



Gesture Recognition and Multi-modal Fusion on a New Hand Gesture Dataset

Monika Schak^(✉) and Alexander Gepperth

Fulda University of Applied Sciences, 36037 Fulda, Germany
{monika.schak, alexander.gepperth}@cs.hs-fulda.de

Abstract. We present a baseline for gesture recognition using state-of-the-art sequence classifiers on a new freely available multi-modal dataset of free-hand gestures. The dataset consists of roughly 100,000 samples, grouped into six classes of typical and easy-to-learn hand gestures. The dataset was recorded using two independent sensors, allowing for experiments on multi-modal data fusion at several depth levels and allowing research on multi-modal fusion for early, intermediate, and late fusion techniques. Since the whole dataset was recorded by a single person we ensure a very high quality of data with little to no risk for incorrectly performed gestures. We show the results of our experiments on unimodal sequence classification using a LSTM as well as a CNN classifier. We also show that multi-modal fusion of all four modalities results in higher precision using late-fusion of the output layer of an LSTM classifier trained on a single modality. Finally, we demonstrate that it is possible to perform live gesture classification using an LSTM-based gesture classifier, showing that generalization to other persons performing the gestures is high.

Keywords: Hand gestures · Dataset · Multi-modal data · Data fusion · Sequence classification · Gesture recognition

1 Introduction

We present a freely available multi-modal dataset of freehand gestures that can be used for research on sequence classification, multi-modal fusion, or other domains in Human-Computer-Interaction and Machine Learning. Hand gesture recognition is widely used as a natural way of non-verbal communication. In general, there are two main applications for hand gesture recognition: Communication, e.g. in sign language recognition, and manipulation, e.g. controlling a robot or other technical device. Another typical application is controlling and communicating with a virtual environment [8].

Large, multi-modal, and reliable datasets are needed for modern deep learning techniques to perform at a sufficiently high level. Important characteristics of a good dataset with training data are the number of gesture classes, the number of samples per gesture class, and the number of distinct modalities from (different) sensors. (cf. [24]).

We present results for baseline experiments to show that the dataset can be used to train state-of-the-art machine learning models and can achieve very high prediction rates. For this, we use Long Short-Term Memory (LSTM) networks as well as deep

Convolution Neural Networks (CNN). LSTM networks [9] are widely used recurrent neural networks with feedback connections to process sequences instead of just single data points. CNN are feed-forward neural networks consisting of fully-connected layers, pooling layers, and normalization layers. They are most common in image classification.

When working with multiple modalities, each sensor produces a separate data stream, also called sensory modality. Each sensory modality contains unique and independent information. But since all sensors observe the same situation – in our scenario the same hand gesture – the information from all sensors at least partially correlates. The goal of multi-modal fusion is to exploit that correlation to obtain more precise and reliable observations. In general, sensor data can be fused at three stages. Early fusion approaches combine data from sensors without preprocessing or after features have been extracted from raw data and use collaborative representation classifiers [14]. Late fusion strategies often combine the output score provided by multiple classifiers each trained on a single modality [28], i.e. by transforming the output to a probability score by a softmax layer and combining it by sum rule, product rule, or max rule.

This paper is an extended version of our previous work [25]. In this version, we conducted additional experiments using CNNs for uni-modal classification. We also more thoroughly explained the reasoning behind our choices regarding the single-user approach as well as the selection of our gesture classes.

2 Related Work

2.1 Hand Gesture Datasets

Several hand gesture datasets have been made available to researchers in recent years. Regardless, we find that there are no publicly available datasets that include a large enough number of gesture samples needed to suit the high requirement of machine learning methods for an extensive dataset for training. On the other hand, we require a dataset with a reasonable number of modalities recorded from independent sensors since present-day sensors are increasingly cheap and universally available and we find that gesture recognition can highly profit when including information from several modalities.

The SHGD dataset [13] consists of 15 gesture classes recorded from 27 persons with 96 sequence samples per class, resulting in a total size of 4,500 gesture samples. It only contains depth data recorded by an RGB-D camera.

A multi-modal dataset is presented in [16]. It consists of 10 gesture classes recorded from 14 persons with 140 sequence samples per class, resulting in a total size of 1,400 gesture samples. It contains depth data recorded by an RGB-D camera and data from a LeapMotion sensor.

In the next dataset [19], the approach is to render gesture samples using an advanced computer graphics pipeline instead of recording them. The dataset consists of 11 gesture classes with about 3,000 sequence samples per class, totaling 35,200 gesture samples. It only contains depth data.

The Cambridge dataset [12] contains 10 gesture classes recorded from two people with about 100 sequence samples per class, resulting in a total size of 1,000 gesture

Table 1. Comparison of the MMHG dataset with other hand-gesture datasets provided in literature. (Source: [25].)

Dataset	Classes	Samples/Class	Persons	Total samples	Modalities
SHGD [13]	15	96	27	4,500	Depth
Cambridge dataset [12]	10	100	2	1,000	RGB
n.A. [16]	10	100	14	1,400	Depth, Motion
IsoGD [30]	249	190	21	50,000	RGB, Depth
EgoGesture [32]	83	300	50	24,000	RGB, Depth
SKIG [15]	10	360	6	1,080	RGB, Depth
ChaLearn [6]	20	390	27	13,900	Audio, RGB, Depth
n.A. [18]	11	3,000	–	35,200	Rendered Depth
MMHG (this paper)	6	≈13,300	1	79,881	RGB, Depth, Motion, Audio

samples. The Sheffield Kinect Gesture Dataset [15] contains 10 gesture classes recorded from six persons with 360 sequence samples per class, resulting in a total size of 1,080 gesture samples. It also only contains depth data recorded by an RGB-D camera.

One of the first large-scale hand gesture datasets is the ChaLearn-2013 dataset [6], which consists of 20 gesture classes recorded from 27 persons with an average of 360 gesture samples per class, resulting in a total size of roughly 14,000 sequence samples. It contains audio, RGB, and depth modality.

The IsoGD dataset [29] is even larger but only includes an RGB and a depth modality. It contains about 50,000 gesture samples, grouped into 249 gesture classes with an average of 190 gesture samples per class, and it was recorded by 21 persons.

The next dataset [32] is of similar size and also includes only an RGB and a depth modality, but it is egocentric and recorded from a head-mounted camera. The dataset consists of 83 gesture classes with about 300 sequence samples per class, totaling about 24,000 gesture samples.

Table 1 shows a comparison of the MMHG dataset with the hand-gesture datasets introduced in this section.

2.2 Multi-modal Fusion

Research in the field of psychology and neurophysiology [1,2] shows multi-sensory fusion to be a common concept. This means, that the human brain is capable of probabilistically combining different modalities [5,7].

