A Visual-Based Driver Distraction Recognition and Detection Using Random Forest

Amira Ragab(✉), Celine Craye, Mohamed S. Kamel, and Fakhri Karray

Center for Pattern Analysis and Machine Intelligence Electrical and Computer Engineering Department, University of Waterloo, 200 University Avenue, Waterloo, ON N2L 3G1, Canada
{amira.ragab,ccraye,mkamel,karray}@uwaterloo.ca

Abstract. Driver distraction and fatigue are considered the main cause of most car accidents today. This paper compares the performance of Random Forest and a number of other well-known classifiers for driver distraction detection and recognition problems. A non-intrusive system, which consists of hardware components for capturing the driver's driving sessions on a car simulator, using infrared and Kinect cameras, combined with a software component for monitoring some visual behaviors that reflect a driver's level of distraction, was used in this work.

In this system, five visual cues were calculated: arm position, eye closure, eye gaze, facial expressions, and orientation. These cues were then fed into a classifier, such as AdaBoost, Hidden Markov Models, Random Forest, Support Vector Machine, Conditional Random Field, or Neural Network, in order to detect and recognize the type of distraction. The use of various cues resulted in a more robust and accurate detection and classification of distraction, than using only one. The system was tested with various sequences recorded from different users. Experimental results were very promising, and show the superiority of the Random Forest classifier compared to the other classifiers.

1 Introduction

Many efforts have been made recently to ensure the driver safety and to decrease car accidents. According to [12], around 80% to 90% of accidents involving fatalities or injuries are mainly related to the driver's absence of alertness. Specifically, the driver's alertness is affected by distraction and fatigue. In order to detect whether the driver is distracted or fatigued, many car manufacturing companies have started to embed audio-visual sensors in intelligent vehicle systems. These sensors are either intrusive or non-intrusive, and the non-intrusive systems are much more appealing to drivers for their naturalness.

This paper studies the classification performance of various well-known classifiers in a non-intrusive computer vision system for monitoring drivers distraction. This system started by capturing the driver sessions while driving a car simulator, followed by a feature extraction module, which consisted of five sub-modules:

A. Campilho and M. Kamel (Eds.): ICIAR 2014, Part I, LNCS 8814, pp. 256–265, 2014.
DOI: 10.1007/978-3-319-11758-4_28

analyzing eye gaze and closure, arm position, facial expressions, and facial orientation. Finally, the extracted features were merged together and classified using a number of well-known classifiers, such as AdaBoost, Hidden Markov Models, Random Forest, Support Vector Machine, Conditional Random Field, and Neural Network. Experimental results from six subjects were promising for both detection and recognition problems (82.9% accuracy for the type of distraction and 90% for distraction detection).

The rest of the paper is organized as follows: Section 2 discusses related work, then the system used is described in Section 3. Section 4 depicts the experiments and results. Conclusions and future work are presented in Section 5.

2 Related Work

Typically, the main causes of driver inattention are distraction and fatigue. However, according to the study in [7], the main contributor for 10% to 25% of vehicle accidents is distraction, which is our main focus in this work. According to [1], distraction can be classified into three main categories:

- Visual: The driver takes his eyes off the road for some reason, such as reading or watching a video.
- Manual: The driver takes his hands off the wheel for some reason, such as text messaging, eating, using a navigation system, or adjusting the radio.
- Cognitive: The driver's mind is taken away from driving. This can happen when talking on the phone, text messaging, or simply thinking.

Generally, systems for detecting driver distraction are non-intrusive (i.e., do not require attaching cumbersome devices to the driver). These systems detect distraction based on driver's behavior using camera(s), driving or car behavior using sensors that measure steering, braking, lane keeping, etc. or both.

A wide range of sensors and classifiers have been utilized in the literature for capturing and detecting driver distraction. In [6], Neural Network, with a back propagation algorithm and 80 nodes in the hidden layer, was used to detect distraction using eye closure only. Murphy-Chutorian et al. [9] used Support Vector Machine with Localized Gradient Orientation histograms to estimate the orientation of the driver's head. Earlier in [11], driver visual attention was modeled with three independent Finite State Machines, in order to monitor both eye and head movements.

