Two Phase Classification for Early Hand Gesture Recognition in 3D Top View Data

Aditya Tewari[1,2(✉)], Bertram Taetz[1], Frederic Grandidier[2], and Didier Stricker[1]

[1] Augmented Vision, Technische Universität Kaiserslautern,
Kaiserslautern, Germany
aditya.tewari@dfki.de
[2] IEE S.A., Contern, Luxembourg

Abstract. This work classifies top-view hand-gestures observed by a Time of Flight (ToF) camera using Long Short-Term Memory (LSTM) architecture of neural networks. We demonstrate a performance improvement by a two-phase classification. Therefore we reduce the number of classes to be separated in each phase and combine the output probabilities. The modified system architecture achieves an average cross-validation accuracy of 90.75% on a 9-gesture dataset. This is demonstrated to be an improvement over the single all-class LSTM approach. The networks are trained to predict the class-label continuously during the sequence. A frame-based gesture prediction, using accumulated gesture probabilities per frame of the video sequence, is introduced. This eliminates the latency due to prediction of gesture at the end of the sequence as is usually the case with majority voting based methods.

Keywords: Driver assistance · Hand gesture · LSTM networks · Hand features · Neural networks

1 Introduction

The touch and tactile based systems in cars cause visual distraction which affects the attention while driving [1]. The work by [2] shows that simple and natural interactions with multimedia devices in cars improve the driver's safety. [3] has compared various in-vehicular interaction systems and reported that the gesture based interaction requires least eye contact. Work by [4] also shows that the performance of the driver can degrade sharply with small increase in the shift of attention. It can thus be argued that a robust, touch-sensor free gesture based interactions improve driver safety.

The vision based Hand Gesture Recognition (HGR) techniques can be distributed into two broad classes, static and dynamic. The first [5,6] only recognizes a static pose of a hand while the second uses the changing hand pose and hand

This work is supported by the National Research Fund, Luxembourg, under the AFR project 7019190.

G. Bebis et al. (Eds.): ISVC 2016, Part I, LNCS 10072, pp. 353–363, 2016.
DOI: 10.1007/978-3-319-50835-1_33

motion over frames in addition. The later scheme supports a potentially larger and more natural set of gestures. The primary challenge for an HGR system is the rapidly changing global illumination. Further, defining an optimal location for a camera that minimises the palm occlusion is a difficult task. It has been observed that an overhead location is best suited for such problems [7] because it minimises occlusion due to objects inside the car, however the self occlusion of the hand remains significant especially when a gesture is performed with vertically downward pointing palm. It is desirable to have a flexible system which can be modified to identify gestures which were not originally built into it. The problem of illumination is suppressed by the choice of sensor, on the other hand the problem of occlusion and flexibility require algorithmic solutions.

The early solutions for HGR used Finite State Machines (FSM) [8], a gesture was distributed into phases and a set of twelve gestures were classified. Inspired by the results on handwriting recognition [9] and speech analysis various adaptations of a Hidden Markov Model (HMM) have been used [10]. Another branch of solution includes neural networks and Recurrent Neural Networks (RNN) [11]. Most often, both the FSM and RNN strategies use the information of the instantaneous hand-pose for identifying gesture sequences.

The Long Short-Term Memory (LSTM) network [12,13] is a variation of the traditional RNNs and has been shown to outperform the traditional RNN. It has been extensively used for hand-writing and speech recognition tasks recently [14]. In contrast to the HMM where some prior experiments are required to identify the number of states, it is easier to construct an LSTM model. [15] have used LSTM for gesture identification and demonstrated that it performs better than HMM and SVM.

Location, orientation and velocity of the palm have been used as features for gesture recognition problems [16]. This work reaffirms that features like palm and finger positions along with their velocity are useful for gesture classification. A two-phase classification scheme using three LSTMs is introduced. It is demonstrated that distributing the learning in which one phase learns from the hand pose and the other learns from the direction of motion, simplifies the learning tasks.

An early-detection system which is capable of predicting gesture class while the gesture is being completed is introduced. This is an important requirement for an interaction system. To achieve this a one to one labelling scheme between the gesture frames and gesture class is used. Some sequences are sub-sampled for learning fast sequences. The proposed cumulative probability addition scheme for prediction also help stabilise the system response during discontinuous hand movements. The HGR with this LSTM architecture demonstrates an overall frame-wise accuracy of over 90.5%. We observed that, with an equal sized data the proposed two-phase early hand gesture recognition system outperforms a single all-class, but larger LSTM based system which provides accuracy of 86%.

