 shows an example of an extracted pose skeleton for a video frame from the *UTD-MHAD* dataset [1]. The neck and shoulder key points are highlighted due to their importance for the normalization.

3.4 Train Examples

We employ a simple One-Nearest-Neighbor classifier, which has proven to work well with Dynamic Time Warping in the context of time-series classification [11][12]. A classification that relies on comparing directly against a set of labeled examples does not need a training stage per se. New gestures could be added simply by adding an example of it. Yet it can be beneficial to incorporate a training step to find the best examples to include for the classifier. Such a method is described by Gillian et al. [2]. A number of examples for the same gesture is recorded and the examples are compared to each other using the same DTW algorithm used by the classifier. The example with the minimum total warping distance to all other examples for the same gesture is chosen. This can be thought of as choosing the example which represents the gesture the best.

Instead of selecting a single example from the recorded ones for each gesture, you could also use all of them and switch to using a *k-nearest-neighbor* classifier instead of 1NN. The key argument against this approach is that the computational complexity of the classification grows linearly with the number of examples each sequence has to be compared to. Therefore, we try to limit the number of examples where possible.

3.5 Smoothing and Dimension Selection

For most gestures, only parts of the body are relevant. Hence, only a few of the key points might be relevant for each gesture. Take for example a *wave-with-left-hand* gesture: only the key points of the left arm are of relevance here and the others will usually be uninformative. This observation can be used to reduce the dimensionality of the problem at this step. If all key point sequences were to be used in the DTW, this would total up to 36 dimensions (18 key points with an x- and y-coordinate each). The neck key point coordinates will always be uninformative due to the normalization. To further reduce the number of signals to be processed by DTW, we perform a dimension selection. This step is greatly inspired by the work of ten Holt et al.

The criterion to select a dimension is the variance of its signal. Key points that do not move significantly during a gesture will cause the signals of the respective coordinates to be roughly constant with only little variance. All signals whose variance is below a threshold will be filtered out and are assumed to be uninformative. This filtering is done for the sequence to be classified as well as for the example sequences of each gestures. The set of dimensions for which DTW algorithm is then performed is the union of those dimensions for which the variance is above the threshold for either the sequence to be classified or the example sequence. If only those dimensions were considered where the variance is above the threshold for the sequence to be classified, some combinations of gestures could pose problems. Consider for example if there were a *wave-with-left-hand*, *wave-with-right-hand* and a *wave-with-both-hands* gesture. Classify a newly recorded *wave-with-left-hand* correctly is problematic if only its salient dimensions would be used in the DTW. Variance in the signal might also be due to noise. A noticeable source of noise was observed caused by the limited spatial resolution of the output from OpenPose. The quantization error caused sudden spikes in the signal. We therefore smooth the signal first before determining whether it should be included for the DTW. We use a median filter with radius $r = 3$ for the smoothing. The decision whether a dimension will be included for the DTW is done on the median filtered signal. However, the signal used for further processing is instead filtered using a Gaussian filter with $\sigma = 1$. This is done because the median filter is very effective at removing the noise spikes, but edges in the resulting signal are overly brought out. This worsened classification performance in our experiments, while the Gaussian filter is able to mitigate noise without these adverse effects.

In a last step before the DTW, the mean of the signal is subtracted from it, thus making it *zero-mean.* A common step for feature scaling is to also normalize the signal to have *unit-variance* by dividing the signal by its standard deviation. However, this had an adverse effect on classification performance in our experiments, possibly because differences in the amplitude of key point coordinate signals is relevant for classification. We therefore only transform the signals to to zero-mean, but *not* to unit-variance.

3.6 Dynamic Time Warping

We employ the *FastDTW* method by Salvador and Chan [16] to perform DTW on each selected dimension separately. Their method is aimed at providing an approximation of DTW with less computational cost compared to the classical DTW algorithm. Finding the optimal warping path is not guaranteed with this method, but we consider this limitation to be outweighed by the superior computational performance. For the internal distance metric FastDTW uses we chose Euclidean distance. The result is the warping distances of the sequence to be classified to all gesture examples of the classifier.

