Gesture MNIST: A New Free-Hand Gesture Dataset

Monika Schak[1] and Alexander Gepperth[1]

Fulda University of Applied Sciences, Leizpiger Str. 123, 36037 Fulda, Germany
{monika.schak, alexander.gepperth}@cs.hs-fulda.de

Abstract. We present a unimodal, comprehensive, and easy-to-use dataset for visual free-hand gesture recognition. We call it GestureMNIST because of the 28×28 grayscale format of its images, and because the number of samples is approximately 80,000, similar to MNIST. Each of the six gesture classes is composed of a sequence of 12 images taken by a 3D camera. As a peculiarity w.r.t. other datasets, all sequences are recorded by a single person, ensuring high sample uniformity and quality. A particular focus is to provide a vision-based dataset that can be used "out of the box" for sequence classification without any preprocessing, segmentation, and feature extraction steps. We present classification experiments on GestureMNIST with different types of DNNs, establishing a performance baseline for sequence classification algorithms. We place particular emphasis on ahead-of-time classification, i.e., the correct identification of a gestures class *before* the gesture is completed. It is shown that CNN and LSTM-based deep learning achieves near-perfect performance, whereas ahead-of-time classification performance offers ample scope for future research with GestureMNIST. GestureMNIST contains visual samples only, but other modalities, namely acceleration and sound data, are available upon request.

Keywords: Hand Gesture · Dataset · Sequence Classification · LSTM · CNN · Outlier Detection

1 Introduction

This work is in the context of free-hand gesture recognition, a field of machine learning that has profound application relevance in, e.g., human-machine-interaction (HMI). More generally, we target the wider field of *sequence classification*, where data samples consist of several elements that are presented one after the other. Typical sequence classifiers are given by Hidden Markov models (HMM) often used in speech classification. Recurrent neural networks (RNNs) have a long tradition in this domain as well. With the advances in Deep Learning, deep recurrent neural networks have been proposed, perhaps most prominently represented by bi-directional LSTM networks [3] which reach state-of-the-art performance in several application domains.

RNNs offer the intriguing possibility to obtain a decision before a sequence has been completely presented, which we denote as **ahead-of-time** classification.

Especially in free-hand gesture recognition, this ability seems crucial for seamless and intuitive interaction with humans.

Successful hand gesture recognition requires datasets that are large, diverse, and reliable. These requirements are partially conflicting since large and diverse datasets are usually ensured by involving many different persons. Although this promotes sample diversity, this diversity is, partially at least, an undesirable one, since each person performing the gestures needs to learn how to perform them correctly. As a consequence, the recorded data will contain many gestures that are inconsistent with others or even plain inappropriate. Thus, the **sample quality and quantity** of a dataset plays an important role in successfully training gesture classifiers.

Free-hand gesture recognition is often performed on RGB images, which requires extensive **image pre-processing** and **feature extraction** techniques to be applied. In particular, these steps are necessary if robustness to illumination and background is a goal.

Lastly, the acquisition of a **sufficiently large number** of gesture samples is a tedious and expensive process, which maybe explains why most public gesture recognition datasets are rather small as compared to image classification benchmarks.

1.1 Related Work

Hand gesture recognition is a long-standing subject of academic and industrial interest and thus has a long history in machine learning, please see [10, 5, 14] for surveys. Surprisingly, the number of large-scale public datasets is rather low. Concretely, the nvGesture dataset [8] contains 1,532 samples grouped in 25 categories ($\sim$ 60 samples per category). The EgoGesture dataset [15] contains $\sim 24,000$ samples in 83 classes ($\sim$ 300 samples/class). Lastly, the ChaLearn ISO/ConGD 2016 datasets [13] contains $\sim 47,000$ gesture samples grouped into 249 classes, or $\sim$ 200 samples per class. In most of these datasets, the emphasis is on realistic settings, so background, clutter, and subject diversity form an integral part of the problems addressed in these work, namely robustness and subject invariance. On the other hand, given the low number of gesture samples per class, it may be asked whether this is actually sufficient for training DNNs, which require a large number of data samples for training. All of these datasets (except [15]) require significant pre-processing of images since the full background is included and no foreground/background segmentation is provided.

