2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)
Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4, 2016

In-Vehicle Hand Gesture Recognition using Hidden Markov Models

Nachiket Deo, Akshay Rangesh and Mohan Trivedi

Abstract— **In this work we explore Hidden Markov models as an approach for modeling and recognizing dynamic hand gestures for the interface of in-vehicle infotainment systems. We train the HMMs on more complex shape descriptors such as HOG and CNN features, unlike typical HMM based approaches. An analysis of the optimal hyperparameters of the HMM for the task has been carried out. Also, dimensionality reduction and data augmentation have been explored as methods for reducing overfitting of the HMMs. Finally we experiment with the CNN-HMM hybrid framework which uses a trained Convolutional Neural Network for estimating the emission probabilities of the HMM. We obtain a mean recognition accuracy of 57.50% on the VIVA hand gesture challenge, which while not the best result on the dataset, shows the feasibility of the approach.**

Index Terms— ***Hand Gesture Recognition, naturalistic drive setting, Hidden Markov Models (HMM), Convolutional Neural Networks (CNN) for feature extraction, CNN-HMM hybrid***

I. INTRODUCTION

A contact-free interface for an in-vehicle infotainment system can potentially reduce the visual load on the driver as compared to a tactile interface, leading to fewer distractions and improving the safety and comfort of the driver. A vision based hand gesture recognition system can lead to an interface that is both intuitive and non-intrusive. In this paper, we explore one such approach based on Hidden Markov Models.

The volatile in-vehicle environment introduces many challenges for gesture recognition as compared to a controlled indoor environment. There can be rapid illumination changes and shadow artifacts. There can be considerable temporal and postural variability in gestures performed by different users, which the system needs to be robust to. The system could be engaged by either the driver or the passenger sitting next to them and needs to be able to handle either case. Finally, to allow for multiple functionalities of the infotainment system, a realistic gesture set needs to be considerably large, involving hand and finger movements. The system thus needs to be able to classify a diverse set of gestures. The VIVA hand gesture dataset provides a realistic setting taking these factors into account and has thus been used for the evaluation of this work.

Notable previous work on the VIVA hand gesture challenge includes [1] by Ohn-Bar and Trivedi. The authors use an SVM based gesture classifier and compare various hand crafted spatio-temporal features such as HOG [3], HOG2 [4], HOG3D [5] and Dense Trajectories [6]. They report their best recognition accuracy with a combination of concatenated HOG features and HOG2 features. Molchanov et al.[2] report the highest recognition accuracy to date on the VIVA hand gesture dataset using a 3D Convolutional Neural Network. Both methods handle the temporal variability of the gestures by first resizing the videos to a fixed length by interpolating frames and then extracting spatio-temporal features from each video.

An alternative to this approach would be to use generative models inherently capable of modeling time series. These can be trained on spatial features extracted from each frame from the video without having to resize it. Hidden Markov Models (HMMs) are an example of such generative model. HMMs have been extensively used in Automatic Speech Recognition due to their ability to model both the spectral and temporal variability of speech signals. Analogously, in case of hand gesture recognition, HMMs can be expected to model the spatial variability ie. variations in hand posture at any frame of the video, and the temporal variability of the dynamic gesture, if trained on shape descriptors.

HMMs have previously been used for hand gesture recognition. Zobl et al.[7] and Althoff et al.[8] use HMMs for hand gesture recognition in vehicles. Both approaches involve segmenting the hand region and extracting the hand position, area and Hu's moments[11] as features for training the HMM. Starner et al.[9] use HMMs for sign language recognition. They track a gloved hand and extract features such as hand location, area, axis of least inertia and eccentricity of a bounding ellipse for the hand. Minnen and Zafrulla[10] detect hand blobs and extract features based on the blob contour for training the HMM. Each of the aforementioned works use simple feature sets with HMMs. In this work, we explore the use of more complex shape descriptors, namely, HOG features and CNN features, for training the HMM, since results from [1] and [2] suggest that these contain useful cues for discriminating between the VIVA hand gestures. Finally, we also experiment with the Neural Network-HMM hybrid framework that has been employed successfully in speech recognition systems [16], [17] , where a trained fully connected or Convolutional Neural Network is used for generating the emission probabilities of the HMM.

In particular, we seek to answer the following: (1) What are appropriate hyperparameters to be used in an HMM for hand gesture recognition? (2) How do HOG and CNN features compare in the HMM framework? (3) How can we reduce overfitting in the HMM?