Identifier	Action description
a1	right arm swipe to the left
a6	cross arms in the chest
a7	basketball shoot
a9	right hand draw circle (clockwise)
a24	sit to stand
a26	forward lunge (left foot forward)

Table 1: Selected actions from the *UTD-MHAD* [1] dataset to perform classification on

3.7 Classification

The classification is done using a simple *One-Nearest-Neighbor Classifier* (1NN). The metric used for determining the nearest neighbor is the warping distance. A new sequence is classified to a gesture class by calculating the warping distance to all training examples and choosing the class of the training sample for which the warping distance is minimal. An additional threshold can be used in order not to classify a gesture sequence to any class if it does not resemble any of the example gesture sequences. If the minimal warping distance is still very high, this sequence can be considered to contain none of the known gestures.

4 Experiments

4.1 Dataset

To evaluate the performance of our method we chose the multi-modal human action dataset of the University of Texas at Dallas (*UTD-MHAD* [1]). Each gesture is performed by eight subjects four times each. Since we want to operate on RGB data, we use the color videos they provide. These videos feature a resolution of 640x480 pixels at around 30 frames per second. Due to the limitations of the normalization method, we specifically selected gestures where the shoulder-to-shoulder key point distance remains roughly constant throughout the sequence. The selected gestures are given by Table 1.

4.2 Key Point Signals

Fig. 3 shows the signals for every normalized coordinate of the extracted pose for a video sequence consisting of 44 images, i.e. it shows a separate signal for each x- and y-coordinate of each key point. Since the COCO body model has 18 key points, there are 36 individual signals. The video sequence shows a person performing the *right arm swipe to the left* gesture. Most signals are roughly constant throughout the sequence. However, four of the dimensions are highlighted in Fig. 3, since they can be considered salient and provide especially

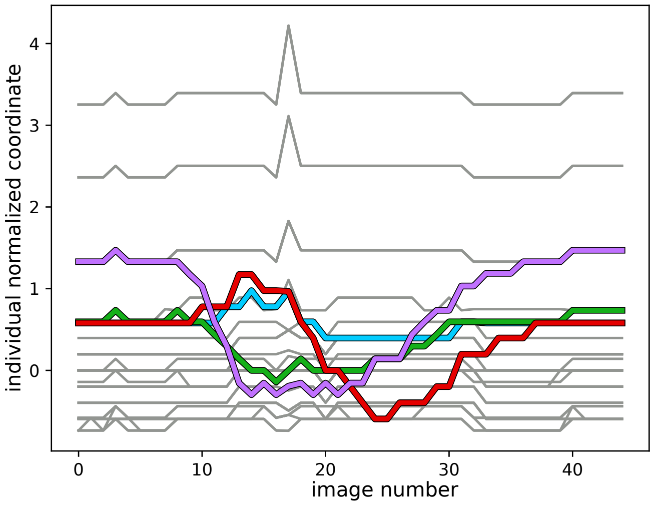


Fig. 3: Normalized key point coordinates for a sequence of 44 images from a person performing the *right arm swipe to the left* gesture in the *UTD-MHAD* dataset. The salient dimensions are highlighted.

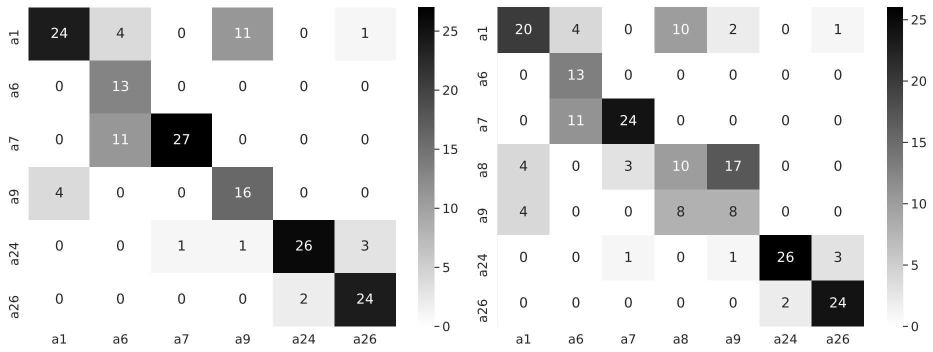
good signal shapes for DTW to work with. Unsurprisingly, these dimensions belong to the x- and y-coordinate of the left hand and left arm key point.

