Driving Posture Recognition by Convolutional Neural Networks

Chao Yan, Bailing Zhang
Department of Computer Science & Software Engineering
Xi'an Jiaotong-Liverpool University
Suzhou, 215123, China

Frans Coenen
Department of Computer Science
The University of Liverpool
Liverpool, UK

***Abstract*—Driver fatigue and inattention have long been recognized as the main contributing factors in traffic accidents. Development of intelligent driver assistance systems with embeded functionality of driver vigilance monitoring is therefore an urgent and challenging task. This paper presents a novel system which applies convolutional neural network to automatically learn and predict four driving postures. The main idea is to monitor driver hand position with discriminative information extracted to predict safe/unsafe driving posture. In comparison to previous approaches, convolutional neural networks (CNN) can automatically learn discriminative features directly from raw images. In our works, a CNN model was first pre-trained by an unsupervised feature learning called using sparse filtering, and subsequently fine-tuned with four classes of labeled data. The Approach was verified using the Southeast University Driving-Posture Dataset, which comprised of video clips covering four driving postures, including normal driving, responding to a cell phone call, eating and smoking. Compared to other popular approaches with different image descriptor and classification, our method achieves the best performance with a overall accuracy of 99.78%.**

***Index Terms*—Driving posture recognition; Driving assistance system; Deep learning; Convolutional neural network**

I. Introduction

With the ever-growing traffic density, the number of road accidents is anticipated to further increase. Unsafe and dangerous driving accounts for the death of more than one million lives and over 50 million serious injuries worldwide each year [1]. Finding solutions to reduce road accidents and to improve traffic safety has become a top-priority for many government agencies and automobile manufactures alike. It has become imperative to the development of Intelligent Driver Assistance Systems(IDAS) which is able to continuously monitor, not just the surrounding environment and vehicle state, but also driver behaviours.

Previous works on vision-based automatic monitoring of unsafe driving behaviours can be categorized into three main streams of activities: (i) gaze and head poise analysis for the prediction of driver behaviour and intention [2], [3], [4], (ii) extraction of fatigue cues from diver facial image [5], [6], [7] and (iii) characterization (in the context of safe versus unsafe driving behaviour) of driver body postures, including the positioning of arms, hands and feet [8], [9], [10], [11]. Despite the encouraging performances under appropriate conditions, the proposed approaches share a common disadvantage of being ad hoc. Most of the vision-based methods follow a two-step framework: (i)extraction of hand-crafted features from raw data, usually with certain assumptions about the circumstances under which the data was taken, (ii)learning classifiers based on the obtained feature. Methods under such a framework cannot reach an optimal balance between the discriminability of the extracted features and the robustness of the chosen classier. The reason is the uncertainty of what features are important for the task at hand since the choice of features is highly problem-dependent in real-world scenarios.

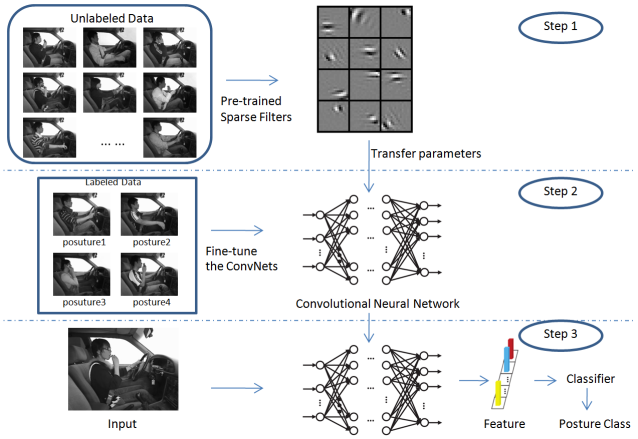


Fig. 1. The frameworks of our method.

Recently, there has been growing interest in the development of deep learning models for various vision tasks [12], [13], [14]. Deep learning models generally features of learning multiple layers of feature hierarchies, with increasingly abstract representations extracted at each stage. Such learning machines can be trained using either supervised or unsupervised approaches, and the resulting systems have been shown to yield competitive performance in speech recognition [15], natural language sentences recognition [16], visual classification task[17], [18], [19], [20], visual detection task[21], and other visual task[22], [23], [24], [25], [26].