1.2 Contribution

We present GestureMNIST, a large, publicly available dataset of high-quality visual hand gesture samples (see Figure 1 for a visualization of typical samples). All gestures are recorded from a single person to ensure uniformity and sample quality. Two of the gesture classes are very similar visually ("one snap" and "two snaps"), which poses strong challenges on ahead-of-time classification. We place great emphasis on providing a sufficient amount of gesture samples

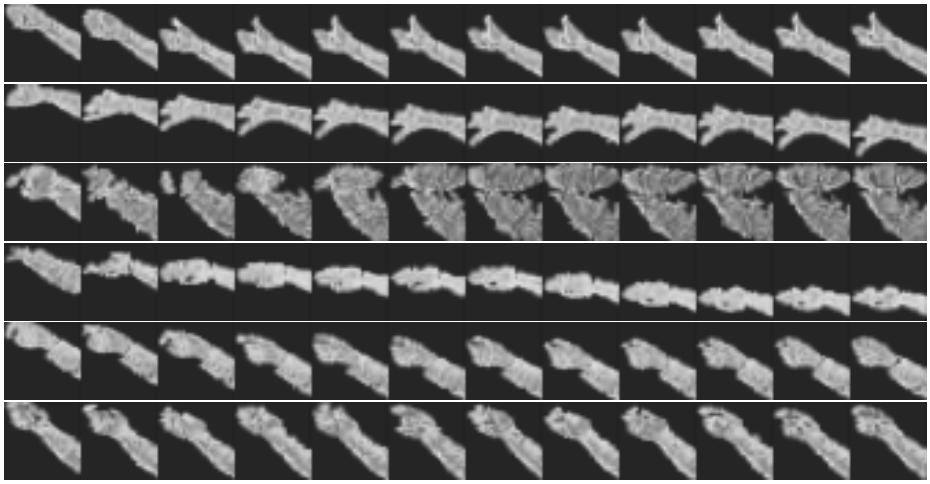


Fig. 1: Samples taken from each GestureMNIST class. From top to bottom: Thumbs up, Thumbs down, Swipe Left, Swipe Right, One Snap, Two Snaps.

per class (∼13,000) to enable efficient training of DNN models. The hand has already been segmented and the background has been removed, so the dataset can be directly used for machine learning, without having to resort to complex image processing pipelines. We describe experiments with deep CNN and LSTM sequence classifiers that establish a performance baseline for this new dataset. As a particular focus of the examination of LSTM classifiers, we investigate ahead-of-time classification for different time points in the sequence.

2 Dataset

Gesture MNIST is an MNIST-like [7] dataset of six free-hand gestures, consisting of 79,881 samples. Each sample is a sequence of twelve 28×28 grayscale images. All gestures are performed by a single person to ensure carefully curated data with little to no errors in how the correct gesture for each class is performed.

The samples for this dataset are recorded in a fixed setting to ensure that the hand gesture is always performed in a predefined volume of interest. For this reason, we built a setup: We bolted the camera to a board and marked the area in which to conduct the gesture. This setup can be seen in Figure 2.