Recently, Butakov et al. [4] suggested using a Gaussian Mixture Model to analyze the driver or vehicle response in the vehicle following case to create a normal behavior model, which can then be used to detect distraction if the driving behavior deviates from the saved model. In [13], two subsets of features were extracted. The first one included accelerator pedal position and steering wheel position, while the second subset included both of these elements, as well as the Collision Avoidance Systems (CAS) sensors (lane boundaries and upcoming road curvatures). This data was then classified using Random forest (with 75 trees). The results revealed that adding the CAS sensors features increased

the accuracy considerably. However, depending only on driving behavior can be misleading, as it can be affected by external factors such as driver experience, road type, weather, and outdoor lighting.

A more promising way for modeling driver distraction is to combine information from both the driver and driving behaviors. Liang et al. [8] extracted the driver's eye movements as well as driving performance data, such as lane position, steering wheel angle, and steering error calculated from steering wheel angle, to capture distraction. This data was then classified using Support Vector Machine. A distraction detection system which infers visual driver information about head position, head pose, and eye pose, as well as car information using a lane-keeping module, was presented in [10]. No training was included in this system, however.

Almost none of the works in the literature have aimed to detect the type of distraction created by the driver, and have instead focused only on recognizing whether the driver is distracted or not. Determining the type of driver distraction provides higher level information which can be used for a number of applications related to intelligent transportation systems. The applications can be implemented in smart cars to provide statistics on the driver's behavior, which could increase the help the vehicle can provide to keep the driver safe.

3 Methodology

The non-intrusive system used in this study consisted mainly of three phases: (1) the data acquisition phase, during which the driving sessions were recorded, (2) the feature extraction phase, during which certain features that reflect distraction were extracted, and (3) the classification phase, during which a classification model was learned using the extracted features.

3.1 Data Acquisition

In this phase, the driving sessions from six drivers of different ethnic backgrounds, genders, ages, and with or without glasses, were recorded. Driving sessions were captured using infrared (IR) and Kinect cameras mounted in front of the driver while he or she drove a car simulator. Each driver was first introduced to the car driving simulator, during which time they were asked to drive for a few minutes in order to familiarize themselves with the simulator. Then, during the driving sessions, instructions for each of the different actions were displayed on the screen. Four driving sessions were recorded for each subject. Each session lasted for around ten minutes. The actions involved in the experiments were a phone call, a text message, drinking, object distraction, and normal driving. For each driver, normal driving represented around 40% to 50% of each sequence, while the remaining were distraction actions. Each of the distraction actions represented between 10% to 20% of the each sequence. An image of the driving simulator is shown in Fig.1.

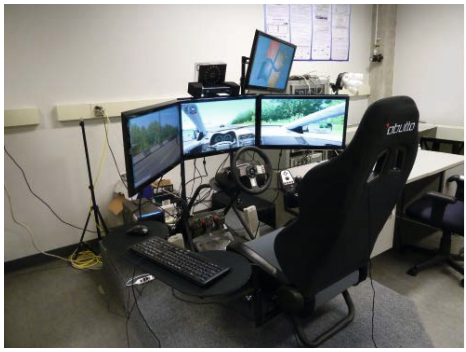

Fig. 1. Driving simulator used in the experiments

3.2 Feature Extraction

The feature extraction module consisted of five main sub-modules:

- **Arm Position.** After segmenting the body from the background by combining the output from the Kinect segmentation with the output from the background removal, the arm position was represented using the segmented depth map acquired using Kinect. Since Kinect records drivers with a frontal view, their right arm is therefore on the right side of their body. Based on this, the features were extracted based on foreground contours. First, the marching squares algorithm was applied to the binary foreground image, which outputs an ordered list of contour pixels. Then, the left section of the contour was removed (since the right arm was the one used for the distraction actions). The remaining "half" contour was then divided into twenty successive segments. Using the depth map, each pixel of the contour was associated with a 3D point, such that each segment of the contour corresponded with a 3D point cloud. For each point cloud, a principal component analysis was applied, and the eigenvector of the main principal component was kept.
  However, using only the right contour from the frontal view was not enough, as some actions, such as texting, cannot be detected from the frontal view. In order to overcome this problem, the aforementioned approach was applied to the profile view as well, resulting in a 120 feature vector. This vector was then fed into a 1-vs-all AdaBoost to create a model which was used to classify the rest of the data. The output of the classifier was a feature vector of size 4, which represented the estimated position among four possible states: arm up, arm down, arm right, and arm forward, as depicted in Fig.2.
- **Facial Orientation.** First, the face was extracted using the face tracking algorithm provided by Kinect SDK [2]. Then, based on the coordinates of the face 3D vertices, the face tracking provided a feature vector of size 3, which represented the head orientation angles, namely the pitch, roll, and yaw angles, whose values were between -180 and 180 degrees.

