Science and Technology Publishing (SCI & TECH)
ISSN: 2632-1017
Vol. 6 Issue 10, October - 2022

Development Of Facial Stress Level Detection System For Driving Safety Using Deep Learning

Udoiwod, Emmmanuel Nsese[1]
Department Of Electrical/Electronic And Computer Engineering,
University of Uyo, Akwa Ibom State Nigeria
eudoiwod@gmail.com

Ubong Linus Okure[2]
Department Of Electrical/Electronic And Computer Engineering,
University of Uyo, Akwa Ibom State Nigeria

Ofonime Dominic Okon[3]
Department Of Electrical/Electronic and Computer Engineering
University of Uyo, Akwa Ibom State Nigeria

***Abstract*—In this research work, a deep learning approach using a YOLO convolutional neural network (YCNN) algorithm was used to determine the facial stress level of drivers for their overall safety and that of others. A camera is placed on the dashboard that continuously tracks the face of the driver's image at real time and the model extracts basic features that helps to determine if the driver is drowsy or distracted. An alarm is triggered that alerts the driver when his/her face is off the car screen. Eye aspect ratio is used to calculate when the driver is gradually sleeping off or when eyes are closed. 10,000 images of drivers were obtained and splitted for the training, testing and validation phases in the ratio of 60: 20: 20. The results obtained after testing indicates 94% accuracy of the model. The model has a wide application in the areas of human computer communication, facial expression recognition, driver fatigue determination and autonomous or self - driving vehicles.**

Keywords: Facial Stress Level, Haar-Cascades Classifiers , Deep Learning, YOLO Convolutional Neural Network, ReLU Activation Function, Google Collaborator

1. Introduction

An accident is an unfavorable event that occurs inadvertently and unexpectedly, usually resulting in human and property damage as well as possible total annihilation [1,2,3,4,5,6]. Given the potential repercussions of an accident, it could be concluded that a driver would not deliberately drive with the intent of causing an accident. Again, a driver's license is a basic requirement to drive in any country of the world, and would-be drivers are taught and tutored about different driving rules, signs and safety precautions during the licensing procedure [7,8,9,10,11,12]. Despite these attempts, accidents still occur, and the human factor is surprisingly attributed as one of the leading causes of these accidents.

Remarkably, car driving is a difficult process in which small lack of attention and concentration can lead to catastrophic occurrences, and many researchers have undertaken researching on drivers' physical and emotional status, which reflects their internal state. However, studying and understanding the facial state of drivers in the driving environment would help promote and reduce the percentage of accidents that is recorded on the road. Statistically, it is recorded that in the driving environment, factors like stress, fatigue, anger, sadness of drivers contribute to a large extent, the increase in the rate of accidents [13,14,15,16,17,18,1920]. Therefore systems and models need to be built and developed to help the drivers while on the steering to avoid loss of life and property. Therefore, in this paper, a deep learning approach using a YOLO convolutional neural network (YCNN) algorithm was used to determine the facial stress level of drivers for their overall safety and that of others 21,22,23,24,25,26,27,28,29,30,31,32]. The system was trained and implemented in Google Colaboratory environment using an open source framework called tensor flow and kerass [33,34,35,36,37,38,39,40].

During the implementation stage, inside the car, a camera is mounted facing the driver. The camera is utilized to record driver's face photos in real-time. The YCNN-based system is able to determine from the driver's face photos if the eye of a vehicle driver is closed or when there is drowsiness or when the driver is not concentrating on the car wind screen. This is done by determining the eye aspect ratio of the driver. The model is then programmed to sound an alarm if the values are consistently below the fixed threshold for at least 35 photo frames. In this way, the driver can be assisted by this system to avoid accidents that are associated with drowsiness, sleeping or lack of concentration on the part of the driver.

2. Methodology

The focus in this paper is to develop a deep learning-based Convolution Neural Networks (CNN) system that can be trained and then used to detect the face stress level of car driver and then trigger alarm to alert the driver that he or she is sleepy or drowsy or loosing concentration, and hence he or she should respond to the alert by stopping the car to take some rest or he should concentrate and avoid whatever is distracting him or her. The flow diagram used in the study is shown in Figure 1. According to the procedure in Figure 1, the phases in the model development include: data collection, pre-processing phase, network model phase, training phase and validation and testing.