Similar concepts are available in the field of multi-modal gesture or activity recognition. There is not the one right way to carry out multi-modal fusion, but there are several possibilities that have been used in experiments in recent years. Each possibility has its own advantages and disadvantages depending on the data used and tasks at hand.

In general, multi-modal fusion techniques can be clustered into three categories: early fusion, intermediate fusion, and late fusion. Early fusion describes that data from different sensory modalities are combined either before any preprocessing steps [14] or after features have been extracted from raw data [4] by using a collaborative representation classifier. The fused data is then passed onto a machine learning model.

In late fusion techniques, there usually are multiple machine learning models for single modalities which are fused at a later stage, very often right before output score prediction. In [27], single modality-based classifiers each provide an output score and the final score is produced by searching the maximum. Another possibility is to use a collaborative representation classifier to combine classification outcomes of different modalities [4]. A very easy and therefore commonly used late-fusion technique is softmax score fusion [11]. Here, multiple classifier outputs are seen as probability scores by a softmax layer. Afterward, they are combined using either the sum rule, the product rule, or the max rule.

Intermediate fusion happens between data level and output level. A possible technique is feature fusion [10]. Features that are the output from fully connected layers are combined and then forwarded to a linear support vector machine or any other classifier.

2.3 Contribution

This work, like the original work [25], has its focus on describing our Multi-Modal Hand Gesture Dataset. Almost 80,000 samples with over 13,000 samples per gesture class in four modalities. There are six gesture classes that are all supposed to be easy to perform by the user but also specifically chosen to make the dataset beneficial for research on multi-modal fusion. All samples have been recorded by just one very well trained and instructed user, therefore the dataset does not contain corrupted data samples. The recording and preprocessing steps were carefully designed to ensure high quality. Thus, the dataset is perfectly suited for machine learning.

Additionally, we present experiments that prove the consistency of the dataset. The first set of experiments shows that plausible classification accuracies can be achieved on each of the four modalities when trained on two state-of-the-art sequence classification models: LSTM networks and CNNs. The second set of experiments shows that even with relatively simple late multi-modal fusion approaches it is possible to improve the classification accuracies achieved by networks trained on uni-modal data.

In Sect. 6 we introduce an implementation based on the Robot Operating System and the results for our experiments to determine the generalization capabilities of our dataset to other people. This shows that although only a single user recorded all data samples for the dataset, it is still possible to train a gesture classifier capable of correctly classifying gestures performed by other users.

3 Dataset

In this paper, we present the Multi-Modal Hand Gesture Dataset (MMHGD). It is a large-scale dataset with only six classes but a high number of samples for each class. Each sample consists of modalities from an RGB and 3D camera, a microphone, and an acceleration sensor.

All gesture samples are recorded and performed by just one single person. This is an unusual choice but since only a well-instructed person performs all gestures, there will be no incorrectly performed gesture samples in the dataset. Thus, we ensure a high quality of gesture recordings and little to no corrupted data.

The dataset contains about 13,300 recordings of each of the six classes. Therefore, it contains a total of almost 80,000 samples. Table 2 shows the exact distribution of each class. Each gesture sample consists of data from a two-second window. The dataset (raw as well as preprocessed data) can be downloaded at <http://data.informatik.hs-fulda.de>.

Table 2. Distribution of the six gesture classes in the MMHG dataset.

Class	C_1	C_2	C_3	C_4	C_5	C_6	Total
Samples	13,440	13,410	13,228	13,233	13,308	13,262	79,881

Although the gestures were recorded with varying background and lightning, we used a fixed setup to ensure that each sample is recorded within a predefined distance of 0.5 m to 0.75 m from the camera. Thus, simplifying the preprocessing step (Fig. 1).

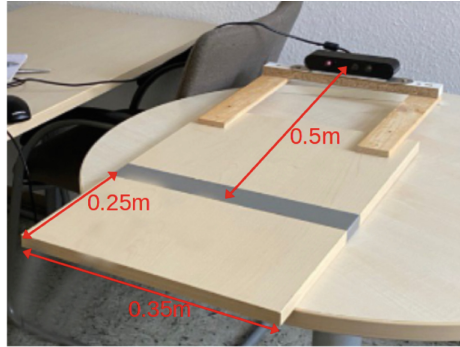


Fig. 1. The setup used for recording the gesture samples for the MMHG dataset, ensuring a fixed distance to the camera (Source: [25]).

During the time of recording, the user is told which gesture to perform and the recorded sample is immediately assigned the correct class label. Each recording therefore consists of RGB images, 3D point clouds, an mp3 file, the acceleration data and the correct class label. We also have preprocessed data available to immediately use for training and testing. The preprocessing step happens independently of the recording and is described in Sect. 3.3.

3.1 Gesture Classes

We use easy-to-use gesture classes that are commonly used in human-machine-interactions. The gesture classes are specifically designed to offer challenging tasks for multi-modal fusion. Therefore, some of the classes rely heavily on fusing different modalities for correct classification. For example, the last two classes – Snap Once and

Snap Twice – do not differ much in their motion. Thus, using only visual modalities will probably not lead to very good classification results. However, fusing those modalities with the audio modality will make them easily distinguishable.

Thumbs Up. (Class 1, denoted as C_1): The first class describes a thumbs-up gesture, which is a very typical gesture to show approval or agreement. The gesture starts with a closed fist in front of the camera. During the gesture, the thumb is extended upwards. There is no distinctive sound for this gesture and it is rather stationary with very little movement of the whole hand. Four frames of the RGB modality of one gesture sample are shown in Fig. 2 exemplarily.



Fig. 2. Four frames of the RGB modality of one Thumbs Up gesture sample.

Thumbs Down. (Class 2, denoted as C_2): The second class is similar to the first class as it describes a thumbs-down gesture. This gesture is commonly used to show rejection or disapproval. This gesture also starts with a closed fist in front of the camera, but this time the thumb is extended downwards. Just like the thumbs-up gesture class, there is no distinctive sound. It shows more movement since the fist is tilted forwards while the thumb is extended downwards. Figure 3 shows four frames of the RGB modality of one gesture sample as an example.



Fig. 3. Four frames of the RGB modality of one Thumbs Down gesture sample.

Swipe Left. (Class 3, denoted as C_3): The third class shows a swiping gesture of the whole hand in a horizontal direction from the right side to the left. Therefore, it is a dynamic gesture. It can be used to switch or forward to the next element or move an object from right to left. Again, this gesture does not have a distinctive sound. Four example frames from the RGB modality of one gesture sample can be seen in Fig. 4.



Fig. 4. Four frames of the RGB modality of one Swipe Left gesture sample.

