Gesture Recognition on a new Multi-Modal Hand Gesture Dataset

Monika Schak[1], Alexander Gepperth[1]
[1]*Fulda University of Applied Sciences, 36037 Fulda, Germany*
{*monika.schak, alexander.gepperth*}*@cs.hs-fulda.de*

Keywords: Hand Gestures, Dataset, Multimodal Data, Data Fusion, Sequence Detection.

Abstract: We present a new large-scale multi-modal dataset for free-hand gesture recognition. The freely available dataset consists of 79,881 sequences, grouped into six classes representing typical hand gestures in human-machine interaction. Each sample contains four independent modalities (arriving at different frequencies) recorded from two independent sensors: a fixed 3D camera for video, audio and 3D, and a wearable acceleration sensor attached to the wrist. The gesture classes are specifically chosen with investigations on multi-modal fusion in mind. For example, two gesture classes can be distinguished mainly by audio, while the four others are not exhibiting audio signals – besides white noise. An important point concerning this dataset is that it is recorded from a single person. While this reduces variability somewhat, it virtually eliminates the risk of incorrectly performed gestures, thus enhancing the quality of the data. By implementing a simple LSTM-based gesture classifier in a live system, we can demonstrate that generalization to other persons is nevertheless high. In addition, we show the validity and internal consistency of the data by training LSTM and DNN classifiers relying on a single modality to high precision.

1 INTRODUCTION

This work is in the context of multi-modal hand gesture recognition. That is a field of machine learning that has profound application relevance in, e.g., human-machine-interaction (HMI). Since modern deep learning methods are powerful but require a large amount of data to reach their peak performance, successful hand gesture recognition requires sufficiently large and reliable datasets. Furthermore, datasets should reflect the following: Present-day sensors are increasingly cheap and universally available and gesture recognition can profit hugely when including information from several sensory sources (or modalities).

Training data for hand gesture recognition may be characterized by the number of distinct modalities, the number of included gesture classes, and the available gesture samples per class. Another important characteristic is sample diversity: this can be promoted by, e.g., choosing different illumination or background conditions. Sample diversity can be further enhanced by choosing a large number of different persons performing the gestures.

In the Multi-Modal Hand Gesture Dataset (MMHG) described here, we chose to create a large-scale dataset with relatively few classes but a large number of samples per class. We include audio, RGB, depth, and IMD sensing as modalities into all gesture samples. For diversity, we make a rather unusual choice: all gestures are performed by a single person. This will certainly reduce diversity, although the advantages are significant as well: a single person will be well instructed in performing the gestures, so there are few corrupted or incorrectly performed samples in the data. This would certainly be the case if a large number of persons were to perform the gestures, with little time for each person to learn the correct way of performing them. Furthermore, this is a typical application setting, where, e.g., an infotainment system in a modern vehicle, is mainly interacting with, and adapted to, a single user. In this context, additional performance boosts can be obtained by specializing *to* that user. We therefore model an "educated" user here, one that is acquainted with the gesture recognition system being used. We show that it is possible to train a system with our dataset and still obtain good results when classifying hand gestures performed by different users.

Table 1: Overview of already available hand-gesture datasets and the MMHG dataset presented in this article.

Dataset	Classes	Samples/ Class	Persons	Total Samples	Modalities
SHGD (Kopuklu et al., 2019)	15	96	27	4,500	Depth
Cambridge Dataset (Kim and Cipolla, 2008)	10	100	2	1,000	RGB
n.A. (Marin et al., 2016)	10	100	14	1,400	Depth, Motion
IsoGD (Wan et al., 2016)	249	190	21	50,000	RGB, Depth
EgoGesture (Zhang et al., 2018)	83	300	50	24,000	RGB, Depth
SKIG (Liu and Shao, 2013)	10	360	6	1,080	RGB, Depth
ChaLearn (Escalera et al., 2013)	20	390	27	13,900	Audio, RGB, Depth
n.A. (Memo et al., 2015)	11	3,000	-	35,200	rendered Depth
MMHG (this paper)	6	$\approx$ 13,300	1	79,881	RGB, Depth, Motion, Audio

2 RELATED WORK

Numerous gesture datasets have been proposed in recent years: The SHGD dataset (Kopuklu et al., 2019) contains only depth data recorded by an RGB-D camera. It consists of 15 gesture classes with 96 sequence samples per class, recorded from 27 persons. The total dataset size is about 4,500 gesture samples. (Marin et al., 2016) present a multi-modal dataset containing depth data from an RGB-D and LeapMotion sensor. It consists of ten gesture classes with 100 gesture samples per class recorded from 14 persons. In total, this amounts to a dataset size of 1,400 gesture samples. An interesting approach is pursued in (Memo et al., 2015): instead of recording gesture samples, the authors propose to render them using an advanced computer graphics pipeline. The resulting dataset contains depth data grouped into eleven gesture classes, with about 3,000 gesture samples per class. In total, the dataset contains 35,200 gesture samples. The Cambridge dataset (Kim and Cipolla, 2008) contains 1,000 RGB samples grouped in ten gesture classes, with 100 gesture samples per class recorded from two persons. The Sheffield Kinect Gesture Dataset (Liu and Shao, 2013) contains 1,080 RGB-D sequences, grouped into ten classes, with 360 gesture samples per class recorded from six subjects.