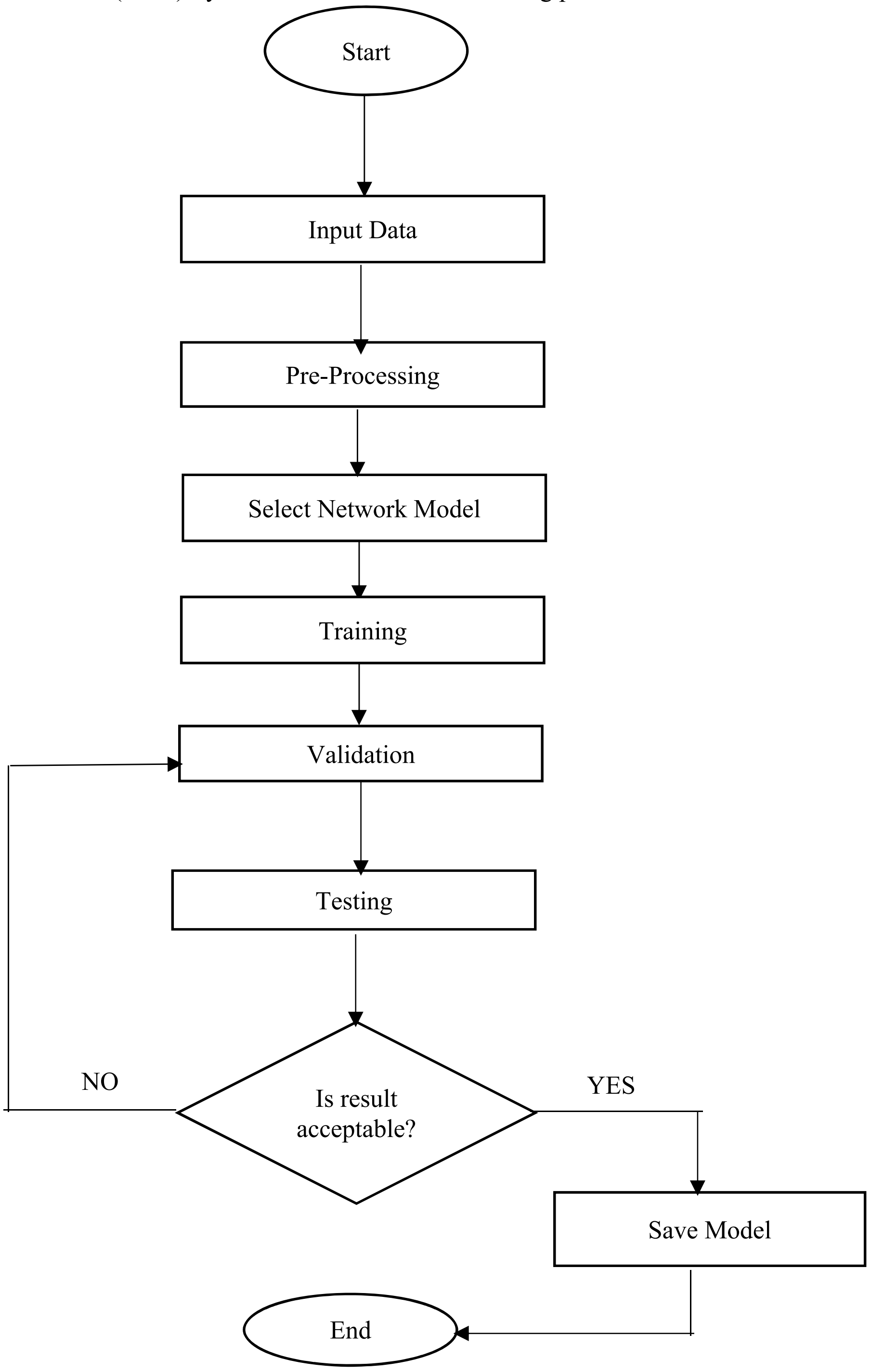

Figure 1 Flow diagram of the system model

Dataset of drivers who are stressed or not focused on the steering was sourced on the Internet. About 10,000 of relevant images of drivers were downloaded from Kaggl website available at https://www.kaggle.com/uciml/datasets and https://archive.ics.uci.edu/ml/index.php. The dataset includes images of drivers that were stressed, drivers that were distracted, drivers whose eyes were closed (a case of sleeping while driving), and drivers whose eyes were not focused on the car screen. These images were divided into the training datasets and the testing datasets. The validation dataset was gotten from the training/testing datasets using the Shutil Operation in python environment.

The network was developed using YOLOv3 Convolutional Neural Network (YCNN). The YCNN was used because it is mostly suitable for object detection and it consists of 53 convolutional layers otherwise called darknet – 53. For detection tasks, additional 53 layers are stacked to the original architecture making it 106 layers. Yolov3 detection is made in three layers: 82, 94 and 106 layers.
The experiment was performed on google colaboratory with RAM size of GPU12gb. The framework is for recognizing facial stress level for driving safety. The CNN model is trained using the Kaggle dataset by adjusting the hyper parameters like epochs, learning rate, batch size, and so on. The batch size for this study was 34, and the training lasted around 2200 epochs. A test face detected image from the front camera sensor is sent into the trained CNN-based facial stress detection system. Open-CV is used to process the input camera images, and the Haar-Cascade classifier is used to recognize faces, because it uses integral pictures, which decrease duplicate operations. Classification approach was adopted in this network for predicting likelihood of facial features; softmax activation function was used in the final layer to detect if a certain feature corresponded to features of interest which include , nose , eye and jaw line, then a bounding block was drawn enclosing those features.
Particularly, in this paper, a deep learning system is used to construct the face stress level detection system by training Convolution Neural Networks (CNN) with face image dataset of drivers. When compared to other pattern recognition systems, deep learning-based systems provide a substantial advantage in accuracy and performance. In particulalr, Haar-Cascades classifiers are used to process the camera's incoming input image and detect the face (ViolaJones Algorithm). Later, the image of the recognized face is supplied into the CNN input for feature extraction and learning. Inside the CNN, feature extraction is done via a cascade of convolutional layers followed by a pooling layer.