Swipe Right. (Class 4, denoted as C_4): The fourth class is a swipe right gesture, similar to the previous gesture class. It also is a dynamic hand gesture that describes a swiping gesture of the whole hand in a horizontal direction from the left side to the right. It can be used to switch or rewind to the previous element or move an object from left to right. Unsurprisingly, this gesture also does not have a distinctive sound. Figure 5 shows four frames from the RGB modality of one gesture sample.



Fig. 5. Four frames of the RGB modality of one Swipe Right gesture sample.

Snap Once. (Class 5, denoted as C_5): The fifth class again is a rather stationary gesture, where the hand remains almost still while only two fingers show motion. For this gesture, the middle finger pushes hard against the thumb and then gets released so fast that it creates a snapping sound. Therefore, this gesture class has a distinctive sound. Four frames from the RGB modality of one gesture sample from this class are shown in Fig. 6.

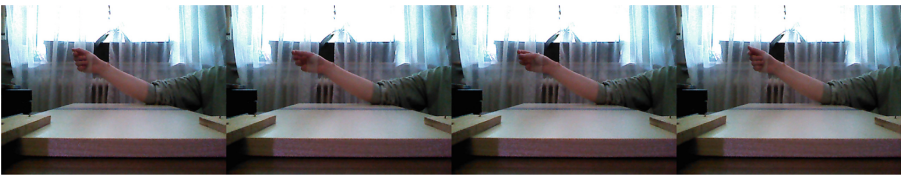


Fig. 6. Four frames of the RGB modality of one Snap Once gesture sample.

Snap Twice. (Class 6, denoted as C_6): The sixth gesture class is very similar to the previous class and describes two snaps. Accordingly, the hand remains almost still while only two fingers are moving. For this gesture, the middle finger and the thumb create two snapping sounds consecutively. This gesture class also has a distinctive sound. Figure 7 shows four frames of the RGB modality of one gesture sample as an example.



Fig. 7. Four frames of the RGB modality of one Snap Twice gesture sample.

3.2 Modalities

RGB. The sensory modality of RGB images is recorded with an Orbbec Astra 3D sensor. It outputs a stream of images at a resolution of 800×600 pixels with a frequency of 30 fps. To reduce the load and still be able to process all modalities at the same time, we save images at a frequency of 6 fps. This leads to twelve images for every gesture sample.

3D. The sensory modality of 3D images is also recorded with an Orbbec Astra 3D sensor. The 3D camera outputs a stream of depth images with a size of 640×480 pixels. Those images are converted to point clouds before being saved for further preprocessing. Due to performance reasons and to get matching RGB images to every 3D image, we also save point clouds at a frequency of 6 fps. Thus, receiving twelve point clouds for every gesture sample.

Audio. The audio data is also provided by the Orbbec Astra 3D sensor. The sensor has a sensitivity of 30 dB and works with audio between 20 Hz and 16 kHz. We save the raw wave data for the entire two-second window for every gesture sample.

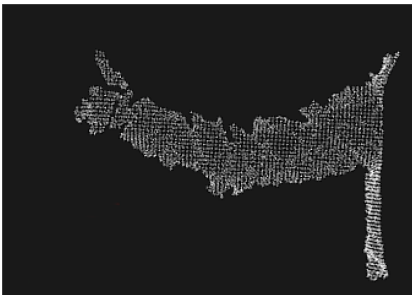
Acceleration. The sensory modality of the acceleration data is recorded using an acceleration sensor (BWT901CL from Bitmotion) attached to the users right wrist. The acceleration sensor offers 9-axis: acceleration data in three axis, yaw rates also in three axis, gyroscopic measurements, and magnetic field measurements. The sensor has a frequency of 200 Hz, which means we get 400 measurements for each gesture sample.

3.3 Preprocessing

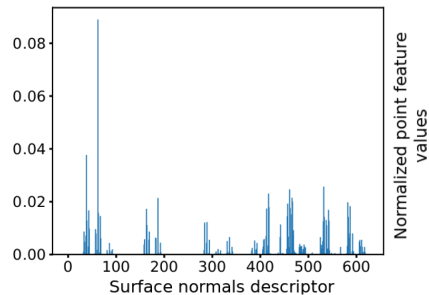
RGB. To reduce the computational costs and remove unnecessary overload, we crop the 800×600 px RGB images to the part where the hand is visible. Examples for the original RGB images can be seen in Figs. 2, 3, 4, 5, 6 and 7. Since we always perform the hand gesture in a predefined area in front of the camera, the complexity of this step is reduced: The hand is always in the same area in the RGB image for every frame for every gesture. Therefore, we do not have to perform object detection on every single image but instead can define the area to which to crop and it will work for all images. Afterward, we scale the cropped image to 72×48 pixels and then calculate

the Histogram of Oriented Gradients (HOG) [17,31] descriptor. For this, we use the OpenCV implementation with default parameters. The only parameters we set ourselves are the cell size, which we set to 8×8 pixels, and the block size, which we set to 16×16 pixels. The calculated HoG descriptor has 756 entries. The resulting NumPy array consists of twelve HoG descriptors for every gesture sample and thus has a shape of $(N, 12, 756)$.

3D. During the recording phase we store twelve point clouds for every gesture sample. One exemplary frame from a sample of the Thumbs Up gesture is shown in Fig. 8a. Each of these point clouds goes through the same three steps during the preprocessing phase. The first step is downsampling the point cloud to reduce the size and computational costs of the following steps. We begin by removing measurement errors by deleting all points where one or more of the x -, y , or z -value is not a number (NaN), then we perform downsampling using the 3D-voxel grid technique. Using this downsampled point cloud, we perform conditional removal to delete all points that are outside of our predefined volume of interest. Again, this can be done using the same volume of interest for every single gesture frame since we ensured that the gesture is always performed in the same area in front of the camera. The result is a point cloud of just the hand without any background data. In the second step, we infer surface normals by using approximation and use those to calculate Point Feature Histograms (PFH) [22,23] in the third step. With PFHs, we are able to receive a descriptor with the same size for every point cloud - although they have a high variability in size. This is important since machine learning models often require a fixed input size. According to [21], we randomly select two surface normals and compute the “four values based on the length and relative orientation of the surface normals” [25]. By dividing each value into five intervals, we receive 625 possible discrete values, which then get normalized. The resulting histogram consisting of 625 dimensions is “able to feasibly characterize the hand and fingers” [25]. Figure 8a shows a single point cloud (frame) of the Thumbs Up gesture class, while Fig. 8b shows the corresponding PFH. The resulting NumPy array has a shape of $(N, 12, 625)$.