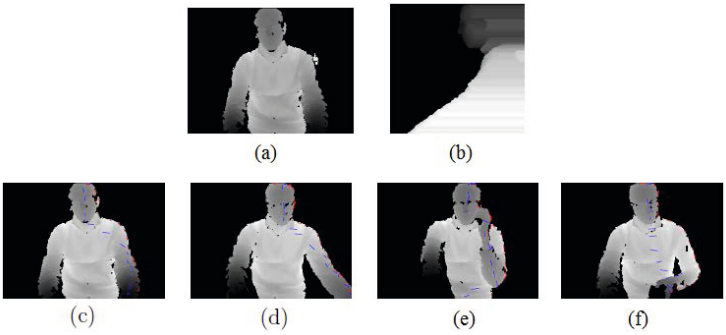

Fig. 2. An example of (a) frontal view and its (b) associated profile view, as well as their forward, right, up and down arm positions and associated features in (c)-(f) respectively. In (c)-(f) the red dots represent projections of a point clouds' local orientations from the frontal view, while blue dots are from the profile view.

- **Facial Expressions.** The face tracking algorithm provided by Kinect SDK was also used to provide four animation units (AUs). AUs were expressed as coefficients, and represented how strongly distorted features of the face were. The four AUs extracted were the ones related to the mouth only, such as upper lip raiser (AU10), jaw lowerer (AU26/27), lip stretcher (AU20), and lip corner depressor (AU13/15).
- **Eye Gaze.** First, the eye position was extracted using the SDK face tracking algorithm. Then, an efficient iris detection method based on cost function maximization and spatio-temporal considerations was applied. The cost function was the result of two main filters: circular Hough transform and circular Gabor filter. The cost function was also inspired by the filter introduced by [8], and depends on the high intensity difference between the iris and its neighborhood. Finally, the iris center was estimated as the summation of the three normalized filter responses. Then, an approximate gaze estimation was carried out by calculating the position of the iris relative to the eyes' corners. The output of this module was a feature vector of size 4.
- **Eye Closure.** In order to determine whether the eye was opened or closed, a database of opened and closed eyes was constructed. In turn, this database constructed an SVM model with Radial Basis Function (RBF) kernel, which was used afterward to classify the data. The output of this module was the decision for each eye: open(1), closed(0), or something else(-1).

3.3 Classification

Both sequential and non-sequential classifiers were deployed in this work. The non-sequential classifiers used were the Support Vector Machine (SVM), the Random Forest (RF) and the AdaBoost (Adaptive Boosting). The strength of the SVM mainly depends on the selection of the kernel, as well as its parameters.

In this work, the best value for C was selected by searching with the exponentially growing sequences of C, e.g., $C\epsilon\{10^{-2}, 5^{-1}, ..., 5^{3}, 10^{2}\}$. A C-SVM was deployed for its efficiency, and after experiments with the different kernels, the Radial Basis Function kernel was chosen for producing the best results. Random Forest is an ensemble of many decision trees, and its strength relies on combining diverse classifiers. In this work, several values for the number of trees were experimented and the size 75 was chosen. Whereas the number of features used to train each tree was set to $\sqrt{M}$ (where M is the total number of features), as proposed by Breiman [3]. For the Adaboost, a simple real 1-vs-all AdaBoost initialized with a decision tree of depth four and 300 iterations, was used.

