



Driver Drowsiness Detection and Tracking Based on Yolo with Haar Cascades and ERNN

Belmekki Ghizlene Amira¹, Mekkakia Maaza Zoulikha^{1*}, Pomares Hector²

¹ SIMPA Laboratoty, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf (USTO-MB), Oran 31000, Alegria

² Dept. of Computer Architecture and Technology, CITIC-UGR Research Center, University of Granada, Granada 18071, Spain

Corresponding Author Email: zoulikha.mekkakia@univ-usto.dz

<https://doi.org/10.18280/ijssse.110104>

ABSTRACT

Received: 4 October 2020

Accepted: 26 December 2020

Keywords:

driver drowsiness detection, Yolo, haar cascade, Convolutional Neural Network, Elman recurrent neural network, multi-layer perceptron, PerClos, assistant agent

When it comes to dangerous drowsiness, the security of the driver and peoples surrounding him depends only on his decisions. This paper expose both of driver drowsiness detector and driving behaviour corrector method based on a conversational assistant agent able to discern and try to avoid driver sleepiness on the wheel, by using a camera to get face's images of the driver in real time, and an agent displayed in the screen and monitors the driver's face in order to warn of drowsiness and to avoid a possible accident. For that, we used Haar cascade with a simplified Yolo-Lite merged with a tree word for detection, followed by the proposed PerStat method with MLP instead of PerClos method which gave a difference of (20%). For the recognition, and helped to raise the problem of the vanishing gradient being used as sequential pre-processing for an ERNN which will generate the agent feedbacks.

1. INTRODUCTION

The road is not only a traffic for cars it can be the workplace for some transporters with heavy nights travel alternation, those are usually struck by fatigue, which is responsible of driver's lack of alertness and awareness and make him vulnerable in front of a situation as vicious as the drowsiness which gently leads him to unconsciousness. In fact the statistics on E-Survey of Road users Attitudes (ESRA) Goldenbeld and Nikolaou [1] show that drowsiness is the cause of crash at 74% in Europe, and 64% in Africa, it represent by itself; more than the half percentage of all types of accidents. In this driver awareness issue there are many methods according to the parameters used to measure the sleepiness, some are based on the respiratory signal of the driver such as Yauri-Machaca et al. [2] using the algorithm thoracic effort derived drowsiness index (TEDD), others focus on the mouth Akrouit and Mahdi [3] by studying the spatio-temporal descriptors of a non-stationary and nonlinear signal to detect the frequency of yawning, or the eyes as much as Oliveira et al. [4] makes a comparison between an (EOG) electrooculogram detection and its combination with an (ECG) electrocardiogram, or artificial vision like Amodio et al. [5] uses the circular Hough transform to detect the pupil opening diameter to tell if the driver is conscious or not, when several other techniques use computer vision and a simple typed alarm to wake up the driver such as Islam et al. [6]. The problematic that inspired us to start this work is that the available methods are more focusing on the detection of fatigue than its evolution or handling it; we suggest a continuity of our previous framework [7] in this interface for the detection of fatigue as well as its evolution over time and its handling with a conversational assistant to detects the state of the driver via a camera by remaining discreet enough, but who in dangerous

situations marks a vigilant and moral presence to try help to take the right decisions.

This paper is divided into four other sections, after this introduction the section 2 is survey of similar works, the section 3 covers the proposed approach and its methodologies, section 4 presents the experiments and obtained results, and section 5 determines the conclusion of this work as well as a projection of the future perspectives.

2. RELATED WORKS

The Human Machine Interface (HMI) with an assistant agent that covers the driver's security and comfort can be divided in three groups:

The ones focusing on the state of the car and its conditions those are more likely used for self-driving cars.

Secondly, the systems oriented towards the scene that is the road such as the method [8] is based on a hybrid ADAS system (Advanced driver assistant agent) with a road oriented monocular camera to detect the lane and the vehicles by using a cascade of classifiers. Xing et al. [9] Uses the data driven control with a mobile assistant agent to avoid optimally and rapidly obstacles. Fan and Zhang [10] Propose an ADAS with as first action the detection of the traffic signs with HOG features (histogram of oriented gradient) then a feed forward neural network to recognize the signs. Cho et al. [11] uses machine learning of the images of the surveillance camera of the parking and a random forest to select the available places as the method. Tariq et al. [12] helps the driver to find a place to park by using the graph representation of the parking and provide the shortest path to each place.

