2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

Real-time Hand Posture and Gesture-based Touchless Automotive User Interface using Deep Learning

V. John[1], M. Umetsu[1], A. Boyali[1], S. Mita[1], M. Imanishi[2], N. Sanma[2] and S. Shibata[2]

***Abstract*— In this study, a vision based in-car entertainment user interface is presented. The user interface is designed using a hand posture and gesture recognition algorithm in deep learning framework. The hand posture recognition algorithm is formulated using the convolutional neural network to perform the fundamental tasks in the user interface. The hand gesture recognition algorithm is formulated using the long-term recurrent convolutional neural network to intuitively interact with the touchless automotive user interface in a detailed manner. In the recurrent deep learning framework, typically, the gesture frames are taken from a uniformly sampled image sequence. In this work, the recurrent structure is enhanced using a reduced number of input frames captured from the image sequence. The reduced input frames or key frames represent the action present in the video sequence. Sparse dictionary learning provide reliable key frame extraction from video sequences. However, sparse dictionary learning is computationally expensive, and are individually optimized for every video sequence. In this paper, we propose to approximate sparse dictionary learning using a non-linear regression framework. The multilayer perceptron is utilized to model the non-linear regression framework. The optimal neural network architecture is identified after a detailed evaluation. We evaluate the proposed recognition methods on public datasets. The proposed methods yield a recognition accuracy of 92% and 90% for pose and gestures, respectively. The combined hand posture and gesture recognition takes 82ms which is a reasonable for real time implementation.**

I. Introduction

Touch-less in-car interfaces have been in demand for decades in automotive industry. Touchless interfaces assist the drivers in seamlessly interacting with the car, without distracting them. To intuitively interact with the interface, typically, camera-based systems are used. In such systems, hand gesture-based intuitive interaction is suitable for dynamic tasks such as selecting the mode, changing the channel, volume etc. The hand gesture recognition algorithm estimates the hand gesture label from the video sequences. While gesture is used for intuitive dynamic interactions, we propose to utilize vision-based hand pose recognition for the fundamental static interactions in the touchless interface. The fundamental static interactions include initializing-terminating the hand gesture recognition system, rejecting phone calls etc. Compared to the gesture recognition, the hand posture recognition estimates the hand pose label from images. Solving the vision-based hand posture and gesture recognition is not straightforward. Some of the challenges include appearance, pose and motion variation among people, illumination variation and background noise. The appearance variations are inter-person, while the pose and motion variations are both inter-person and intra-person. The appearance variations are common to both the pose and gesture recognition algorithms, while the pose variation and motion variations correspond to pose recognition and gesture recognition, respectively. Finally, an important challenge in the development of touchless user interfaces is real-time computational efficiency.

We estimate the pose and gesture label of the image sequences captured from by video. The convolutional neural network [1] (CNN) estimates the pose label from the image. While the long-term recurrent convolution network (LRCN) [2] is adopted and extended to identify the gestures labels from the image sequence. Typically, given a image sequence, the gesture is recognised by the LRCN framework using uniformly sampled image frames. We propose to estimate the gesture label, and improve the performance of the LRCN, using fewer frames identified from the video. These limited frames are selected to encode the entire video's gesture information.

The sparse dictionary learning algorithm (SMRF) proposed in [3] is among the efficient methods in identifying the representative signal patterns. However, the sparse algorithm is a computationally demanding algorithm, and it requires solving an individual optimization function to find representative patterns for every signal sequence. Consequently, the SMRF is not suitable for the proposed study, as the key frames would have to be estimated for every gesture performed. We propose to "learn" the SMRF's dictionary learning function as a remedy. By modelling the dictionary learning function over a number of training video sequence, we approximate the SMRF's learning mechanism. The SMRF approximation is performed using a multilayered perceptron-based non-linear regression function. The trained multilayered perceptron is then used to estimate the key frames for every test gesture sequence, without the need for individual optimization. Using the key frames, the LRCN estimates the gesture for the image sequence.

The proposed algorithms are validated on the Cambridge [4] and HGR datasets [5]. For the gesture recognition, we report comparable classification accuracy. However, unlike the baseline algorithms, we report real-time computational efficiency. In case of the pose recognition, we report better classification accuracy with real-time efficiency. We obtain a gesture classification accuracy of 90% and a

[1] V. John, M. Umetsu, A. Boyali and Seiichi Mita are with the Toyota Technological Institute, Japan {vijayjohn,sd13017,ali-boyali, smita}@toyota-ti.ac.jp.