One of the most successful deep learning models is the Convolutional Neural Network (CNN) [14], [17], a hierarchical multi-layered neural network able to learn visual patterns directly from the image pixels. In CNNs, small patches of the image (dubbed a local receptive field) are inputted to the

978-1-4673-7679-2/15/$31.00 ©2015 IEEE

first layer of the hierarchical structure. Information generally passes on the different layers of the network, and at each layer trainable filters and local neighborhood pooling operations are exploited in order to produce salient features for the data observed. In addition, the method provides a level of invariance to shift, scale and rotation as the local receptive field allows the processing unit access to elementary features such as oriented edges or corners. It has been repeatedly proved that CNN is powerful to learn rich features from the training set automatically.

In this paper, we apply Convolutional Neural Network architecture to represent and recognise driving postures, which aims at building high-level feature representation from low-level input automatically with minimal domain knowledge of the problem. Our work focus on the characterization of driving posture based on driver hand position, with high-level features extracted hierarchically using the convolutional layers and the max-pooling layers directly from raw input image. Each convolutional layer generates feature maps using sliding filters (templates) on a local receptive field in the maps of the preceding layer (input or max-pooling layer). The map sizes decrease layer by layer such that the extracted feature becomes more complex and global. Then, the output is inputted to a fully connected multilayer perceptron (MLP) classifier. The proposed approach was evaluated on the Southeast University Driving-Posture Dataset[5], demonstrating competitive performance.

The key contributions of this work can be summarized as follows:

1) To recognise driving posture, this paper proposed to build a deep convolutional neural network in which trainable filters and local neighborhood pooling operations are applied alternatingly for automatically exploring salient features. Using CNN to learn rich features from the training set is more generic and requires minimal domain knowledge of the problem compared to hand crafted feature in previous approaches.
2) We using sparse filter to pre-train the filters in our networks, with advantages including (i)acceleration of training for faster convergence, (ii) a better generalization and performance. In addition, we setup experiment to evaluate the CNN architecture selection, with max-pooling and ReLU identified as better options for pooling operation and activation function, respectively.
3) The proposed approach was evaluated on the Southeast University Driving-Posture Dataset, with best performance achieved with an overall accuracy of 99.78%.

The rest of the paper is organized as follows. Section II presents an overview of our proposed method and the SEU driving posture dataset, while Section III gives a detailed introduction to the convolutional neural network followed by the training details in Section IV. Section V reports the conducted evaluation and the experiment results, followed by some conclusions presented in Section VI.

II. System Overview

The proposed driving posture recognition system comprises three steps: (i)unsupervised pre-train the network with unlabeled data (ii)fine-tune the network with four class of labeled data, (iii) use the network to extract feature from input for classification . A schematic illustrating the operation of the proposed driving posture recognition system is shown in Fig.1.

A. Southeast University Driving-Posture Dataset

Fig. 2. Example images of from the driving dataset. The first column is normal driving posture; The second column is the posture of operating the shift gear; The third column is the posture of eating or smoking; The forth column is the posture of responding a cell phone

To test the proposed driving posture recognition approach, the Southeast University Driving-Posture Dataset(SEU dataset) was used. This data was first created by Zhao [5]. Each video included in the dataset was obtained using a side-mounted Logitech C905 CCD camera under day lighting conditions with a resolution of 640×480. Ten male drivers and ten female drivers participated in the creation of the dataset. Each video was recorded under normal day light conditions, poor illuminated night time conditions were not considered. We extract all frames in there videos and manually labeled four pre-defined posture including :

1) normal driving (posture1)
2) operating the shift gear (posture2)
3) eating and smoking (posture3)
4) responding a cell phone (posture4)

Some selected samples are shown in Fig.2. Each posture from (1) to (4) contains 46081, 12000, 18181, 16211 samples, respectively.