The authors are affiliated with the department of Electrical and Computer Engineering at the University of California San Diego
ndeo arangesh mtrivedi @ucsd.edu

978-1-5090-1889-5/16/$31.00 ©2016 IEEE

TABLE I

GESTURE INVENTORY

No.	Gesture	No.	Gesture
1	Swipe Right	11	Scroll Up
2	Swipe Left	12	Tap once
3	Swipe Down	13	Tap thrice
4	Swipe Up	14	Pinch
5	Swipe V	15	Expand
6	Swipe X	16	Rotate Counter Clock-wise
7	Swipe +	17	Rotate Clock-wise
8	Scroll Right	18	Open
9	Scroll Left	19	Close
10	Scroll Down		

II. METHOD

This section details the data and methods used in this work. Section II.A briefly describes the VIVA hand gesture dataset. Section II.B describes the structure of the HMMs used and their training process. Section II.C describes the features used. Section II.D describes the methods attempted for reducing overfitting in the HMM. Finally section II.E describes the CNN-HMM hybrid system.

A. Data:

The VIVA hand gesture dataset [1] consists of grayscale and depth videos of dynamic hand gestures performed near the infotainment unit of a moving vehicle. The videos were captured using a Microsoft Kinect device and have a resolution of 115 × 250 pixels. The dataset consists of 19 different gestures involving hand and finger movements given in Table I. The gestures were performed by 8 different subjects. Each subject performed every gesture two to three times with their right hand, while sitting in the drivers seat and with their left hand, while sitting in the passenger's seat, giving a total of 885 gesture videos. The dataset was designed in order to test the robustness of systems to fast illumination changes, subject variability, position of the subjects and the unstable environment of the moving vehicle.

B. HMM topology and training:

For each of the 19 gestures, we use a left-right HMM topology. The state transitions of a left-right HMM are restricted only to self transitions and forward transitions to the next state. This considerably simplifies the model and is a reasonable assumption since each gesture can be considered to be a sequence of hand postures and positions that follow the same order every time. The number of states is a hyperparameter that we vary. This is explained in greater detail in section III.A. The emission probability distribution of each state of the HMM is modeled as a mixture of Gaussians. The number of mixture components is also a hyperparameter that is varied. Diagonal covariances are used for each mixture component instead of full covariances to reduce the model complexity and possibility of overfitting.

The HMM is trained using the Baum-Welch algorithm[12], with each gesture's data being used to train the respective HMM. Finally, the Viterbi algorithm is used for classifying a test gesture using the trained HMMs. HMM training and testing was carried out using the HTK toolkit [13].

C. Features:

Features are extracted from each frame of the depth and grayscale videos from the dataset. The features are subjected to a discrete cosine transform after being extracted in order to decorrelate them. This makes them more suitable to be modeled by diagonal covariance Gaussian mixture models. We consider two features in particular:

1) **HOG features:** We extract modified HOG features as described in [1]. The entire 115 × 250 pixel frame is divided into a 4 × 4 grid of blocks with a 50% overlap between any adjacent blocks. HOG features are extracted from each of the blocks. 8 unsigned orientation bins are used for generating the histograms. Finally all the histograms from the 16 blocks are concatenated to form the 128 dimensional modified HOG feature vector for that frame.
2) **CNN features:** Razavian et al.[14] showed that features extracted from a Convolutional Neural Network trained for an object recognition task can be used as a generic image representation for a variety of different unrelated vision tasks. We use this concept here. We use the ImageNet trained VGG-16 network[15] as a feature extractor. Each depth and grayscale frame is first resized to match the input size of the VGG-16 network. It is then subjected to z-scoring and then given to the network as input. The activation of the second last (fully connected) layer of the network, consisting of 1000 units is treated as the feature vector to be used in the HMM.

D. Reducing Overfitting in the HMMs:

The HMMs have a tendency to overfit due to the limited size of the VIVA hand gesture dataset. We thus consider two approaches to reduce overfitting in the HMM:

1) **Dimensionality reduction using PCA:** We apply Principal Component Analysis for reducing the dimensionality of the feature vectors in order to reduce overfitting. The number of principal components retained is determined experimentally
2) **Data Augmentation:** We make use of data augmentation methods described in [2] for increasing the size of the training set. We consider the following transformations for data augmentation:
   - **Ordering / orientation based transformations:** The videos are reversed, mirrored or both reversed and mirrored to obtain three new gesture instances. This often ends up changing the gesture label. eg. The swipe up label on reversal becomes the swipe down label. The scroll left label on mirroring becomes the scroll right label.
   - **Affine transformations:** The videos are subjected to affine transformations such as translation or rotation. We use a vertical shift of ± 5 pixels,

a horizontal shift of ± 10 pixels and a rotation of ± 5 degrees for each of the videos.