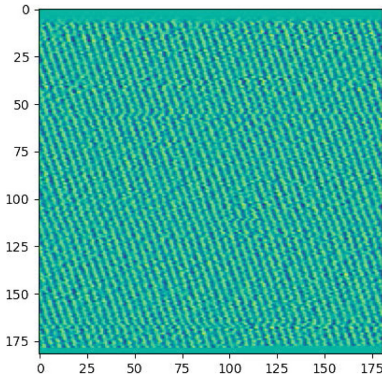
(a) Example of one point cloud of a sample for the gesture class Thumbs Up.



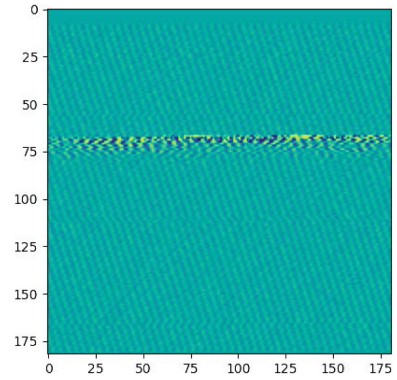
(b) Example of the corresponding point feature histogram to Figure 8a.

Fig. 8. Example of one frame of a gesture sample from the gesture class Thumbs Up (C_1) before (a) and after (b) preprocessing (Source: [25]).

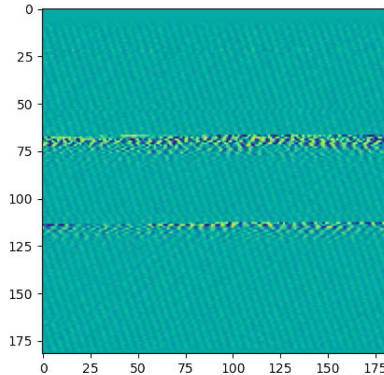
Audio. We downsample the raw wave data to a frequency of 8,000 Hz and ensure an equal length by using zero-padding with a randomly picked offset. Afterwards, Short-Time Fourier Transform (STFT) [3, 20] with the following parameters is performed: a window of 455 data points with an overlap of 420 data points. Using STFT we can visualize the frequency information in 2D, more precisely the change of frequency during certain time frames. The result is a NumPy array with a shape of $(N, 182, 181)$. Exemplarily, STFT data for a gesture sample with no distinct audio (Swipe Left, C_3) is shown in Fig. 9a. Figure 9b shows the STFT data for a gesture sample with one snap, while Fig. 9c shows the STFT data for a gesture sample with two snaps. While Fig. 9a



(a) Plotted STFT data for a gesture sample without distinct audio of gesture class Swipe Left (C_3).



(b) Plotted STFT data for a gesture sample with distinct audio of gesture class Snap Once (C_5)



(c) Plotted STFT data for a gesture sample with distinct audio of gesture class Snap Twice (C_6).

Fig. 9. Examples of the STFT data of gesture classes without (a) and with (b, c) distinct audio (Source: [25]).

shows no distinct change in frequency, Figs. 9b and 9c show distinct changes in frequency during one or two time frames respectively.

3.4 Acceleration

The acceleration data is rather noisy. Thus, we calculate statistical values over the 20 tuples from each 200-millisecond window.

$$\bar{x} = \frac{1}{N} \left(\sum_{i=1}^N x_i \right) = \frac{x_1 + x_2 + \dots + x_N}{N} \quad (1)$$

$$\text{Var}(x) = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (2)$$

$$S(x) = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (3)$$

In our two-second window for every gesture sample, we receive a total of ten descriptors. Each descriptor contains the mean (cf. Eq. 1), the variance (cf. Eq. 2) and the standard deviation (cf. Eq. 3) for each of the six axes. The resulting NumPy array has a shape of $(N, 10, 3, 6)$.

4 Unimodal Classification

To show that every single modality of the MMHG dataset can be used to train state-of-the-art machine learning models to be able to perform with high gesture classification accuracies, we provide results of unimodal classification experiments with different architectures. We conducted experiments on the modalities with sequential data (RGB, 3D, and acceleration) using LSTM networks. Then, we conducted experiments on all four modalities using CNNs.

4.1 LSTM

We provide results of our experiments (cf. [25]) using LSTM networks for the RGB, 3D, and acceleration modalities, since after preprocessing those three modalities consist of sequential data, while the audio data is only one 2D plot per gesture sample. LSTM networks [9] are recurrent neural networks capable of learning dependencies over time (long-term). They are often used in sequence classification since due to the feedback connections they are able to process sequences of data instead of single data points such as in images.

We randomly select 20% of our dataset to use as a test set and train the LSTM network on the remaining 80%. Preliminary experiments were used to determine network parameters for every modality that result in the highest classification accuracies.

The gesture classification accuracy is the fraction of correct predictions compared to all predictions, as shown in Eq. 4. Precision defines the proportion of correct positive

classifications compared to all positive classifications for a class, as shown in Eq. 5. Recall defines how many samples were classified correctly compared to all predictions of that class, as shown in Eq. 6. The F1-score is the harmonic mean between precision and recall, as shown in Eq. 7. (tp = true positive, tn = true negative, fp = false positive, fn = false negative)

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \quad (4)$$

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \quad (5)$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \quad (6)$$

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = \frac{2 \cdot \text{tp}}{2 \cdot \text{tp} + \text{fp} + \text{fn}} \quad (7)$$

RGB. For the RGB modality, we use an LSTM network with $S = 200$ cells each on $L = 2$ hidden layers and train it in $I = 3,000$ iterations with a batch size of $b = 250$. After training the network on our training set with those parameters, we achieve an average gesture classification accuracy on the test set of 85.55%. Table 3a shows the confusion matrix for the unimodal gesture classification, while Table 3b shows the precision, recall and F1-score for every class.

Table 3. Results for the unimodal classification on the RGB modality using an LSTM network (Source: [25]).

		Predicted class $C_{[1-6]}$					
		2662	26	0	0	0	0
Target $C_{[1-6]}$	0	2335	90	38	26	193	
	0	165	2430	26	25	0	
	0	76	64	2430	76	0	
	0	166	0	25	1806	664	
	0	318	0	38	293	2004	

(a) Confusion matrix for an LSTM network trained on RGB data.

Class	Precision	Recall	F1-Score
C_1	1.00	0.99	0.99
C_2	0.75	0.86	0.80
C_3	0.94	0.92	0.93
C_4	0.95	0.92	0.93
C_5	0.80	0.67	0.73
C_6	0.69	0.74	0.72

(b) Classification report for an LSTM network trained on RGB data.

