ISSN 2278-3091
Volume 9 No.2, March - April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse158922020.pdf
https://doi.org/10.30534/ijatcse/2020/158922020

Drowsiness Detection System for Online Courses

Umang Lahoti[1], Rakshita Joshi[1], Nupur Vyas[1], Kashmira Deshpande[1], Sweta Jain[2]
[1]Student, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, lahotiug@rknec.edu
[1]Student, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, joshirj@rknec.edu
[1]Student, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, vyasnm_1@rknec.edu
[1]Student, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, deshpandekk@rknec.edu
[2]Assistant Professor, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, jains@rknec.edu

ABSTRACT

Online courses are gaining currency these days. But they aren't as effective as classroom teaching. Hence, we designed an application to make the online courses interactive. A drowsiness detection system is implemented which uses the concept of Ear Aspect Ratio and Mouth Aspect Ratio and it is validated using ResNet50 pre-trained model. Once the course attendee is detected drowsy, questions are generated using Natural Language Processing based on the video watched till then. If the attendee answers the question incorrectly a penalty would be imposed, else the video resumes. The highest training accuracy achieved was 97% and highest validation accuracy achieved was 94%.

Key words: Deep Learning, NLP, ResNet50, CNN, Ear Aspect Ratio, Dlib, question generation, spaCy, drowsiness detection, online course.

1. INTRODUCTION

The primary source of gaining knowledge is online courses because online courses are economical, flexible, have plethora of courses to choose from, provide comfortable environment and permits everybody to access courses conducted by renowned professors from esteemed universities. However, online courses are less communicative as compared to classroom teaching. Whenever a teacher finds that the student is inattentive because of drowsiness, they are able to capture the students' interest back by asking them questions relevant to the topic. On online courses, there is no such provision of two-way communication between the instructor and the attendee. In other words, there is no one to keep a track of the students' behaviour during the conduction of course. To overcome this issue, we have developed a module. This module is capable of detecting drowsiness.

The dataset consisting of 167 images of 50 subjects was collected. The dataset was annotated into two categories namely drowsy and not drowsy. The images in the dataset were pre-processed. Since, the dataset was quite small we augmented the images to create a larger dataset consisting of 1670 images.

To detect whether a person is drowsy, his/her Ear Aspect Ratio and Mouth Aspect Ratio was calculated [1]. This was done by extracting relevant facial action unit. To validate whether the Ear and Mouth Aspect Ratios were calculated accurately, the snapshot of that instance was captured. The captured image acts as a test case for our pre-trained Convolution Neutral Network (CNN) model. It was observed that ResNet50 performed comparatively better than the other pre-trained models[2] that we implemented (MobileNet, GoogleNet, etc). ResNet50 is a 50 layered Convolution Neutral Network which was trained on more than a million images in the ImageNet Database [3, 4].Originally the network classified the images into 1000 object categories. We modified the network to classify the images into our required categories (drowsy and not drowsy).

Once the user is detected drowsy the question generating module is called. This module generated fill in the blank type questions relevant to the content of the video covered till that instance. The GUI enables the users to answer multiple questions if they wishto do so because one question might not be sufficient to capture their attention.

2. LITERATURE REVIEW

Researchers have used many different strategies for detecting drowsiness. In [5, 6], the authors proposed non-image-based methods based on electro-oculography. In this method, electrical signals of the muscles around the eyes are captured and eye movement signals are analysed. Similarly, to analyse electrical signals around the eyes, [7, 8, 9] use electromyography and electroencephalograms by attaching sensors to the muscles around the eyes. These non-image-based methods lead to faster collection of data. However, the problem arises when the sensors catch noises due to the user's movements resulting in lower accuracy. Also, attaching sensors to a user for drowsiness detection every time pose an inconvenience problem.

In [10, 11], image-based methods are used to capture the eyes using Haar based cascade classifier and blink detection is implemented using Histogram of Oriented Gradient (HOG) based features along with SVM classifier. Subsequently, percentage of eye closure (PERCLOS) is

calculated. If value of PERCLOS is above a threshold, then a person is detected as drowsy.