III. Deep Convolutional Neural Network Architecture

The overall convolutional net architecture is shown in Fig.3. The network consists of three convolution stages followed by three fully connected layers. Each convolution stage includes convolutional layer, non-linear activation layer, local response normalization layers and max pooling layer. The non-linear activation layer and local response normalization layers were not illustrated in Fig.3 as data size was not changed in these two layers. Using shorthand notation, the full architecture is C(12,5,1)-$\tilde{A}$-N-P-C(16,5,1)-$\tilde{A}$-N-P-C(20,4,1)-$\tilde{A}$-N-P-FC(512)-$\tilde{A}$-FC(128)-$\tilde{A}$-FC(4)-$\tilde{A}$, where C(d,f,s) indicates a convolutional layer with d filters of spatial size $f \times f$, applied

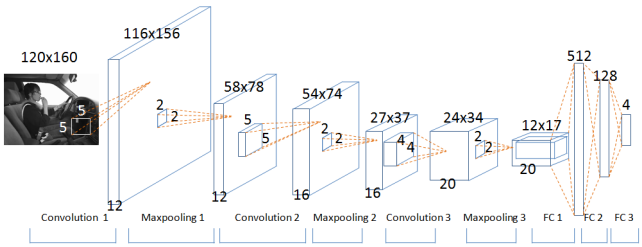


Fig. 3. The architecture of our unsupervised convolutional neural network. The network contains three stages, each of which is consisted of convolution layer, non-linear activation layer, local response normalization layer, and max-pooling layer. Only convolution and max-pooling layers which change the data size during operating, are illustrated here.

to the input with stride s. $\tilde{A}$ is the non-linear activation function, which uses ReLU [27] activation function. $FC(n)$ is a fully connected layer with n output nodes. All pooling layers P use max-pooling in non-overlapping 2×2 regions and all normalization layers N are defined as described in Krizhevsky et al. [17] and use the same parameters: $k = 2$, $n = 5$, $\alpha = 10^{-4}$, $\beta = 0.5$. The final layer is connected to a softmax layer with dense connections. The structure of the networks and the hyper-parameters were empirically initialised based on previous works using ConvNets, then we setup cross-validation experiment to optimize the selection of network architecture in sectionV-A.

IV. Training Details

In this section, We first briefly introduced how to apply back-propagation technique to optimize the parameters in CNN architecture. Then we discussed the pre-train problem. Finally, we give the implementation details including hardware, software, regularization rules, and parameter updating details in our experiment.

A. Learning through Back-propagation

We use Back-propagation[28] to propagate errors in the network back through the feedforward architecture and hence to train the weights based on back propagated errors in each layer. Training the convolutional neural network is composed of two steps: (i)feed forward the training data through the network till the final output layer, and finally calculate the error/loss (ii)back propagate the error/loss layer by layer from top to bottom, and therefore calculate the gradients and update the weights in respective layers based on the back propagated errors.

With respect to the loss estimation based on the output from the final layer, cross-entropy loss is widely used in recent CNN architectures[17], [18], [19], [20], [25], [26], which has been demonstrating better in avoiding learning slowdown than conventional mean square error. In the output layer, the cross-entropy loss function is given by :

$$L = -\sum_{j=1}^{K}[t^j log(p^j) + (1 - t^j) log(1 - p^j)] \tag{1}$$

where x^j and p^j are the j-th input and output respectively, K is the number of output neuron(class) and t^j is the j-th class's one-hot encoded target label.

When performing back propagation, the first step is to calculate the loss gradient by partial derivative as follows:

$$\begin{aligned} \delta^j = \frac{\partial L^j}{\partial p^j} \cdot \frac{\partial p^j}{\partial x^j} &= (-\frac{t^j}{p^j} + \frac{1 - t^j}{1 - p^j}) \cdot p^j(1 - p^j) \\ &= \frac{p^j - t^j}{p^j(1 - p^j)} \cdot p^j(1 - p^j) \\ &= p^j - t^j \end{aligned} \tag{2}$$

where δ^j is the loss gradient in the output layer, which will be back propagated to the topmost full connection layer as an error.