[2] M. Imanishi, N. Sanma and S. Shibata are with Nippon Soken, Japan masayuki_i_imanishi@soken1.denso.co.jp norio_sanma@soken1.denso.co.jp syunsuke_shibata@soken1.denso.co.jp

pose classification accuracy of 92% along with a combined computational efficiency of 82ms. Based on the review of literature, the main contributions are: the approximation of sparse dictionary learning-based key frame extraction; the real-time vision-based touchless user interface using deep learning. We structure the remainder of the paper as the following. In Section II, we review the literature. In Section III the algorithm is presented. Finally, in Section IV and Section V, we present the experiments and conclusion, respectively.

II. Literature Review

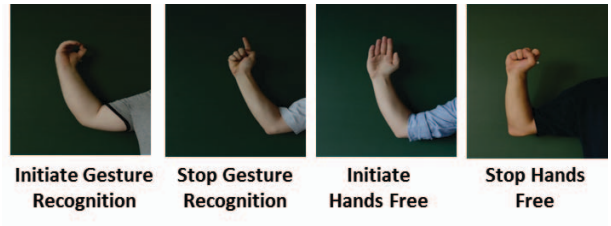
The hand gesture and posture recognition by optical sensors and cameras is an important research area. In hand posture recognition, various salient image features are used to classify the pose [6]–[9]. In case of gesture recognition, apart from appearance, spatio-temporal video features are used to classify the gesture [10]–[13]. These features are then used within various classification frameworks to identification of gesture labels. For example in the works by Ahmed et al. [11], the authors utilize the support vector machine and neural network, respectively, to classify SIFT features. In [14], the hidden Markov model is trained with skin colour-based features to classify the gestures. Alternatively, researchers have also utilized compressed sensing techniques to perform gesture classification [15], [16]. The main challenges in vision-based systems are illumination, appearance and motion variations. Researchers address these issues by either integrating multiple sensors [10], [17] or by using the deep learning framework [18]–[21].

The deep learning framework has been reporting successful results with high performance in image detection [1]. Consequently, researchers have extended deep learning for activity recognition [22], [23] as well. In case of the gesture recognition problem, Pigou et al. [23] perform the recognition using a multiscale CNN. This is extended by Neverova et al. [10], where the authors use a multimodal framework in addition to the multi-scale deep learning framework. Apart from the CNN, researchers also use the recurrent network to perform gesture recognition. Both these networks are used to model the inter-frame temporal information [19]. For example, Nishida et al. [19] propose a multimodal LSTM with color and depth information to classify the gestures. Recently, John et al. [18] utilize the long recurrent convolutional neural network (LRCN) to classify the gestures. The authors utilize the deconvolutional neural network to extract the representative video frames. These frames function as the LRCN input.

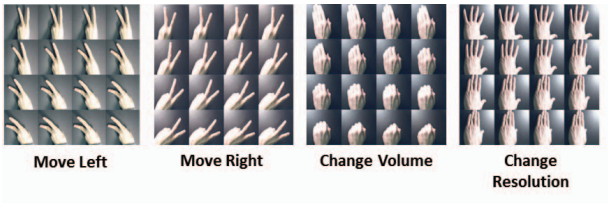
In our proposed work, we extend the work by John et al. [18] by approximating the sparse dictionary learning representative frame extraction algorithm. [3] with a non-linear regression model. Compared to [18] we report comparable classification accuracy with real-time computational efficiency, which is suitable for autonomous user interfaces. Additionally, we investigate the suitability of deep learning for hand pose recognition within touchless user interfaces.

III. Algorithm

Given an input image I with hand pose, the vision-based hand posture recognition algorithm estimates the pose label p using the CNN. On the other hand, given an gesture-based image sequence V, the LRCN estimates the gesture label l. The estimated pose label p is used for the fundamental static tasks, while the estimated label of gesture l is used for the dynamic intuitive tasks. Examples of the fundamental static and dynamic intuitive interface tasks are shown in Figure 1.



(a)



(b)

Fig. 1. (a) Illustrative examples of static poses along with their pose labels used for fundamental interface tasks. (b) Illustrative examples of dynamic gestures along with their gesture labels used for dynamic intuitive interface tasks.