E. CNN-HMM hybrid:

In a CNN-HMM hybrid, a trained CNN replaces Gaussian mixture models as the state emission probability estimator of the HMM. The outputs of a trained CNN correspond to the class posterior probabilities $P(C|x)$ given the input x. If a CNN is trained to classify the HMM state s_t given a gesture frame x_t at time t, its outputs correspond to the probabilities $P(s_t|x_t)$. These can then be scaled by the HMM state prior probabilities $P(s_t)$ to give the HMM emission probabilities $P(x_t|s_t)$.

We use the ImageNet trained VGG-16 network in the CNN-HMM hybrid framework. Since the VGG-16 network has been trained on a different set of outputs than our HMM states, we replace the final layer of the network with output units corresponding to our HMM states and retrain only the final layer of the network with the VIVA hand gesture data. The labels required for training the CNN are obtained by running our best performing HMM in forced alignment mode on the VIVA hand gesture data.

III. RESULTS AND DISCUSSION

For each of the following subsections, we test our system on the VIVA hand gesture dataset in a leave one subject cross-validation setting. Thus all recognition accuracies reported are the average accuracies across the 8 folds.

A. HMM parameter sweep:

We initially experiment with the HMM hyperparameters, namely the number of states and number of Gaussian mixture components in order to decide the optimal values for them. We use only the depth videos for these experiments.

1) **Number of States:** We vary the number of states of each left-right HMM from 5 to 35 using increments of 5 for both HOG and CNN features. The number of Gaussians per mixture is fixed at 2. Fig.1 shows the plot of recognition accuracy vs number of states for both HOG as well as CNN features. We can see that the number of states in the HMMs greatly affects the recognition accuracy. Having very few states per HMM (5 or 10) leads to poor accuracies and so does having too many states per HMM. This trend seems to hold irrespective of the feature used. To better understand the reason behind this, we analyze the accuracies for individual gestures for the case of HOG features and 25 HMM states. Fig. 2 shows the results. The x-axis corresponds to the gesture numbers as given in Table I. The bar plot corresponds to the recognition accuracy for the specific gestures. The red plot shows the average lengths, in number of frames, of each gesture and their standard deviations. This is superimposed with the black plot corresponding to the number of states ie. 25. The lowest recognition accuracies are obtained for gestures 2, 3, 4, 12 and 13. These gestures can be seen to have the lowest average lengths, with values lower than 25. This implies that there are a considerable number of training instances of these gestures that are smaller than 25 frames and cannot be used for training an HMM with 25 states. This loss of training data leads to a sharp drop in accuracy. On the other hand, the accuracies also drop for gestures 6, 7 and 16, 17. These gestures have the highest average lengths, well over 25 frames. This shows that an HMM with 25 states is not sufficient to model the temporal variability of these gestures. Thus, there is a trade off involved while selecting the number of HMM states. Fig. 1 suggests that using 20 or 25 states for the HMM gives the best accuracies. We fix the number of HMM states to 20 for the rest of the experiments.

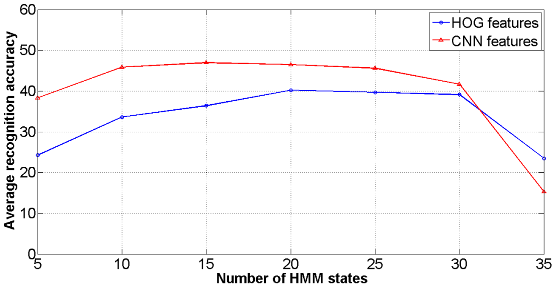

Fig. 1. Effect of varying number of HMM states

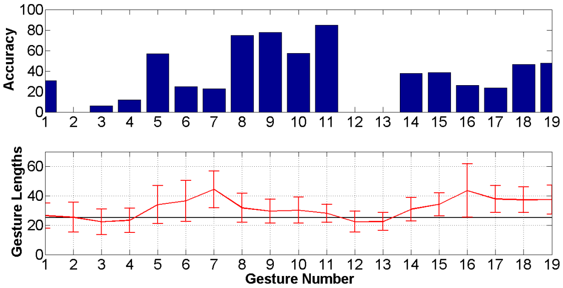

Fig. 2. Comparison of gesture wise recognition accuracies and average gesture lengths for 25 HMM states