In [12], a real time algorithm is proposed for eye blink detection in a video. Facial landmark points are used to extract eye aspect ratio (EAR) to determine eye openness. The SVM classifier is used to detect eye blinks as a pattern of EAR values and the Hidden Markov Model is used to estimate the state of the eyes.

3. METHODOLOGY

The system for drowsiness detection is divided into two parts: 1. Aspect Ratios module 2. CNN module

1. Aspect Ratios module

In this module, the webcam keeps recording the state of the student. For each frame, eye aspect ratios (EAR) and mouth aspect ratios (MAR) are calculated. This is done using Dlib [13], which isa software library written in C++. Dlib detects the face and then generates 68 facial landmarks. The coordinates of the left eye, right eye and the mouth were extracted. Contours were also plotted on these points for better visualisation.

The coordinates detected for eyes are labelled as p1, p2, p3, p4, p5, p6. The Euclidean distance between vertically opposite points (p2-p6 and p3-p5) and horizontally opposite points(p1-p4) are calculated [14]. Ratio between the distance of the vertically and horizontally opposite points is termed as the Eye Aspect Ratio [15]. Similarly, mouth aspect ratio was calculated.

Threshold for the eye aspect ratio (EYE_AR_THRESH), mouth aspect ratio (MOUTH_AR_THRESH) and consecutive number of frames (CONSEC_FRAMES) was set.
If either EAR < EYE_AR_THRESH or MAR > MOUTH_AR_THRESH is detected consecutively for longer than CONSEC_FRAMES [16], drowsiness alert is generated.

The aspect ratios module was chosen as a preliminary step in drowsiness detection because we wanted to limit the number of images being fed to the CNN as test cases. Only when the Aspect ratios module detects the user as drowsy, the snapshot of that particular instant was fed as a test case to the CNN. Also, the Aspect ratios module gave accurate results only when the student faced the camera from the front but performed poorly when the user viewed the screen from another angles, hence the CNN was used to validate whether the Aspect ratios module detected drowsiness correctly.

2. Convolution neural network

Convolution Neural Networks (CNN) [17] outperforms other classical machine learning algorithms. It acts as an automatic feature extractor of the image, hence the pre-processing required is comparatively less [18]. We collected our own dataset, pre-processed and augmented the images and trained them over a pretrained CNN.

3.1 Dataset

We had collected the data of 50 subjects. The data collected had three different expression – neutral expression, eyes closed and yawning as shown in the Figure 1. The collected data was then annotated into two categories – *drowsy and alert.* The drowsy class had 104 images including eyes closed and yawning and the alert class had 63 images with the neutral expression. Then we divided the data into two parts – training set and validation set in the ratio 80:20. So, the training set had 83 images of class drowsy and 50 images of class alert and the validation set had 21 images of class drowsy and 13 images of class alert.

Fig. 1.1 Neutral | Fig. 1.2 Eye Closed | Fig. 1.3 Yawn

Figure 1: Dataset

3.2 Pre-processing

The data collected cannot be used directly, hence we need to pre-process it. This pre-processing is done in two steps – cropping and augmenting.

3.2.1 Haar Cascade

It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Haar Cascade Detection in OpenCV comes with both a trainer and a detector. We can train our own classifier for any object like table, book, car, etc. using OpenCV [13, 19]. But, in this project we only need to detect face for cropping the images.

So, the code will detect the faces in the input image in the form of a rectangle (x co-ordinate, y co-ordinate, width, height). Using this data, we cropped an image and obtained faces as shown in the Figure 2.

Fig. 2.1 Neutral | Fig. 2.2

Figure 2: Cropped Image using Haar Cascade

3.2.2 Augmentation

Augmentation helps increase the diversity of the data available for training models. Since we had data for only 50 subjects, that was not enough for training a model using deep learning. So, we augmented the data to increase the number of images. The augmentation was done in the following ways –
Images were shifted via the width_shift_range=0.3. They were flipped horizontally and rotated by a rotation_range=10 and zoomed by a factor=0.1. Images were sheared via the shear_range argument.
Using the above ways, we increased the data 10 times i.e. each image was converted into 10 images. So, as a result we had 1350 images for training set and 320 images for validation set.