In [2] (LRCN), the gesture label is estimated using features derived from 16 video frames. CNN [1] extracts the features from the 16 frames, and the LSTM identifies the gesture. In this work, we enhance LRCN by using fewer input frames. More specifically, we estimate the gesture label using 3 identified key frames. These key frames are identified using a learnt neural network-based regression model. The neural network is formulated in order to "learn" the dictionary learning of the SMRF [3]. Given the extracted key frames, the CNN extracts the corresponding image features. The LSTM subsequently estimates the gesture label.

A. Brief Overview

Here we review the sparse dictionary learning framework, the CNN and the LRCN, before presenting the algorithm in detail.

1) Sparse Modeling: Given a data matrix $Y = \{y_i\}_{i=1}^{N} \in \mathbb{R}^{m \times n}$ with column-wise data points $y_i \in \mathbb{R}^m$, the sparse dictionary learning framework is given as,

$$\sum_{i=1}^{N} \|y_i - Dx_i\|_2^2 = \|Y - DX\|_F^2 \tag{1}$$

where $D = \{d_i\}_{i=1}^{U} \in \mathbb{R}^{m \times u}$ corresponds to the basis dictionary and $X = \{x = i\}_{i=1}^{N} \in \mathbb{R}^{u \times t}$ is a column vectors of the

sparse coefficients. D and X, which efficiently represent the data Y are obtained by optimizing Eqn 1.

The sparse framework is utilized by Elhamifar et al. [3] to identify representative frames in an image sequence. These k frames represent a given image sequence. The learning framework used to identify the key frames is given as,

$$\sum_{i=1}^{N} \|y_i - Yc_i\|_2^2 = \|Y - YC\|_F^2 \tag{2}$$

where Y is the data matrix and $C \in \mathbb{R}^{n \times n}$ represents the coefficient matrix. In Eqn 2, the reconstruction error of the frames are minimised. During the optimisation, each frame is reconstructed as a linear combination of all the other frames. Following the solving of Eqn 2, the k representative frames correspond to the top K ranked non-zero rows of C.

2) Convolutional Neural Network: CNN simultaneously performs feature extraction and classification using multiple learnable layers. The initial layers of CNN extract the features from the image using learnable filters. The CNN filters extract these features from the preceding layer using the convolution operation. The final layers of the CNN perform the feature classification using fully connected layers. The CNN filters and fully connected layers are learnt using the back-propagation algorithm. The CNN filters or weights in the different layers are updated using back-propagation and stochastic gradient descent algorithm. In this paper, a pre-trained CNN architecture [1] is fine-tuned for the pose label estimation. Consequently, we utilize the same architecture as proposed by Krizhevsky et al. [1].

3) Long Recurrent Neural Network: Deep learning-based activity recognition is performed using the LRCN [2]. This framework uses the CNN to extract image features, and the LSTM to estimate the gesture. The gates include the input, forget, input modulation and output gates. The input gate and forget gate are used to selectively forget the memory from the previous unit. While, the output gate is used to selectively transfer the memory to the next hidden unit. By selectively forgetting and transferring the memory, the LSTM improves the accuracy of the RNN in modeling the temporary dynamics.

Donahue et al. [2], utilize the LRCN to recognize action from video. First, the authors sample 16 frames from the video sequence. The CNN extracts the deep features from the sampled frames. These set of CNN-based features from the 16 sampled frames are then given as an input to the LSTM, which performs the action classification.

B. Training Phase

In this phase, firstly, the SMRF algorithm is approximated using the neural network. Secondly, the CNN is trained to estimate the hand pose. Finally, the LRCN is trained and the gesture is estimated. The training phase is illustrated in Fig 2.

1) Sparse Dictionary Learning Approximation: The sparse dictionary learning algorithm is approximated using a set of training sampled image sequences. Given a training set of T video sequences (sampled), $\mathbf{V} = \{V_t\}_{t=1}^{T}$, where each t-th sequence contains 16 frames uniformly sampled from the image sequence. The corresponding video features for the training set are obtained as $\Phi = \{\phi_t\}_{t=1}^{T}$. The video feature, $\phi_t \in \mathbb{R}^{16 \times 64 \times 64}$, for the t-th video sequence is extracted from the corresponding 16 sampled gray scaled images with size 64×64 . The training set of video features are then used by the SMRF algorithm to estimate the sparse coefficient matrix $\mathbf{C} = \{C_t\}_{t=1}^{T}$. Each C corresponds to the coefficient matrix in Eqn 2. Given the training set of sparse coefficient matrices and corresponding video features, we formulate a non-linear regression model to approximate the SMRF dictionary learning. This is given as,