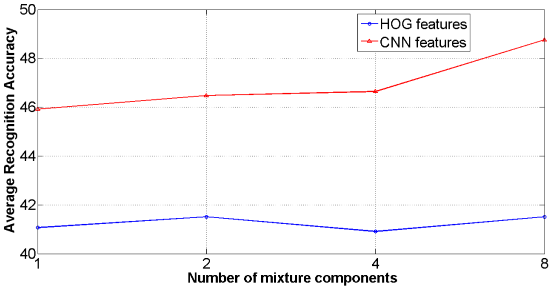

Fig. 3. Effect of varying number of mixture components

2) **Number of mixture components:** We vary the number of Gaussian mixture components per state of the HMM, keeping the number of states fixed at 20. The number of mixture components compared are 1,2,4

and 8. Fig. 3 shows the plot of recognition accuracy vs number of mixture components. We can see that the number of mixture components does not greatly affect the recognition accuracy. However, having a greater number of mixture components slows down the training and classification process which would be detrimental to a real-time gesture interface. Thus, we fix the number of mixture components to 2 for the rest of the experiments.

B. Comparison of features and modalities

We note from Fig.1 and Fig.3 that the CNN features consistently outperform the HOG features as the HMM hyperparameters are varied. This trend is also observed across modalities. The first two columns of Table II show the average recognition accuracies and standard deviations for the two features extracted from either the depth or the grayscale frames. In either case, the CNN features considerably outperform the HOG features. We also note that for either feature, the depth frames lead to better recognition accuracies than the grayscale frames. A reason for this could be that the depth frames are invariant under the illumination changes in the video, unlike the grayscale frames. The drop in the accuracy from depth to grayscale is much more severe for the HOG features, than the CNN features, suggesting that the CNN features could be more robust to illumination variation. The last column of the table shows accuracies for feature vectors formed by concatenating both the depth and grayscale features. For the HOG features, this leads to a slight drop in accuracy as compared to just using the depth frames due to the noise introduced by the badly performing grayscale HOG features. In case of the CNN features, however, using both modalities considerably improves the accuracy over using only depth or grayscale frames, suggesting that the two modalities contain complementary cues for hand gesture recognition.

Figures 4 and 5 show the confusion matrices for the best performing HMMs trained on HOG and CNN features respectively. In general, the CNN confusion matrix has much larger diagonal entries than the HOG confusion matrix, as well as fewer and smaller off-diagonal entries. This shows that the CNN features lead to better recognition accuracies for almost all the hand gestures. We also observe that certain specific errors made in case of the HOG features are considerably reduced in case of the CNN features. For example, gestures 1, 2, 3 and 4 are often confused with gestures 8, 9, 10 and 11 respectively in case of HOG features. These correspond to the swipe and corresponding scroll gestures as shown in Table I. These gestures are very similar, save for slight differences in hand posture while performing them. Similarly, gestures 5 and 6 viz. the 'swipe V' and 'swipe X' gestures are confused by the HOG features. Both of these errors are considerably reduced in case of the CNN features, suggesting that the CNN better encodes subtle variations in shape than the HOG features.

TABLE II
AVERAGE RECOGNITION ACCURACIES(%) AND THEIR STANDARD DEVIATIONS

Features	**Modality**		
	Depth	**Grayscale**	**Both**
HOG	**41.51 ± 11.56**	18.02 ± 7.22	38.27 ± 11.83
CNN	46.48 ± 10.31	39.14 ± 10.09	**54.76 ± 12.7**