3.3 Algorithm

We used the pretrained Imagenet [20] model, Resnet-50 for training our dataset. Resnet-50 has a 50-layer deep architecture and uses the concept of skip connections to reduce the degrading accuracy. Resnet-50 was trained on more than a million images and could classify images into 1000 categories [18].
Since, our dataset was significantly smaller, overfitting was observed. To reduce overfitting, dropout layers were added before the output layer [21, 22]. Dropouts simply mean that some neurons from the layer are dropped randomly and are ignored during the training phase. This adds a penalty to the loss function and the model is trained such that it does not learn interdependent set of feature weights. [23]

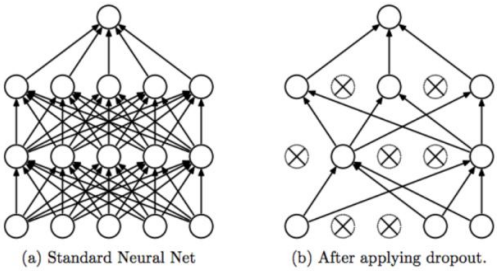

Figure 3: Importance of dropouts [24]

Tuning the hyperparameters-

Epochs: Epochs were tuned and finally set to 20. As the numbers of epochs were increased, the network overfit the data, meaning it did not generalise well on test data. Whereas if the number of epochs were too low, the network underfit the data.

Number of trainable layers: Since the architecture of resnet50 is too deep for our training data, it was important to reduce the architecture. We did this by freezing some layers of the network. As greater number of layers were frozen, it was observed that the network underfit the data. Hence, the first 10 layers were frozen.

Dropout: The dropout rate was set to 0.5. This means that every neuron had a 50% probability to be dropped.

Final result: A training accuracy of 94% and validation accuracy of 91% was observed. The network is still overfitting but this can be reduced by increasing the dataset.

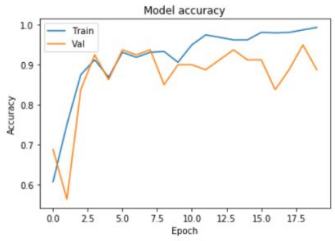

Figure 4: Model Accuracy

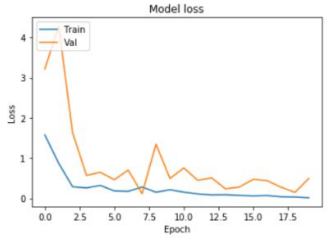

Figure 5: Model Loss

3.4 Question generation

The secondary aim of this module is to get back the attention of the viewer when he is drowsy. So, to draw back his attention the viewer is asked to answer a few questions based on the video watched till that moment. The questions are generated at that instant and a random question is picked from the generated questions which the learner is supposed to answer.

3.4.1 Retrieval of Online Course Video Transcripts

We made use of "youtube_transcript_api" to get the transcripts of the video. It is a python API which allows one to get the transcripts/subtitles for a given YouTubevideo. It returns a dictionary with following keys 'text', 'start', 'duration' which contains the text, the start time of that text and the duration of it respectively. We used this because apart from text it also provides the start time and duration that are needed to generate questions dynamically.

3.4.2 Analysis of Various Transcripts

We analysed the content of the videos and observed that while delivering a lecture, the part of speech of utmost importance for question generation was nouns [25]. Apart from this we observed that pre-processing the transcript text was necessary for better results. But some unimportant words like 'the', 'something' also fall under the category nouns. So instead of nouns, noun phrases could give better results by eliminating all such words [26]. We also used the concept of term frequency to determine the words that aren't much related to the content but are used frequently.Such words were added to the list of stop words. Few of the stop words are *'she', 'i', 'am', 'thereby', 'seems', 'they', 'somewhere', 'with', 'otherwise', 'bottom', 'enough', 'against', 'is', 'toward', 'myself', 'seemed', 'give', 'seem', 'each', 'hence', 'indeed', 'does', 'however'*.