$$\mathbf{C} = f(\Phi, [\mathbf{W}, \mathbf{B}]) \tag{3}$$

where the weight is represented by $\mathbf{W}$ and the bias is represented by $\mathbf{B}$ of the neural network used to model the non-linear regression framework. By learning the regression framework in Eqn 3, we approximate the SMRF's dictionary learning over the entire set of training sequences. Subsequently, unlike the SMRF, the trained neural network can be used for any test video sequence, without the need for any re-training or individual dictionary learning. Moreover, the trained neural network is also computationally inexpensive compared to the SMRF algorithm. The neural network architecture used for modeling the regression framework was identified after a detailed parameter analysis on the experimental dataset (Sec IV). We utilized the neural network-A shown in Table I for modeling the non-linear regression function.

2) CNN Training: The pre-trained Alexnet [1] is fine-tuned to perform hand pose recognition, we first modify the pre-trained CNN architecture. More specifically, the final output layer in the pre-trained Alexnet is modified to reflect the number of pose labels in the recognition algorithm. Since we validate the algorithm with the HGR dataset [5], which contains 32 pose labels, we have 32 neurons in the output layer. This CNN model is referred to as the hand-pose model. The weights and biases of the pre-trained Alexnet are used to initialise the weights and biases of the hand-pose model. Consequently, the CNN learning rate, the momentum, weight and bias multipliers are set to 0.01,0.9, 1 and 2 for these layers. For the final layer, which is not pre-initialized, only the CNN learning rate is lowered to 0.001.

3) LRCN Training: To estimate the gesture, the pre-trained LRCN algorithm is fine-tuned [2]. The pre-training is done with the UCF-101 dataset [2]. To facilitate the fine-tuning, we adopt the original architecture with a few changes. Compared to the original LRCN, we have 9 output channels, 3 input channels corresponding to the CNN-based image features, and 64 LSTM units. The Caffenet is used to extract the features from the 3 key frames for a given video sequence. The weights learning multiplier is set to 1, the momentum to 0.9 and the bias learning multiplier is set to 2. For the final layer, the weight learning multiplier is set to 10 and the bias learning multipliers is set to 20. Finally,