The variations in the signals for key points of body parts which are not being moved during the *right arm swipe to the left* gesture (such as legs etc.) are mostly due to the noise caused by the limited resolution of the extracted pose. A conspicuous noise spike can be seen at frame 17. It is caused by the neck key point being located at a slightly higher position for one frame. Since the neck key point is the origin of our normalized coordinate system, it has a noticeable effect across multiple dimensions. A median filter with radius $r = 3$ will filter out most of these spikes, which is the reason why we introduce this filtering step.

4.3 Classification Performance

We selected six different gestures from the *UTD-MHAD* dataset. The selected gestures are given in Table 1. To select an example for each gesture, we only considered the gesture performances by subject one, i.e. from the four sequences for subject one for each of the gestures, one is selected as described in Section 3.4. The other three sequences are not considered for the classification. From the 168 sequences which were classified, 130 were classified correctly. This equates to approximately 77.4%. The confusion matrix is given by Fig. 4 (a).

To further test the discriminative strength of the classification, we added another gesture to the classification: gesture *a8, right hand draw x*. The confusion matrix for this experiment is illustrated in Fig. 4 (b). As can be seen, the classification performance deteriorated significantly. 125 of 196 sequences were



(a) (b)

Fig. 4: Confusion matrix for the classification of the actions given in Table 1. Actual classes are on the horizontal axis, predicted classes on the vertical axis (a). Confusion matrix for the classification for performance of *a8* (b).

	Correctly classified	
t_{var}	**without a8**	**with a8**
0.05	72.0%	63.3%
0.10	77.4%	63.8%
0.15	74.4%	67.9%
0.20	76.2%	66.8%

Table 2: Percentage of correctly classified gestures for different variance thresholds t_{var}

classified correctly (63.8%). Most notably, *a9* was classified as *a8* more often than it was classified correctly. This clearly shows the limitations of the method. Gestures that are too similar to each other can not be distinguished.

An important parameter for the processing is the variance threshold t_{var}. Choosing a very low threshold will result in many dimensions being selected for the DTW step, which makes computation slow. If the threshold is set too high on the other hand, possibly none of the signals will exceed it and the classification will fail because no data reaches it. Table 2 shows the classification performance for different values of t_{var}. Finding the appropriate value for t_{var} is not part of the method, so it has to be chosen a priori. We can not derive any general advice for how to choose t_{var} from this data, this question could be addressed in future research.

5 Conclusion

We presented a method for gesture recognition on RGB videos using OpenPose to extract pose key points and Dynamic Time Warping in conjunction with a

One-Nearest-Neighbor classifier to perform classification. We showed how this can be used to perform gesture recognition with only very little training data and without the need for special hardware. Our first tests using this method yielded promising results if the gestures where sufficiently different, but also revealed limitations in case of attempting to classify more similar gestures.

Recent methods for gesture recognition using multi-modal data are often able to outperform our results in terms of accuracy, even more so considering our focus on only few selected gestures. Examples include the method by Rwigema et al. [24] with an accuracy of 99.40% on the *UTD-MHAD* dataset, Celebi et al. [23] with an accuracy of 96.70% on their own dataset or Molchanov et al. [20] with up to 83.8% accuracy, also using their own custom dataset. Nonetheless, we find our results promising considering the substantially reduced amount of data available to our method by restricting ourselves to only RGB videos, which is often the most easily obtainable data in a real-world scenario.

6 Future Work

A variety of modifications to the original DTW algorithm have been presented through the years. Some have already been mentioned in Section 2. Others include for example methods for adding feature weighting [9], *Derivative Dynamic Time Warping (D-DTW)* [28] or *Multi-Dimensional* DTW [10]. The effect these modifications have on the performance of nearest neighbor classifiers based on warping distance is often times not obvious and also dependent on factors like noise in the signal [10]. Future research could try to work out general guidelines for when to use which variant of DTW. In addition to these fundamental algorithmic options, there are a number of other factors which can impact the classification performance parameters, such as the window size of DTW, the variance threshold or the choice of example gestures.