Finally the HMI oriented towards the actors that are the drivers. Ho [13] used the Haptic warning signals warning to

interrupt driver distraction or redirect the focus on the direction that needs them immediate attention, while Choi et al. [14] is about getting the driver characteristics as much as age, gender, driving ability to adjust the interface to the driver and provide cognitive assistance and continual follow-up through the camera and DAVIS (Drive-Adaptive Vehicle Interaction System) that controls the interaction between the vehicle and the driver by monitoring driver's conditions and feedbacks. Yang et al. [15] proposes an HMI based on virtual affective agent as an interactive robot system to estimate the driver status through the car operational commands. Park et al. [16] suggests an assistant companion for events prediction from the online stream of sensory measurements by providing voice assistance. Gull and Mogali [17] proposed an agent that takes in consideration a sensor set with a central server and human expert to study new events and takes corrective measures to handle it. An HMI with an emotional voice alert system adapted to the driver's emotions by using a CNN (Convolutional Neural Network) is used to detect the driver's emotions [18]. Tipprasert et al. [19] suggests using infrared camera with Haar cascade for eyes closer and yawning detection in low light condition. Baek et al. [20] employs an integrated camera on the dashboard with MCT (modified census transform features) on ada boost classifier with LBF (Local Binary Features) mapping and global linear regression with random forest to detect the landmarks of the face than PerClos (PERcentage of CLOsure) is applied for the eyes. Manu [21] tracks the eyes and the mouth by using correlation coefficient template matching and a binary SVM to make the classification of the status of each landmark detected. Nguyen et al. [22] uses a camera with Haar cascade for eye tracking and an alarming board that displays the history of status of the eye brought by the PerClos to detect the opening percentage of the eyes for 3 minutes.

However the literature focuses more on how to detect the somnolence than how to deal with it; being awaked by a beep is not enough to keep safe, on the other hand being attended and aware of the seriousness of the state in which a driver can be at any time and trying to help him to take a reasonable decision about the risks, this kind of assistance could save lives, the issue is to deal with the state of the driver whether it's good or not after detecting the state of face and eyes that could tell so.

Concerning the detection comes the Schroff et al. [23] that uses an end to end learning by employing harmonic embeddings and harmonic triplet loss that reflects face verification and recognition and clustering the used architecture is a start by batch normalization succeeded by Convolutional Neural Network (CNN) followed by a normalization resulting in the face embeddings and triplet loss. CNN is the most used architecture in deep learning for image processing, that include the pre-treatment as convolutions layer to create a feature maps those are latter on all concatenated to create an input tensor for the last fully connected to do the classification. With the technological advancements the architecture CNN became adjustable to make multi object detection, various method exists however due to the fastness and accuracy the inspiring model for this method is Yolo (You Only Look Once) [24] based on selection of bounding boxes surrounding probable objects for one shot multi detection objects and a CNN marked by a differentiation of the filters used in the convolution layers followed up by a word tree for specification of the classification. Updated to Yolo 9000. [25] is made by using Darknet and tree word and

a modified bounding boxes process, after some modification they launch Yolo V3 [26] which will have inspired others for the creation of Yolo version 4 [27] by changing the model skeleton composed of a long spine with many layers a neck for a multi-level pyramidal extraction of the characteristics and finally a head for the classification, the latest version is Yolo V5 launched by a U.S. based start-up (Ultralytics LLC).

This work presents drowsiness management whose motivation for its improvement was the limits of using an instance detection and combination of cases of condition to induce to the final state of the driver that was not suitable and couldn't handle exactly the real state that was hiding in the iteration of some states and the redundancy of some of them, that in longer time would induce a dangerous driving state. In this new version this point is raised using PerStat method based on proportionality estimation and a Multi-layer Perceptron that allows us to make a closer approximation of the state of the driver that will strengthen the robustness of the method, besides the obstacle of redundancy was lifted by using a Elman recurrent neural network (ERNN) which once paired with the pre-treatment of PerStat gives good performance and overcome the problem of the disappearing gradient while increasing the size of the ERNN. We also replaced the ontology by a simple a word tree that will allow us to gain some time calculation and make the diagnostic part faster. We also reorganized some feedbacks regarding to the kind of situation but also its duration and iteration that sometimes can be alerting more than its own direct detection. We simplified the structure of the CNN on Yolo by using Yolo-Lite with less layers to make faster calculation and consume less time and memory.

3. PROPOSED APPROACH

The DRD method (Detection, Recognition, Decision) is inspired by the human reflex when faced with a dangerous situation (The myolitic reflection system), first to detect the danger then to recognize the type of danger it is and finally to treat it. In order to follow the evolution of fatigue and more precisely of driver drowsiness, we present below the developed assistant agent (Figure 1). This agent depends on 3 modules: the detection of the driver state, recognition of the danger of the situation, and decision of the appropriated agent feedback.

We present architecture of the proposed approach of DRD method (Detection, Recognition, and Decision) description as shown in (Figure 2). The beginning is a video frame It that is pre-treated by Haar to detect the face landmarks followed by Yolo-Lite to determine the state of the landmarks, after that the proposed PerStat method is used to determine the most frequent state of each landmarks observed in period T, those states are afterwards combined with an multi-layer perceptron (MLP) that will estimate the state of the driver, once this state is determined it will be fed into an Elman recurrent neural network (ERNN) which will generate the most appropriate feedback agents in ways sequentially relating to these states and their evolution through time.

The goal of tracking that will be made by this approach is how the landmarks look like so the feedback can be generated, we have determined 9 classes for the face state in all principal 3D direction which comes down to { straight, left, right, down, down left, down right, rest back, rest back left, rest back right} regarding the eyes with and without glasses { open , closed,

blinking}, for the detection of those features as shown in Figure 1 the used modules in the proposed approach are detailed below.