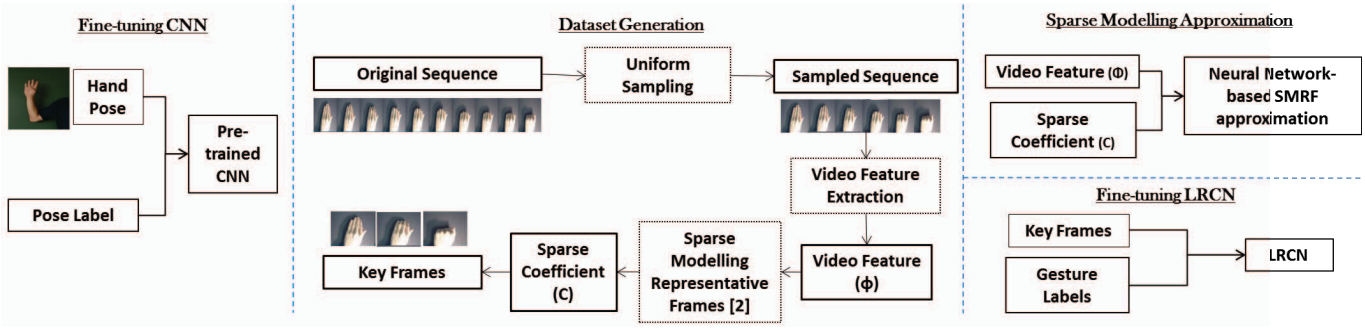


Fig. 2. The training phase of the proposed algorithm

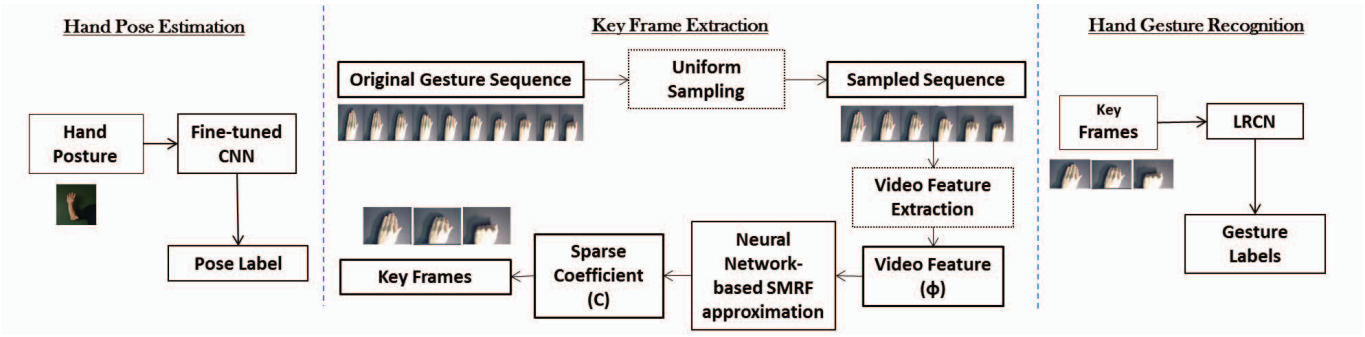


Fig. 3. The testing phase of the proposed algorithm

TABLE I

DIFFERENT NEURAL NETWORK ARCHITECTURES. *FC* REPRESENTS FULLY CONNECTED LAYERS WITH RELU ACTIVATION FUNCTION. WE REPRESENT THE NUMBER OF NEURONS FOR EACH LAYER.

NN – A	Input Layer (ϕ) video feature	$FC1$ (1000)	$FC2$ (1000)	Output 256 (C)				
NN – B	Input Layer (ϕ) video feature	$FC1$ (1000)	$FC2$ (1000)	$FC3$ (1000)	$FC4$ (1000)	Output 256 (C)		
NN – C	Input Layer (ϕ) video feature	$FC1$ (1000)	$FC2$ (1000)	$FC3$ (1000)	$FC4$ (1000)	$FC5$ (1000)	$FC6$ (1000)	Output 256 (C)

we set the number of iterations to 7500 with a learning rate of 0.001.

C. Testing Phase

The hand pose is estimated by the fine-tuned hand pose CNN. Similarly, the gesture is estimated using the trained neural network and LRCN model. The testing phase is illustrated in Fig 3.

Given a test video sequence with N original frames, $\hat{V}$, uniform sampling of 16 frames is performed to generate the "sampled" test sequence. Subsequently, we extract video features, $\hat{\phi}$, from the sampled test sequence. Given these input video features, the trained neural network model predicts the sparse coefficient matrix $\hat{C}$ using Eqn 3. From the predicted sparse matrix, k key frames represented by the top k non-zero rows are obtained [3]. Using these k test key frames, the trained LRCN identifies the gesture.

IV. EXPERIMENTS

We validate our methods on the HGR [5] and the Cambridge [4] dataset . The Cambridge dataset contains video sequences with 9 gestures, which are performed by 2 subjects. Each gesture is performed 10 times with variations over 5 different illuminations. The HGR dataset contains 32 hand pose images performed by 18 different subjects. Both the algorithms are validated with a 5-fold cross validation over the videos sequences and images. We compare our method with baseline methods. Moreover, we analyse the performance of the algorithm with varying parameters. We implement the proposed algorithms on a Linux desktop using Caffe [24] and Keras libraries. The hand posture recogntion algorithm reports a computational complexity of 40ms, while the hand gesture recognition reports a time of 42ms using the Nvidia Geforce GTX 980 graphics card.

A. Comparative Analysis

1) Hand Posture Algorithm: We compare our method with baseline methods. The histogram-of-oriented gradient features (HOG) are extracted from the hand pose image, and the random forest classifier with 200 trees is used to estimate the pose label. An illustration of the HOG features and the deep features obtained from the first convolutional layer ($C1$) in the hand pose CNN model is shown in Fig 4. We report the pose classification accuracy in Table II, where the classification of the proposed method is better. We report a computational complexity of 40ms.