Every recording consists of ten repetitions r of one gesture before continuing to the next gesture class c. Therefore, each recording produces $r \cdot c = 10 \cdot 6 = 60$ samples. The samples are simultaneously assigned class labels while saving them to the disk. For this dataset, we only used the 3D point clouds obtained by an Orbbec Astra 3D sensor. In fact, we also record three other modalities – RGB images, audio, and acceleration data. Since this is supposed to be an MNIST-like dataset for benchmarking uni-modal sequence detection models, those are not

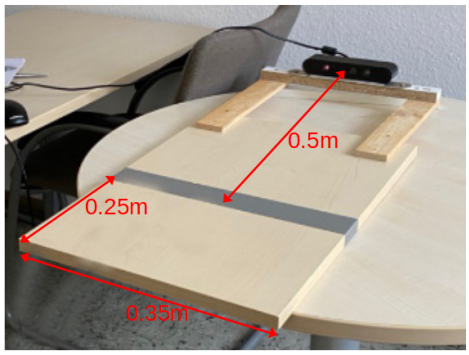


Fig. 2: The fixed setup to record hand gestures for our gesture dataset.

used here but are available upon request for further research on multi-modal models.

The Gesture MNIST dataset consists of six classes C_i with $i = [1, 6]$ as follows: Thumbs Up, Thumbs Down, Swipe Left, Swipe Right, One Snap, Two Snaps. Both swiping gestures are performed with the whole hand instead of one or multiple fingers. The snapping gestures focus on the thumb and middle finger to make one or two snapping sounds, respectively.

To record our data, we use the stream of depth images provided by an Orbbec Astra 3D sensor. The depth images have a size of 640×480 pixels and are converted to point clouds, then stored. Each gesture lasts for two seconds. During this time frame, we receive a total of twelve depth images. Therefore, the length of each sample is twelve frames.

After recording the gestures, we conducted an automated preprocessing step. At first, we downsample the point cloud. We create a 3D-voxel grid over the input data. Then, we compute the centroid of all the points in that voxel and use this to represent the voxel. Thus, we reduce the size to lower computational costs. In the next step, we crop the point cloud to a predefined volume of interest to remove unnecessary data. That is the reason why performing the gesture in a predefined area during recording is so important. By performing a Principal Component Analysis [12], we determine which points belong to the hand and remove all others. After these steps, we only keep the downsampled points that describe the hand performing the gesture.

Afterward, we project these points onto a 2D plane and remove all color information. Thus, we receive a grayscale image of just the hand. This image is further processed: We resize the image to 28×28 pixels and invert the colors to correspond to the MNIST data format and style. A randomly picked sequence for each gesture class is shown in Figure 1.

In total, the Gesture MNIST dataset contains approximately 13,300 recordings of each gesture class, totaling almost 80,000 samples. Table 1 shows the exact distribution of each class. All gestures are performed by a single person to ensure a consistently high quality of the data. Since this person is well-instructed

and experienced, mistakes or incorrect gestures are very unlikely to happen. That is not necessarily a sign of invariability since all background and size data are removed during preprocessing and there are strict guidelines on how to perform a gesture to ensure consistency. Experiments with a live demonstrator [11] show that an LSTM model trained on our dataset is able to correctly classify most gestures performed by users that are not the one who recorded the dataset.

Table 1: Distribution of the six gesture classes in the Gesture MNIST 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**

The dataset will be publicly available to be used for research at the following website: `http://data.informatik.hs-fulda.de/`. We provide the data in dependence on the format given by the well-known MNIST dataset. The data are available in a Numpy-Array of shape $(N, 12, 28, 28)$, where $N = 79,881$ is the total number of samples, 12 is the number of frames per sequence, and 28×28 is the size of each frame. The labels are also available in a Numpy-Array of shape $(N, 6)$ in one-hot-encoded format.

3 Experiments

We conduct benchmark experiments with state-of-the art classification networks for sequence detection: a deep Long Short-Term Memory (LSTM) network [4] and a deep Convolutional Neural Network (CNN) [6].

3.1 LSTM Network

By using preliminary experiments to establish network parameters that lead to the highest classification accuracy, we choose a deep LSTM network with 5 hidden layers, 800 cells per layer, a learning rate of 0.001, a batch size of 1,000, and we run 1,000 iterations. We train the network on 80% randomly picked samples and test the performance on the remaining 20%.