In an l-th full connection layer, an error ${\delta^j}_{l+1}$, is back propagated from an upper $(l+1)$-th layer, then the error ${\delta^i}_l$ and the weight gradient ${\Delta w^{ij}}_l$ in this layer is given by Equ.3 and Equ.4, respectively.

$${\delta^i}_l = \sum_{j=1}^{n} {w^{ij}}_l \cdot \Delta {f_{acti}(\cdot)}_l \cdot {\delta^j}_{l+1} \tag{3}$$

$${\Delta w^{ij}}_l = {x^i}_l \cdot \Delta {f_{acti}(\cdot)}_l \cdot {\delta^j}_{l+1} \tag{4}$$

where w^{ij} is the $m \times n$ weight matrix, n and ${x^i}_l$, is the total number of the output neuron and input neuron when feed-forwarding,respectively. $\Delta {f_{acti}(\cdot)}_l$ is the gradients of the nonlinearity activation function, the error ${\delta^i}_l$ will be back propagated to the lower $(l-1)$-th layer.

B. Pre-train

CNN architecture strongly depends on large amounts of training data for good generalization. When the amount of labeled training data is limited, directly training a high capacitor CNN from only a few thousand training images is problematic. Researches [29] have shown an alternative solution to compensate the property of CNN, that is, choosing a optimised stating point which can be pre-trained by transferring parameters either supervised or unsupervised, as opposite to random initialized start.

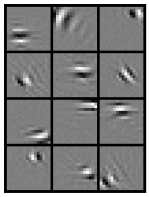


Fig. 4. The unsupervised pre-trained sparse filters of the first convolution layer

To speed up the training for faster convergence, we introduce the Sparse Filtering [30] to learn the filter in each convolution layer of our network. Sparse filtering is an unsupervised feature learning algorithm which optimizes for the sparsity in the feature distribution and avoids explicit modeling of the data

distribution. The details can be found in [30]. Compare to other unsupervised technique such as Autoencoder, Sparse coding and Sparse RBMs, sparse filtering is easy to implement and hyperparameter free. In experiment, we randomly extracted patches size of 5×5 from the dataset video and the sparse filtering method is utilized to learn the filters. The learnt filters of the first layer are shown in Fig.4. As we expected, the filters mainly preserve the edge, point and junction information of the driver.

C. Implementation Detail

We train our models using stochastic gradient decent with a batch size of 128 examples, momentum of 0.6, and weight decay of 0.0005. The learning rate is initialized as 0.01 for all trainable layers and adapted during training. How to adaptively control the learning rate in a reasonable value is an important issue in CNN learning. A too small learning rate makes the convergence rather slow, while a too big learning rate would make the network parameters vibrated. We proposed a adaptive learning rate by monitoring the loss function value and the validation error. To further prevent possible overfitting, we apply dropout and data augmentation as performed in [17].

The experiments were implemented on our GPU CNN package in C++ language based on NVIDIA CUDA and cuDNNv2. Our experiments are conducted on a NVIDIA GTX Titan GPU and a 4-core Intel(R) Core i7-3770 3.40-GHz computer.

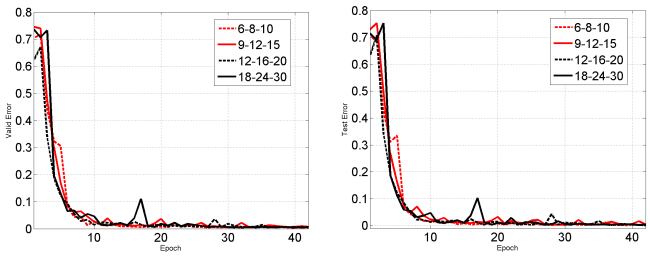
V. Experiment

In this section, we first conduct three evaluation experiments to select the CNN structure, activation function, pooling method and other hyperparameters in sectionV-A. Then the CNN model is applied to verify the effectiveness of the proposed algorithm on the SEU driving posture database in sectionV-B. Finally, some other hand-coded feature approaches that applied on SEU data are compared in sectionV-C.