The convolutional layer is made up of many filter banks with varying orientations that are used to extract the features in an image using a 2D convolution operation that extracts all of the directional characteristics. The feature picture is supplied into the pooling layer, which reduces the dimension of the feature map by down sampling it. To bring non-linearity into the system, a non-linear ReLU activation function was applied to the down sampled image. The image's dimension is greatly reduced because to the cascaded layers of convolutional and pooling layers. Finally, the derived non-linearly activated feature map is input into a softmax activation function output layer.
In operation, a camera is utilized to record photos in real-time during the implementation stage. Inside the dashboard, a camera is mounted facing the driver. To track a set of facial landmarks, a face tracker is used. To classify facial emotion classes in each frame, the holistic or local texture features are retrieved. A pose normalization step is used to compensate for the pose mismatch caused by head movements and camera view by generating a virtual frontal image of the face. The system is able to determine if the eye of a vehicle driver is closed or when there is drowsiness or when the driver is not concentrating on the car wind screen. This is done by determining the eye aspect ratio of the driver. The system was trained and implemented in Google Colaboratory environment using an open source framework called tensor flow and kerass.

3.0 Results and discussion

The driver's camera's live video feed is captured and processed into frames. The ocular region is localized with the help of the dlib library. After that, the eye is calculated using Euclidean distance for each frame. The model is then programmed to sound an alarm if the values are consistently below the fixed threshold for at least 35 frames. Sample snapshot of image showing a closed eye is given in Figure 2 while Sample snapshot of image showing a drowsy face is given in Figure 3.

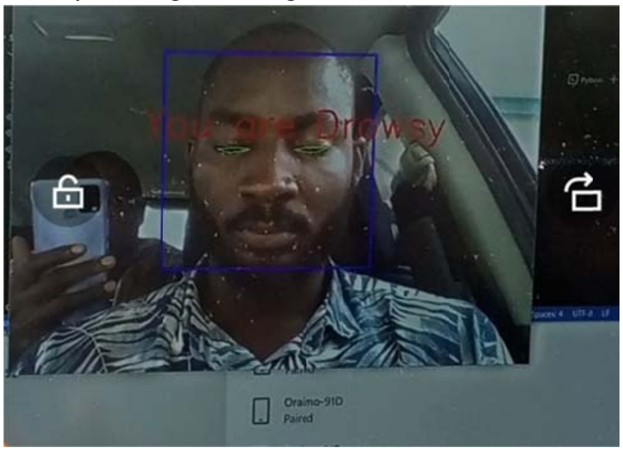

Figure 2Snapshot of image showing a closed eye

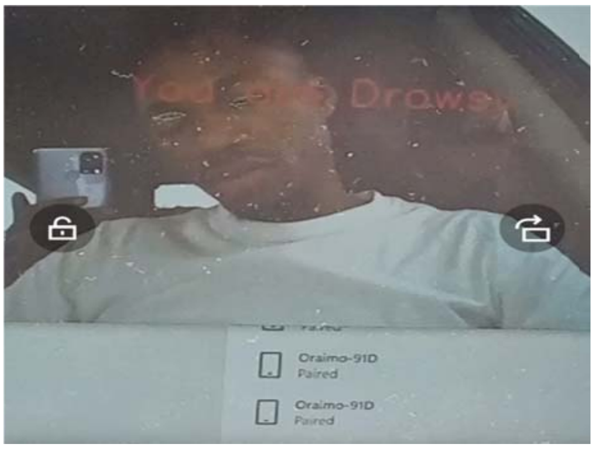

Figure 3 Snapshots of image showing a drowsy face

In order to know the systems level of compliance when exposed to other factors, the system was further subjected to test in different scenarios. The following factors were considered and the corresponding findings from the results are as follows:

i. Eye colour and skin colour: the system was not affected by the eye colour and skin colour of the driver.
ii. Glasses: the model did not work with drivers that wear classes because the eyes region, which is a major feature that was extracted during the training, was not being able to be captured by the camera.
iii. Facemasks: the system works perfectly with drivers wearing facemask so far the facemask does not cover the eye region of the faces.
iv. Beard: the model was still working perfectly with beard because it was not a feature of interest as far as this research is concerned.