3.1 Face landmarks detection and classification

To provide the face and eyes detection, Haar is used to replace the batch normalization and bounding boxes process in Yolo methods, because the maximum number of possible objects in the image is only 3 objects (face, left and right eye) with 9 classes for the face and 6 for both eyes, and so we've avoided wasting calculation time by dividing the entire image into boxes, then we can partially sweep it until we detect a single face with the eyes that are inside the face region. We'll use with the simplified CNN over Yolo-Lite [28] since it works perfectly with non GPU and allows us to make the experiments without consuming a lot of time in the additional

calculations which are pressed in the additional convolution layers of the others models of Yolo. The architecture of the used CNN is shown in Figure 3.

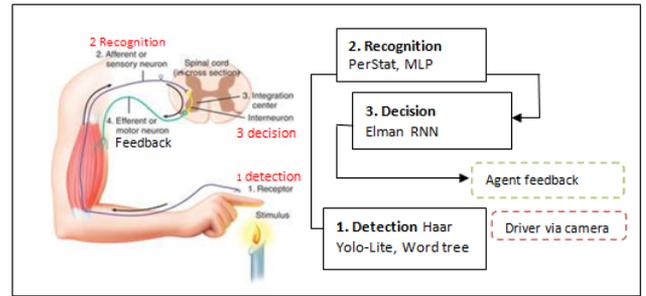


Figure 1. The main modules composing the assistant agent inspired from the myolitic reflection system

