



Deep learning-based car seatbelt classifier resilient to weather conditions

Osama Hosam *

¹ Graphics Department, IRI, SRTA City Egypt. As with the collage of computer science and engineering in Yanbu, Taibah University, Saudi Arabia

*Corresponding author E-mail: mohandesosama@yahoo.com

Abstract

Deep Learning is a very promising field in image classification. It leads to the automation of many real-world problems. Currently, Car seatbelt violation detection is done manually or partial manual. In this paper, an approach is proposed to make the seat belt detection process fully automated. To make the detection more accurate, sensors are set to detect the weather condition. When specific weather condition is detected, the corresponding pre-trained model is assigned the detection task. In other words, a research is conducted to check the possibility of dividing the big-sized deep-learning model - that can classify car seatbelt, into sub-models each one can detect specific weather condition. Accordingly, a single specialized model is used for each weather condition, Deep convolutional neural network (CNN) model AlexNet is used in the detection/classification process. The proposed system is sensor based AlexNet (S-AlexNet). Results support our hypothesis that "Using single model for each weather condition is better than general model that support all weather conditions". On average, previous approaches that trained single model for all weather conditions have accuracy less than 90%. The proposed S-AlexNet approach successfully reaches 90+% accuracy.

Keywords: Seatbelt Detection; Deep Neural Network; Classification; True Positive.

1. Introduction

To avoid injuries and casualty resulting from drivers and passengers not wearing the seat belt, a research is committed to detect seatbelt violations. There is restricted research with respect to seat belt detection. The authors in [1] proposed a method for detecting the seat belt in a monitoring image which contains the full scene information of the moving car. The proposed method is based on two steps. In the first step, the driver area location is extracted, the driver area is the boundary of driver where the potential seat belt is located. This step is done through vertical boundary detection, and horizontal boundary detection. The vertical boundary is the left boundary of the driver area, it is considered as the right edge of license plate. The horizontal boundary is the top boundary of the driver area and is considered as the top edge of windscreen. The second step is seat belt detection in the driver area. This step is done through edge detection, then obtaining a candidate region and finally verification.

Most surveys directly employed a line detection technique relied on Hough transform. This technique is affected by vehicle factors like drivers clothing and steering wheel [13]. There are three keys to detect seat belt: vehicle detection, windshield detection, and seat belt detection. To detect vehicle and windshield, most algorithms are based on edge detect in which they firstly detect horizontal and vertical vehicle edge. Then they eliminate nonvehicle edges and estimate the vertical edge symmetry. At end, they obtain the region of interest (ROI) to extract desired features for the classifier [2]. Major challenges faced windshield detection such as vehicle color and lighting condition. Furthermore, an algorithm to detect windshield based on HIS color model has been proposed [3]. The drawback of this algorithm is using many iterations and not verify real-time requirements. Convolutional Neural Networks use convolutional layers for multiresolution analysis of the image. CNN get rid of segmentation phase (line segmentation, intensity-based segmentations, etc.). In CNN, that segmentation phase which is found in classical approaches are replaced with convolutional layers.

In this paper, we proposed novel deep neural network model to detect violation in seatbelt. The architecture of the proposed model contains multiple trained NN's, weather sensors, and decoder. The sensors are connected to the decoder and one neural network is activated according to which weather condition is sensed.

Our model depends on two main human behaviors:

- Multiple senses
- Specialization

It is well known that human does have 5 senses, namely smelling, sight, hearing, touching, and taste [4]. To recognize objects, a person may use one or more of his senses. Most of deep learning image classification algorithms use only the input from cameras [5]. Our idea is to use more than one input types, in this paper input from camera and input from hardware weather detection sensors are used. The same analogy of a human using more than one sense to comprehend and detect objects.

The idea of specialization is derived from human learning specialization. If there is a computer engineer, he might be specialized in micro-controller or software engineering. In this paper, we conduct research on the possibility of dividing the big-sized deep-learning model that can classify car seat-belt into sub-models each one can detect specific weather condition (specialized). Hence, a deep learning model [6] for classifying seat-belt at night is developed, another one for detecting seat-belt in rainy weather, a third one for detecting seat-belt in clear weather and a fourth one for detecting seat-belt in night-weather. Notice that the difference between weather conditions is not mainly an intensity variation difference like the case of different levels of brightness. In our case, the difference came from adding objects such as fog or rain to the image.