The SEU driving posture dataset contains twenty videos. All experiment are conducted using five-fold cross-validation. The original twenty videos are randomly divided into five folds, each contains four videos. In five-fold cross-validation , one of the folds is retained for testing while the remaining four folds are used for training. The cross-validation process will then be repeated five times, which means that each of the fold will be used exactly once as the testing data. In each training time, 5% of the training data are randomly selected as validation data.

A. Architecture Selection

This sub-section introduces three evaluation experiments, which are used to select network architectures including convolution layer capacity, nonlinear activation function, and pooling method. At first, we empirically choose a default architecture as C(9,5,1)-$\tilde{A}$-N-P-C(12,5,1)-$\tilde{A}$-N-P-C(15,4,1)-$\tilde{A}$-N-P-FC(512)-$\tilde{A}$-FC(128)-$\tilde{A}$-FC(4)-$\tilde{A}$, the notation meaning has been explained in sectionIII. Then we use cross-validation to optimize all the hyperparameters one by one.



(a) Validation Error (b) Testing Error

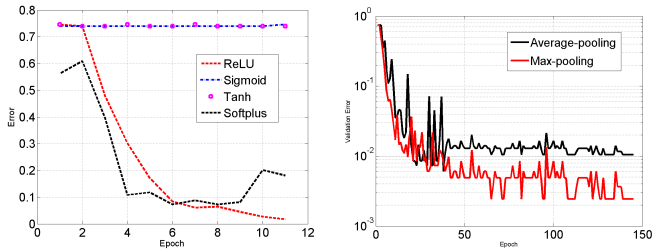
Fig. 5. Selecting filter numbers in each convolution layer

1) Architecture Capacity: According to learning theory, if the architecture has too much capacity, it tends to overfit the training data and has poor generalization. If the model has too little capacity, it underfits the training data, and both the training error and the test error are high. The capacity here means (i)the depth of our deep convolutional neural networks, and generally stands for the number of repeated convolution stages, and (ii)the width of our network, which means the filter numbers in each convolutional layer.

The depth is mostly depends on the size of raw input image and the complexity of the task. Compare to those 10,000 face image classification [26] task, the complexity of our work is lower. The original image is 480×640. To save computation resources, we resize the image directly to 120×160, which still holds enough discriminative information to classify the posture in human level perception. We select the filter size of 5×5 for first and second convolution layers ,and the filter size of 4×4 for the third convolution layers. Each pooling layer that follows the convolution layer, uses a non-overlapping 2×2 kernels as most CNN-based approaches. Therefore, the output feature map is 12×17 after three stages of convolution as shown in Fig.3. We think the resolution of feature map is small enough to terminate convolution, because a continuing convolution is meaningless and wasting computation resources.

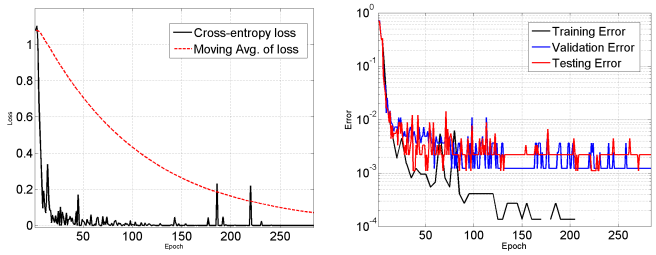
With respect to the width of the network. We empirically set the filter size as 9, 12 and 15 in the three convolution layers respectively. The pooling layer which follows the convolution layer, will decrease resolution of the feature map. To prevent the information from being lost too quickly, the filter size will generally increased. Here, we setup experiment to compare with three other groups of filter number, that is 6-8-10, 12-16-20 and 18-24-30, where the notation of x-y-z means x number filters, y number filters and z number filters in the fist, second and third convolution layers respectively.