3.4.3 Question Generation

We have used spaCy which is a library for advanced Natural Language Processing in Python and Cython [27]. It's built on the very latest research, and was designed to be utilized in real products. spaCy comes with pretrained statistical models and word vectors. It features state-of-the-art speed, convolutional neural network models for tagging, parsing, named entity recognition and many other things [28, 29]. It's also easy to use and is the best choice for analysing text. It helps tokenize a statement and get the part of speech of each word/token.

Using spaCy's 'noun chunks' we directly got the required noun chunks that were present in the text. We had to pre-process these chunks to get better and more useful noun chunks. We used spaCy's stop words' list and added a few more words found while analysing and if the noun chunk contained a stop word, we removed it. We then searched the transcript for the sentence corresponding to the noun chunk and generated fill in the blanktype question.

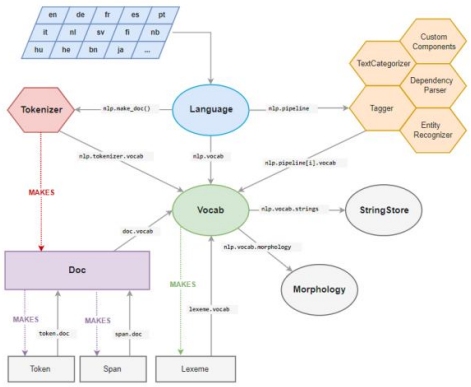

Figure 6: Question Generation [30]

4. CONCLUSION

This paper presents a module which detects drowsiness of an online course attendee. The module automatically generates fill in the blank type pertinent questions about the video if the attendee is found drowsy. Depending upon the answers of all the questions either incentive would be awarded or penalties would be imposed on the user. This is an attempt to make online courses more effective and useful.

Facial indicators of drowsiness such as position of eyelid and yawning were detected using eye and mouth aspect ratio and a CNN model ResNet50. Our module was trained on self-made dataset with a training accuracy of 97% and validation accuracy of 94%. The questions were generated using an NLP library namedSpacy. The produced questions were based on the video's transcript.

This module can be used as an extension with online course providers to keep a check on the user's attentiveness and make the courses more interactive and efficient. The module can also be used to generate analysis of the courses. For example, a part of a video, where most of its viewers were detected drowsy, can be reviewed by the course providers. So, the analysis report of the videos can be generated for the course providers along with the improvement suggestions.

REFERENCES

1. Joseph L. Sazon. **Driver's Attention Monitoring System with Low Light Capability**, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No. 4, pp. 1153-1158, 2019. https://doi.org/10.30534/ijatcse/2019/50842019

2.Md ZahangirAlom, Tarek M. Taha, Chris Yakopcic, Stefan Westberg, PahedingSidike, MstShamima Nasrin, Brian C Van Essen, Abdul A S. Awwal, and Vijayan K. Asari. **The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches.** 2017.

3.A. Berg, J. Deng, and L. Fei-Fei. **Large scale visual recognition challenge** 2010. www.imagenet.org/challenges. 2010.

4.J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. **ImageNet: A Large-Scale Hierarchical Image Database.** In CVPR09, 2009. https://doi.org/10.1109/CVPR.2009.5206848

5.Hsieh, C.-S.; Tai, C.-C. **An improved and portable eye-blink duration detection system to warn of driver fatigue.** Instrum. Sci. Technol. 2013.

6.Bulling, A.; Ward, J.A.; Gellersen, H.; Tröster, G. **Eye movement analysis for activity recognition using electrooculography.** IEEE Trans. Pattern Anal. Mach. Intell. 2011. https://doi.org/10.1109/TPAMI.2010.86

7.Chittaro, L.; Sioni, R. **Exploring eye-blink startle response as a physiological measure for affective computing.** In Proceedings of the Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, Switzerland, 2–5 September 2013.

8.Champaty, B.; Pal, K.; Dash, A. **Functional electrical stimulation using voluntary eyeblink for foot drop correction.** In Proceedings of the International Conference on Microelectronics, Communication and Renewable Energy, Kerala, India, 4–6 June 2013.