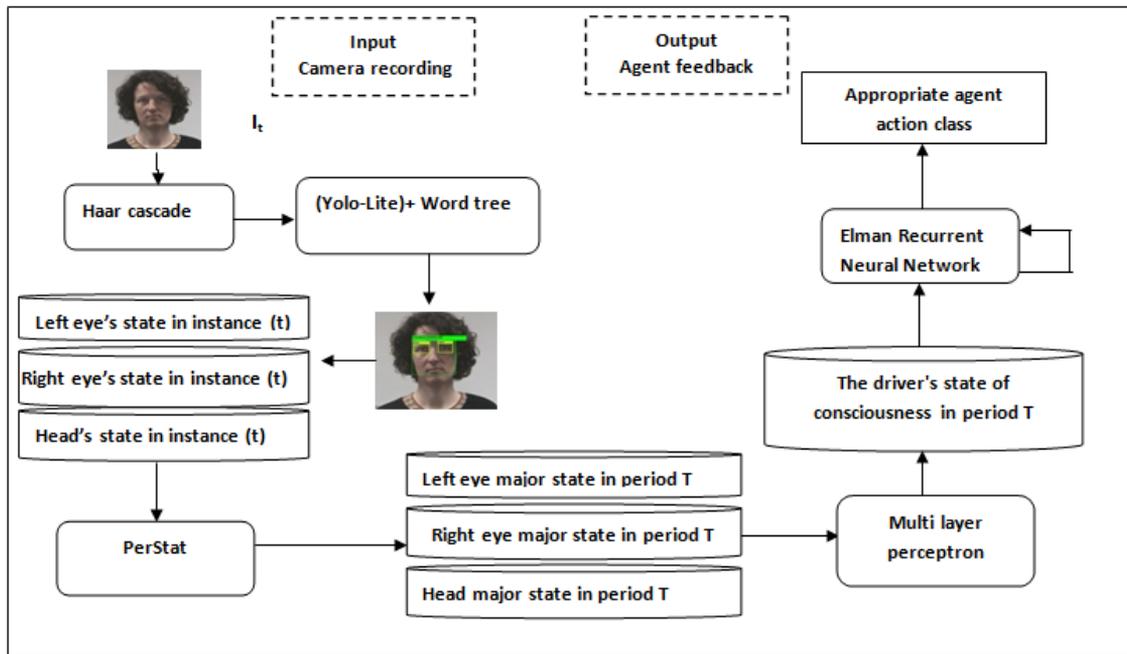


Figure 2. Proposed approach architecture

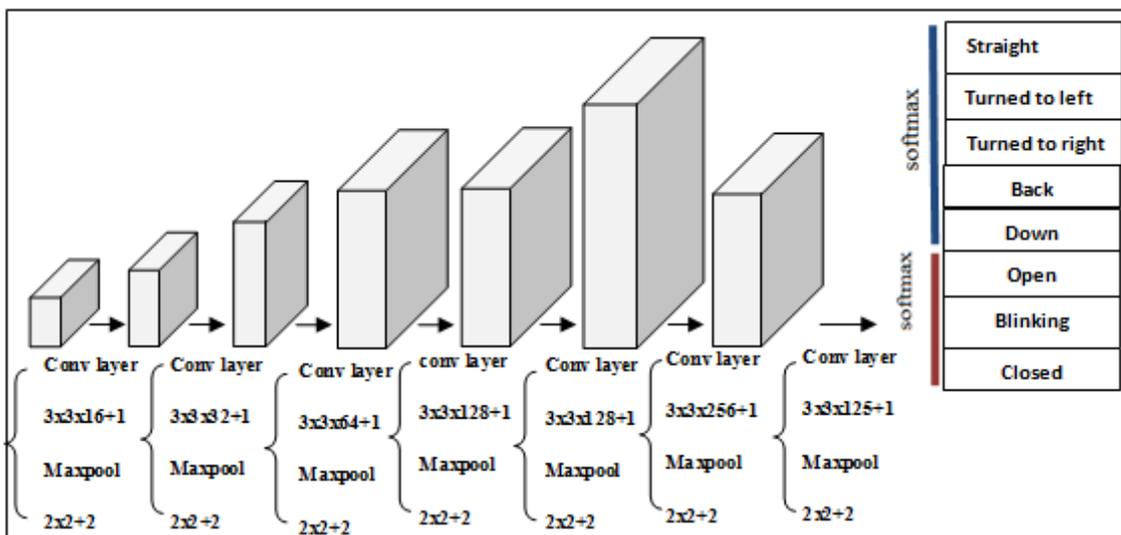


Figure 3. Architecture of the model Yolo-Lite merged with word tree used for drowsiness detection, a CNN features extraction for the object detected from Haar, than specification with word tree

However we merged it with the concept of Yolo 9000 [25] that made it stronger by using a word tree that allowed to make multiple logistic regression (Softmax) over two hyponyms (eye and face), only from the specific classification part, and as long as the Haar dispatches the two principal roots (physical objects: face and eyes) and CNN of Yolo-Lite extracts the characteristics the back propagation would be starting from this part generating of nine hyponyms for the face and three for the eyes, for the appropriated concepts (state) to get and probability of the right class we kept the same equation as Yolo 9000 (Eq. (1)).

$$\Pr(\text{state}) = \Pr(\text{state} | \text{Physical Object}) \quad (1)$$

If we had a face looking down regions detected already by Haar the probability would be as the following Eq. (2).

$$\Pr(\text{face looking down}) = \Pr(\text{face looking down} | \text{face}) \quad (2)$$

The result of this first module represents the detected state of the landmarks of the driver and represented in a vector of 3 words corresponding to the state of each of the head the left and right eye, in the instance t. This vector representation allows having an entry for MLP neural network and makes the treatment sequencing of the more fluent.

3.2 Recognition

The recognition and measurement of drowsiness is done according to the following two stages. The recognition and measurement of drowsiness is done according to the following two stages.

The recognition and measurement of drowsiness is done according to the following two stages.

3.2.1 Estimation of major states of each landmark

Usually for fatigue measurement over the eyes the most used method is PerClos [20] which indicate that the eyes are considered as closed if the percentage is higher than 70% (which is defined as threshold for PerClos) as it is shown in equation (Eq. (3)) PerClos takes the number of the images of the eye captured per minute as (N_m) and the number of the images of the attentive open eye (N_a)

$$PerClos = \left[\frac{(N_m - N_a)}{N_m} \right] \times 100 \quad (3)$$

But it still doesn't take in considerations the other inattentive states as turned left or right, or even blinking eyes. However, as the blinking case cannot be considered as closed or an open one; it seems more efficient to see it as a third state which have also its importance that can be formulated in the fatigue announcement in the redundant repetition of its appearance, for that Nguyen et al. [22] is based only on the number of blinking during a lapse time to measure fatigue. So To have a more exact estimation of the real state of the eyes, the proposed method is based on the rule of proportionality, where each state of the eye whether it is "open", "closed" or "blinking" is considered and calculated proportionally by using the equation (Eq. (4)) in which (N_m) represents the number of the image per instance T, and (N_s) represents the number of the image where the state of the eye S is detected.

$$Perstat = \left[\frac{N_s}{N_m} \right] \times 100 \quad (4)$$

Only the higher probability of estimation is kept and this state is considered as the major state in the sequence (m), if the distribution is equally likely we will take the median case as being the major state.

Using PerStat instead of PerClos provide the use of the blinking category which is not possible in PerClos since this one uses only the closed cases and the open ones, the obtained results show a difference on the estimation of almost 20% compared to PerClos, besides PerStat allowed us to also calculate the state of the head, and so the out coming of this step is a set of periodic vectors with 3major states of the head and two eyes for each sequence (m).

3.2.2 Determination of the overall state of the driver

In this phase we will be able to say what is the exact state of the driver in a period by combining the staff of each landmarks and for that we will use a simple MLP (multi-layer perceptron) type neuron network which will have as input a vector of 3 states major and as output a global state of the driver revealing without level of consciousness the model used and shown in Figure 4.

The classes used for the determination are shown in Table 1, and so for each period this MLP will determine the state of the driver the treatment is done sequentially.

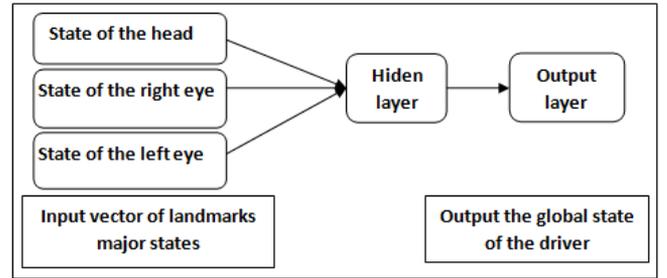


Figure 4. Multi-layer perceptron used for the determination of the overall driver state