The first really large-scale dataset on hand gesture recognition to be published was the ChaLearn-2013 dataset (Escalera et al., 2013). It contains roughly 14,000 gesture samples grouped into 20 classes with an average of 360 gesture samples per class and recorded from 27 persons. Included modalities are audio, RGB and depth. An even larger dataset, the IsoGD dataset, was published in (Wan et al., 2016), although it just includes RGB and depth modalities. This dataset contains about 50,000 gesture samples, grouped into 249 gesture classes, with an average of 190 gesture samples per class and recorded from 21 persons. A dataset of similar size that includes RGB and depth was presented in (Zhang et al., 2018). Here, 24,000 gesture samples were recorded from 50 persons and grouped into 83 classes with roughly 300 gesture samples per class. A particularity of this dataset is that it is egocentric and recorded from a head-mounted camera.

To summarize, we find that there are no publicly available datasets that, on one hand, include a large number of gesture samples ($>$ 10,000), and which, on the other hand, contain a reasonable number of modalities recorded from independent sensors. We aim to close this gap with the dataset we are describing here. An overview of the mentioned datasets is given in Table 1.

An interesting point to make here is that all datasets we know aim to capture gestures from different people to include as much diversity as possible. As stated before, our approach is different: we only record from a single person, which results in a less diverse but more reliable dataset that closely reflects

applications of gesture recognition in, e.g., HMI.

For completeness: There exists a large number of multi-modal datasets for human activity recognition (Ranasinghe et al., 2016; Romdhane et al., 2013; Chen et al., 2015; D. Lara and Labrador, 2013; Ni et al., 2011; Zhang and Sawchuk, 2012; Radu et al., 2018; Sharma et al., 2016), a related but less well-defined field. The main feature of these datasets is data from wearable sensors (which play a role in our dataset as well), as well as the inclusion of multiple RGB and RGB-D sensors.

2.1 Contribution

The main contribution of this article is the presentation of a new, large-scale dataset for hand gesture recognition consisting of four modalities. It offers a large number of samples per class. Each class is designed carefully to show the benefits of multi-modal fusion. In addition, we offer carefully curated data coming from a single person that is well-instructed. Thus, we are ensuring a consistently high quality of data for machine learning.

In addition, we present experiments showing the consistency of the dataset by achieving very plausible classification performances on each of the modalities, taken individually. Lastly, we describe a real-time implementation of the 3D-based gesture classifier and demonstrate excellent generalization to other persons. This last point is very important, since it shows that the relative lack of diversity in our dataset is not an obstacle to generalization.

3 DATASET

The dataset contains recordings of six different hand gestures. There are around 13,300 recordings of each gesture class, totaling 79,881 samples. All gestures are performed by a single person as explained above. This approach ensures that the conducted gestures are performed correctly and consistently across the dataset, and also that the variability of each gesture class is not excessive. Each gesture sample, irrespectively of its class, lasts for two seconds. The hand gestures are observed by the following four modalities: a wearable IMD sensor, an RGB sensor, a 3D sensor, and a microphone.
Table 2 shows information about the provided data before and after preprocessing. Both the raw and preprocessed data can be downloaded at `data.informatik.hs-fulda.de`.

3.1 Setup

We used a fixed setup for all recordings to ensure each gesture is recorded with the same distance to the camera. For that, we bolted the camera to a board and marked the area in which to conduct the gesture. This setup is shown in Figure 1.

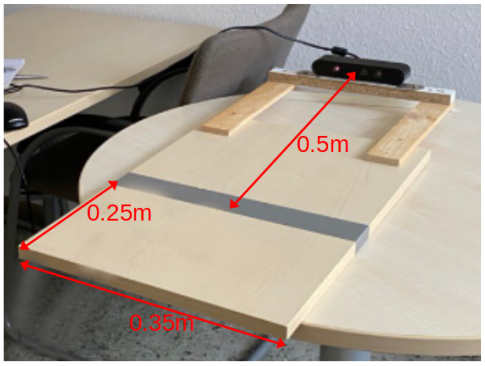

Figure 1: The fixed setup to record hand gestures for our multi-modal gesture dataset.

In every recording, each gesture class is repeated ten times before moving on to the next class. Therefore, each recording produces ten samples for each of the six classes or a total of sixty samples. During the time of recording, the samples are immediately assigned class labels. For each recording, we save the RGB images as PNG files, the 3D point clouds as PCD files, audio as MP3 files, the acceleration data and labels as NumPy arrays. In a separate step, independently of the recording, we preprocess the data into a format that can be used as training and test input. The recorded data and the preprocessing step are described in Sections 3.3 to 3.6.