After training the network, we get the predictions on the test data for each frame of the sequence to receive the gesture classification accuracy for the whole gesture along with the gesture classification accuracy for ahead-of-time classification.

Figure 3 shows the gesture classification accuracy at each frame in a graph. It can be seen that after five frames the accuracy is already over 50%. After two-thirds of the gesture has been processed, the correct class can be predicted with an accuracy of over 80%. And, after nine of twelve frames, we achieve a

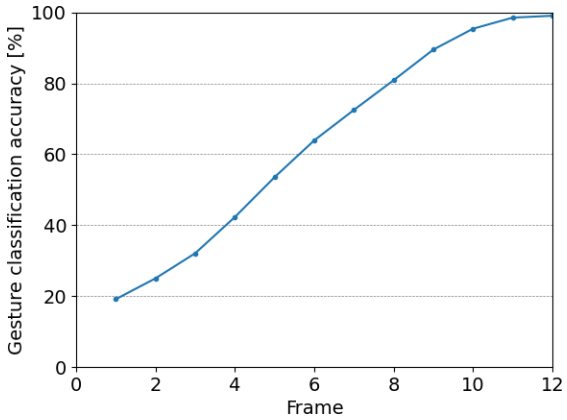


Fig. 3: Gesture classification accuracies on our test data [in %] for the ahead-of-time classification at each of the twelve frames.

classification accuracy of almost 90%. After processing the whole gesture, we can predict the correct gesture class with an accuracy of 99,04%.

Table 2 shows the confusion matrix after half the gesture has been processed (frame 6 of 12). Here, we achieve classification accuracies of 84%, 86%, 79%, 66%, 10% and 63% for the six classes. Therefore, it is visible that the classification of class 5 (Snap Once) seems to be most difficult and requires more frames for reliable classification.

Table 2: Confusion matrix for the ahead-of-time classification at frame 6 of 12.

		Predicted class [1-6]					
		1,987	0	0	27	726	0
		4	2,032	0	618	18	5
Target [1-6]		7	0	1,796	370	356	91
		4	0	103	1,805	738	38
		0	1	1	0	259	2,325
		0	0	1	1	330	2,333

Table 3 shows the confusion matrix after eight of twelve frames have been processed. Now, the first four classes achieve classification accuracies of over 90%, while classes five and six only achieve accuracies of 34% and 65% respectively. This is not surprising, since those two classes are specifically designed to be very difficult to distinguish. Adding additional modalities like acceleration data or sound can help improve the classification.

Finally, Table 4 shows the confusion matrix after all twelve frames have been processed. After being able to see the whole gesture, the LSTM model

Table 3: Confusion matrix for the ahead-of-time classification at frame 8 of 12.

		Predicted class [1-6]					
		2,586	0	0	8	146	0
		1	2,579	0	96	0	1
Target [1-6]		4	0	2,478	82	52	4
		0	0	79	2,365	213	31
		0	1	0	0	716	1,869
		0	0	0	1	468	2,196

achieves an average classification accuracy over all gesture classes of 99.04%. Still, the accuracy for the first four classes is marginally higher than the gesture classification accuracy achieved on the two snapping gestures. Overall, it can be said that an LSTM model can achieve near-perfect results on the Gesture MNIST dataset when classifying the whole gesture. Ahead-of-time classification still requires further research to improve results.

Table 4: Confusion matrix for the gesture classification after all twelve frames of the gesture have been processed.

		Predicted class [1-6]					
		2,739	0	0	1	0	0
		0	2,673	0	4	0	0
Target [1-6]		0	0	2,603	16	0	1
		0	0	6	2,682	0	0
		0	1	0	0	2,538	47
		0	1	0	1	75	2,589

3.2 CNN

Since a Convolutional Neural Network is not specifically designed to classify sequential data we concatenate the twelve frames of each Gesture MNIST sample to create one big image of size $28 \times 28 \cdot 12 = 28 \times 336$ pixels comparable to the ones shown in Figure 1. We choose a standard Deep CNN architecture consisting of the following 17 layers: 3 Conv2D layers, 4 ReLU layers, 3 Max Pooling layers, 4 Dropout layers, 1 Flatten layer, and 2 Dense layers. Further information about each layer and how the model is designed is shown in Figure 4.