3D. For the 3D modality, we use an LSTM network with $S = 250$ cells each on $L = 2$ hidden layers and train it in $I = 5,000$ iterations with a batch size of $b = 1,000$. After training the network on our training set with those parameters, we achieve an average gesture classification accuracy on the test set of 93.43%. Table 4a shows the confusion matrix for the unimodal gesture classification, while Table 4b shows the precision, recall and F1-score for every class.

Table 4. Results for the unimodal classification on the 3D modality using an LSTM network (Source: [25]).

		Predicted class $C_{[1-6]}$					
		C_1	C_2	C_3	C_4	C_5	C_6
Target $C_{[1-6]}$	C_1	2688	0	0	0	0	0
	C_2	2	2667	4	8	1	0
	C_3	0	4	2613	29	0	0
	C_4	0	0	16	2627	3	0
	C_5	1	0	0	0	2350	310
	C_6	0	0	1	0	670	1982

(a) Confusion matrix for an LSTM network trained on 3D data.

Class	Precision	Recall	F1-Score
C_1	1.00	1.00	1.00
C_2	1.00	0.99	1.00
C_3	0.99	0.99	0.99
C_4	0.98	0.99	0.99
C_5	0.74	0.86	0.80
C_6	0.84	0.70	0.77

(b) Classification report for an LSTM network trained on 3D data.

Acceleration. For the acceleration modality, we use an LSTM network with $S = 250$ cells each on $L = 5$ hidden layers and train it in $I = 1,000$ iterations with a batch size of $b = 500$. After training the network on our training set with those parameters, we achieve an average gesture classification accuracy on the test set of 83.66%. Table 5a shows the confusion matrix for the unimodal gesture classification, while Table 5b shows the precision, recall and F1-score for every class.

Table 5. Results for the unimodal classification on the acceleration modality using an LSTM network (Source: [25]).

		Predicted class $C_{[1-6]}$					
		C_1	C_2	C_3	C_4	C_5	C_6
Target $C_{[1-6]}$	C_1	2571	25	20	12	39	21
	C_2	524	2124	17	4	8	5
	C_3	36	273	2250	52	30	5
	C_4	16	7	328	2204	86	5
	C_5	26	3	17	163	2116	336
	C_6	8	1	2	9	533	2100

(a) Confusion matrix for an LSTM network trained on acceleration data.

Class	Precision	Recall	F1-Score
C_1	0.81	0.96	0.88
C_2	0.87	0.79	0.83
C_3	0.85	0.85	0.85
C_4	0.90	0.83	0.87
C_5	0.75	0.80	0.77
C_6	0.85	0.79	0.82

(b) Classification report for an LSTM network trained on acceleration data.

4.2 CNN

We also provide the results of our experiments using CNNs for all four modalities. The modalities with sequential data are passed through the network as one data point with an additional temporal dimension. CNNs are state-of-the-art networks for image classification since they are highly able to recognize patterns in images. CNNs are designed according to multilayer perceptrons to reduce processing requirements and they usually consist of a combination of convolutional layers, pooling layers, fully connected layers, and normalization or reshaping layers. For our experiments, we use a CNN with eight layers: Three convolutional layers, two pooling layers, one reshaping layer, and two fully connected layers. The CNN was trained using the Adam Optimizer and cross-entropy as loss function. Again, we randomly select 20% of our dataset to use as a test set and train the CNN on the remaining 80% in ten epochs.

RGB. After training the CNN on the training set of the RGB modality, we achieve an average gesture classification accuracy on the test set of 85.45%. Table 6a shows the confusion matrix for the unimodal gesture classification of RGB data, while Table 6b shows the precision, recall and F1-score for every class.

Table 6. Results for the unimodal classification on the RGB modality using a CNN.

		Predicted class $C_{[1-6]}$					
		2660	24	1	0	2	1
Target $C_{[1-6]}$	0	2412	21	15	39	195	
	0	138	2464	19	23	2	
	0	55	79	2441	71	0	
	0	202	2	117	1611	729	
	0	271	0	45	274	2063	

(a) Confusion matrix for a CNN trained on RGB data.

Class	Precision	Recall	F1-Score
C_1	1.00	0.99	0.99
C_2	0.78	0.90	0.83
C_3	0.96	0.93	0.95
C_4	0.93	0.92	0.92
C_5	0.80	0.61	0.69
C_6	0.69	0.78	0.73

(b) Classification report for a CNN trained on RGB data.

3D. After training the CNN on the training set of the 3D modality, we achieve an average gesture classification accuracy on the test set of 94.05%. Table 7a shows the confusion matrix for the unimodal gesture classification of RGB data, while Table 7b shows the precision, recall and F1-score for every class.

Table 7. Results for the unimodal classification on the 3D modality using a CNN.

		Predicted class $C_{[1-6]}$					
		2687	1	0	0	0	0
Target $C_{[1-6]}$	3	2666	1	10	1	1	
	1	7	2619	17	2	0	
	0	0	20	2620	1	0	
	3	1	0	2	2376	279	
	0	2	4	0	589	2058	

(a) Confusion matrix for a CNN trained on 3D data.

Class	Precision	Recall	F1-Score
C_1	0.94	1.00	1.00
C_2	1.00	0.99	0.99
C_3	1.00	0.99	0.99
C_4	0.99	0.99	0.99
C_5	0.80	0.89	0.84
C_6	0.88	0.78	0.82

(b) Classification report for a CNN trained on 3D data.

Audio. After training the CNN on the training set of the audio modality, we achieve an average gesture classification accuracy on the test set of 45%. Table 8a shows the confusion matrix for the unimodal gesture classification of audio data, while Table 8b shows the precision, recall and F1-score for every class. As can be seen, there is a high recall for the two gestures depending on sound (C_5 and C_6 , Snap Once and Twice respectively) while there was a very low recall for the four gestures not depending on sound (C_1 to C_4 , Thumbs Up and Down, Swipe Left and Right respectively). As explained in Sect. 3.1, the purpose of the audio modality lies in reinforcing predictions in combination with other modalities [25].

Table 8. Results for the unimodal classification on the audio modality using a CNN (Source: [25]).

		Predicted class $C_{[1-6]}$					
		525	171	207	1771	9	5
Target $C_{[1-6]}$	460	230	217	1772	2	1	
	462	168	264	1750	2	0	
	480	188	178	1798	0	2	
	85	53	41	60	2076	346	
	9	6	10	4	381	2243	

(a) Confusion matrix for a CNN trained on audio data.

Class	Precision	Recall	F1-Score
C_1	0.26	0.19	0.22
C_2	0.28	0.09	0.13
C_3	0.29	0.10	0.15
C_4	0.25	0.68	0.37
C_5	0.84	0.78	0.81
C_6	0.86	0.84	0.85

(b) Classification report for a CNN trained on audio data.