Section 2 introduces the gestures used for the experiments and describes the data collection and feature extraction process along with the data augmentation method and the data distribution scheme for cross-validation. The overall system architecture, the prediction scheme and the cumulative probability method is described in Sect. 3. The analysis of the training process, test results and comparisons with single all-class network prediction is presented in the Sect. 4. Discussion on results and possible future directions are presented in the Sect. 5.

2 Gesture Data and Features

2.1 Gesture Definition

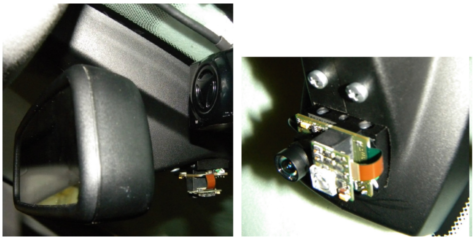
A hand-gesture is a sequence of frames of moving palm. It can involve motion of palm without change in the hand-pose or it could be defined as a sequence of hand-poses where the occurrence of the different hand-poses have a predictable, predetermined order. For this work the recorded hand-gestures include, ‘Clicking’, ‘Swiping’ in Left and Right direction and in Up and Down motion, ‘Accepting’, ‘Declining’, ‘Drop’ and ‘Grabbing’. ‘Clicking’ involves a forward horizontal motion of the pointing finger. Hand motion in horizontal left-right direction is denoted as ‘Swiping’ in left and right direction. The swiping motion may be repeated more than once. Similarly vertical palm motion is denoted as vertical swiping. ‘Accepting’ is a motion of hand outwards from the screen (relative to the camera). ‘Declining’ is the motion of a hand into the screen. ‘Grabbing’ involves a transition of a spread hand with the palm facing vertically downwards to a position of joined fingers accompanied with some vertical motion. ‘Drop’ begins with joined fingers ending in a spread hand with a short downward motion.

2.2 Data Collection and Properties

The output frames from the camera have two channels, the depth and the amplitude. The amplitude value of the pixels are proportional to the reflectance of the surface and inversely proportional to the square of the distance values. The data is recorded with a frame rate of 25 Frames per second.

We use a Photonic Mixer Device (PMD) Nano sensor with a resolution of 120×165 pixel for recording data. This ToF based 3-D camera is attached to the rear-view mirror holder protection. Shown in Fig. 1. Thus the dataset is 3-D top view of hand gestures, it is used for hand pose recognition problem by [17]. The experiments for hand gesture recognition inside the car are conducted with seventeen participants. The data is recorded inside the car and each participant repeats nine gestures around the sat-nav screen of the car. Every participant repeats each gesture six to twelve times. Each frame of the sequence is marked with two labels. ‘Accepting’, ‘Declining’, ‘Drop’, ‘Grabbing’, ‘Clicking’, ‘Horizontal’, and ‘Vertical’ are used as the primary labels. Sequences marked as ‘Horizontal’ are marked with a secondary label ‘Left’ and ‘Right’, and those marked with ‘Vertical’ are marked with secondary labels ‘Up’ and ‘Down’.

As the data is recorded inside a car it allows a combination of depth information with the information about the car environment. This information is utilised to extract the features for hand-shape, location and motion. Note that these features are explicitly utilized in the proposed approach and thus no comparison on a different dataset is shown.



(a) Camera behind rear view mirror.
(b) Camera on the mirror holder.

Fig. 1. The camera setup.

2.3 Training and Testing Strategy

Before training the network, the data was shuffled such that the frames from a complete gesture sequence stay together, while the gesture sequences were placed randomly. This shuffling was essential because the participant continuously repeated the same gesture multiple times during recording. Each frame of the gesture sequence is marked with the label for the entire sequence. This allows us to train the network in way such that it attempts at predicting the sequence-label from the start of the gesture.

The total number of available sequences for training the model are increased by sub-sampling approximately one-fifth of sequences in time. Equal proportion of sequences from each class of gesture are reduced to half duration. Such down-sampling effectively creates sample-points on which the duration for completing a gesture is shorter than the average gesture sequence. The start and end of each sequence including the sub-sampled once are marked. Both training and testing phase of the algorithm use these sequence markers. Table 1 gives a description of the distribution of the data-samples over classes and the number of sub-sampled sequences created for each class.