We train the model with a batch size of 64 for a total of 10 epochs. 80% randomly picked samples from the Gesture MNIST dataset are used for training while the remaining 20% are used to validate the model and evaluate the perfor-

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 336, 32)	320
leaky_re_lu_1 (LeakyReLU)	(None, 28, 336, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 14, 168, 32)	0
dropout_1 (Dropout)	(None, 14, 168, 32)	0
conv2d_2 (Conv2D)	(None, 14, 168, 64)	18496
leaky_re_lu_2 (LeakyReLU)	(None, 14, 168, 64)	0
max_pooling2d_2 (MaxPooling2	(None, 7, 84, 64)	0
dropout_2 (Dropout)	(None, 7, 84, 64)	0
conv2d_3 (Conv2D)	(None, 7, 84, 128)	73856
leaky_re_lu_3 (LeakyReLU)	(None, 7, 84, 128)	0
max_pooling2d_3 (MaxPooling2	(None, 4, 42, 128)	0
dropout_3 (Dropout)	(None, 4, 42, 128)	0
flatten_1 (Flatten)	(None, 21504)	0
dense_1 (Dense)	(None, 128)	2752640
leaky_re_lu_4 (LeakyReLU)	(None, 128)	0
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 6)	774

Total params: 2,846,086
Trainable params: 2,846,086
Non-trainable params: 0

Fig. 4: Description of the 17 layers of the CNN model used for our experiments including layer type, output shape and number of parameters.

mance. Since all frames are fed into the network as one big image, ahead-of-time classification is not possible in this case.

After training the deep Convolutional Neural Network model, we achieve a gesture classification accuracy of 99.61% on the test set. The training and test accuracies during the ten epochs are shown in Figure 5. It can be seen that the test accuracy is already very high after just one epoch of training and then hardly improves. Therefore, it can be said that a CNN model adapts to the GestureMNIST dataset really fast which can reduce computational costs for training the network.

Tables 5 and 6 show the confusion matrix, and the classification report for the test data after the training process is finished. A nearly perfect gesture classification accuracy can be achieved which is not surprising since deep convolutional neural networks are specifically designed to classify images. It can also be seen that the most difficulties – albeit they are not really significant either – happen with gestures from the last two gesture classes: Snapping once or twice. This, as explained above already, is due to the fact of their nature to be designed to add some challenge when only classifying visual modalities.

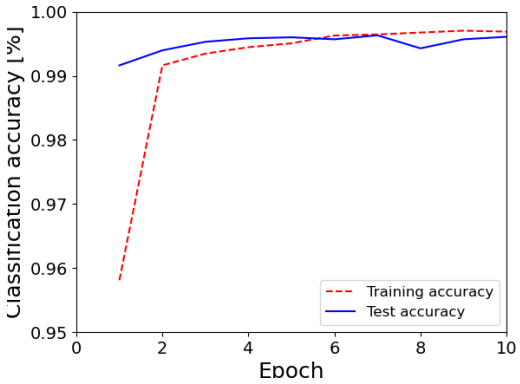


Fig. 5: Gesture classification accuracy for each of the ten epochs during training, comparing the accuracy on the training data and the testing data.

Table 5: Confusion matrix for the gesture classification on our test set using a deep CNN model trained on GestureMNIST.