The validation error and testing error with epoch are shown in Fig.5(a) and Fig.5(b),respectively. From the figure, the network using four groups of filters, converges since 10 epoches and is stable after 20 epoches. However, the black dash line which stands the filter number group of 12-16-20, holds a minor early convergence compared to other three groups. Hence, we choose this group as our filter numbers in each convolution layer.



(a) Validation error of four activation functions (b) Validation error of two pooling method

Fig. 6. Selecting activation function and pooling method



(a) Loss value (b) Errors

Fig. 7. Plots of four activation functions

2) Nonlinear activation function and pooling method: In this section , we evaluate the nonlinear activation function and pooling method in our network. The ReLU has been reported faster convergence than conventional sigmoidal functions, we re-implemented these activation functions, the validation error with epoch are illustrated in Fig.6(a). In the figure, we plot the first ten epoches, which demonstrates a higher training speed of ReLU and softplus. ReLU is reported that gives lower test error than softplus after both convergence in [27]. But we did't observe this point in our dataset within limited epoches.

With respect to pooling method. The validation error with epoch for both pooling method is plotted in Fig.6(b). It is obvious that max-pooling yields a better performance.

B. Experiment Results

As above architecture selection in sectionV-A, we finalize our convolutional neural network for training and testing on the SEU dataset. The experiment is conducted using five-fold cross-validation as told previously. The cross-validation process will then repeated five times. The first time result is illustrated in Fig.7. In Fig.7(a), the black line the global loss versus each epoch, the red line is the moving average of the black line with a learning rate of 0.01. In Fig.7(b), the training error, validation error and testing error are plotted in black, blue and red respectively. The training data are tested as a whole each five epoches and we stop to test training error after 210 epoches. The network is convergence after 250 epoches.

After we repeated five cross-validation experiments, to further evaluate the classification performance, confusion matrix is used to visualize the discrepancy between the actual class labels and predicted results from the classification. Confusion matrix gives the full picture at the errors made by a classification model. The confusion matrix shows how the predictions are made by the model. The rows correspond to the known class of the data, that is, the labels in the data. The columns correspond to the predictions made by the model.The value of each of element in the matrix is the number of predictions made with the class corresponding to the column,for example,with the correct value as represented by the row. Thus, the diagonal elements show the number of correct classifications made for each class, and the off-diagonal elements show the errors made. The confusion matrices of the our results are shown in TableI. The accuracy for normal driving , operating shit gear, eating or smoking and responding a cellphone are 99.47%, 100%, 100% and 99.55% respectively.

TABLE I
CONFUSION MATRIX FOR THE CROSS VALIDATION RESULT

class	pose1	pose2	pose3	pose4
pose1	99.47	0	0.53	0
pose2	0	100	0	0
pose3	0	0	100	0
pose4	0	0	0.45	99.55

C. Comparison

The performance on SEU driving posture dataset using convolution neural networks is evaluated in sectionV-B ,and in this section it will be compared with other six hand-crafted methods including : (i)the baseline method which represents the posture pattern by contourlet transform on skin region[5]; (ii) mutiwavelet transform method to represent skin region and using MLP for classification[31]; (iii)an classifier ensemble method[32]; (iv)a bayesian classifier approach[33]; (v)PHOG descriptor followed by support vector machine and (vi) SIFT descriptor[34] followed by support vector machine.

TABLE II
CLASSIFICATION ACCURACY COMPARED WITH OTHER SIX APPROACHES

	pose1	pose2	pose3	pose4	Avg.
Baseline[5]	97.70	87.55	85.95	89.30	90.63
WT[31]	97.52	92.77	88.99	83.02	89.23
RSE[32]	99.95	91.20	99.20	87.42	94.20
Bayes[33]	94.82	95.20	98.26	92.77	95.11
PHOG+SVM	**99.83**	88.71	89.12	73.20	91.56
SIFT+SVM	99.40	93.52	94.55	91.21	96.12
Proposed	99.47	**100**	**100**	**99.55**	**99.78**

For fair comparison, we have re-implemented the methods in [5], [31], [32] and [33] and carried our experimented on these methods, two popular vision descriptor approaches and our convolutional neural network method on the same training and testing dataset within the SEU driving posture dataset. The ten times repeated overall results are shown in TableII. The results clearly show that our approach outperforms other hand-crafted feature approaches.