Sequential classifiers which predict sequences of labels for sequences of input samples, such as Hidden Markov Model (HMM), Conditional Random Field (CRF) and Neural Networks (NNs) were also experimented with. For the HMM, a different Markov model was trained for each class, and the Viterbi algorithm was used to decide which state each sample belongs to. The best value for the number of hidden states is chosen experimentally. For the CRF, the CRF++ library [1] was utilized in this work. Since this library does not handle continuous features, the features were quantized using a simple quantization method. Finally, the nonlinear property of the NNs allows them to solve some complex problems more accurately than linear methods. Recurrent NNs, with Levenberg-Marquardt training function and hidden layer of size 10 neurons, were chosen in this work, since they have proved their superiority to feedforward networks in modeling time series data with lower errors [5]. However, Recurrent NNs are not suitable for large datasets, so we had to randomly sample the dataset to reduce its size.

4 Experiments and Results

As explained earlier, data was collected from sessions recorded for six subjects. The features from the five different aforementioned sub-modules were combined to form a feature vector of length 17 (i.e. 4 + 3 + 4 + 4 +2). Then, a median filter with a sliding window of size 100 was used to temporarily smooth this feature vector. Also, the standard deviation within the hundred-sample window was computed. The resulting feature vector of length 34 is then classified.

Due to the randomness of the RF and the random sampling used in the NN, the experiments involving the RF and NN were repeated for 5 runs. Then the accuracy average of the runs, as well as the standard deviation, were calculated. Both classification recognition (five classes) and detection (two classes) were computed. The performance measures used to evaluate the system and compare between the different classifiers were accuracy, specificity, precision, recall, f-measure, g-means, and prediction time/sample in msec.

[1] The used library is available at http://crfpp.googlecode.com/svn/trunk/doc/index.html

4.1 Driver Distraction Recognition

Driver distraction was recognized using AdaBoost, RF, CRF, HMM and Recurrent NN classifiers. The evaluation protocol was leave-one-subject-out, wherein each classifier was trained using all sessions except that of the driver to be evaluated, and tested using all sessions involving this driver. A comparison between the performance of the different classifiers for the distraction recognition problem per subject is shown in Table 1. The first six rows show the accuracy for each of the six subjects, while the next rows show the overall performance.

Table 1. A per subject comparison between the different classifiers for the distraction recognition problem

	CRF	**HMM**	**AdaBoost**	**RF**	**NN**
1	68.53	75.98	90.38	88.36±1.59	48.23±14.59
2	73.07	89.29	89.16	89.23±0.13	73.61±2.96
3	66.08	86.6	82.21	81.17±1.03	68.57±3.77
4	70.68	81.41	82.75	76.09±1.55	72.94±8.78
5	73.49	81.78	79.67	81.92±0.94	74.69±2.36
6	53.55	62.01	73.64	78.81±0.87	72.26±2.32
Accuracy	67.57	79.5	**82.97**	**82.78±0.07**	68.38±3.02
Specificity	71.97	84.62	87.26	86.81±0.16	71.61±2.49
Precision	37.47	37.47	43.54	42.26±0.43	19.38±4.62
Recall	59.13	68.34	72.81	71.59±1.25	50.49±11.68
F_measure	32.21	48.4	54.49	53.15±0.65	28±6.54
G-means	65.22	76.04	79.71	78.83±0.67	59.85±7.57
Prediction Time	0.6	0.03	0.6	0.05	0.03

The classifiers' performance for each subject varied significantly. However, the RF was very close to the AdaBoost in producing the highest overall accuracy, besides being computationally efficient. On the other hand, the CRF and NN proved to be inappropriate for this task, producing the worst performance.

Another test was performed to provide more insight into how well each class was recognized. Table 2 provides a few classification metrics for each class, based on the average of the driver's performance. It is clear from the results that actions such as *phone call* and *normal driving* were successfully recognized. The low results for *drinking* were produced due to two main reasons. First, the action was sometimes very fast (the driver held the cup for few moments before putting it away). Second, there was a large variance between the different subjects, in performing this action. The worst performance was for *object distraction*, probably because this action required neither huge visual nor cognitive attention. It was often misclassified as *normal driving* or *text messaging*, which made it harder to be recognized. Fortunately, *object distraction* and *drinking* were the least dangerous among the distraction actions, making the misclassification in these cases less critical.