		Predicted class [1-6]					
		2,710	0	0	0	0	1
		0	2,617	0	0	0	0
Target [1-6]		0	0	2,627	12	0	0
		0	0	3	2,687	0	0
		0	0	0	4	2,636	14
		0	0	0	0	30	2,635

4 Outlier Detection

Outlier detection arguably constitutes another important functionality in the context of gesture recognition, since relevant gestures are often embedded into a continuous stream of non-gestures, or else there may be irrelevant frames before and after a meaningful gesture that need to be ignored. In addition, there should be an additional safeguard against spurious gestures or adversarial attacks, where unknown or absurd gestures may be used to confident classification results. Outlier detection usually relies on unsupervised methods such as Gaussian Mixture Models (GMMs) or k-means (which is really an approximation to GMMs). Here, we report results for GMMs that are fed entire sequences in concatenated form. Based on a thresholding operation performed on the returned score, a gesture is classified as an inlier or an outlier. Notably, GMM training is performed on inliers only.

Table 6: Classification report for the gesture classification on our test set using a deep CNN model trained on GestureMNIST.

Class	Precision	Recall	Accuracy	Support
1	1.00	1.00	1.00	2,711
2	1.00	1.00	1.00	2,617
3	1.00	1.00	1.00	2,639
4	0.99	1.00	1.00	2,690
5	0.99	0.99	0.99	2,654
6	0.99	0.99	0.99	2,665

4.1 Gaussian Mixture Models (GMMs)

GMMs [1] are unsupervised generative models that directly model the data distribution, which is represented as a weighted mixture of K multi-variate Gaussian densities $\mathcal{N}(\vec{x}; \Sigma_k, \vec{\mu}_k) \equiv \mathcal{N}_k(\vec{x})$, each of which is parameterized by a centroid $\vec{\mu}_k$ and a covariance matrix Σ_k. GMM training aims at maximizing the *log-likelihood* $\mathcal{L}$ of the data under the model, with:

$$\mathcal{L} = \sum_n \log \sum_k \pi_k \mathcal{N}_k(\vec{x}_n). \tag{1}$$

The vector $\vec{\pi}$ represents the mixture weights, which are adapted together with the set of all centroids $\{\vec{\mu}_k\}$ and covariance matrices $\{\Sigma_k\}$.

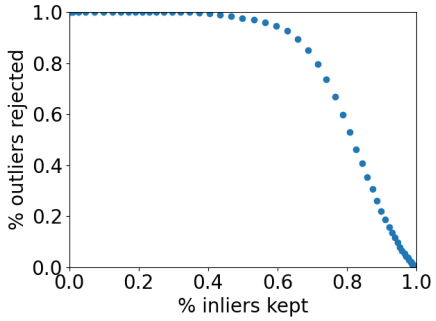
Once a GMM has been trained on data, the log-likelihood computed from a single inlier or outlier sample is taken to be a measure of the GMMs familiarity with that sample. Consequently, the sample is classified as an inlier if $\mathcal{L}(\vec{x}) \geq \theta_{\text{GMM}}$. For each value of this threshold, we can now compute the percentage p_I of inliers (in a test set) that would be accepted as inliers, and a corresponding percentage p_O of outliers that are rejected. By plotting pairs of $p_I(\theta_{\text{GMM}}), p_O(\theta_{\text{GMM}})$ into a 2D plot while varying the threshold θ_{GMM}, we obtain receiver-operator-characteristics (ROCs) as shown in Fig. 6a and Fig. 6b.

4.2 Results

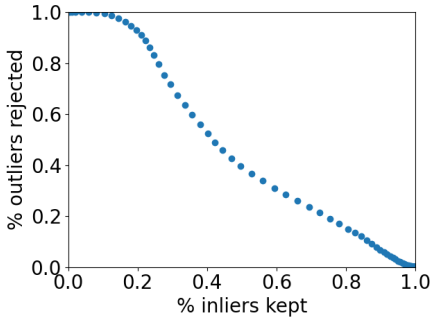
We train a GMM with $K = 100$ mixture components on all classes but one and then perform outlier detection using the remaining class, which, by definition, contains outliers only. The GMM is trained for 10 epochs by SGD using the procedure and the default parameters given in [2]. Table 7 shows the results for these experiments. As can be seen, the best results were achieved performing outlier detection on class 4 (Swipe Right) with an AUC of 0.802. The corresponding ROC can be seen in Figure 6a. The lowest results were achieved performing outlier detection on class 5 (One Snap) with an AUC of 0.480. The corresponding ROC curve can be viewed in Fig. 6b.