Acceleration. After training the CNN on the training set of the acceleration modality, we achieve an average gesture classification accuracy on the test set of 69%. Table 9a shows the confusion matrix for the unimodal gesture classification of the acceleration data, while Table 9b shows the precision, recall, and F1-scores for every class. As can be seen, it is difficult for the CNN to distinguish Snap Once and Snap Twice (C_5 and C_6 respectively) using only the acceleration modality.

Table 9. Results for the unimodal classification on the acceleration modality using a CNN.

		Predicted class $C_{[1-6]}$					
		2422	5	29	150	53	29
Target $C_{[1-6]}$	606	1960	37	56	17	6	
	114	290	1575	616	44	7	
	109	2	350	2109	65	11	
	245	2	43	487	1520	364	
	129	3	15	138	967	1401	

(a) Confusion matrix for a CNN trained on acceleration data.

Class	Precision	Recall	F1-Score
C_1	0.67	0.90	0.77
C_2	0.86	0.73	0.79
C_3	0.77	0.60	0.67
C_4	0.59	0.80	0.68
C_5	0.57	0.57	0.57
C_6	0.57	0.53	0.63

(b) Classification report for a CNN trained on acceleration data.

5 Multi-modal Fusion

The results of our experiments on unimodal prediction show that both the visual modalities and the acceleration modality have difficulties to distinguish the two classes with little movement but distinct audio (C_5 and C_6 respectively). They also show that the audio modality cannot distinguish the four classes with no distinct sound (C_1 to C_4) but leads to acceptable results in the other two classes. Therefore, we investigate the effect of fusing different modalities to increase the prediction accuracy that can be achieved.

Since the sensors output the data in different formats and also in different frequencies, we discard early fusion methods that fuse the data before using them as input to the machine learning model. We choose two commonly used and easy-to-implement late fusion strategies to prove our assumption, other late fusion strategies as well as

intermediate fusion strategies are possible and might even lead to better results but are not within the scope of this work.

We analyse the LSTM or CNN readout layer r^m after the whole gesture sample has been processed, where m denotes the modality $m \in \mathcal{M} = \{\text{RGB}, \text{3D}, \text{Audio}, \text{Acc}\}$. The readout layer provides one entry for every available gesture class, every entry has a value of $r_i^m \in [0, 1]$ and all entries are normalized $\sum_i r_i^m = 1 \forall m \in \mathcal{M}$. \mathcal{C} denotes the decision based on the fused modalities, while \mathcal{C}^m denotes the decisions based on the uni-modal predictions. (Cf. [24])

The first late fusion strategy is called **max-conf**. Here, the uni-modal prediction of the modality with the highest confidence is used, as described in Eq. 8.

$$\begin{aligned} x &= \operatorname{argmax}_{m \in \mathcal{M}} \left(\max_i r_i^m \right) \\ \mathcal{C} &= \mathcal{C}^x \end{aligned} \quad (8)$$

The second late fusion strategy is called **prob**. Here, the readout layer entries are treated as independent conditional probability distributions for a class i given the uni-modal input sequence \mathbf{x}^m (cf. [24]). We denote the probabilities as $r_i^m = p^m(\mathcal{C}^m = i | \mathbf{x}^m)$ and use the class with the highest probability after multiplying the independent conditional probabilities, as shown in Eq. 9. (Cf. [24])

$$\begin{aligned} \mathcal{C} &= \operatorname{argmax}_i \left(\prod_{m \in \mathcal{M}} p(\mathcal{C} = i | \mathbf{x}^m) \right) \\ &= \operatorname{argmax}_i \left(\prod_{m \in \mathcal{M}} r_i^m \right) \end{aligned} \quad (9)$$

Figure 10 exemplarily shows some of the results of our multi-modal experiments. It can be seen that the results – that are already very high – can be further improved, even

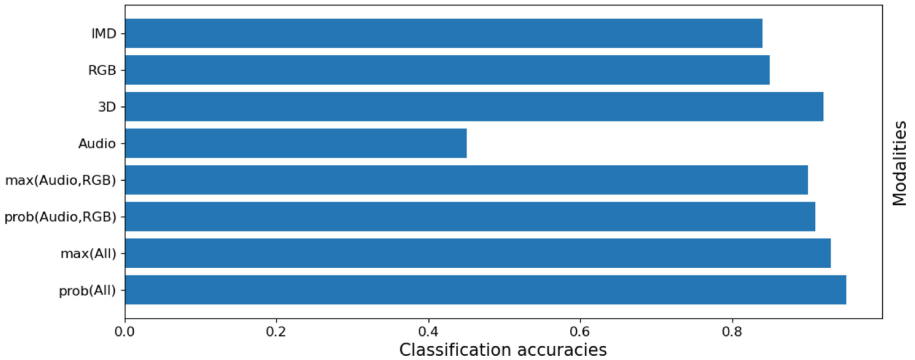


Fig. 10. Gesture classification accuracies achieved by the LSTM networks trained on uni-modal data (Acceleration data, RGB data, 3D data, Audio data) and by performing multi-modal fusion: max-conf of audio and RGB data, prob of audio and RGB as well as max-conf and prob of all four modalities. (Source: [25]).

with rather simple fusion strategies as used here. While fusing two modalities like audio and RGB does not yield better or the same results as uni-modal prediction on 3D data, these results show that fusing two modalities can improve the prediction accuracy, i.e. when a 3D sensor is not available or processing 3D data is not computationally feasible or even possible.

6 Live Demonstrator

Since our choice of only one person performing all the gestures in the dataset, we want to prove that the dataset still can be used to train a real-live system that is able to correctly classify hand gestures that are performed by different people. Thus, experiments on the live system prove that a machine learning model trained on the MMHG dataset is able to generalize to another person apart from the one who recorded all gesture samples.

As a proof of concept, we implement a live system that consists of an LSTM network trained only on the 3D modality of the MMHG dataset and the live classifier based on the Robot Operating Systems (ROS) that receives and processes 3D data from an Orbbec Astra as described in Sect. 3.3 and then feeds it into the trained LSTM network. Afterward, it receives the prediction and outputs it to the user. The other three modalities can be handled accordingly, the network model can be swapped out by another pre-trained model, i.e. a CNN, or fusion can be implemented as well, if needed.

6.1 Implementation

The implementation consists of a Point Cloud Processor that receives the data stream from our 3D sensor and processes it according to the preprocessing steps described in Sect. 3.3. It also consists of an LSTM classifier that receives the preprocessed data from the Point Cloud Processor and feeds it into multiple pre-trained LSTM networks. The third part is the Aggregator which collects the output predictions from the LSTM networks, selects the most likely prediction, and outputs it to the user.

