Drowsy Driver Detection in Video Sequences using LSTM with CNN Features

Nisha Gandhi[1] Tejas Naik[2] Aditya Yele[3]

Abstract— **Around 100,000 accidents per year are caused by driver drowsiness. To add to the seriousness of the matter, no test exists to determine sleepiness as there is for intoxication detection. Detection of driver drowsiness is gaining importance in the field of Computer Vision and Machine Learning. Recurrent Neural Networks (RNNs) and Long Short-Term Memory Units(LSTM) have been very successful in processing of sequential multimedia data. In this project, we propose a novel driver drowsiness detection method using Convolutional Neural Networks (CNNs) to extract information from images and feed a sequence of such information to the LSTMs for prediction.**

Index Terms - **Computer Vision, Machine Learning, Deep Learning, Drowsy Driver Detection, Face Detection, Eye Tracking, Convolutional Neural Network, Recurrent Neural Networks, Long Short Term Memory.**

I. INTRODUCTION

Accidents caused due to drowsy driving are a major problem in the United States. The National Highway Traffic Safety Administration estimates that drowsy driving was responsible for 72,000 crashes, 44,000 injuries, and 800 deaths in 2013[1]. Drowsiness detection technologies have attempted to prevent such incidents by predicting if a driver is falling asleep based on various inputs. Technologies in drowsiness detection can be classified in to three main categories[2]. The first category involves measuring cerebral and muscular signals and cardiovascular activity. These techniques are invasive and not commercially viable. The second category includes techniques of measuring overall driver behavior from vehicle patterns.Examples of this method include monitoring the vehicles position in a lane, steering pattern monitoring. These measurements need to take in to account many parameters such as vehicle type, driver experience, condition of the road[2].Measuring most of these parameters requires significant amount of times and user data. These techniques do not work with microsleeps-when the driver falls asleep for a few seconds without causing any significant changes in the driving patterns.The third category consists of using Computer Vision techniques as a non invasive way to monitor drivers sleepiness. We present a system in the third category for drowsiness detection using CNNs and LSTMs. After face detection using Viola Jones face detector, we track the eyes. These are fed to a pre-trained CNN. The sequences of features extracted by the CNN are then given to LSTM for detecting drowsiness.

II. RELATED WORKS

Efforts reported in literature have focused on all three categories of drowsiness detection systems. Here we present a survey of literature on non intrusive detection using computer vision.

Alshaqaqi et al.[3] have presented a detection system based on edge detection and exploiting the symmetry of facial features for extracting the eyes. The state of the eyes is determined as open or closed by taking the Hough transform for circles and comparing the intersection of the Hough transform and the edge image with a threshold. The state of drowsiness is then determined by using Percentage of Eyelid Closure(PERCLOS)- a scientifically associated measure of drowsiness associated with slow eye closure.

Grace et al. [4] have presented two drowsiness detection methods. In the first method they develop a camera by exploiting the fact that the retina reflects different amount of infrared light at different frequencies.Two images of the drivers face are taken at fixed wavelengths. The difference of this images is used to measure percentage eye closure. The second method although in its infancy uses a neural network to predict PERCLOS by finding the right combinations of driver performance variables.

Malla et al. [5] have built a system for detecting microsleep. The system uses a remotely placed camera with near infra-red illumination to acquire the video. Haar object detection algorithm is used to detect a face. The eyes Region of interest is detected using anthropomorphic parameters. Eye closure is detected by taking ratio of the closed portion of the eye to the average height of the open portion.

Under the light of what has been mentioned above, methods for drowsy detection have involved detection of face, eyes and(or) facial features.

III. PROPOSED APPROACH: CONV-LSTM

The problem of detecting drowsiness is that it is difficult to tell from a single frame if the person is blinking or falling asleep. In order to overcome this problem, we introduce our method Conv-LSTM, which comprises of two submodels: the CNN model for feature extraction and LSTM for interpreting the features across consecutive frames. The procedure for drowsiness detection is thus as follows: First, we extract significant CNN features from the video frames. Then features representing the sequence of the action (Alert or a Drowsy Driver) for a certain time interval (fixed number of frames) are fed to the LSTM as an input. Finally, a softmax layer is used to predict drowsiness/alertness of the entire video sequence.[15]. Figure(1) below explains the flow of our model.