For testing a leave-2 cross-validation was performed on the data. The data from the seventeen participants was distributed into eight sets of two participants and one set of one. Owing to an otherwise small test dataset a 9-fold cross-validation is done to report the average accuracy of the model. The sub-sampled sequences are separately divided into 9 groups and then used in training and testing accordingly.

Table 1. Number of gesture class samples in dataset

	Up	Down	Left	Right	Click	Accept	Decline	Grab	Drop	Total
Data-points	220	226	247	247	160	188	194	172	160	1814
Down-sampled	44	45	49	49	32	47	48	34	32	380
Total	264	271	296	296	192	235	228	204	192	2194

2.4 Segmentation and Feature Extraction

The palm region is segmented by creating a virtual cuboidal space in the region where we wish to observe the hand-gesture. The background was generated by recording a video in the car and keeping the consistent pixels. This 3-D background image was then removed from each incoming images of the video sequences. Furthermore, the palm pixel closest to screen is tracked. Hand region is segmented by assuming a real length of 18 cm, another threshold divides hand and finger and a Mahalanobis distance based K-mean clustering refines palm-finger segmentation. The hand palm centroid and finger-tip are estimated and tracked using Kalman filter. Features are further described in the Table 2. The features were centred and normalised such that the mean of each feature element over the training data was zero and the variance was unity.

Table 2. Description of features used for the experiments

Type	Feature names	Description
Location	Finger coordinates	The X,Y,Z coordinates of the tracked pixel closest to the screen
	Hand coordinates	The X,Y,Z coordinates of the tracked palm centroid
	Finger azimuth	Polar angle of the principal component vector of the finger cluster of the palm
	Finger polar	Azimuth angle of the principal component vector of the finger cluster of the palm
Velocity	Finger Velocity	The X,Y,Z components of the tracked pixel closest to the screen
	Hand velocity	The X,Y,Z components of the tracked palm centroid
Shape	Concave depth	The maximum distance between convex hull and edge of the segmented palm region
	Convex ratio	The ratio of the size of the convex hull around the palm and the segmented palm region
	Active pixels	The number of pixels in the segmented palms provides an indication of palm-size

3 System Architecture and LSTM Networks

A two-phase classification strategy is employed for classification. To this end, three neural network based systems are combined. The first network classifies the seven primary classes describing the nature of motion. The other two networks are trained to classify the direction of the motion, i.e. Up vs. Down and Left vs. Right. These networks are used in series with the first network. Various neural network architectures were trained and tested for the three classifiers, the network architectures which provided the best cross-validation results separately were used for the classification system. These networks are further described in detail.

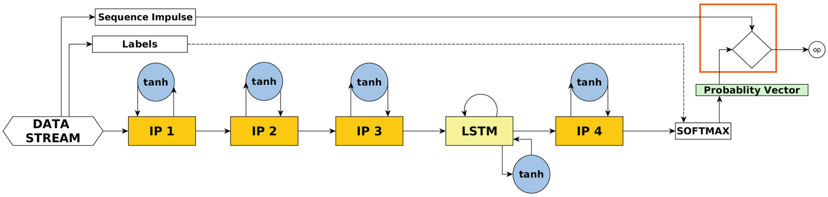


Fig. 2. LSTM network with the output decision unit.

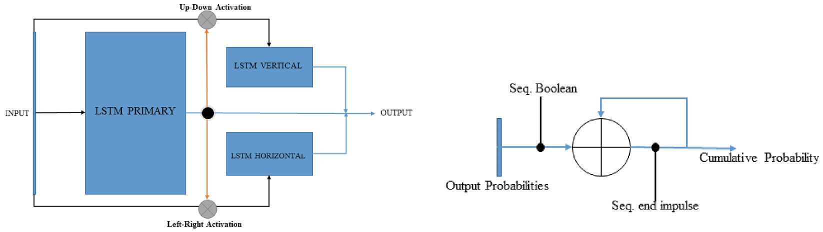
Each network has an *LSTM layer* and several *fully connected dot product layers*. The input layer is connected to a dot product layer. Non-linearity is added to the network by using a *tanh activation function* with each fully connected layers. The network for the primary classifier has five layers apart from the input layer and the output *softmax layer*. The LSTM layer is placed as the fourth layer from the input. The output layer has seven output nodes, each node represents one gesture, see Fig. 2.