Point Cloud Processor. The Point Cloud Processor is implemented as a ROS node and is responsible for receiving the data stream from the Orbbec Astra, processing the point clouds, and publishing the preprocessed data to the LSTM classifier.

According to Sect. 3.3, the Point Cloud Processor subscribes to the 3D camera sensor. It accepts 3D data at 6Hz corresponding to the frequency used in the MMHG dataset. The node receives the 3D image and performs the same preprocessing steps performed on the 3D data in the MMHG dataset. Thus, it downsamples the point clouds, infers surface normals, and then calculates Point Feature Histograms. Those PFHs are then published for further processing by the other nodes.

LSTM Classifier. The LSTM Classifier is also implemented as a ROS node and is responsible for performing gesture classification by passing the PFHs published by the Point Cloud Processor through multiple pre-trained LSTM networks. Since gestures can

start at any given moment in time, we use an approach called *Shifted Recognizer* [26]: N identical classifiers or recognizers are run simultaneously. Each classifier is pre-trained on the MMHG dataset and has learned to classify gestures with a fixed length $T = 12$ which determines the Temporal Receptive Field (TRF) (according to [26]). The LSTM Classifier node feeds the PFHs it receives from the Point Cloud Processor to all N classifiers. Each classifier has a delay of $\Delta = \frac{T}{N}$ frames compared to the other classifiers. Therefore, “if we run enough parallel classifiers, a gesture of length $l \leq T$ will always correlate with the TRF of a single classifier which will then classify it and report its prediction” [25]. Figure 11 shows $N = 4$ parallel classifiers or recognizers denoted as R_n with $n \in [1, N]$. The delay between the classifiers is set to $\Delta = \frac{T}{N}$. A performed gesture will fall into exactly one classifier’s TRF (indicated as a green bar in the figure), therefore this classifier will predict the gesture and output the results. All other classifiers (indicated as red bars) will only receive part of the gesture in their TRF, therefore they will not predict the gesture correctly.

In our system, we use $N = 12$ LSTM classifiers since the gesture samples in the MMHG dataset have 12 frames. Therefore, as described above, a gesture will always correlate to exactly one classifier with no onset or offset.

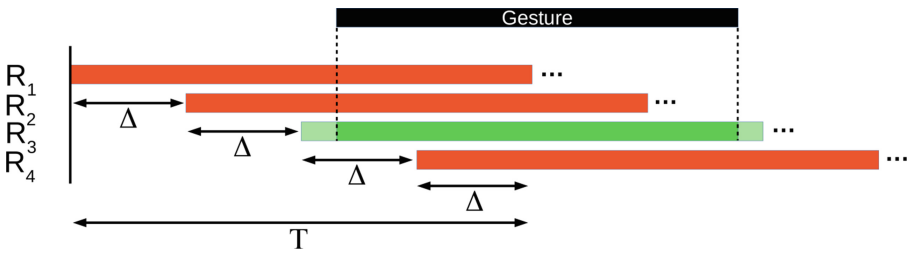


Fig. 11. Shifted Recognizer with $N = 4$. The delay is set to $\Delta = \frac{T}{4}$. The red and green bars indicate the TRF of the four Shifted Recognizers (denoted $R_{1...4}$). The currently performed gesture is shown as a black bar on the top. The current gesture fits in exactly one classifier’s TRF, thus it can predict the gesture correctly. (Source: [25]). (Color figure online)

Aggregator. The Aggregator is the third ROS node. It gathers the predictions from the readout layers from all $N = 12$ LSTM classifiers. According to previous research [26] LSTM networks are able to classify sequences with varying onset and offset to some extent. Thus, it is very likely that not just the LSTM classifier in whose TRF the gesture fits in completely but also some of the other LSTM classifiers will predict the correct gesture. Therefore, the Aggregator chooses the gesture class with the highest prediction score, but only if it exceeds a predefined threshold and has been stable for the past three frames. Thus, no prediction is chosen if there is no gesture in the data stream.

6.2 Experiments

We asked four people to perform gestures and had them classified by our live system. Neither of those people were the person who conducted the gestures in the MMHG

dataset. Each person was given a short introduction on how to perform the six different gestures correctly, then they performed gestures from every gesture class and we recorded the prediction made by the live system.

There were two male and two female users with different hand sizes and skin colors to present as much variation as possible to our system and test the capability of our system to generalize to different users (cf. [25]). Table 10 shows the results of our experiments. P_i denotes the ratio of correct classifications of a gesture class C_j (as described in Sect. 3.1) for the i -th user with $i \in \{1, 2, 3, 4\}$.

Table 10. Resulting gesture classification accuracies using our live system trained on the 3D modality of the MMHG dataset (Source: [25]).

	P_1	P_2	P_3	P_4	Total
C_1	5/5	1/1	1/1	3/3	100%
C_2	5/5	1/1	1/1	2/3	90%
C_3	5/5	1/1	0/1	2/3	80%
C_4	2/5	0/1	1/1	1/3	40%
C_5	5/5	1/1	1/1	2/3	90%
C_6	0/5	0/1	0/1	1/3	10%
Σ	73.3%	66.7%	66.7%	61.1%	66.7%

Since our live system is meant as a proof of concept and depends only on the 3D modality, it is not surprising that the system is not able to distinguish between the gesture classes Snap Once and Snap Twice (C_5 and C_6 respectively). Using multi-modal fusion of the 3D data with the audio data will most likely improve those results. Also, for our system Swipe Left and Thumbs Up (C_3 and C_1 respectively) are difficult to distinguish since the angle and movement of the hand is similar (cf. [25]). Again, using multi-modal fusion of the 3D data with – for example – acceleration data, could possibly improve the gesture classification accuracies for that gesture class (C_3).

Nevertheless, the experiments on our live system show that a system trained on the MMHG dataset is able to generalize to other people performing the gestures. The results can be improved by the use of multi-modal fusion with one or more other modalities.

7 Conclusion

We provide an in-depth description of the new, freely available Multi-Modal Hand Gesture Dataset consisting of almost 80,000 samples in six gesture classes with the four sensory modalities RGB, 3D, audio, and acceleration. The gesture classes of the dataset were carefully chosen to be easy to perform by all users and also suitable for application oriented experiments on sequence classification and multi-modal fusion.

It can be seen that even very simple late-fusion techniques can be combined with state-of-the-art sequence classification models such as LSTM and CNN models, thus

improving the results of uni-modal gesture classification. Unsurprisingly, the audio modality alone leads to disappoint gesture classification accuracies, but can improve the quality of gesture classification when being fused with other modalities. Of course this stems from the fact that our gesture classes were specifically chosen to show this kind of behavior, to allow the dataset to be well suited for research on multi-modal fusion.