Only single-person gesture recognition has been regarded in this paper. Since OpenPose is also capable of detecting the poses of multiple people at once, upgrading to multi-person gesture recognition is a possible subject for future research. The *UTD-MHAD* dataset we used for our experiments was recorded in a very controlled environment, further tests should be conducted to find out how our results generalize to more realistic scenarios.

Another topic of research is how this method can be sped up, desirably up to the point where it reaches real-time capability.

References

1. C. Chen, R. Jafari, and N. Kehtarnavaz, “UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor,” in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 168–172.
2. N. Gillian, B. Knapp, and S. O’Modhrain, “Recognition of multivariate temporal musical gestures using n-dimensional dynamic time warping.” in *Nime*, 2011, pp. 337–342.

3. A. Rosa-Pujazón, I. Barbancho, L. J. Tardón, and A. M. Barbancho, "Fast-gesture recognition and classification using Kinect: An application for a virtual reality drumkit," *Multimedia Tools and Applications*, vol. 75, no. 14, pp. 8137–8164, 2016.
4. F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao, "Multi-layered gesture recognition with Kinect," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 227–254, 2015.
5. A. Ribó, D. Warchol, and W. Oszust, "An approach to gesture recognition with skeletal data using dynamic time warping and nearest neighbour classifier," *International Journal of Intelligent Systems and Applications*, vol. 8, no. 6, pp. 1–8, 2016.
6. M. A. Bautista, A. Hernández-Vela, V. Ponce, X. Perez-Sala, X. Baró, O. Pujol, C. Angulo, and S. Escalera, "Probability-based dynamic time warping for gesture recognition on RGB-D data," in *International Workshop on Depth Image Analysis and Applications.* Springer, 2012, pp. 126–135.
7. S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.
8. N. Y. Y. Kevin, S. Ranganath, and D. Ghosh, "Trajectory modeling in gesture recognition using cybergloves® and magnetic trackers," in *2004 IEEE Region 10 Conference TENCON 2004.* IEEE, 2004, pp. 571–574.
9. M. Reyes, G. Dominguez, and S. Escalera, "Feature weighting in dynamic time warping for gesture recognition in depth data," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1182–1188.
10. G. A. Ten Holt, M. J. Reinders, and E. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *Thirteenth Annual Conference of the Advanced School for Computing and Imaging*, vol. 300, 2007, p. 1.
11. X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd International Conference on Machine Learning.* ACM, 2006, pp. 1033–1040.
12. A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.
13. H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.
14. F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975.
15. C. A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," in *Proceedings of the 2004 SIAM International Conference on Data Mining.* SIAM, 2004, pp. 11–22.
16. S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.
17. M. Müller, *Information retrieval for music and motion.* Springer, 2007.
18. P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, pp. 1–23, 2008.

19. X. Chai, Z. Liu, F. Yin, Z. Liu, and X. Chen, "Two streams recurrent neural networks for large-scale continuous gesture recognition," in *23rd International Conference on Pattern Recognition (ICPR).* IEEE, 2016, pp. 31–36.
20. P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4207–4215.
21. H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, 2018.
22. R. Memmesheimer, I. Mykhalchyshyna, and D. Paulus, "Gesture recognition on human pose features of single images," in *Intelligent Systems (IS), 2018 9th International Conference on.* IEEE, 2018, pp. 1–7.
23. S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici, "Gesture recognition using skeleton data with weighted dynamic time warping." in *VISAPP (1)*, 2013, pp. 620–625.
24. J. Rwigema, H.-R. Choi, and T. Kim, "A differential evolution approach to optimize weights of dynamic time warping for multi-sensor based gesture recognition," *Sensors (Basel, Switzerland)*, vol. 19, no. 5, p. 1007, 2019.
25. S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.
26. T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.
27. Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," in *arXiv preprint arXiv:1812.08008*, 2018.
28. E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proceedings of the 2001 SIAM International Conference on Data Mining.* SIAM, 2001, pp. 1–11.