3.2 Gesture Classes

Our dataset consists of six classes: two are rather stationary, two are rather dynamic and two rely on sound to be distinguished from each other.

- **Thumbs Up (0)** The first class is a thumbs-up gesture. It is a stationary gesture with very little movement.
- **Thumbs Down (1)** The second class is a thumbs down gesture. It is also a stationary gesture with very little movement.
- **Swipe Left (2)** The third class is a dynamic gesture. For this, the whole hand swipes from right to left.
- **Swipe Right (3)** The fourth class is also a dynamic gesture. In contrast to the third class, the whole hand swipes from left to right.

Table 2: Information about the provided data in our new multi-modal hand gesture dataset before and after preprocessing.

	Before preprocessing		After preprocessing	
Modal.	**Format**	**Size**	**Format**	**Size**
RGB	PNG, 640×480 pixel, 12/gest.	546.0 GB	Numpy, $(N, 12, 756)$	39.8 GB
3D	PCD, 12/gesture	372.6 GB	Numpy, $(N, 8, 625)$	2.0 GB
Audio	MP3, 16 kHz, 1/gesture	1.8 GB	Numpy, $(N, 182, 181, 1)$	21.2 GB
IMD	Numpy, $(N, 20, 7)$, 10/gesture	124.0 MB	Numpy, $(N, 10, 3, 6)$	114.9 MB

- **One Snap (4)** The fifth class is a rather stationary hand gesture as well. The thumb and middle finger make a snapping sound, therefore sound is important for this gesture.
- **Two Snaps (5)** The sixth class is comparable to the fifth class, with the difference of snapping twice instead of once. Therefore, sound is also important for this gesture.

3.3 RGB Data

To record RGB data, we use the video stream provided by an Orbbec Astra 3D sensor. It sends a stream of 800×600 RGB images which we record. Preprocessed as follows: after cropping the images to the part in which the hand is visible, we scale them to 72×48 pixels. Afterward, we calculate the histogram of oriented gradients (HOG) descriptor (McConnell, 1986; William T. Freeman, 1994) for each image, using the OpenCV implementation. We use the default parameters except for cell size which is set to be 8×8 pixels, and block size which is 16×16 pixels, resulting in a descriptor of 756 entries. We set the frame rate such as to receive twelve images per gesture. Thus, a gesture is characterized by twelve HOG descriptors, each having a fixed size of 756 values. The preprocessing results in a NumPy array for the RGB data with a shape of $(N, 12, 756)$.

Figure 2: Example of one frame for an RGB sample of class 5 (One Snap).

Figure 2 shows one of the recorded RGB images for class 5 (One Snap). The setup – as in position and distance to the camera – is always the same, the background as well as the lighting might vary.

3.4 3D Data

To record 3D data, we use the stream of depth images provided by an Orbbec Astra 3D sensor. The depth images have a size of 640×480 and are converted to point clouds, then stored. During the two-second window for each gesture, we receive a total of twelve point clouds. Each of these point clouds is passed through the five steps of processing:

- *Downsampling:* At first, we reduce the size of the point clouds to lower the computational costs. Therefore, we create a 3D-voxel grid over the input point cloud data. For every voxel, we calculate the centroid of all its points and use this to represent the voxel.
- *VoI Filtering:* In the second step, we use conditional removal to crop the point cloud to a defined volume of interest. Thus, removing background data and leaving just the area in which the hand is present.
- *Removing NaN:* Afterwards, we remove measurement errors by deleting all points whose x-, y-, or z-value is equal to NaN.
- *Computing Normals:* In the fourth step, we use approximations to infer the surface normals for all remainng points in the point cloud.
- *Creating a Point Feature Histogram (PFH):* To get a descriptor of fixed length, regardless of the size of the point clouds, that can be fed to a machine learning model, we decided on a representation with PFH (Sachara et al., 2017; Sarkar et al., 2017). These descriptors characterize the phenomenology of hand, palm, and fingers in a precise manner while remaining computationally feasible at the same time. PFH is based on the surface normals computed in the previous step. Now, we repeatedly select two points and compute their descriptor (Rusu et al., 2008), which provides four

values based on the length and relative orientation of the surface normals. Each of the four values is subdivided into five intervals, giving a total of 625 discrete possibilities. The result is a 625-dimensional histogram for each point cloud. Lastly, we normalize the histogram.

We receive eight frames for every gesture. Each frame consists of 625 values. Figure 3 shows the point feature histogram of one frame of one gesture (bottom) and the corresponding point cloud (top). The resulting NumPy array after preprocessing has a shape of $(N, 12, 625)$.