The system was also evaluated with varying distance of driver position from the camera, the results show that at a distance of 4 cm between the camera and the driver the system performs significantly poor. And at a distance of 3cm, the object was detected but the system did not pop up an alarm indicating when the driver is drowsy. However, at a distance of 2.5cm, the system works optimally and was able to detect a drowsy driver and a driver that is distracted by popping an alarm if the eye aspect ratio is below the threshold that was specified.

The system was also evaluated at different angles ranging from -90° to +90°. The results show that at an angle of -90, the system did not work optimally. This is because, the camera could not extract the needed feature for processing, but when the angle was set at +90°, the system was able to capture the images of drivers and also extract the necessary features for preprocessing and computation. The system was further evaluated under varying conditions of light; particularly, the night time and the day time. The results show that during the day time, the system perform normal but during night, the system performs poorly.

The confusion matrices of the system model were carried out to ascertain the level of accuracy of the model. Figure 4 and Figure 5 illustrate the matrices when a driver is drowsy or not drowsy and the matrix to illustrate when the driver is concentrating and when he is not. The result shows 90% accuracy for the YOLOv3 model.

n = 35	**Not drowsy**	**Drowsy**
Not drowsy	10	5
Drowsy	5	15

Figure 4 Matrix showing the drivers state of drowsiness

n = 35	Not distracted	Distracted
Not distracted	10	4
Distracted	5	11

Figure 5Matrix showing drivers distraction level

In this study, YOLOv3 was adopted as a model because of its peculiarities as compared to the earlier version and other object detection algorithm. Some of these peculiarities include speed, precision and specificity of classes. Particularly, the Darknet-19 was used as the backbone feature extractor in YOLOv2, but Darknet-53 is currently used in YOLOv3. Darknet-53 contains 53 convolutional layers, compared to 19 in Darknet-19, making it more powerful and efficient than other rival backbones (ResNet-101 or ResNet-152).

In terms of Mean Average Precision (mAP) and Intersection Over Union (IOU) values, YOLOv3 is also quick and precise. It is substantially faster than the other methods of detection with comparable results. Furthermore, by simply modifying the model's size, it can easily be traded-off between speed and accuracy with no retraining necessary. The results of the models evaluation showing billions of operations per second (Ops), billion floating-point operations per second (BFLOP/s), and frames per second (FPS) for various backbone frameworks are shown in Table 1. The results show that the Darknet-53 is about 1.5 times faster than ResNet101. Darknet 53 is still as precise as ResNet-152 while being two times faster. Using

Nvidia GPU, YOLOv3 is substantially faster than other detection algorithms with comparable performance.

Table 1 Results of the models evaluation showing billions of operations per second (Ops), billion floating-point operations per second (BFLOP/s), and frames per second (FPS) for various backbone frameworks

Backbone	Ops	BFLOP/s	FPS
Darknet-19	7.29	1246	171
resNet-101	19.7	1039	53
ResNet-152	29.4	1090	37
Darknet-53	18.7	1457	78

Again, as shown in Figure 6, YOLOv3 achieved equivalent mAP@0.5 with a substantially faster inference time than RetinaNet. For example, YOLOv3 achieved 57.9% mAP in 51 milliseconds, but RetinaNet-101 achieved 57.5 percent mAP in 198 milliseconds, which is 3.8 milliseconds faster. During training of the model for extracting eyes from image frames sing YOLOv3, it took 2200 epochs for the model to sit at 70.1 mean average precision (mAP), as shown in Figure 7. This result showed some signs of over-fitting because when the models' validation score stood at 66.8 mAP and test results at 64.3 mAP. However, this amount of over fitting was acceptable for the purpose of this study as the training set was relatively very small compared to the standard coco dataset which was initially used for training YOLOv3 and other state of the art object detection algorithms. In this study, there are 5000 images of which, 1000 images were saved for validation , 1000 images for testing and 3000 images preserved for training. The results showing the training data, training size, classes, mAP and FPS of different system models are presented in Table 2.