9.Chang, W.-D.; Lim, J.-H.; Im, C.-H. **An unsupervised eye blink artifact detection method for real-time electroencephalogram processing.** Physiol. Meas. 2016. https://doi.org/10.1088/0967-3334/37/3/401

10.L. Pauly and D. Sankar, "**Detection of drowsiness based on hog features and svm classifiers**," in 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 181–186, Nov 2015.

11.Cesar A. Llorente, Elmer P. Dadios. **Development and Characterization of a Computer Vision System for Human Body Detection and Tracking under Low-light Condition**, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No. 2, pp. 251-254, 2019. https://doi.org/10.30534/ijatcse/2019/24822019

12.T. Soukupova and J. Cech, "**Real-time eye blink detection using facial landmarks**," in 21st Computer Vision Winter Workshop (CVWW2016), 2016.

13.N. Boyko, O. Basystiuk and N. Shakhovska, "**Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and Opencv Library**," 2018 IEEE Second International Conference on

Data Stream Mining & Processing (DSMP), Lviv, 2018, pp. 478-482. doi: 10.1109/DSMP.2018.8478556.

14.X. Zhu and D. Ramanan, "**Face detection, pose estimation, and landmark localization in the wild**," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 2879-2886. doi: 10.1109/CVPR.2012.6248014.

15.Novie Theresia Br. Pasaribu, AgusPrijono, RatnadewiRatnadewi, Roy PramonoAdhie, and Joseph Felix, **Drowsiness detection according to the number of blinking eyes specified from eye aspect ratio value modification**, 1st International Conference on Life, Innovation, Change and Knowledge (ICLICK 2018), Atlantis Press, 2019/07.
https://doi.org/10.2991/iclick-18.2019.35

16.Mehta, Sukrit and Dadhich, Sharad and Gumber, Sahil and Jadhav Bhatt, Arpita, **Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio** (March 20, 2019). Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur - India, February 26-28, 2019.
https://doi.org/10.2139/ssrn.3356401

17.JiuxiangGua, ZhenhuaWangb, Jason Kuenb, Lianyang Mab, Amir Shahroudyb, Bing Shuaib, Ting Liub, XingxingWangb, Li Wangb, Gang Wangb, JianfeiCaic, TsuhanChenc. **Recent Advances in Convolutional Neural Networks.** Oct 2017.

18.K. He, X. Zhang, S. Ren, and J. Sun. **Deep residual learning for image recognition.** In CVPR, pages 770–778, 2016.

19.B. K. Savaş, S. İlkin and Y. Becerikli, "**The realization of face detection and fullness detection in medium by using Haar Cascade Classifiers**," *2016 24th Signal Processing and Communication Application Conference (SIU)*, Zonguldak, 2016, pp. 2217-2220.doi: 10.1109/SIU.2016.7496215

20.Krizhevsky, A., Sutskever, I., and Hinton, G. **ImageNet classification with deep convolutional neural networks.** 2012.

21.Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. **Dropout: A Simple Way to Prevent Neural Networks from Overfitting.** 15(Jun):1929−1958, 2014.

22.Nitish Srivastava**. Improving Neural Networks with Dropout.** 2013.

23.G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov. **Improving neural networks by preventing co-adaptation of feature detectors**, CoRR abs/1207.0580.

24.Amar Budhiraja - **Dropout in Machine Learning** https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5

25.Michael Heilman. 2011. **Automatic factual question generation from text.** Ph.D. Dissertation. Carnegie Mellon University, USA.

26.Ali, Husam, Yllias Chali and Sadid A. Hasan. "**Automation of Question Generation From Sentences.**" (2011).

27.F. N. A. Al Omran and C. Treude, "**Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments**," 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), Buenos Aires, 2017, pp. 187-197.
https://doi.org/10.1109/MSR.2017.42

28.Colic N, Rinaldi F. **Improving spaCy dependency annotation and PoS tagging web service using independent NER services.** Genomics Inform. 2019;17(2):e21. doi:10.5808/GI.2019.17.2.e21

29.Jiang, Ridong & Banchs, Rafael & Li, Haizhou. (2016). **Evaluating and Combining Name Entity Recognition Systems**. 21-27. 10.18653/v1/W16-2703.

30.spaCy 101: **Everything you need to know**, https://spacy.io/usage/spacy-101