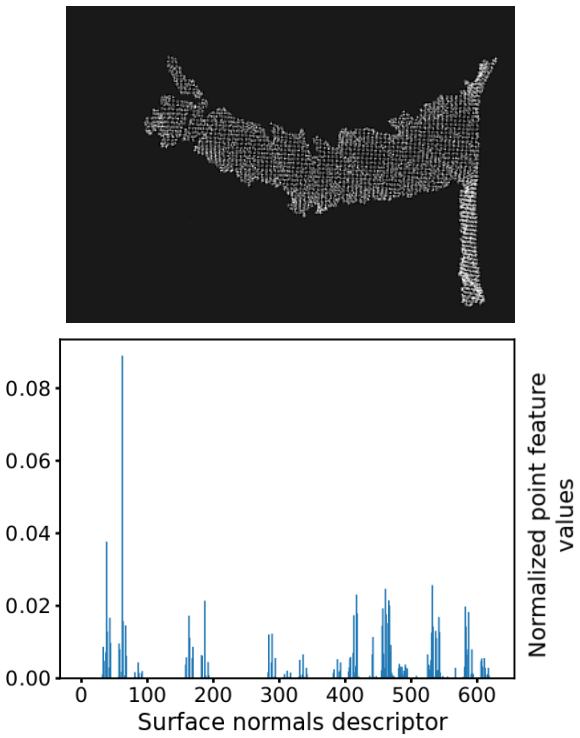

Figure 3: Example of a point feature histogram (bottom) corresponding to one frame of a "Thumbs Up" gesture (top: point cloud from which the histogram was computed).

3.5 IMD Data

To record the acceleration data, we use a 9-axis acceleration sensor (BWT901CL von Bitmotion) attached to the wrist of the user's hand. It can record 3-axis acceleration data and 3-axis yaw rates as well as gyroscopic and magnetic field measurements at a frequency of 200 Hz. We store a 7-tuple for each of the 400 measurements containing the timestamp, the acceleration data and the yaw rates. Preprocessing cleans the rather noisy signals: for this, we gather all $N = 20$ 7-tuples from each consecutive 200-millisecond window into a block and calculate statistical values for each entry except the timestamp, as shown in equations 1-3.

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

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

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

Thus, we receive ten descriptors ($200ms \cdot 10$ frames $= 2s$) for every gesture sample. Each descriptor consists of 18 values: three statistical values for each of the six axes. This results in a NumPy array for the preprocessed IMD data with the shape of $(N, 10, 3, 6)$.

3.6 Audio Data

To record the audio data, we use the audio stream provided by the Orbbec Astra 3D sensor. The sensor allows to record audio between 20 and 16,000 Hz and has a sensitivity of 30 dB.

For the entire length of the recording, we save the wave data and later down-sample it to a frequency of 8,000 Hz. To ensure each sample has the same length, we use zero-padding and randomly pick the offset for each sample.
Afterward, we compute the Short-Time Fourier Transform (Nasser, 2008) (STFT) for each data sample. Over a window of 455 data points with an overlap of 420 points, the STFT provides the frequency information and displays how much the frequency varies during that time frame. The result of this computation is 2D data in the shape of 181×182. The preprocessing step is based on data conversion conducted on the AudioMNIST dataset (Becker et al., 2018) and results in a NumPy array for the audio data with shape $(N, 182, 181, 1)$.
Figure 4 shows three examples of the plotted STFT data: one for a sample without any particular noise, one for a sample with one snapping sound, and one for a sample with two snapping sounds. This shows the differences between the classes that can be used by an algorithm to perform classification.

4 EXPERIMENTS

We provide uni-modal classification results as a benchmark for our dataset. Each experiment is repeated five times, and the average classification accuracy on a test set is reported. We train a distinct deep