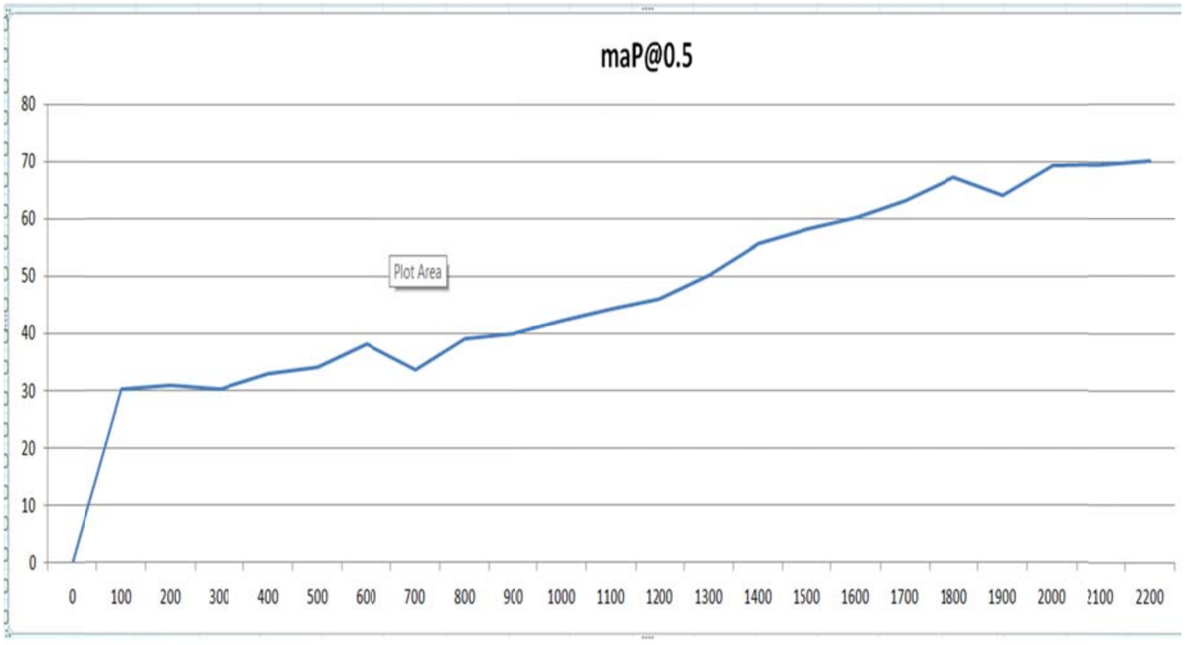

Figure 6 The plot of mAP versus the inference time for YOLOv3 model

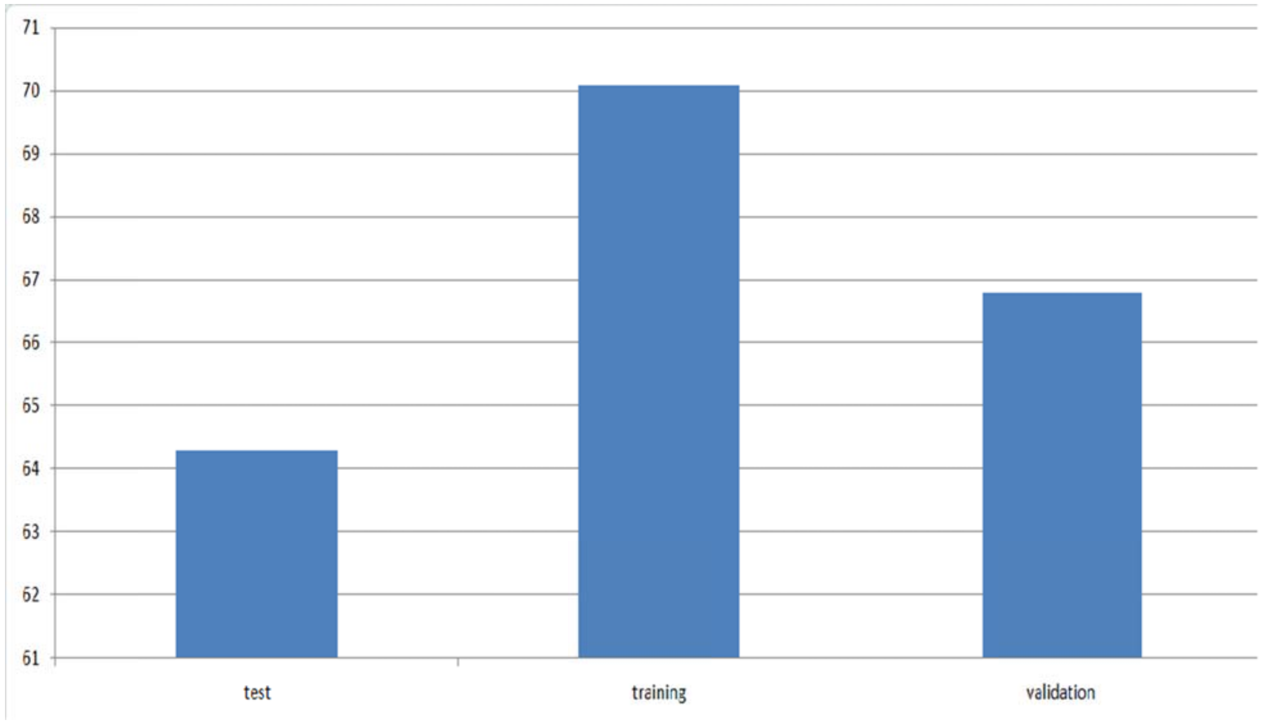

Figure 7 The bar chart showing the mAP of training, validation and test results

Table 2 Results showing the training data, training size, classes, mAP and FPS of the different models considered in the study