Fig.3 displays a frame-by-frame classification for a given sequence. The blue lines represent the ground truth, while the red lines represent the estimated classes. In this example, *phone call* and *text message* were almost accurately detected, *drinking* produced some false positives, and *object distraction* was often considered *text message*.

Table 2. A per class comparison between the different classifiers for the distraction recognition problem

Action		CRF	HMM	AdaBoost	RF	NN
Phone Call	**precision**	63.96	81.04	90.98	81.44±1.88	70.52±6.95
	recall	64.02	68.34	72.81	72.22±1.19	50.49±11.68
	f-measure	63.99	74.15	80.89	76.53±0.64	58.52±9.8
Text Message	**precision**	61.55	79.08	79.52	75.94±0.76	40.16±6.61
	recall	40.94	74.96	76.05	79.31±1.4	39.64±7.46
	f-measure	49.17	76.96	77.74	77.58±1.01	39.81±6.75
Drinking	**precision**	4.39	47.21	68.21	67.94±1.58	21.94±4.39
	recall	1.6	91.89	68.67	79±1.84	40.1±5.95
	f-measure	2.35	62.37	68.44	73.04±1.22	28.25±4.89
Object Distraction	**precision**	17.94	54.42	58.24	53.34±4.06	23.23±5.05
	recall	6.6	21.89	27.4	27.86±2.11	27.72±5.41
	f-measure	9.65	31.22	37.27	36.56±2.4	25.22±5.08
Normal Driving	**precision**	90.12	90.62	88.32	90.32±0.25	94.01±0.82
	recall	88.4	92.76	97.54	95.41±0.5	84±3.49
	f-measure	89.25	91.67	92.7	92.79±0.33	88.69±1.73

4.2 Driver Distraction Detection

The distraction detection was also classified using the aforementioned classifiers, in addition to the SVM. The evaluation protocol is leave-one-subject-out also. In this case, all the distraction classes were merged into a single class and compared to the normal driving class. A comparison between the performance of the different classifiers for the distraction detection problem per subject is depicted in Table 3. The first six rows show the accuracy for each of the six subjects, while the next rows show the overall performance. The results show the superiority of the RF to the other classifiers in producing the best overall accuracy in a reasonable time. It can also be deduced that decreasing the number of classes enhances the performance of classifiers such as CRF and NN significantly, as for this task, they produce results much closer to the other classifiers.

5 Conclusions and Future Work

A comparison between the performance of Random Forest and other well-known classifiers was investigated for evaluating a visual-based distraction detection and recognition system. The system was based on five modules for extracting

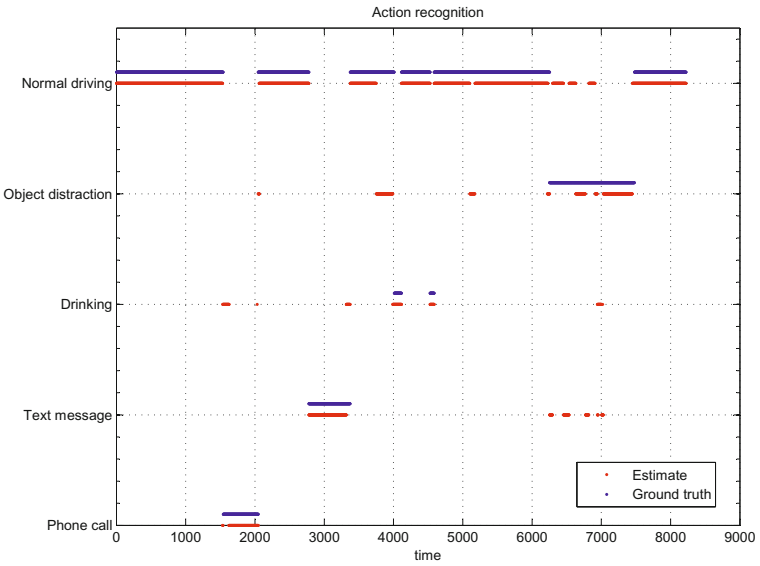

Fig. 3. Results of action recognition for a given sequence using RF. Ground truth (red) and estimated actions (blue) are displayed for each frame (x-axis).