Table 1. Global state classes of the driver

Overall driver state
Correct
Distracted
Critical
drowsy
unawake

3.3 Decision

he goal of this phase is to have a reaction from the agent which could alarm or try to rectify the state of the driver, this can be done in a static way but the agent would lose that consistency when it's about dealing with redundant states which separated do not mean big thing but seen sequentially can reflect the evolution of a critical state and required more interactive reactions; and so to detect this recurrence through temporal sequences, inspired from the recursive methods [29, 30]; we had recourse to the Elman recurrent neural network (ERNN), this model of network is similar to an MLP (multi-layer perceptron) which has an additional temporal connection through its hidden layer which allows it to perform recurrent

treatments the used model is shown in Figure 5, in which X_t is the input, H_t is the hidden layer, Y_t is the output, and W_{xh} and W_{hy} are the weights of the full connections between the respective layers ($X_t H_t$) ($H_t Y_t$), and W_{hh} is the weight of the temporal recurrent connexion over the hidden layers. H_t and Y_t are defined respectively by the equations (Eq. (5)) and (Eq. (6)); Where σ is the activation function the used one is softmax.

$$h_t = \sigma_h(W_{xh}x_t + W_{hh}h_t) \quad (5)$$

$$y_t = \sigma_y(W_{hy}h_t) \quad (6)$$

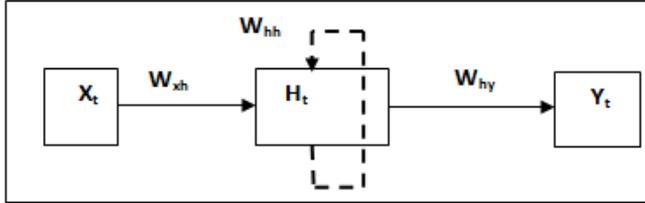


Figure 5. Used ERNN model for decision

The entry of the ERNN X_t is a word vector which describes the state of the conductor which is at the origin of the exit of the Multi-layer Perceptron of the recognition phase, as for the exit of the ERNN is a class which represents the action of the agent corresponding to the entered word. The model used runs on sequential tempo states as shown in the Figure 6, the output of the ERNN in the instance t depends on its input and the value of the previous activation due to the existing connection between the previous hidden layer and the current one.

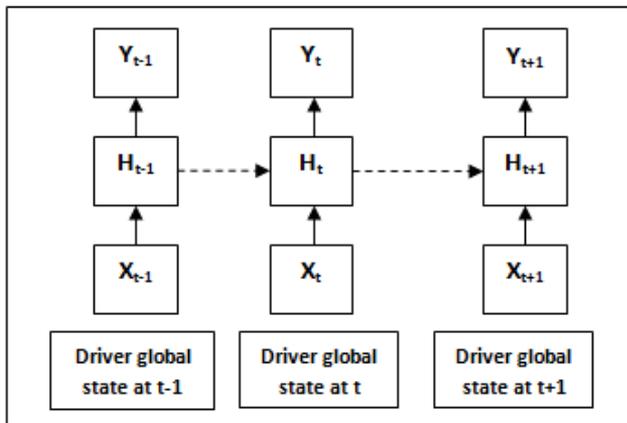


Figure 6. The representation of the used ERNN model unfold through time similar to a stack of sequential MLP

Just as the MLP the ERNN needs a back propagation for the gradient but due to the temporal connection the gradient will be going back over all the previous sequences, known as back propagation through time (BPTT) and estimated by following the headlines in the algorithm (Figure 7), this phase is the big limits of the ERNN if the number of sequences or iterations increases the Gradient will disappear; this problem was raised partially and relatively to our context by adding the PerStat method, using this pre-treatment helped a lot in the fact that the number of the occurrence will be increased in the PerStat instead than in the ERNN, the information remains kept and relative to reality and the feedback remains all the more appropriate, so the previous PerStat combined with MLP could be seen as a memory cells for the ERNN.

Repeat

- Feed a sequence of time steps of X and Y pairs to the network.
- Unroll the network
- Calculate the error in each time step
- Accumulate errors across each time step.
- Roll-up the network and update weights.

Until convergence of gradient.

Figure 7. The main lines of the back propagation through time algorithm

3.4 Agent feedback

Finally, the classes of the feedbacks agent used as outputs of the ERNN are shown in Table 2 those feedbacks are made as follow:

- Standby: as long as the driver is awake the agent is asleep and keeps silent.
- Bip: is the same bip for the short lapse time dangerous states.
- Voice message: try to awake the driver by starting a conversation with him and disturb the installation of his sleepiness as will do a passenger aboard a basic conversation followed by weightier information and the rest of the distance (intelligent vocal chat-bot for human machine conversations).
- Conditional voice message: is to make the driver aware of his situation and make some suggestion which requires agent action such as turn on the radio and look for the nearest coffee shop or station in the Map.
- Make moralizing conversations: set sensitizing phrases such as "it will be safer to rest than continue", "did you know that sleeping on the wheel is the most important cause of fatal crashes after speeding"
- Advise the police: if the driver is sleeping for more than 9 seconds the agent will advise the police, this is the most alarming signal him, knowing this the driver will do his best not to fall asleep and if ever he feels tired he will stop instead of driving drowsing.