MODEL	TRAINING DATA	TRAINING DATA SIZE	NO OF CLASSES	GPU USED AT TEST	mAP	Fps
yolo v3 tiny	eye region dataset	3000	1	no	70.1	15
faster-RCNN	coco dataset	328000	91	yes	73.2	7
yolo v3 tiny	coco dataset	328000	91	yes	52.7	155

The three models used for comparison in this study (as shown in Table 2) are meant to highlight the major delima faced in selecting the right model for this work. According to the results in Table 2, YOLOv3, which is a light weight version of YOLOv2 (which is also used in this study) , boast of faster inference time than any of the other models. However, this speed is still greatly influenced by the availability of a hardware accelerator such as GPU. It can be seen in Table 2 that the YOLOv3 model trained for this system was run on a computer without a GPU but still boasted of frame rates greater and faster than the RCNN.

However , the accuracy of the models tell a different story all together; faster RCCN outperforms both YOLO models even as the YOLOv3 model in this study is proven to be over fitted. The faster RCNN outperforms the YOLOv3 model in terms of accuracy because the research model only needs to classify a single class, as opposed to the aster RCNN having to classify 91 classes. This outclassing was too significant not to explore. So, in order to further compare the results, YOLOv3 model was used to train the same dataset and the same class. It was discovered (as shown in table 2) that, the YOLO tiny model trained on the 91 classes was also significantly outperformed by faster RCNN model in terms of precision. But, nevertheless,YOLOv3 was greatly faster at inference.

Notably, from Figure 8, it is obvious that accuracy and speed become very essential while detecting facial stress level of a driver for driving safety. However, in driving environment, the safety of the driver depends on how fast a driver is alerted when he/she is drowsy or distracted. So it is said that how fast the warning happens is important than how accurate the warning happens, because if a system detects that a driver is about to sleep when in reality the driver is not about to sleep, that false positive does not actual have any negative effect or impact on the driver or there is no mortality involve, but on the other hand, if it takes a longer time for a True positive warning to come as the case may be as seen with the Faster-RCNN model, then the car will most likely crash before the system warns the driver. Based on the fact that this system is deployed on little edge devices with little computing resources, speed becomes a very important parameter and as such very essential to deploying such a system.

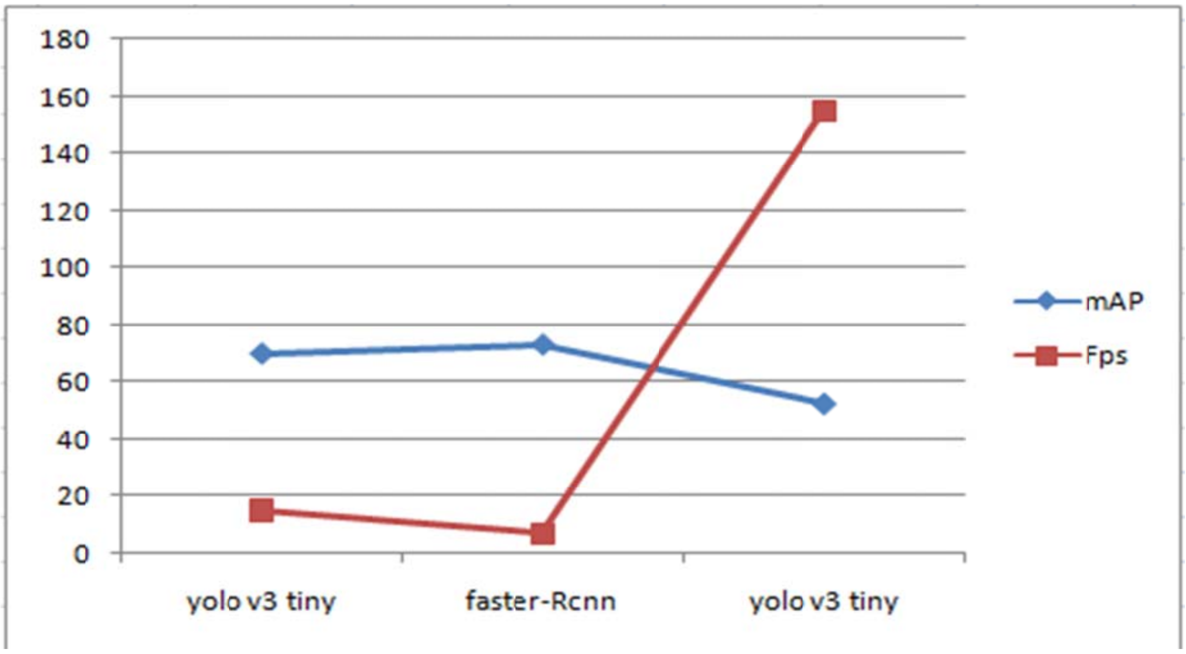

Figure 8 Speed versus precision trade–off for the different models considered in the study