Table 3. A per subject comparison between the different classifiers for the distraction detection problem

	CRF	**HMM**	**AdaBoost**	**RF**	**SVM**	**NN**
1	84.46	82.32	89.81	89.69±0.16	87.47	86.07±1.58
2	82.48	93.46	92.22	92.39±0.72	90.1	87.6±1.22
3	84.24	93.72	93.39	92.25±0.2	89.15	88.49±1.37
4	72.97	88	87.8	90.08±0.85	91.63	87.25±2.93
5	82.53	89.13	87.37	87.3±0.43	89.85	90.66±0.73
6	80.99	82.26	82.84	88.35±3.46	83.09	89.84±5.9
Average	82.55	88.15	88.9	**90.47±0.28**	88.54	88.32±1.65
Specificity	86.95	87.38	94.28	94±0.38	93.51	94.07±1.22
Precision	74.95	79.59	88.39	88.7±0.64	86.39	87.38±2.34
Recall	74.18	93.53	82.66	85.12±0.27	78.28	77.83±1.31
F_measure	74.56	86	85.43	86.87±0.21	82.14	82.32±1.55
G-means	80.31	90.4	88.28	89.58±0.1	85.56	85.56±1.08
Prediction Time	0.63	0.04	0.13	0.08	0.01	0.014

data: arm position, face orientation, facial expression, eye gaze, and eye closure. A real dataset was collected from six subjects using IR and Kinect cameras, while the subjects drove a simulator and performed different distraction actions. The classifiers employed included AdaBoost, HMM, RF, SVM, CRF, and NN,

and the results for detecting and recognizing the drivers distraction show the superiority of the RF for the two tasks in real-time.

This work can be extended by increasing the dataset, adding more subjects to create a more generalized system and more reliable results. Another extension would be increasing the number of sensors, such as ones that measure the pressure on the steering wheel.

References

1. Distracted driving, http://www.cdc.gov/Motorvehiclesafety/Distracted_Driving/index.html
2. Microsoft kinect face tracking, http://msdn.microsoft.com/en-us/library/jj130970.aspx
3. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
4. Butakov, V., Ioannou, P., Tippelhofer, M., Camhi, J.: Driver/vehicle response diagnostic system for vehicle following based on gaussian mixture model. In: 2012 IEEE 51st Annual Conference on Decision and Control (CDC), pp. 5649–5654. IEEE (2012)
5. Connor, J.T., Martin, R.D., Atlas, L.E.: Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks **5**(2), 240–254 (1994)
6. D'Orazio, T., Leo, M., Guaragnella, C., Distante, A.: A visual approach for driver inattention detection. Pattern Recognition **40**(8), 2341–2355 (2007)
7. Holahan, C.J.: Relationship between roadside signs and traffic accidents: A field investigation, Research Report 54. Council for Advanced Transportation Studies, Austin, TX (1977)
8. Liang, Y., Reyes, M.L., Lee, J.D.: Real-time detection of driver cognitive distraction using support vector machines. IEEE Transactions on Intelligent Transportation Systems **8**(2), 340–350 (2007)
9. Murphy-Chutorian, E., Doshi, A., Trivedi, M.M.: Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In: IEEE Intelligent Transportation Systems Conference, ITSC 2007, pp. 709–714. IEEE (2007)
10. Pohl, J., Birk, W., Westervall, L.: A driver-distraction-based lane-keeping assistance system. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering **221**(4), 541–552 (2007)
11. Smith, P., Shah, M., da Vitoria Lobo, N.: Determining driver visual attention with one camera. IEEE Transactions on Intelligent Transportation Systems **4**(4), 205–218 (2003)
12. Stanton, N.A., Salmon, P.M.: Human error taxonomies applied to driving: A generic driver error taxonomy and its implications for intelligent transport systems. Safety Science **47**(2), 227–237 (2009)
13. Torkkola, K., Massey, N., Wood, C.: Driver inattention detection through intelligent analysis of readily available sensors. In: Proceedings of the The 7th International IEEE Conference on Intelligent Transportation Systems, pp. 326–331. IEEE (2004)