The binary classifier identifies the intended direction of the motion when the palm moves in horizontal or vertical direction. Since the swiping motion may be repeated more than once while completing the gesture the identification of the intended gesture is more sophisticated problem than merely identifying the direction of motion. The binary classifier LSTM network has *three hidden layers* along with the one LSTM layer. The output layers have two nodes and a softmax activation function. The connection weights and bias are independent of each other in all networks. The three networks are trained independently using the samples belonging to the corresponding classes from the same training dataset. The training uses the RPROP Algorithm for the optimisation process [18].

3.1 Prediction

The system is shown in Fig. 3, it can be broadly separated into a frame classification part Fig. 3a which produces a nine dimensional probability vector $\overrightarrow{p(t)}$ at time t, and an output probability combination part Fig. 3b which results in another nine dimensional probability vector $\overrightarrow{P(t)}$.

In the classification part of the system the primary classifier is connected with the two motion-direction classifiers, see Fig. 3. It provides a seven dimension probability vector, $\overrightarrow{p^1(t)}$. $\overrightarrow{p^1(t)}$ has five gesture probabilities $\overrightarrow{p(t)^g} = p(t)^{1-5}$



(a) 2-phase gesture recognition setup.

(b) Prediction output for online gesture detection.

Fig. 3. The system architecture and the cumulative probability addition scheme.

and probability for horizontal and vertical direction of motion $p(t)^h, p(t)^v$. On identifying vertical or horizontal swiping of the hand the vertical or horizontal motion classifier is activated with binary activation signals A_v and A_h. The activated binary classifiers then detect the intended direction of the swiping gestures resulting in the two dimension probability vectors $\overrightarrow{p(t)^v}$ and $\overrightarrow{p(t)^h}$ for vertical and horizontal direction respectively. The output from these classifiers replace the motion probabilities in the primary classifier output, (1). The output probabilities from LSTM units are combined (2) to form a nine dimensional vector $\overrightarrow{p(t)}$ and are weighted by the there values in the primary probability vector, the resulting output vector is re-normalised to form a probability vector $\overrightarrow{\hat{p(t)}}$, (3).

$$\overrightarrow{p(t)'^k} = \begin{cases} [\frac{p(t)^k}{2}, \frac{p(t)^k}{2}] & \text{if } A_k = 0. \\ \overrightarrow{p(t)^k} & \text{if } A_k = 1 \text{ where } k \in [v, h]. \end{cases} \tag{1}$$

$$\overrightarrow{p(t)} = [\frac{\sum_{j=1}^{5} p(t)^j}{5}.\overrightarrow{p(t)^g};\ p(t)^v.\overrightarrow{p(t)'^v};\ p(t)^h.\overrightarrow{p(t)'^h}]. \tag{2}$$

$$\overrightarrow{\hat{p(t)}} = \overrightarrow{p(t)}/|\overrightarrow{p(t)}|. \tag{3}$$

The early predictions of the LSTM based system are stabilised by using a cumulative probability addition scheme Fig. 3b. The cumulative addition of the probability regularizes the estimates while making an early prediction. This adds robustness towards jerks, stops and change in hand direction, during the completion of the hand-gesture sequence. Also a strategy based on maximum-probability or majority decision approach predicts the gesture at the end of the sequence. The described method makes a probability estimation for the gesture at every frame.

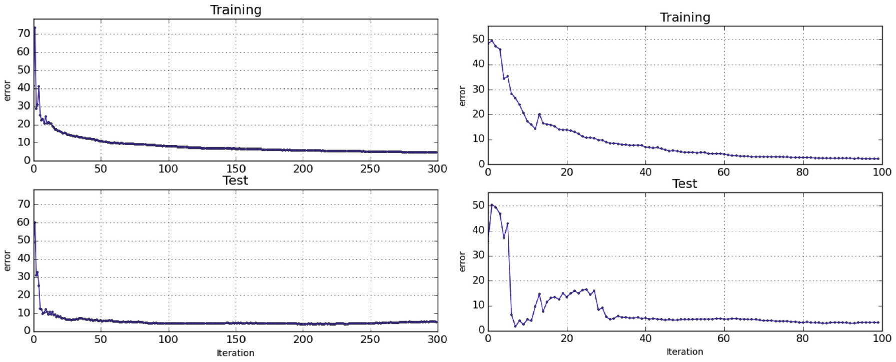
The system output probability is given as $\overrightarrow{P(t)}$, (4). The sum is reset to zero whenever an end of sequence impulse is seen. I_n is the impulse corresponding to the n^{th} sequence. The impulse has a value 1 and the impulse time is given by t_{I_n}. The probability addition is initiated again with a sequence-begin impulse. The n^{th} prediction G_n, corresponds to the index i of the maximum value in the probability vector $\overrightarrow{P(t)}$ (5). Since the initial frames of the sequence have little or no temporal context the predictions made during these first t_d frames of the input stream are not reliable and thus are not read at the output. This scheme allows continuous predictions unlike majority-vote like decisions where prediction is made after viewing the entire sequence.