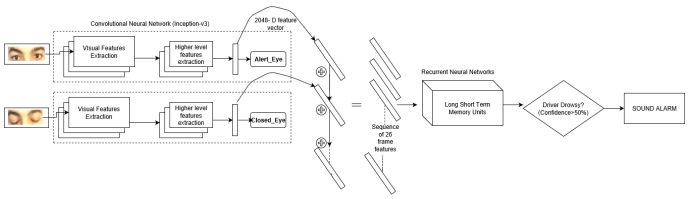

Fig. 1. Flow Diagram for Conv-LSTM

A. Dataset Collection

Videos of eight subjects (6 males and 2 females) imitating signs of alertness and drowsiness were recorded under ambient recording conditions. During the recording of the videos, the subjects were asked to perform certain actions to imitate drowsiness such as slow eyelid closure, and droopy eyes followed by a quick recovery of head posture to imitate micro-sleep. In order to imitate alertness, the subjects were asked to gaze in different directions with/without head movement.

The dataset consists of 16 Training and 3 testing videos, both containing classes: Alert-Eyes and Drowsy-Eyes. Videos were recorded with a CMOS front web-camera 1280x720p at 30fps with a flicker reduction of 50 Hz.

B. Face ROI Detection and Eye Detection module

We use Viola-Jones Haar-Feature based Cascade Classifiers[6] for face detection. In order to avoid false positives, we first detect the face Region of Interest(fROI) and then apply eye detection on this ROI to obtain a rectangular localized patch containing a pair of eyes. After detecting the face and eyes in the first frame, we track them using CAMShift (Continuously Adaptive Mean-shift). Below figures demonstrate detection of closed as well as open eyes.

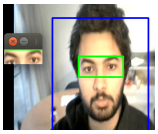

Fig. 2. Alert-Eye detection

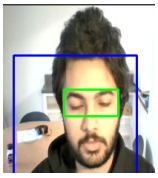

Fig. 3. Drowsy-Eye detection

C. Convolutional Neural Network (Inception-v3) module

We manually created an image dataset for feature extraction. Two classes were made with approximately 120 images each for Alert-Eyes and Drowsy-Eyes. To extract significant visual features from these images, we use Convolutional Neural Networks (CNNs), which are state-of-the-art for image classification and feature extraction.We adapted a pre-trained model, Inception-v3[12], which is trained on the Image-Net Dataset comprising of 1000 classes for Large Scale Visual Recognition Challenge(2012)[10]. Using transfer learning we retrain the final layer of this model on our dataset with Tensorflow[11].

At the completion of 4000 training steps, our model reported an accuracy of 96.5% on the validation set. Then, we ran each frame(image) of every video through Inception model and saved the output from the final pooling layer (pool-3:0). This results in a 2048-Dimensional vector of features, which we passed to the sequential neural models. Finally, we convert these extracted features into sequences of extracted features.

D. Long Short Term Memory Units (LSTM)

Long Short Term Memory Networks are a special kind of Recurrent Neural Networks, capable of learning long-term dependencies while avoiding the vanishing and exploding gradients problems. Each block contains one or more recurrently connected memory cells and three multiplicative units, the input, output and forget gates, which control the information flow inside the memory block.

The LSTM framework enables the prediction (textual description) for visual time series problems. In Drowsy Driver Detection, the stitched features (16 videos x 26 frames x 1024 feature vectors) are used to train the sequential model.

We used a single, 4096-wide LSTM layer, followed by a 1024 Dense layer, with some dropout in between. We trained the model for 10 epochs, with a batch-size of 4, using Keras and Tensorflow as the back-end.[13] We used Adam Optimizer configured with a learning rate of 0.00005 to train and optimize our network weights. Figure 4 below shows the architecture of our LSTM model.

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 26, 2048)	33562624
flatten_1 (Flatten)	(None, 53248)	0
dense_1 (Dense)	(None, 1024)	54526976
dense_2 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 2)	1026

Fig. 4. LSTM Architechture

IV. Results Obtained

We tried and tested our model with various parameters. Inception-v3 retrained on our dataset of eye patches obtained an approximate training accuracy of : 96.5%. Our testing accuracy was 87.5% for 10 epochs for the LSTM model. The model was able to correctly classify a sequence of consecutive frames from unseen videos, it detected a drowsy person with 93.65% confidence and a alert driver with

99.63% confidence in most of our test runs. To visualize the loss function we ran over 30 epochs which resulted in the graph shown in Figure 5. We performed hyperparameter tuning on learning rate with ADAM and SGD optimizers. Results obtained with ADAM optimizer were significantly better than SGD.