Also, experiments conducted on a live system prove that a generalization to other persons is high even though only a single person recorded all gesture samples available in the dataset.

Future research will include further experiments on generalization capabilities and the possible bias in recognition due to the single subject in the dataset. Also, we will perform experiments using probabilistic models for multi-modal sequence classification, outlier detection, and sampling. Another focus in future work will be more complex – intermediate – fusion strategies, i.e. with an end-to-end learned fusion contribution at multiple stages in a network.

References

1. Angelaki, D.E., Gu, Y., DeAngelis, G.C.: Multisensory integration: psychophysics, neurophysiology, and computation. *Curr. Opin. Neurobiol.* **19**(4), 452–458 (2009)
2. Beauchamp, M.S.: See me, hear me, touch me: multisensory integration in lateral occipital-temporal cortex. *Curr. Opin. Neurobiol.* **15**(2), 145–153 (2005)
3. Becker, S., Ackermann, M., Lapuschkin, S., Müller, K.R., Samek, W.: Interpreting and explaining deep neural networks for classification of audio signals (2018)
4. Chen, C., Jafari, R., Kehtarnavaz, N.: Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE Trans. Hum.-Mach. Syst.* **45**, 51–61 (2014). <https://doi.org/10.1109/THMS.2014.2362520>
5. Ernst, M.O., Banks, M.S.: Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* **415**(6870), 429–433 (2002)
6. Escalera, S., et al.: Chalearn multi-modal gesture recognition 2013: grand challenge and workshop summary. In: *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, pp. 365–368 (2013)
7. Gepperth, A.R., Hecht, T., Gogate, M.: A generative learning approach to sensor fusion and change detection. *Cogn. Comput.* **8**(5), 806–817 (2016)
8. Guan, Y., Zheng, M.: Real-time 3D pointing gesture recognition for natural HCI. In: *2008 7th World Congress on Intelligent Control and Automation*, pp. 2433–2436 (2008). <https://doi.org/10.1109/WCICA.2008.4593304>
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
10. Imran, J., Raman, B.: Evaluating fusion of RGB-D and inertial sensors for multimodal human action recognition. *J. Ambient. Intell. Humaniz. Comput.* **11**(1), 189–208 (2019). <https://doi.org/10.1007/s12652-019-01239-9>
11. Khaire, P., Kumar, P., Imran, J.: Combining CNN streams of RGB-D and skeletal data for human activity recognition. *Pattern Recognit. Lett.* **115**, 107–116 (2018)
12. Kim, T.K., Cipolla, R.: Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(8), 1415–1428 (2008)
13. Kopuklu, O., Rong, Y., Rigoll, G.: Talking with your hands: scaling hand gestures and recognition with CNNs. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2019)

14. Liu, K., Chen, C., Jafari, R., Kehtarnavaz, N.: Fusion of inertial and depth sensor data for robust hand gesture recognition. *IEEE Sens. J.* **14**(6), 1898–1903 (2014). <https://doi.org/10.1109/JSEN.2014.2306094>
15. Liu, L., Shao, L.: Learning discriminative representations from RGB-D video data. In: *Twenty-Third International Joint Conference on Artificial Intelligence* (2013)
16. Marin, G., Dominio, F., Zanuttigh, P.: Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimed. Tools Appl.* **75**(22), 14991–15015 (2015). <https://doi.org/10.1007/s11042-015-2451-6>
17. McConnell, R.: *Method of and apparatus for pattern recognition* (1986)
18. Memo, A., Minto, L., Zanuttigh, P.: Exploiting silhouette descriptors and synthetic data for hand gesture recognition. In: Giachetti, A., Biasotti, S., Tarini, M. (eds.) *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association (2015). <https://doi.org/10.2312/stag.20151288>
19. Memo, A., Zanuttigh, P.: Head-mounted gesture controlled interface for human-computer interaction. *Multimed. Tools Appl.* **77**(1), 27–53 (2016). <https://doi.org/10.1007/s11042-016-4223-3>
20. Nasser, K.: *Digital Signal Processing System Design: LabVIEW Based Hybrid Programming* (2008)
21. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3384–3391. IEEE (2008)
22. Sachara, F., Kopinski, T., Gepperth, A., Handmann, U.: Free-hand gesture recognition with 3D-CNNs for in-car infotainment control in real-time. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 959–964 (2017). <https://doi.org/10.1109/ITSC.2017.8317684>
23. Sarkar, A., Gepperth, A., Handmann, U., Kopinski, T.: Dynamic hand gesture recognition for mobile systems using deep LSTM. In: Horain, P., Achard, C., Mallem, M. (eds.) *IHCI 2017*. LNCS, vol. 10688, pp. 19–31. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72038-8_3
24. Schak, M., Gepperth, A.: On multi-modal fusion for freehand gesture recognition. In: Farkaš, I., Masulli, P., Wermter, S. (eds.) *ICANN 2020*. LNCS, vol. 12396, pp. 862–873. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61609-0_68
25. Schak, M., Gepperth, A.: Gesture recognition on a new multi-modal hand gesture dataset. In: *ICPRAM* (2022)
26. Schak, M., Gepperth, A.: Robustness of deep LSTM networks in freehand gesture recognition. In: Tetko, I.V., Kůrková, V., Karpov, P., Theis, F. (eds.) *ICANN 2019*. LNCS, vol. 11729, pp. 330–343. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30508-6_27
27. Tran, T., et al.: A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality. In: *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1947–1952 (2018). <https://doi.org/10.1109/ICPR.2018.8546308>
28. Tran, T.H., et al.: A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality. In: *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1947–1952 (2018). <https://doi.org/10.1109/ICPR.2018.8546308>
29. Wan, J., Li, S.Z., Zhao, Y., Zhou, S., Guyon, I., Escalera, S.: ChaLearn looking at people RGB-D isolated and continuous datasets for gesture recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 761–769 (2016). <https://doi.org/10.1109/CVPRW.2016.100>
30. Wan, J., Zhao, Y., Zhou, S., Guyon, I., Escalera, S., Li, S.Z.: ChaLearn looking at people RGB-D isolated and continuous datasets for gesture recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 56–64 (2016)

31. William T. Freeman, M.R.: Orientation histograms for hand gesture recognition. Technical report TR94-03, MERL - Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 (1994)
32. Zhang, Y., Cao, C., Cheng, J., Lu, H.: EgoGesture: a new dataset and benchmark for egocentric hand gesture recognition. *IEEE Trans. Multimed.* **20**(5), 1038–1050 (2018). <https://doi.org/10.1109/TMM.2018.2808769>