4. Conclusion

In this paper, a deep learning technique is used to develop a face stress level detection system by training Convolution Neural Networks (CNN) with face image data from drivers. The network is developed using YOLOv3Convolutional Neural Network (YCNN) as the deep learning algorithm. The YCNN model developed is used for real-time monitoring system that detects the driver's facial stress level, by studying facial expressions on the live (real-time) face image captured from the driver. The deep CNN model is used to track special features and the general face in order to use it to determine the stress level. The system is designed to monitor the eye if it is closed or about to close or how many times it is closed in a single seconds. This help the system to determine if the driver is stress or about to sleep.

Specifically, the YCNN algorithm is used to detect the eye and other features of interest, the CNN is also used to detect if the features extracted are that of a stressed person. The network is train to recognize the difference between a close eye and an open eye. A period when the eye is open and when the eye is about to close. During this monitory phase, it alerts the driver by beeping an alarm for the driver to stop driving and pack to have rest. In all, the results showed that the trained YOLOv3Convolutional Neural Network (YCNN) is quite fast in detecting the stress level and hence enabling the system to promptly trigger the alarm on time for the driver to take the right decision before it becomes late.

References

1. Utriainen, R., & Pöllänen, M. (2021). The needed features of connected and automated vehicles to prevent passenger car crashes caused by driving errors. *Future transportation*, *1*(2), 370-386.
2. Agustin, I. W. (2019, October). Analysis of Car Accident at the Location of Black-Spot and Rating for Accident-Prone Roads in Surabaya. In *IOP Conference Series: Earth and Environmental Science* (Vol. 328, No. 1, p. 012023). IOP Publishing.
3. Fiksel, J. (2015). From risk to resilience. In *Resilient by design* (pp. 19-34). Island Press, Washington, DC.
4. Caban, J., Karpiński, R., & Barta, D. (2018, April). Road traffic accident injuries—Causes and biomaterial related treatment. In *2018 XI International Science-Technical Conference Automotive Safety* (pp. 1-7). IEEE.
5. Mohammed, A. A., Ambak, K., Mosa, A. M., & Syamsunur, D. (2019). A review of traffic accidents and related practices worldwide. *The Open Transportation Journal*, *13*(1).
6. van Pernis, K. A. (2019). *Influence of Semi-Autonomous cars on road safety in the Netherlands* (Bachelor's thesis).