Table 2. The ERNN output classes of the agent feedbacks

Agent feedback
Stand by
Bip/Alarm
Voice message
Conditional voice message
Make moralizing conversation
Advise the police

4. EXPERIMENTAL RESULTS AND DISCUSSION

IDE 8.2 java net beans development software with is used with the open CV version 3.4 library for the implementation of Haar and cascades, and bazel for Yolo and vertical and horizontal threads to allow parallelism. The data base used: for the head we use head pose data base Goldenbeld and Nikolaou [1] which represents the decomposition of 15 video sequences in images of the movement of the head in all the probable directions of people with and without glasses of feminine and masculine gender, and for the eyes we used the MRL eye database by injecting the blinking class and this by modifying certain instances supposed to be blinking but annotated as closed to add the blinking class in our experiments.

- As preliminary experience for the use of PerStat instead of PerClos we used instance image caption of one image per instance and period of 3 instances for the estimation of the major states, those are taken relatively to the fact that the agent cannot have one feedback per second, so it is reduced to this suitable periodic minimum. the results of both methods in sequence of 10 periods, shows that the limit with the proposed method is when the higher probabilities are all the same in one period with 33.34% of presence of each state, in this case we take the blinking state as the major state, besides we noticed that with PerClos if the number of image should be high by sequences, since in this experiment of 3 images per period the higher probability does not exceed 70% if not 100%, while the threshold to confirm the opening is limited at 75% in PerClos, we therefore took the greatest probability as in our method, we noticed in some periods where the major state in PerClos is detected as not open whereas in reality the blinking state or closed as detected by the proposed method, this represents a significant gain of information which will increase the precision in the recognition of the major state. as shown in Figure 8, the difference on performance is 20% more accuracy for PerStat.

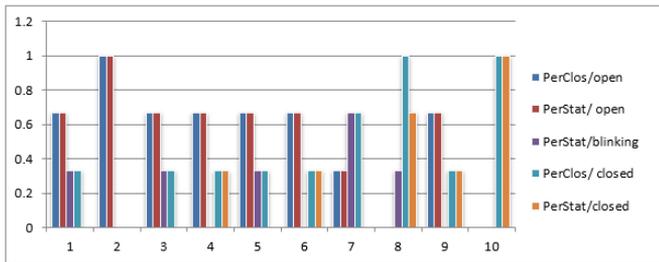


Figure 8. Performance of PerClos and PerStat

Figure 9 shows that PerClos represents only two states of the eyes while PerStat presents three states. The figure illustrates the results obtained in sequence of 10 periods.

- For the training in our experiments. We used two types of training.

Normal training: We divided dataset into two parts, 8 frames of people were used in the training phase and 7 were kept for the testing phase. We created a set of categories referring to the classes of the learning thus according to the directional characteristic present in the base. We made the experiments according to the frames of images of the 15 people in movement.

Personalized training: The main idea of this kind of training is to consider each personal set as a global dataset, and so for each person is made an experience composed of the bought of the training and tests phases. The training consists in taking 3 images of each category of 9 postures of the head own to a single person then we create the personalized training base regarding the eyes. We keep the same data set as the normal training, after that the frame for tests is tacked from the same single person set in full as a video sequence.

Obtained results of both training experiences present an overall improvement of 13.53% in the personalized training compared to the normal one.

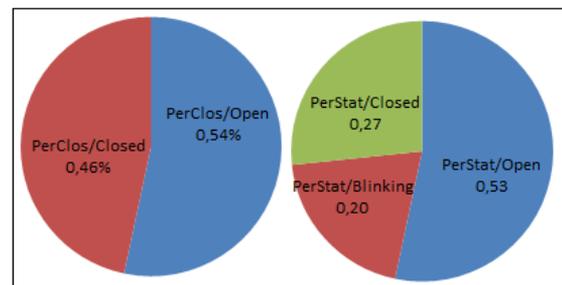


Figure 9. PerClos and PerStat difference



Figure 10. The virtual representation of the assistant agent on the screen as he will see it the driver changing his color by changing feedbacks