$$\overrightarrow{P(t)} = \overrightarrow{\hat{p(t)}} + (1 - I) \times (\overrightarrow{P(t-1)}) \tag{4}$$

$$G_n = arg\max_i(\overrightarrow{P(t)}) : t - t_{I_{n-1}} > t_d \tag{5}$$

4 Results and Comparisons

This section describes the training progression of the three models and presents the performance of the entire system. As mentioned in the last sections the first few frames of the prediction made by the system are not considered for output, we also skipped these frames for the evaluation analysis. The output probabilities for sequences beyond the eighth frame of the gesture, which corresponds to a time-period of 0.3 s are considered for the analysis.



(a) Primary LSTM classifier.

(b) Left-Right classifier.

Fig. 4. Train-Test error progression.

The train-test error progression by learning epochs during a sample cross validation for primary phase of classification and the left-right binary classifiers are depicted in Fig. 4. The network was trained for 600 epochs and evaluation was conducted for every second epoch. The average misclassification rate for the given training was 5%. The misclassification rate for the test data at the end of the training was 7%, see Fig. 4a. Both up-down and left-right classifiers were trained as binary classifiers for 400 epochs. It is observed that the misclassification rate on training data after the completion of the training for the up-down motion classifier is 6%, and 1.5% for the left-right classifier. The misclassification rate for

Table 3. Confusion matrix proposed system.

%	U	D	L	R	C	A	De	G	Dr
U	**84**	4	2	0	8	2	0	0	0
D	4	**85**	0	0	0	0	8	0	3
L	0	0	**92**	1	0	0	3	3	1
R	0	0	0	**93**	0	0	3	4	0
C	0	0	0	0	**96**	0	4	0	0
A	8	4	0	0	1	**82**	4	1	0
De	3	7	0	0	0	4	**84**	2	0
G	4	0	0	0	0	5	0	**89**	2
Dr	1	4	0	0	0	0	4	0	**91**

Table 4. Confusion matrix single all-class LSTM

%	U	D	L	R	C	A	De	G	Dr
U	**77**	8	0	0	0	5	3	3	4
D	7	**78**	0	0	0	0	9	2	4
L	0	0	**88**	6	0	2	4	0	0
R	0	0	4	**89**	0	4	3	0	0
C	0	0	0	0	**96**	0	2	2	0
A	8	5	0	0	1	**78**	5	1	3
De	2	9	0	0	0	7	**80**	0	2
G	2	3	0	0	0	2	0	**91**	2
Dr	3	1	0	0	0	0	5	0	**91**

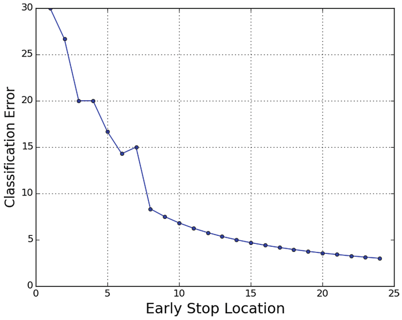


Fig. 5. Test error with early start location.

the test data is 8% and 3%, respectively. Figure 4b shows the left-right classifier error progression. On combining the three networks as the described system, the observed misclassification rate for the full system is 9.25%. The Table 3 shows the confusion matrix for the classification of the nine gesture classes in case of the architecture following the two level classification strategy.

In comparison with a larger all-class single LSTM, chosen after experiments on multiple LSTM models, the performance was considerably better. The improvement in the gestures where direction is important is large. In other gestures the performance improves in all classes apart from 'Drop' where accuracy remains the same and 'Grab' which has a small decrement. The performance of the compared LSTM model is shown in Table 4. Confusion matrices are calculated at each step of the 9-fold cross validation and the mean confusion matrix are reported.