7. Sergeev, I., Gayko, E., & Pirtovsek, D. (2019). THE AMMOUNT OF COST NEEDED FOR A STUDENT TO TAKE THE DRIVING TEST IN SLOVENIA, CROATIA, NORTHERN MACEDONIA AND RUSSIA.
8. Waehlte, S. (2018). Individuals with Intellectual and Developmental Disabilities Obtaining Washington State Driver's Licenses.
9. Driessen, T., Picco, A., Dodou, D., de Waard, D., & de Winter, J. (2021). Driving examiners' views on data-driven assessment of test candidates: An interview study. *Transportation research part F: traffic psychology and behaviour*, *83*, 60-79.
10. Castile, M. (2015). *Driver's license*. Bloomsbury Publishing USA.
11. Rosenblat, A. (2018). *Uberland: How algorithms are rewriting the rules of work*. Univ of California Press.
12. Dernbach, J. C., Singleton, R. V., Wharton, C. S., Wasson, C. J., & Ruhtenberg, J. M. (2021). *A practical guide to legal writing and legal method*. Aspen Publishing.
13. Magaña, V. C., Scherz, W. D., Seepold, R., Madrid, N. M., Pañeda, X. G., & Garcia, R. (2020). The effects of the driver's mental state and passenger compartment conditions on driving performance and driving stress. *Sensors*, *20*(18), 5274.
14. Dingus, T. A., Guo, F., Lee, S., Antin, J. F., Perez, M., Buchanan-King, M., & Hankey, J. (2016). Driver crash risk factors and prevalence evaluation using naturalistic driving data. *Proceedings of the National Academy of Sciences*, *113*(10), 2636-2641.
15. Halim, Z., & Rehan, M. (2020). On identification of driving-induced stress using electroencephalogram signals: A framework based on wearable safety-critical scheme and machine learning. *Information Fusion*, *53*, 66-79.
16. Alonso, F., Esteban, C., Gonzalez-Marin, A., Alfaro, E., & Useche, S. A. (2020). Job stress and emotional exhaustion at work in Spanish workers: Does unhealthy work affect the decision to drive?. *PLoS one*, *15*(1), e0227328.
17. Hatami, A., Vosoughi, S., Hosseini, A. F., & Ebrahimi, H. (2019). Effect of co-driver on job content and depression of truck drivers. *Safety and Health at Work*, *10*(1), 75-79.
18. Kandeel, A. A., Abbas, H. M., & Hassanein, H. S. (2021, January). Explainable model selection of a convolutional neural network for driver's facial emotion identification. In *International Conference on Pattern Recognition* (pp. 699-713). Springer, Cham.
19. Alavi, S. S., Mohammadi, M. R., Souri, H., Kalhori, S. M., Jannatifard, F., & Sepahbodi, G. (2017). Personality, driving behavior and mental disorders factors as predictors of road traffic accidents based on logistic regression. *Iranian journal of medical sciences*, *42*(1), 24.
20. Bowen, L., Budden, S. L., & Smith, A. P. (2020). Factors underpinning unsafe driving: A systematic literature review of car drivers. *Transportation research part F: traffic psychology and behaviour*, *72*, 184-210.
21. Zaghari, N., Fathy, M., Jameii, S. M., Sabokrou, M., & Shahverdy, M. (2021). Improving the learning of self-driving vehicles based on real driving behavior using deep neural network techniques. *The Journal of Supercomputing*, *77*(4), 3752-3794.
22. Oh, G., Ryu, J., Jeong, E., Yang, J. H., Hwang, S., Lee, S., & Lim, S. (2021). Drer: Deep learning–based driver's real emotion recognizer. *Sensors*, *21*(6), 2166.
23. Sirohi, D., Kumar, N., & Rana, P. S. (2020). Convolutional neural networks for 5G-enabled intelligent transportation system: A systematic review. *Computer Communications*, *153*, 459-498.
24. Jiang, Y., & Li, C. (2020). Convolutional neural networks for image-based high-throughput plant phenotyping: a review. *Plant Phenomics*, *2020*.
25. Wu, J. D., & Chang, C. H. (2022). Driver Drowsiness Detection and Alert System Development Using Object Detection. *Traitement du Signal*, *39*(2).
26. Ankile, L. L., Heggland, M. F., & Krange, K. (2020). Deep Convolutional Neural Networks: A survey of the foundations, selected improvements, and some current applications. *arXiv preprint arXiv:2011.12960*.
27. Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., ... & Kohno, T. (2018). Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*.
28. Fayyad, J., Jaradat, M. A., Gruyer, D., & Najjaran, H. (2020). Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors*, *20*(15), 4220.
29. Gupta, A., Anpalagan, A., Guan, L., & Khwaja, A. S. (2021). Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, *10*, 100057.
30. Thakare, V., Moundekar, P., Chamat, P., Sangode, S., & Lande, Y. A SURVEY ON THERMAL FACE RECOGNITION USING MACHINE LEARNING.

31. Singh, A. K., Ganapathysubramanian, B., Sarkar, S., & Singh, A. (2018). Deep learning for plant stress phenotyping: trends and future perspectives. *Trends in plant science*, *23*(10), 883-898.
32. Li, Y., Sun, S., Zhang, C., Yang, G., & Ye, Q. (2022). One-Stage Disease Detection Method for Maize Leaf Based on Multi-Scale Feature Fusion. *Applied Sciences*, *12*(16), 7960.
33. Lux, M., & Bertini, M. (2019). Open source column: deep learning with Keras. *ACM SIGMultimedia Records*, *10*(4), 7-7.
34. Gulli, A., Kapoor, A., & Pal, S. (2019). *Deep learning with TensorFlow 2 and Keras: regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API*. Packt Publishing Ltd.
35. Christa, G. H., Jesica, J., Anisha, K., & Sagayam, K. M. (2021, May). CNN-based mask detection system using openCV and MobileNetV2. In *2021 3rd International Conference on Signal Processing and Communication (ICPSC)* (pp. 115-119). IEEE.
36. Basnet, R. B., Shash, R., Johnson, C., Walgren, L., & Doleck, T. (2019). Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks. *J. Internet Serv. Inf. Secur.*, *9*(4), 1-17.
37. Balbin, J. R., Sese, J. T., Ang, J. G., Tapang, J. S. S., & Valenzuela, J. D. (2021, July). Detection of Anal Sac Disease, Atresia Ani and Rectal Prolapse for Canis Lupus Familiaris Using Image Processing and Convolutional Neural Network. In *2021 5th International Conference on Imaging, Signal Processing and Communications (ICISPC)* (pp. 16-20). IEEE.
38. Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.".
39. Planche, B., & Andres, E. (2019). *Hands-On Computer Vision with TensorFlow 2: Leverage deep learning to create powerful image processing apps with TensorFlow 2.0 and Keras*. Packt Publishing Ltd.
40. Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, *6*, 61677-61685.