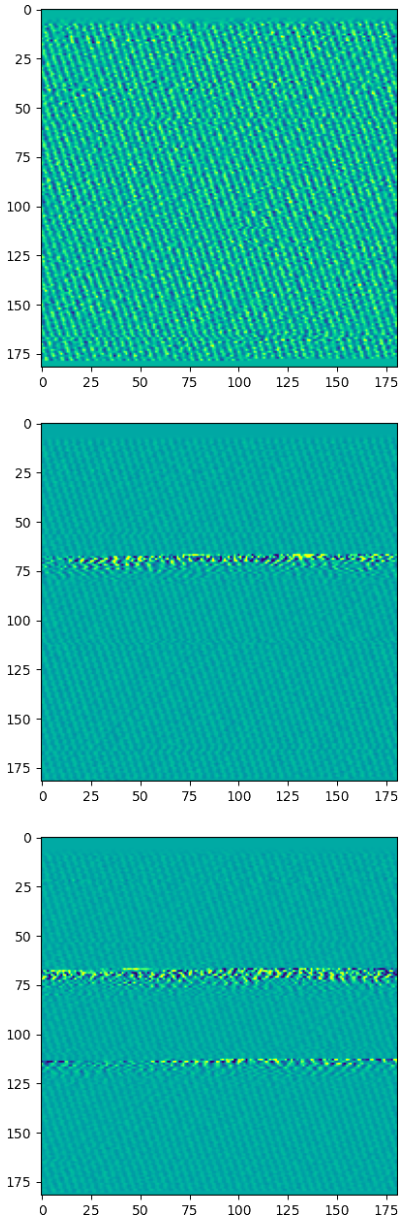

Figure 4: Examples of plotted STFT data. Sample without sound (top, class 2 - swipe left), with one snapping sound (middle, class 4 - snap once) and with two snapping sounds (bottom, class 5 - snap twice). STFT provides the frequency information and displays how much the frequency varies during a time frame.

LSTM network (Hochreiter and Schmidhuber, 1997) on preprocessed data for each modality. The recorded gestures from the dataset are randomly split into training and test sets at a proportion of 80:20 before training and subsequently used as training and test data in all uni-modal experiments.

In preliminary experiments, we identify the network parameters that result in the highest classification accuracy. We vary the learning rate ε, the number of hidden Layers L, the number of cells per layer S, the batch size b, and the number of iterations I. Then, we compare the resulting accuracy and use the parameters that achieved the best results for our experiments thus resulting in the network architectures used for our experiments as shown in Table 3.

Table 3: Architectures of the LSTM networks used for our experiments.

Modality	ε	L	S	b	I
Accel. Data	0.001	5	250	500	1,000
RGB Data	0.001	2	200	250	3,000
3D Data	0.001	2	250	1,000	5,000

Tables 4 and 5 show the results of our uni-modal experiments for acceleration data. It shows a unimodal gesture classification accuracy for the acceleration data at 84%.

Table 4: Confusion matrix for the unimodal gesture classification of acceleration data.

		Predicted class [0-5]					
Target [0-5]							
	3778	37	29	17	58	31
	772	3128	25	5	13	7
	54	408	3358	78	45	7
	24	11	489	3290	128	8
	39	4	25	242	3141	499
	12	1	3	14	794	3126

Table 5: Classification report for the unimodal gesture classification of acceleration data.

Class	**Precision**	**Recall**	**F1-Score**
0	0.81	0.96	0.88
1	0.87	0.79	0.83
2	0.85	0.85	0.85
3	0.90	0.83	087
4	0.75	0.80	0.77
5	0.85	0.79	0.82

Tables 6 and 7 show the results of our uni-modal experiments for RGB data, which shows that the gesture classification accuracy reaches 85%.

The results for the uni-modal experiments for 3D data can be seen in Tables 8 and 9. The gesture classification accuracy reaches 92% for this modality.

We are not using an LSTM network to classify audio data since the preprocessed data are not sequential but a single 2D image for each gesture (cf. Figure 4). Therefore, we use a Deep Convolutional Neural Network which is the state-of-the-art technique for image classification. We chose the Adam Optimizer and

Table 6: Confusion matrix for the uni-modal gesture classification of RGB data.

		Predicted class [0-5]				
Target [0-5]	3912	38	0	0	0	0
	0	3447	133	57	38	285
	0	247	3637	38	38	0
	0	114	95	3637	114	0
	0	247	0	38	2687	988
	0	475	0	57	437	2991

Table 7: Classification report for the unimodal gesture classification of RGB data.

Class	**Precision**	**Recall**	**F1-Score**
0	1.00	0.99	0.99
1	0.75	0.86	0.80
2	0.94	0.92	0.93
3	0.95	0.92	0.93
4	0.80	0.67	0.73
5	0.69	0.74	0.72

Table 8: Confusion matrix for the uni-modal gesture classification of 3D data.

		Predicted class [0-5]				
Target [0-5]	3950	0	0	0	0	0
	2	3928	6	12	2	0
	0	6	3900	44	0	0
	0	0	24	3922	4	0
	2	0	0	0	3488	460
	0	0	2	0	998	2950

Cross-Entropy Loss Function to train our Deep CNN in 10 epochs. The used CNN consists of the following eight layers:

1. A Convolutional layer with the input shape $(182, 181, 1)$, a filter size of 32, and a kernel size of $(3, 3)$.
2. A Max Pooling layer with a pooling size of $(2, 2)$.
3. A Convolutional layer with a filter size of 64, and a kernel size of $(3, 3)$.
4. A Max Pooling layer with a pooling size of $(2, 2)$.
5. A Convolutional layer with a filter size of 64, and a kernel size of $(3, 3)$.
6. A Reshaping Layer that flattens the input.
7. A densely connected Neural Network layer with 64 units.
8. A densely connected Neural Network layer with 10 units.