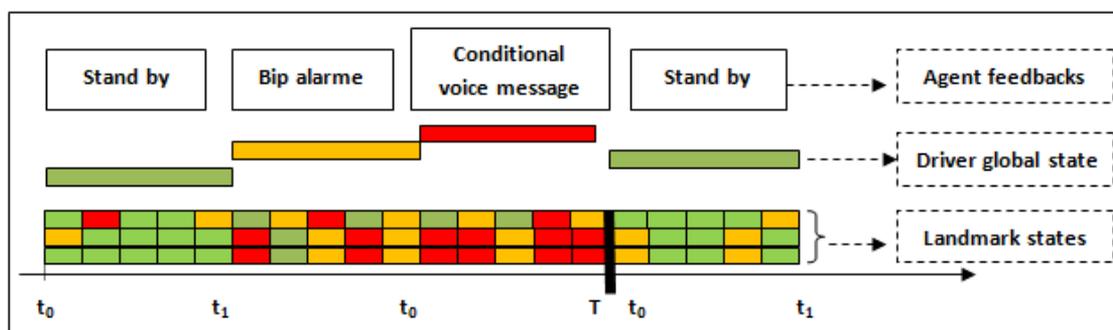


Figure 11. Agent feedback performance with the chosen temporization

- For the virtual representation of the agent, we use a simple wheel with an eye inside which changes colors according to the risk level of the global state of the driver. So used sphinx library for speak recognition and free TTS lib for the text to sound translation to provide the messages of the agent. We tried to make the agent as much as possible discreet so as not to disturb the driver, moreover the agent should not attract should his attention from the road or distract him visually but should be present in way to remind him that he is not alone, for this the graphic representation of the agent is simply a wheel by way of 'an observant eye that changes a color with the feedback as shown in Figure 10.
- Elman recurrent neural networks turned out to be suitable to generate feedbacks of the agent over driver states, besides by doing a pre-treatment using multi-layer perceptron (MLP) before the ERNN allowed us to minimize the size of the ERNN and avoid the problem of the vanishing Gradient through long back propagation, instead of having a back propagation 9 times its reduced to only 3, If the sequences are too short than the feedbacks don't have enough time to be displayed and if the sequences get larger than it'll make the agent slow and may miss some dangerous cases, and so to make it more suitable we made feedbacks needing more time play as the last output of the ERNN, besides we used PerStat over 5 seconds with one image per second the performance of this combination are presented in Figure 11, illustrate the agent's feedback on the driver's states which are green when not alarming, orange when worrying and red when risky, and t_i with i from 0 to T are the sequences temporization chosen for ERNN. The after that the ERNN forget everything and starts over again, we see in this sequence that the agent was able to capture the driver's vigilance and modified his state of consciousness and this is the main goal.

5. CONCLUSION

This work presents technique of managing driver drowsiness using an assistant agent with DRD (detection, recognition, decision) approach. We used a combination of Haar cascades and Yolo-Lite merged with the tree word; Haar is used to replace the bounding boxes process in Yolo since we are doing a trio-detection instead of multi detection, this gives the same detection rate as if we are using bounding boxes process Yolo, but saves time in the detection process.

Whereupon we used PerStat a method of proportionality to detect the state of the eyes and the head, this method works better than the PerClos method by allowing to take into consideration the blinking state, which is ignored in PerClos, what decreases the probability of consciousness and increases the level of overall risk which refers to the real state of the driver that is estimated by using a multi-layer perceptron.

Secondly the out coming of the comparison between a normal training, when we merged all the persons, and a personalized training, when we used experiences of tests and training by person, made the recognition of the face state more precise. So it'll be asked from the user to first set all needed poses for better detection.

And finally the agent feedbacks are generated by a Elman recurrent neural network (ERNN) preceded by PerStat, it turned out that this pre-processing allowed us to reduce the

size of the ERNN and avoid the loss of the vanishing gradient and this while retaining the information.

As perspective, we foresee add a second eye to the agent to get more efficiency, we will add a focus on the road by another camera. Both of the collected information will be paired in real time, this will allow us to create a connection between vehicles on the road which can be really helpful for individual and general security.

ACKNOWLEDGMENT

This work was funded by Algerian Ministry of Higher Education and Scientific Research, General Direction of Scientific Research and Technological Development.

REFERENCES

- [1] Goldenbeld, C., Nikolaou, D. (2019). Driver fatigue. ESRA2 thematic report Nr. 4. ESRA project (E-Survey of Road users' attitudes). The Hague, Netherlands Institute for Road safety Research SWOV.
- [2] Yauri-Machaca, M., Meneses-Claudio, B., Vargas-Cuentas, N., Roman-Gonzalez, A. (2018). Design of a vehicle driver drowsiness detection system through image processing using matlab. In 2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII), pp. 1-6. <https://doi.org/10.1109/CONCAPAN.2018.8596513>
- [3] Akrouf, B., Mahdi, W. (2016). Yawning detection by the analysis of variational descriptor for monitoring driver drowsiness. In 2016 International Image Processing, Applications and Systems (IPAS), pp. 1-5. <https://doi.org/10.1109/IPAS.2016.7880127>
- [4] Oliveira, L., Cardoso, J.S., Lourenço, A., Ahlström, C. (2018). Driver drowsiness detection: a comparison between intrusive and non-intrusive signal acquisition methods. In 2018 7th European Workshop on Visual Information Processing (EUVIP), pp. 1-6. <https://doi.org/10.1109/EUVIP.2018.8611704>
- [5] Amodio, A., Ermidoro, M., Maggi, D., Formentin, S., Savaresi, S.M. (2018). Automatic detection of driver impairment based on pupillary light reflex. IEEE Transactions on Intelligent Transportation Systems, 20(8): 3038-3048. <https://doi.org/10.1109/TITS.2018.2871262>
- [6] Islam, M.M., Kowsar, I., Zaman, M.S., Sakib, M.F.R., Saquib, N. (2020). An algorithmic approach to driver drowsiness detection for ensuring safety in an autonomous car. In 2020 IEEE Region 10 Symposium (TENSYMP), pp. 328-333. <https://doi.org/10.1109/TENSYMP50017.2020.9230766>
- [7] Ghizlene, B., Zoulikha, M., Pomares, H. (2019). An efficient framework to detect and avoid driver sleepiness based on YOLO with Haar cascades and an intelligent agent. In International Work-Conference on Artificial Neural Networks, pp. 699-708. https://doi.org/10.1007/978-3-030-20518-8_58
- [8] Kang, J.S., Kim, J., Lee, M. (2014). Advanced driver assistant system based on monocular camera. In 2014 IEEE International Conference on Consumer Electronics (ICCE), pp. 55-56. <https://doi.org/10.1109/ICCE.2014.6775905>