VI. Conclusion

This paper addresses the importance of automatic understanding and characterization of driver behaviours in the scenario of reducing motor vehicle accidents, and presents a novel system for vision-based driving behaviour recognition. We verify our approach on the SEU driving dataset which includes postures of normal driving, operating the shift gear, eating or smoking, and responding a cell phone. The proposed approach applied deep convolutional neural network, which learns feature from raw image automatically. We have described the details of each layer in our network and evaluated the selection of networks architecture and hypeparameters. The final results demonstrate better performance than hand-coded features including four previous approaches and two popular feature descriptor approaches, achieving an overall accuracy of 99.78%.

References

[1] Online, "Who world report on road traffic injury prevention," 2004, http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/en/.

[2] P. Watta, S. Lakshmanan, and Y. Hou, "Nonparametric approaches for estimating driver pose," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 2028–2041, July 2007.

[3] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 300–311, June 2010.

[4] A. Doshi and M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 453–462, Sept 2009.

[5] C. Zhao, B. Zhang, J. He, and J. Lian, "Recognition of driving postures by contourlet transform and random forests," *Intelligent Transport Systems, IET*, vol. 6, no. 2, pp. 161–168, June 2012.

[6] L. Bergasa, J. Nuevo, M. Sotelo, R. Barea, and M. Lopez, "Real-time system for monitoring driver vigilance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 63–77, March 2006.

[7] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1052–1068, July 2004.

[8] S. Y. Cheng, S. Park, and M. M. Trivedi, "Multi-spectral and multi-perspective video arrays for driver body tracking and activity analysis," *Computer Vision and Image Understanding*, vol. 106, no. 2C3, pp. 245–257, 2007.

[9] H. Veeraraghavan, N. Bird, S. Atev, and N. Papanikolopoulos, "Classifiers for driver activity monitoring," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 1, pp. 51–67, 2007.

[10] S. Cheng and M. Trivedi, "Vision-based infotainment user determination by hand recognition for driver assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 759–764, Sept 2010.

[11] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and prediction of driver behavior by foot gesture analysis," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 435–445, 2012.

[12] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[13] Q. Le, W. Zou, S. Yeung, and A. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 3361–3368.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[15] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[16] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2042–2050.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[18] J. Krause, T. Gebru, J. Deng, L.-J. Li, and L. Fei-Fei, "Learning features and parts for fine-grained recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, Aug 2014, pp. 26–33.

[19] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 568–576.

[20] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, "Panda: Pose aligned networks for deep attribute modeling," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1637–1644.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 580–587.

[22] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[23] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 1385–1392.

[24] D. Yi, Z. Lei, S. Liao, and S. Li, "Deep metric learning for person re-identification," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, Aug 2014, pp. 34–39.

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1701–1708.

[26] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1891–1898.

[27] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks." *Journal of Machine Learning Research*, vol. 15, no. Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, AISTATS 2011, pp. 315–323, 2011.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors." *Nature*, vol. 323, no. 6088, p. 533, 1986.

[29] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?." *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[30] J. Ngiam, Z. Chen, S. A. Bhaskar, P. W. Koh, and A. Y. Ng, "Sparse filtering," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 1125–1133.

[31] C. Zhao, Y. Gao, J. He, and J. Lian, "Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1677 – 1686, 2012.

[32] C. Zhao, B. Zhang, X. Zhang, S. Zhao, and H. Li, "Recognition of driving postures by combined features and random subspace ensemble of multilayer perceptron classifiers," *Neural Computing and Applications*, vol. 22, no. 1, pp. 175–184, 2013.

[33] C. Zhao, B. Zhang, and J. He, "Vision-based classification of driving postures by efficient feature extraction and bayesian approach," *Journal of Intelligent and Robotic Systems*, vol. 72, no. 3-4, pp. 483–495, 2013.

[34] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.