Tables 10 and 11 show the results of our unimodal experiments for audio data. It shows an unimodal gesture classification accuracy for audio data at 45%. Unsurprisingly, the data show a high recall for gestures 4 (One Snap) and 5 (Two Snaps) and a low recall for the other four gestures that do not depend on sound to be distinguished. The purpose of the audio modality lies in reinforcing predictions in combination with other modalities.

Table 9: Classification report for the unimodal gesture classification of 3D data.

Class	**Precision**	**Recall**	**F1-Score**
0	1.00	1.00	1.00
1	1.00	0.99	1.00
2	0.99	0.99	0.99
3	0.98	0.99	0.99
4	0.74	0.86	0.80
5	0.84	0.70	0.77

Table 10: Confusion matrix for the uni-modal gesture classification of audio data.

		Predicted class [0-5]				
Target [0-5]	772	251	304	2602	14	7
	678	339	319	2609	3	2
	690	251	394	2612	3	0
	717	280	266	2684	0	3
	126	79	61	89	3081	514
	13	9	15	6	568	3339

Table 11: Classification report for the uni-modal gesture classification of audio data.

Class	**Precision**	**Recall**	**F1-Score**
0	0.26	0.19	0.22
1	0.28	0.09	0.13
2	0.29	0.10	0.15
3	0.25	0.68	0.37
4	0.84	0.78	0.81
5	0.86	0.84	0.85

We therefore also investigated the effect of fusing different modalities using two commonly used late fusion methods: **max-conf**, where we use the most certain uni-modal class prediction as output, and **prob**, where we treat the uni-modal output layer predictions as independent conditional probability distributions for a class, multiply and renormalize them. We discard early fusion methods because the four sensory modalities have different numerical formats, arrive at different frequencies, and – in the case of audio data – are processed by a different network type.

Figure 5 shows the averaged results of our unimodal and multi-modal experiments.

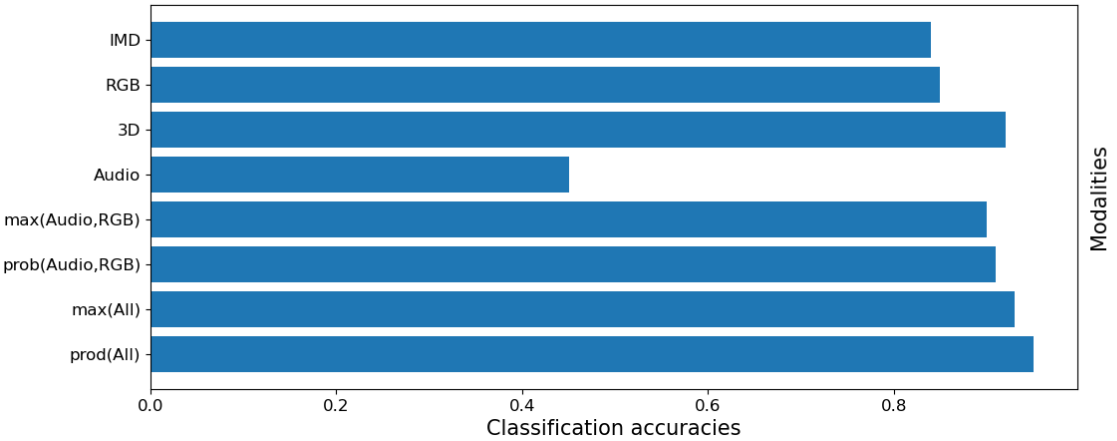

Figure 5: Gesture classification accuracies achieved by the uni-modal experiments as well as selected results for the two multi-modal fusion approaches max-conf and prob.

5 LIVE SYSTEM

We confirm that it is possible to train a real-time system on our dataset so that it can be used to correctly classify hand gestures done by different people. For this, we implement a live demonstrator based on the 3D modality. Our live system is split into two parts: One part is the implementation of an LSTM network as used for the experiments described in Section 4. The second part is an implementation for the Robot Operating System (ROS) that receives the sensor data and feeds it to the trained LSTM network.

5.1 Implementation Details

As a proof of concept, we only show the details for processing, training, and classifying 3D data. All other modalities can be handled analogously.

5.1.1 Point Cloud Processor

The ROS node responsible for processing the point clouds subscribes to the 3D camera sensor. The camera publishes `PointCloud2` messages at about 6 Hz to correspond to the number of frames in the dataset. Each message includes meta-information (i.e. height, width, step size) and the point cloud data itself. In this node, the point cloud is extracted from the message and then converted according to the preprocessing steps described in Section 3.4. Afterward, the PFH is published for further processing.

5.1.2 LSTM Classifier

The second ROS node contains the LSTM classifier. This node listens to PFH messages published by the Point Cloud Processor. Since gestures can start at any given moment in time, we use an approach called *Shifted Recognizer* (Schak and Gepperth, 2019): N identical classifiers or recognizers are run in parallel, each of them is trained with the same dataset consisting of gestures with a fixed length T which determines their Temporal Receptive Field (TRF). Each of them receives the same data from the Point Cloud Processor. However, the classifiers are delayed by $\Delta = \frac{T}{N}$ frames w.r.t. to the other classifiers, as shown in Figure 6, and they are reset after a full TRF. If there are enough classifiers, a gesture of length $l \leq T$ will correlate with the TRF of a single classifier which will then classify and report it.