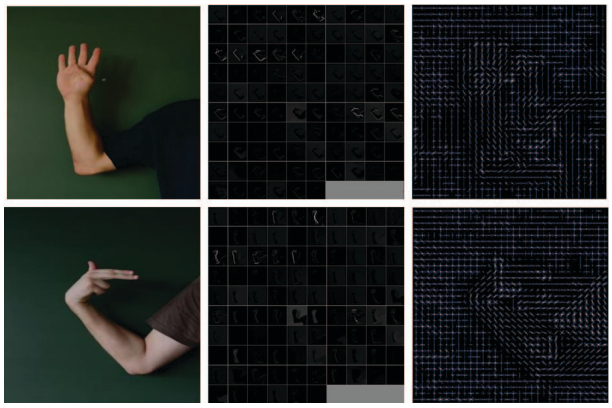


Fig. 4. An illustrations of the hand pose features. (Left) Hand pose from HGR dataset, (Middle) $C1$ filter maps from the hand pose CNN, and (Right) HOG features.

TABLE II

THE POSE CLASSIFICATION ACCURACIES FOR THE HGR DATASET .

Algo.	Class. Acc. (Mean and Std.Dev) %
Proposed	91.9 $\pm$ 6.7
HOG-Random Forest	83.6 $\pm$ 5.2

2) Hand Gesture Algorithm: The performance of our algorithm is compared with state-of-the-art baseline algorithms. We compare with the following baseline algorithms: 1) Deconvnet-based LRCN algorithm proposed by John et al. [18]. In this algorithm, the authors extract the key frames using the deconvnet-based semantic segmentation framework. Tiled patterns generated from the video are used to identify the LRCN-input key frames; 2) The LRCN algorithm proposed by Donahue et al. [2], where 16 uniformly sampled frames are the LRCN input; 3) The SMRF-based LRCN algorithm, where the key frames are extracted using the SMRF algorithm are the LRCN input. In all the baseline algorithms, the LRCN estimates the gesture label.

We report the classification accuracies and computational complexities in Table III. The proposed algorithm reports 90% classification accuracy, which is similar to the performance of the Deconv-LRCN. Moreover, unlike the Deconv-LRCN, we report real-time computational complexity. We can also observe that the computationally expensive SMRF-LRCN reports the best classification accuracy. However, the computational time is not suitable. The performance of the 16-frame LRCN is inferior to the key frame-based LRCN's. Additionally, the computational time is also high. As shown in the experiments, we can conclude that our method reports high classification accuracy at real-time computational complexity.

B. Parameter Analysis

In the parameter analysis, we evaluate different neural network models to estimate the non-linear regression model in Eqn 3. We evaluate three different neural network architectures on the Cambridge dataset [4]. The different network architectures are shown in Table I.

1) Different Architectures: The neural network models are evaluated on Eqn 3 using a 5-fold cross validation. The Φ is estimated from the video sequences, while the corresponding $\mathbf{C}$ is estimated using the SMRF algorithm. We train the neural network model with a mean square error loss, ADAM optimization and 7200 iterations. Additionally, the input features where normalized to unit norm. The training and testing errors corresponds to the Euclidean distance computed between the ground truth and estimated coefficient matrix. The errors are shown in Table IV. Based on these errors we selected NN-A to learn the regression function and predict the coefficient matrix.

TABLE IV

EVALUATION OF DIFFERENT NEURAL NETWORK ARCHITECTURES

Arch	Train. Loss	Test. Loss
NN-A	32.9 $\pm$ 0.35	16.7 $\pm$ 0.17
NN-B	34.2 $\pm$ 2.04	17.4 $\pm$ 1.13
NN-C	34.1 $\pm$ 0.4	17.2 $\pm$ 0.11

2) Different Activation Function: We also further evaluate different activation function. As shown in Table V-Table VII, the relu activation function is marginally better than the sigmoid and tanh activation function. We also performed an evaluation with linear activation function. However, the resulting NN models were not successfully trained.

TABLE V

EVALUATION OF DIFFERENT ACTIVATION FUNCTIONS WITH NN-A

Activ	Train. Loss	Test. Loss
Relu	32.9 $\pm$ 0.35	16.7 $\pm$ 0.17
Sigmoid	33.9 $\pm$ 0.1	17.1 $\pm$ 0.2
Tanh	35.9 $\pm$ 0.4	18.2 $\pm$ 0.9

TABLE VI

EVALUATION OF DIFFERENT ACTIVATION FUNCTIONS WITH NN-B

Activ	Train. Loss	Test. Loss
Relu	34.2 $\pm$ 2.04	17.4 $\pm$ 1.13
Sigmoid	34.1 $\pm$ 0.13	17.2 $\pm$ 0.18
Tanh	37.4 $\pm$ 0.4	18.9 $\pm$ 0.4

TABLE III

THE GESTURE RECOGNITION ACCURACIES WITH COMPUTATIONAL TIME.