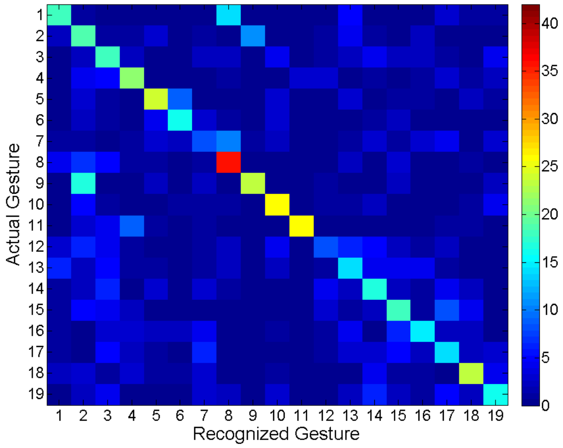

Fig. 4. Confusion matrix for HOG features

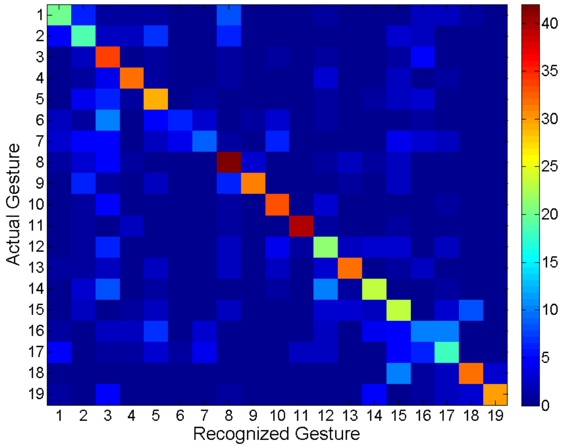

Fig. 5. Confusion matrix for CNN features

C. Reducing overfitting in the HMM:

We work with only the CNN features for the remainder of this paper, since the previous section shows that they clearly outperform HOG features in this framework

1) **Dimensionality reduction using PCA:** We subject the feature vectors to dimensionality reduction using PCA. We vary the number of principal components retained from 20 to 90 and plot the mean accuracies for the HMMs trained on them. Fig. 6 shows the results. We can see that when the number of retained

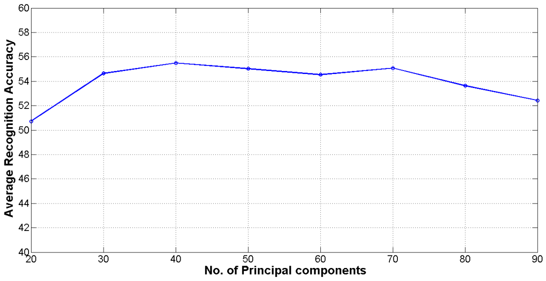

Fig. 6. Effect of varying number of retained principal components

principal components is reduced below 30, the mean accuracy begins to drop. This could be because we end up discarding useful information in the feature vectors. Also, as the number of principal components is increased beyond 70, we see a drop in accuracy since the models start to overfit. We obtain the best accuracy of 55.49% when we retain 40 principal components.

2) **Data Augmentation** We apply the data augmentation methods described in section II.D to increase the size of the training data. The ordering and orientation based transformations generate three new videos for each video in the dataset. The affine transforms generate two translated and two rotated videos for each video in the dataset. Table III shows the effect of data augmentation. We consider the effects of the two types of transformations separately for each modality, with PCA being applied and the first 40 principal components retained, in each case. We observe that the affine transformation based data augmentation improves the recognition accuracies across modalities. The results are more ambiguous in case of ordering and orientation based data augmentation. We get the best average recognition accuracy of 55.71% with the CNN features using both the depth and grayscale data and affine transformation based data augmentation.

D. CNN-HMM hybrid:

We use the HMM trained on both depth and grayscale data, with CNN features and affine transformation based data augmentation for generating the labels for the CNN-HMM hybrid. We use only 10 states per HMM to reduce the total number of output classes due to the limited size of the VIVA gesture dataset. We use Viterbi forced alignment for generating HMM state labels for each frame in the augmented training data. The labeled data is then used for training the final layer of the VGG-16 network. Table IV shows the average recognition accuracy and standard deviation for the CNN-HMM hybrid system compared with the best performing HMM using the CNN as a feature extractor. We get an improvement in recognition accuracy of about 1.5% with the CNN-HMM hybrid system. This can be attributed to the discriminative training in the CNN.

TABLE III

EFFECT OF DATA AUGMENTATION: AVERAGE RECOGNITION ACCURACIES AND THEIR STANDARD DEVIATIONS

Modality	**Without data augmentation**	**Ordering / Orientation transfromation**	**Affine transformation**
Depth	53.55 ± 12.23	49.03 ± 14.53	55.32 ± 14.13
Grayscale	42.28 ± 10.97	44.37 ± 7.73	46.75 ± 9.44
Both	55.49 ± 12.65	53.52 ± 10.11	**55.71 ± 10.40**