5 Discussion and Conclusion

The performance of the system improves when decisions were taken after a longer delay from the beginning of the sequence. Figure 5 shows the accuracy performance when the latency period for the frame-wise prediction is changed. The decision after a longer latency gained from larger temporal context and is usually more accurate. Some gestures with similar shape and short motion were misclassified, which was reflected in the occasional misclassification of 'Accept' and 'Decline' as 'up', 'down', respectively. This explains the lower accuracy of the up-down gestures in the combined system even though the binary classification accuracy is high. The accumulated regularization of the system output also resulted in missing of fast-very short gestures.

5.1 Conclusion

This work presented a one to one gesture-sequence to label-sequence training procedure to make an immediate decision for a gesture label when the gesture sequence begins. A performance improvement in the two phase classification, when one phase classifies gestures by modification in shape and the other by the direction of motion, is demonstrated. A Modified system architecture achieves an average cross-validation accuracy of 90.75% on the dataset.

This work introduced an accumulated probability based solution for predicting gesture per frame, this eliminates the requirement of delaying the classification until the end of the sequence and also stabilises the prediction outcome to hand-jerks and motion-discontinuity.

As future work we plan to solve the problem of the misclassification of gestures with similar shapes for which we plan to develop more robust shape descriptors. Moreover, spotting of short intended motion of the palm might help in identifying the beginning of the sequences. The prediction performance for short gesture sequences is comparatively worse, using Bayesian filtering approaches with pose identification solutions may help improve this performance.

References

1. Lansdown, T.C., Brook-Carter, N., Kersloot, T.: Distraction from multiple in-vehicle secondary tasks: vehicle performance and mental workload implications. Ergonomics **47**, 91–104 (2004)
2. Green, P.: Visual and task demands of driver information systems. Technical report (1999)
3. Jæger, M.G., Skov, M.B., Thomassen, N.G., et al.: You can touch, but you can't look: interacting with in-vehicle systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1139–1148. ACM (2008)
4. Horrey, W.J.: Assessing the effects of in-vehicle tasks on driving performance. Ergonomics **19**, 4–7 (2011)
5. Freeman, W.T., Roth, M.: Orientation histograms for hand gesture recognition. In: International Workshop on Automatic Face and Gesture Recognition, vol. 12, pp. 296–301 (1995)

6. Liu, Y., Gan, Z., Sun, Y.: Static hand gesture recognition and its application based on support vector machines. In: Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 2008, pp. 517–521 (2008)
7. Alpern, M., Minardo, K.: Developing a car gesture interface for use as a secondary task. In: Extended Abstracts on Human Factors in Computing Systems, CHI EA 2003, pp. 932–933. ACM, New York (2003)
8. Davis, J., Shah, M.: Recognizing hand gestures. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 800, pp. 331–340. Springer, Heidelberg (1994). doi:10.1007/3-540-57956-7_37
9. Hu, J., Brown, M.K., Turin, W.: Hmm based online handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. **18**, 1039–1045 (1996)
10. Chen, F.S., Fu, C.M., Huang, C.L.: Hand gesture recognition using a real-time tracking method and hidden Markov models. Image Vis. Comput. **21**, 745–758 (2003)
11. Yang, J., Horie, R.: An improved computer interface comprising a recurrent neural network and a natural user interface. Image Vis. Comput. **60**, 1386–1395 (2015)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
13. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **6**, 107–116 (1998)
14. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
15. Neverova, N., Wolf, C., Paci, G., Sommavilla, G., Taylor, G.W., Nebout, F.: A multi-scale approach to gesture detection and recognition. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 484–491. IEEE (2013)
16. Yoon, H.S., Soh, J., Bae, Y.J., Yang, H.S.: Hand esture recognition using combined features of location, angle and velocity. Pattern Recogn. **34**, 1491–1501 (2001)
17. Tewari, A., Grandidier, F., Taetz, B., Stricker, D.: Adding model constraints to CNN for top view hand pose recognition in range images. In: Proceedings of the ICPRAM 2005, pp. 170–177 (2016)
18. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: 1993 IEEE International Conference on Neural Networks, pp. 586–591. IEEE (1993)