In our live system, we use $N = 12$ LSTM classifiers. Since the number of parallel classifiers equals the number of frames per gesture, a gesture will always correlate to exactly one classifier and there will be neither an onset nor an offset to the gesture in that classifiers TRF.

Each node follows the same steps: First, it receives the PFH message and converts it to a Numpy array which is then fed into the LSTM model. Then, it publishes the classification result as a message. Each LSTM Classifier node predicts a gesture class for every frame they receive and publishes the result. All predictions are further handled by the Aggregator.

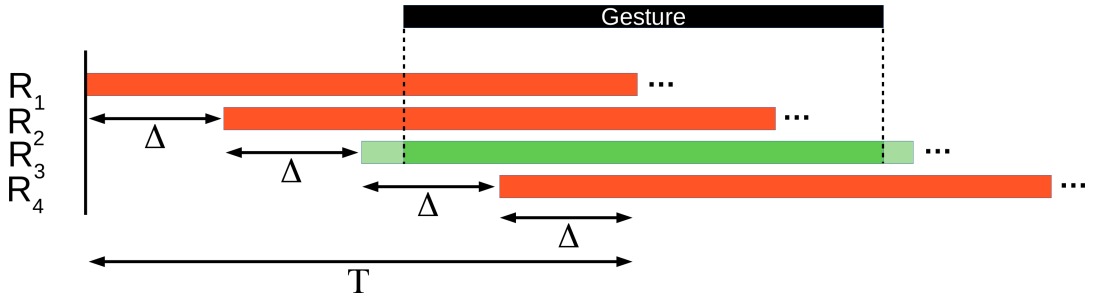

Figure 6: Example for our *Shifted Recognizer* approach with $\Delta = \frac{T}{4}$ frames. The temporal receptive fields (TRFs) of each recognizer ($R_{1...4}$ are indicated by red or green bars, while the received PFH of a conducted gesture is indicated by a black bar. One recognizers TRF will correlate with the gesture and therefore classify the gesture correctly (green bar).

Table 12: Results for our experiments on gesture recognition with a live system trained on the MMHG dataset. P_i notates the ratio of correct classifications for the i-th user with $i \in \{1,2,3,4\}$, whereas C_k notates the k-th class as described in Section 3.2.

	P_1	P_2	P_3	P_4	Σ
C_0	5/5	1/1	1/1	3/3	100%
C_1	5/5	1/1	1/1	2/3	90%
C_2	5/5	1/1	0/1	2/3	80%
C_3	2/5	0/1	1/1	1/3	40%
C_4	5/5	1/1	1/1	2/3	90%
C_5	0/5	0/1	0/1	1/3	10%
Σ	73.3%	66.7%	66.7%	61.1%	66.7%

5.1.3 Aggregator

The Aggregator node collects and aggregates the predictions from the twelve LSTM Classifier nodes. The prediction with the highest prediction score will be used. The result, i.e., the predicted gesture class, is only presented to the user if the following conditions apply: first of all, the prediction score must be above a predefined threshold. In addition, the predicted class of the LSTM node with the highest prediction score must be the same for the last three frames. Otherwise, no prediction is given. Thus, we ensure that there is no prediction if the user is not performing a gesture.

5.2 Experiments

To prove that the described system can correctly classify gestures performed by persons which it was not trained by, we conducted a series of experiments. Four persons performed each gesture multiple times, and the number of correct classifications was recorded. There were two female and two male users, with differing hand sizes and skin colors to present as much variation as possible. Every user got instructed on how to correctly perform the gestures. The results are shown in Table 12.

Since we only discuss results for depth data exemplarily, it is not surprising that the prediction for Two Snaps does not work. When performing Two Snaps, the system always predicts One Snap because it does not consider sound, and the depth data is very similar for both classes. Also, Swipe Left often gets confused with Thumbs Up since the angle and movement of the hand can be pretty similar. Including IMD data could possibly increase the number of correct classifications for this class. Recapitulatory, our experiments show that our live system trained with gestures performed by just a single person can achieve an acceptable classification accuracy for gestures performed by different users.

6 SUMMARY AND CONCLUSION

In this article, we provide an in-depth description of the new publicly available MMHG dataset, as well as the reasoning behind the design of the gesture classes, which is to support large-scale experiments on multi-modal data fusion. We support the suitability of this dataset for fusion purposes by conducting experiments using (admittedly very simple) late-fusion strategies and state-of-the-art sequence classification methods like LSTM and CNN networks. These experiments show that, within the limits of statistical accuracy, fusion with one or more other modalities does improve the quality of uni-modal gesture recognition. Notably, the audio modality, which by itself achieves only very disappointing accuracies, can give a strong boost when fused with others, since some gesture classes are best characterized by audio-only. Also, we show that our dataset can be used to train a system to correctly classify hand gestures performed by other users. Lastly, we have shown that multi-modal gesture recognition is possible using techniques that are real-time capable on off-the-shelf hardware.