Algo.	Class. Acc. (Mean and Std.Dev) %	Time Taken
Proposed.	89.9 ± 3.4	**42ms**
Deconv-LRCN	90.9 ± 1.4	110ms
16-frame LRCN.	86.5 ± 2.7	180ms
SMRF-LRCN.	**95.6±1.4**	300ms

TABLE VII

EVALUATION OF DIFFERENT ACTIVATION FUNCTIONS WITH NN-C

Activ	Train. Loss	Test. Loss
Relu	34.1 ± 0.4	17.2 ± 0.11
Sigmoid	34.1 ± 0.1	17.2 ± 0.18
Tanh	37.6 ± 0.5	18.9 ± 0.3

3) Input Normalization: The input feature normalization is also evaluated with NN-A. As shown in Table VIII, the absence of input normalization affects the performance of the algorithm.

TABLE VIII

EVALUATION OF INPUT NORMALIZATION

Algo	Train. Loss	Test. Loss
Normalization	32.9 ± 0.35	16.7 ± 0.17
Without Normalization	2451.1 ± 1266.3	1219.1 ± 614.1

V. CONCLUSION

A hand pose and gesture classification algorithm for touchless automotive user interface is proposed. The pose is identified using the convolutional neural network. The gesture is identified using the recurrent deep learning framework. The input to the recurrent deep learning framework corresponds to fewer image sequence frames. These frames are identified using a neural network-based approximation of the sparse dictionary learning framework. Sparse learning is computationally expensive, which is addressed by the neural network approximation with a non-linear regression function. We validate the proposed method on public datasets and we report high classification accuracy with real-time computational efficiency. In our future work, we will deploy the pose recognition and gesture recognition on the actual vehicle.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[2] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.

[3] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *CVPR*, 2012.

[4] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1415–1428, 2009.

[5] M. Kawulok, "Fast propagation-based skin regions segmentation in color images," in *FG*, 2013.

[6] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 12, pp. 52 – 73, 2007.

[7] H. Cheng, L. Yang, and Z. Liu, "Survey on 3d hand gesture recognition," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1659–1673, 2016.

[8] S. Bilal, R. Akmeliawati, M. J. E. Salami, and A. A. Shafie, "Vision-based hand posture detection and recognition for sign language; a study," in *International Conference on Mechatronics*, 2011.

[9] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Computing and Applications*, pp. 1–11, 2016.

[10] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *ECCV 2014 Workshops*, 2015.

[11] T. Ahmed, "A neural network based real time hand gesture recognition system," *International Journal of Computer Applications*, vol. 59, no. 4, pp. 17–22, 2012.

[12] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, 2015.

[13] J. J. LaViola, Jr., "A survey of hand posture and gesture recognition techniques and technology," Tech. Rep., 1999.

[14] T. Starner, A. Pentland, and J. Weaver, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, 1998.

[15] A. Boyali and M. Kavakli, "A robust gesture recognition algorithm based on sparse representation, random projections and compressed sensing," in *Conference on Industrial Electronics and Applications*, 2012, pp. 243–249.

[16] S. Georgiana and C. D. Cleanu, "Sparse feature for hand gesture recognition: A comparative study," in *International Conference on Telecommunications and Signal Processing*, 2013.

[17] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," in *FGR*, 2015.

[18] V. John, A. Boyali, S. Mita, M. Imanishi, and N. Sanma, "Deep learning-based fast hand gesture recognition using representative frames," in *DICTA*, 2016.

[19] N. Nishida and H. Nakayama, "Multimodal gesture recognition using multi-stream recurrent neural network," in *PSIVT*, 2015.

[20] D. Wu, L. Pigou, P. J. Kindermans, N. D. H. Le, L. Shao, J. Dambre, and J. M. Odobez, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, 2016.

[21] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *CVPR Workshops*, 2015.

[22] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[23] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *ECCV Workshops*, 2015.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guaddarrame, and T. Darrel, "Caffe: Convolutional architecture for fast feature embedding," in *arXiv preprint arXiv:1408.5093*, 2014.