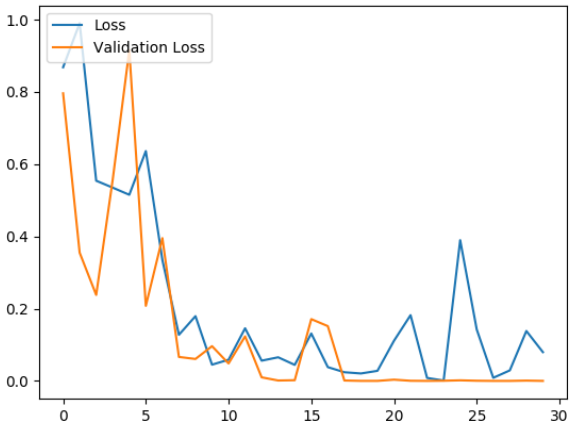

Fig. 5. Loss, Validation Loss for 30 epochs

V. Challenges Faced

Unavailability of an apt dataset led us to creating our own video dataset for driver drowsiness detection. This was quite time consuming and tedious.

Figuring out the exact procedure for reshaping the stitched sequence of frames to connect the output layer of the CNN Inception-v3 model to the LSTM model was challenging for us.

Taking care of corner cases such as not predicting the driver as drowsy for normal eye-blinks proved to be demanding.

VI. Conclusion

Thus our model warns drowsy drivers with an alarm, after successful eye-detection and tracking with computer vision and deep learning techniques (CNN and LSTM models) with an accuracy of 87.5%.

VII. Future Scope

Our model can be improvised by the following methods: Learning to detect faces and eyes in varied lighting conditions, such as at night with infrared lights. In addition to this, the model should also be able to recognize drowsy eyes with sunglasses.

With some modification this system can be used in combination with real time cameras to provide alert a driver while he is driving.This will however require exhaustive testing on a larger dataset.

ACKNOWLEDGMENT

We thank Professor Roy Shilkrot, for his constant guidance and support.

We would also like to thank our fellow batch-mates: Noopur Maheshwari, Rahul Rane, Bhushan Sonawane, Nishant Borude, Mihir Chakradeo and Dhanashree Patil all graduate students at Stony Brook University for helping us create our video dataset.

References

[1] Center for Disease Control and Prevention https://www.cdc.gov/features/dsdrowsydriving/index.html.

[2] Optimised Co-modal Passenger Transport for Reducing Carbon Emissions. COMPASS Handbook of ICT Solutions

[3] B. Alshaqaqi, A. S. Baquhaizel, M. E. Amine Ouis, M. Boumehed, A. Ouamri and M. Keche, "Driver drowsiness detection system," 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA), Algiers, 2013, pp. 151-155. doi: 10.1109/WoSSPA.2013.6602353

[4] R. Grace et al., "A drowsy driver detection system for heavy vehicles," 17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No.98CH36267), Bellevue, WA, 1998, pp. I36/1-I36/8 vol.2. doi: 10.1109/DASC.1998.739878

[5] A. M. Malla, P. R. Davidson, P. J. Bones, R. Green and R. D. Jones, "Automated video-based measurement of eye closure for detecting behavioral microsleep," 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, 2010, pp. 6741-6744. doi: 10.1109/IEMBS.2010.5626013

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-511-I-518 vol.1. doi: 10.1109/CVPR.2001.990517

[7] OpenCV.Open Source Computer Vision Library Reference Manual, 2001

[8] Franois Chollet, Keras, 2015, Github, https://github.com/fchollet/keras

[9] Martn Abadi et al., TensorFlow: Large-scale machine learning on heterogeneous systems,2015. Software available from tensorflow.org.

[10] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L.Fei-Fei. ILSVRC-2012, 2012.http://www.image-net.org/challenges/LSVRC/2012/

[11] Retraining Inception's final layer for New Categories $https : //www.tensorflow.org/tutorials/image_retraining$

[12] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2818-2826).

[13] Video classification methods https://blog.coast.ai/five-video-classification-methods-implemented-in-keras-and-tensorflow-99cad29cc0b5

[14] CNN-LSTMs https://machinelearningmastery.com/cnn-long-short-term-memory-networks/

[15] Donahue, Jeffrey, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. "Long-term recurrent convolutional networks for visual recognition and description." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2625-2634. 2015.