Table 7: Outlier detection results for each class on a GMM trained on the other five classes.

T1	T2	AUC
$\{2,3,4,5,6\}$	1	0.613
$\{1,3,4,5,6\}$	2	0.758
$\{1,2,4,5,6\}$	3	0.772
$\{1,2,3,5,6\}$	4	0.802
$\{1,2,3,4,6\}$	5	0.480
$\{1,2,3,4,5\}$	6	0.592



(a) ROC curve for outlier detection with $T1 = \{1,2,3,5,6\}$ and $T2 = 4$. The AUC is 0.802.



(b) ROC curve for outlier detection with $T1 = \{1,2,3,4,6\}$ and $T2 = 5$. The AUC is 0.480.

5 Discussion and conclusion

Summarizing the presented experiments, we observe that Gesture MNIST can be used "out of the box" for machine learning, without requiring any pre-processing or feature extraction steps. Sequence classification performance of LSTM and CNN-based sequence classifiers is high, indicating that Gesture MNIST is a rather easy classification problem. This does not impair its value as a benchmark, since ahead-of-time classification remains unsatisfactory, and other applications such as outlier detection, video modeling, or continual learning (see, e.g., [9]) can be performed relying on Gesture MNIST. We emphasize again the value of a visual sequence classification dataset that contains a large number of high-quality samples per class, and in which variability is mainly contributed by the intrinsic differences between the classes, in contrast to background variability, inconsistently performed gestures, and differences in gesture onset.

References

1. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B **39**, 1–38 (1977), http://web.mit.edu/6.435/www/Dempster77.pdf
2. Gepperth, A., Pfülb, B.: Image modeling with deep convolutional gaussian mixture models. In: International Joint Conference on Neural Networks(IJCNN) (2021)
3. Graves, A., Jaitly, N., Mohamed, A.r.: Hybrid speech recognition with deep bidirectional lstm. In: 2013 IEEE workshop on automatic speech recognition and understanding. pp. 273–278. IEEE (2013)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**, 1735–80 (12 1997). https://doi.org/10.1162/neco.1997.9.8.1735
5. Khan, R.Z., Ibraheem, N.A.: Hand gesture recognition: a literature review. International journal of artificial Intelligence & Applications **3**(4), 161 (2012)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 25. Curran Associates, Inc. (2012)
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
8. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J.: Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4207–4215 (2016)
9. Pfülb, B., Gepperth, A.: A comprehensive, application-oriented study of catastrophic forgetting in DNNs. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=BkloRs0qK7
10. Rautaray, S.S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. Artificial intelligence review **43**(1), 1–54 (2015)
11. Schak, M., Gepperth, A.: Gesture recognition on a new multi-modal hand gesture dataset. In: ICPRAM (2022)
12. Tharwat, A.: Principal component analysis - a tutorial. International Journal of Applied Pattern Recognition **3**(3), 197–240 (2016). https://doi.org/10.1504/IJAPR.2016.079733, https://www.inderscienceonline.com/doi/abs/10.1504/IJAPR.2016.079733, pMID: 79733
13. 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)
14. Zhang, Y., Cao, C., Cheng, J., Lu, H.: Egogesture: a new dataset and benchmark for egocentric hand gesture recognition. IEEE Transactions on Multimedia **20**(5), 1038–1050 (2018)
15. Zhang, Y., Cao, C., Cheng, J., Lu, H.: Egogesture: A new dataset and benchmark for egocentric hand gesture recognition. IEEE Transactions on Multimedia **20**(5), 1038–1050 (2018). https://doi.org/10.1109/TMM.2018.2808769