TABLE IV

AVERAGE RECOGNITION ACCURACY AND STANDARD DEVIATION FOR CNN-HMM HYBRID SYSTEM

CNN as feature extractor	**CNN-HMM Hybrid**
55.71 ± 10.40 %	**57.50 ± 13.05** %

IV. CONCLUSIONS

While the results obtained do not match the best accuracy of 77.5% reported on the VIVA hand gesture dataset [2], they do suggest that using HMMs with complex shape descriptors extracted from each video frame is a viable approach to modeling dynamic hand gestures. In particular, we showed that the number of states of the HMM seems to have a greater effect on how well the HMM models each gesture than the complexity of the mixture model for each state, and that features extracted from a trained CNN consistently outperform HOG features irrespective of whether the input is depth or visual data. Using PCA for dimensionality reduction and affine transformation based data augmentation methods improve the HMM performance by reducing overfitting. Finally, we showed that using the CNN-HMM hybrid system leads to further improvement in recognition accuracy as compared to using the CNN as just a feature extractor.

This approach would be worth further exploration, especially since HMMs do not require prior knowledge of the gesture boundaries, and can be run online as in continuous speech recognition. This alleviates the need for batch processing of the gestures as in [1], [2]. Future work could be targeted toward exploring a good gesture set for this framework. In particular compound gestures which are combinations of well defined smaller movements could be considered. These smaller movements could be modeled by HMMs which can then be concatenated to model the gesture. Another possible direction would be to explore the framework on a larger dataset, with greater time resolution, allowing us to model the gestures using a greater number of HMM states

V. ACKNOWLEDGMENTS

We would like to thank the reviewers for their constructive suggestions and comments. We would also like to thank our colleagues from the Laboratory for Intelligent and Safe Automobiles (LISA), UCSD, for their support and useful discussions and feedback.

References

[1] E. Ohn-Bar, and M. Trivedi. ”Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations.” Intelligent Transportation Systems, IEEE Transactions on 15, no. 6 (2014): 2368-2377.

[2] P. Molchanov, S. Gupta, K. Kim, and J. Kautz. ”Hand Gesture Recognition with 3D Convolutional Neural Networks.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1-7. 2015.

[3] N. Dalal, and B. Triggs. ”Histograms of oriented gradients for human detection.” In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886-893. IEEE, 2005. Harvard

[4] E. Ohn-Bar, and M. Trivedi. ”Joint angles similarities and HOG2 for action recognition.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 465-470. 2013.

[5] A. Klaser, M. Marszaek, and C. Schmid. ”A spatio-temporal descriptor based on 3d-gradients.” In BMVC 2008-19th British Machine Vision Conference, pp. 275-1. British Machine Vision Association, 2008. Harvard

[6] H. Wang, A. Klser, C. Schmid, and C. Liu. ”Dense trajectories and motion boundary descriptors for action recognition.” International journal of computer vision 103, no. 1 (2013): 60-79.

[7] M. Zobl, R. Nieschulz, M. Geiger, M. Lang, and G. Rigoll. ”Gesture components for natural interaction with in-car devices.” In Gesture-Based Communication in Human-Computer Interaction, pp. 448-459. Springer Berlin Heidelberg, 2003.

[8] F. Althoff, R. Lindl, L. Walchshausl, and S. Hoch. ”Robust multimodal hand-and head gesture recognition for controlling automotive infotainment systems.” VDI BERICHTE 1919 (2005): 187.

[9] T. Starner, J. Weaver, and A. Pentland. ”A wearable computer based american sign language recognizer.” In Assistive Technology and Artificial Intelligence, pp. 84-96. Springer Berlin Heidelberg, 1998.

[10] D. Minnen, and Z. Zafrulla. ”Towards robust cross-user hand tracking and shape recognition.” In Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, pp. 1235-1241. IEEE, 2011.

[11] M. Hu. ”Visual pattern recognition by moment invariants.” information Theory, IRE Transactions on 8, no. 2 (1962): 179-187.

[12] L. Baum, T. Petrie, G. Soules, and N. Weiss. ”A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains.” The annals of mathematical statistics 41, no. 1 (1970): 164-171. Harvard

[13] G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, and V. Valtchev. The HTK book. Vol. 2. Cambridge: Entropic Cambridge Research Laboratory, 1997. Harvard

[14] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. ”CNN features off-the-shelf: an astounding baseline for recognition.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806-813. 2014.

[15] K. Simonyan, and A. Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

[16] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior et al. ”Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups.” Signal Processing Magazine, IEEE 29, no. 6 (2012): 82-97.

[17] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn. ”Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition.” In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pp. 4277-4280. IEEE, 2012.