In future research we will conduct experiments with more individuals to estimate the bias in recognition due to the single subject in our dataset. Also, we will record more data – also with other subjects – and update the dataset over time. Lastly, we will perform further research and conduct experiments using probabilistic models for multi-modal sequence classification, outlier detection and sampling.

REFERENCES

Becker, S., Ackermann, M., Lapuschkin, S., Müller, K.-R., and Samek, W. (2018). Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals.

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015). 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)*, pages 168–172.

D. Lara, O. and Labrador, M. (2013). A Survey on Human Activity Recognition Using Wearable Sensors. *Communications Surveys & Tutorials, IEEE*, 15:1192–1209.

Escalera, S., Gonzàlez, J., Baró, X., Reyes, M., Guyon, I., Athitsos, V., Escalante, H., Sigal, L., Argyros, A., Sminchisescu, C., Bowden, R., and Sclaroff, S. (2013). Chalearn multi-modal gesture recognition 2013: grand challenge and workshop summary. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 365–368.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Kim, T.-K. and Cipolla, R. (2008). Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1415–1428.

Kopuklu, O., Rong, Y., and Rigoll, G. (2019). Talking with your hands: Scaling hand gestures and recognition with cnns. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*.

Liu, L. and Shao, L. (2013). Learning discriminative representations from rgb-d video data. In *Twenty-third international joint conference on artificial intelligence*.

Marin, G., Dominio, F., and Zanuttigh, P. (2016). Hand Gesture Recognition with Jointly Calibrated Leap Motion and Depth Sensor. *Multimedia Tools Appl.*, 75(22):14991–15015.

McConnell, R. (1986). Method of and apparatus for pattern recognition.

Memo, A., Minto, L., and Zanuttigh, P. (2015). Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition. In Giachetti, A., Biasotti, S., and Tarini, M., editors, *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association.

Nasser, K. (2008). Digital signal processing system design: Labview based hybrid programming.

Ni, B., Wang, G., and Moulin, P. (2011). RGBD-HuDaAct: A Color-Depth Video Database For Human Daily Activity Recognition. *International Conference on Computer Vision Workshops, IEEE*, pages 1147–1153.

Radu, V., Tong, C., Bhattacharya, S., Lane, N. D., Mascolo, C., Marina, M. K., and Kawsar, F. (2018). Multimodal Deep Learning for Activity and Context Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4):157:1–157:27.

Ranasinghe, S., Machot, F. A., and Mayr, H. C. (2016). A review on applications of activity recognition systems with regard to performance and evaluation. *International Journal of Distributed Sensor Networks*, 12(8):1550147716665520.

Romdhane, R., Crispim-Junior, C. F., Bremond, F., and Thonnat, M. (2013). Activity Recognition and Uncertain Knowledge in Video Scenes. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, Krakow, Poland.

Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M. (2008). Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391. IEEE.

Sachara, F., Kopinski, T., Gepperth, A., and Handmann, U. (2017). 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)*, pages 959–964.

Sarkar, A., Gepperth, A., Handmann, U., and Kopinski, T. (2017). Dynamic hand gesture recognition for mobile systems using deep lstm. In Horain, P., Achard, C., and Mallem, M., editors, *Intelligent Human Computer Interaction*, pages 19–31, Cham. Springer International Publishing.

Schak, M. and Gepperth, A. (2019). Robustness of deep lstm networks in freehand gesture recognition. In *Artificial Neural Networks and Machine Learning – ICANN 2019: Image Processing*, pages 330–343. Springer International Publishing.

Sharma, S., Kiros, R., and Salakhutdinov, R. (2016). Action Recognition using Visual Attention. *ICLR*.

Wan, J., Zhao, Y., Zhou, S., Guyon, I., Escalera, S., and Li, S. Z. (2016). 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*, pages 56–64.

William T. Freeman, M. R. (1994). Orientation histograms for hand gesture recognition. Technical Report TR94-03, MERL - Mitsubishi Electric Research Laboratories, Cambridge, MA 02139.

Zhang, M. and Sawchuk, A. A. (2012). USC-HAD: A Daily Activity Dataset for Ubiquitous Acitivity Recognition Using Wearable Sensors. *International Conference on Ubiquitous Computing*, pages 1036–1043.

Zhang, Y., Cao, C., Cheng, J., and Lu, H. (2018). EgoGesture: A New Dataset and Benchmark for Egocentric

Hand Gesture Recognition. *IEEE Transactions on Multimedia*, 20(5):1038–1050.