- [9] Xing, S., Guan, X., Luo, X. (2010). Trajectory tracking and optimal obstacle avoidance of mobile agent based on data-driven control. In Proceedings of the 29th Chinese Control Conference, pp. 4619-4623.
- [10] Fan, Y., Zhang, W. (2015). Traffic sign detection and classification for Advanced Driver Assistant Systems. In 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 1335-1339. <https://doi.org/10.1109/FSKD.2015.7382137>
- [11] Cho, W., Park, S., Kim, M.J., Han, S., Kim, M., Kim, T. (2018). Robust parking occupancy monitoring system using random forests. In 2018 International Conference on Electronics, Information, and Communication (ICEIC), pp. 1-4.
- [12] Tariq, S., Choi, H., Wasiq, C.M., Park, H. (2016). Controlled parking for self-driving cars. In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 001861-001865. <https://doi.org/10.1109/SMC.2016.7844509>
- [13] Ho, C. (2008). Haptic interface for the distracted drivers. In 2008 SICE Annual Conference, pp. 890-893. <https://doi.org/10.1109/SICE.2008.4654780>
- [14] Choi, J.K., Kwon, Y.J., Jeon, J., Kim, K., Choi, H., Jang, B. (2018). Conceptual design of driver-adaptive human-machine interface for digital cockpit. In 2018 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1005-1007. <https://doi.org/10.1109/ICTC.2018.8539644>
- [15] Yang, J.Y., Jo, Y.H., Kim, J.C., Kwon, D.S. (2013). Affective interaction with a companion robot in an interactive driving assistant system. In 2013 IEEE Intelligent Vehicles Symposium (IV), pp. 1392-1397. <https://doi.org/10.1109/IVS.2013.6629661>
- [16] Park, J., Son, H., Lee, J., Choi, J. (2018). Driving assistant companion with voice interface using long short-term memory networks. *IEEE Transactions on Industrial Informatics*, 15(1): 582-590. <https://doi.org/10.1109/TII.2018.2861739>
- [17] Gull, K.C., Mogali, A. (2009). Agent based assistance system with ubiquitous data mining for road safety. In 2009 International Conference on Intelligent Agent & Multi-Agent Systems, pp. 1-2. <https://doi.org/10.1109/IAMA.2009.5228058>
- [18] Sarala, S.M., Yadav, D.S., Ansari, A. (2018). Emotionally adaptive driver voice alert system for advanced driver assistance system (ADAS) applications. 018 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, pp. 509-512. <https://doi.org/10.1109/ICSSIT.2018.8748541>
- [19] Tipprasert, W., Charoenpong, T., Chianrabutra, C., Sukjamsri, C. (2019). A method of driver's eyes closure and yawning detection for drowsiness analysis by infrared camera. In 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), pp. 61-64. <https://doi.org/10.1109/ICA-SYMP.2019.8646001>
- [20] Baek, J.W., Han, B.G., Kim, K.J., Chung, Y.S., Lee, S.I. (2018). Real-time drowsiness detection algorithm for driver state monitoring systems. In 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 73-75. <https://doi.org/10.1109/ICUFN.2018.8436988>
- [21] Manu, B.N. (2016). Facial features monitoring for real time drowsiness detection. In 2016 12th International Conference on Innovations in Information Technology (IIT), pp. 1-4. <https://doi.org/10.1109/INNOVATIONS.2016.7880030>
- [22] Nguyen, T.P., Chew, M.T., Demidenko, S. (2015). Eye tracking system to detect driver drowsiness. In 2015 6th International Conference on Automation, Robotics and Applications (ICARA), pp. 472-477. <https://doi.org/10.1109/ICARA.2015.7081194>
- [23] Schroff, F., Kalenichenko, D., Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815-823. <https://doi.org/10.1109/CVPR.2015.7298682>
- [24] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- [25] Redmon, J., Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263-7271. <https://doi.org/10.1109/CVPR.2017.690>
- [26] Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
- [27] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
- [28] Huang, R., Pedoeem, J., Chen, C. (2018). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE International Conference on Big Data (Big Data), pp. 2503-2510. <https://doi.org/10.1109/BigData.2018.8621865>
- [29] Sun, P., Li, J., Lan, J., Hu, Y., Lu, X. (2018). RNN deep reinforcement learning for routing optimization. In 2018 IEEE 4th International Conference on Computer and Communications (ICCC), pp. 285-289. <https://doi.org/10.1109/CompComm.2018.8780950>
- [30] Bai, R., Zhao, J., Li, D., Lv, X., Wang, Q., Zhu, B. (2020). RNN-based demand awareness in smart library using CRFID. *China Communications*, 17(5): 284-294. <https://doi.org/10.23919/JCC.2020.05.021>