The idea is to use more specific and more specialized model for detection whenever the corresponding weather condition is detected. It is well known from literature that more specific classifier is more accurate than general classifier. Authors in [7] improved the accuracy of their model by training their classifier on more specific weather condition such as rainy weather. With multiple classifier or ensemble classifier, the decision is taken depending on decisions from multiple classifiers. However, ensemble classifiers are trained on random subset of the dataset [8].

In current research, training generalization is the trend. In [9], abstract features are obtained for training the classifier that is more resilient to multiple types and shapes of traffic signs. In [10] the authors made one model for detecting cars in different weather condition. In our research we take the opposite way, generalization or abstraction is replaced with specialization. For traffic sign classification [11], the signs might be divided manually into specific groups with similar properties such as red signs and blue signs. And even the same group can be sub-divided into sub-groups such as rainy and night captured signs. Then a trained classifier is created for each group. The accuracy of the resulting classifiers is expected to be higher than the abstract one.

In the proposed approach, we don't want to create very specific models and accordingly facing the overfitting problem. A tradeoff is needed to balance between generalization and specification. Highly abstracted systems are underfitted, and very specific systems may face overfitting. There are two types of specialized classifications

- **Mutual exclusive classifiers:** In this type of classification, multiple classifiers can't be merged to form single classifier. Such as classifying language words or classifying handwritten character. It depends on the language type so one classifier is configured for each separate language. Classifier for Italian, classifier for Indian, and so forth. It is not feasible to have multiple classifiers for the same language.
- **Non-Mutual exclusive classifiers:** The input or dataset has inherent similarity. Examples are classification of traffic signs in different weather conditions and creating one classifier to recognize all types of cars.

In the second type, even though the dataset has inherent similarity, it still can be sub-divided into smaller datasets, each dataset has common properties. For example, it is easier to classify specific model of cars compared to classifying all models of cars. With this analogy, car seatbelt detection can be done with multiple CNN classifiers. In our proposed Sensor based AlexNet (S-AlexNet), one classifier is specialized to detect specific weather condition. Our contribution is focused on increasing the accuracy of seatbelt detection in different weather conditions. A dataset is collected for car seatbelt in different weather conditions. One AlexNet model is trained for specific weather condition. So, if four weather conditions are considered, then four trained AlexNet's should be obtained. Sensors detect the weather condition and accordingly the corresponding trained AlexNet is selected to do the seatbelt detection task. This paper is organized as follows; next section displays the previous approaches related to seatbelt detection. Section 3 represents the proposed approach. Section 4 shows the results and discussion. The last section shows conclusions and future work.

2. Previous approaches

In [12], the authors used Adaboost technique to compute the angular point positioning of windshield. In addition, Wieli et al. [13] proposed Adaboost classifier based seatbelt detection system. Adaboost classifier is type of boosting classifier which is implemented by taking decisions from multiple weak classifiers. The extracted features can be Haar-like features [14], SIFT, etc. HSV colour model is employed to detect vehicle in addition to detect windshields based on dark coloured vehicle bodies. The abuse of this model is to not well carried out for light coloured vehicles in addition to strong light condition [15]

The Adaboost technique is enhanced in the research [16], [17] depended on boosting. The key advantage is to extensively detect and recognize objects very fast.

According to Yu & et al. [18] seat belt detection is considered one of major recognition field to control traffic inspection, which widely needed in observation operation of traffic rules. The several problems appear to detect a seat belt of drivers such as reflection and shelter in intelligent transportation system. Moreover, in technical the edge between vehicle window and driver is near to detect by methods of detection. However, the authors based on previous algorithms to detect seatbelt and extract seatbelt features of gradient orientation, after that select face recognition method to evaluate the area that assist to detect the drivers fasten seatbelt or not. The proposed method of Yu & et al. [18] is based on five pre-processing steps in first stage to extract the features from right side area of front vehicle window, which assumed by recognize vehicle plate location. However, these pre-steps will not be occurred if the driver sit on lift side, or the detection method is observing both drivers and other vehicle rider are fasten seatbelt. The second stage of proposed method is filtered the face of driver and the variation of illumination colour, which based on three steps (Image Enhancement, Calculation of Gradient Orientation and Seat Belt Feature Matching).

In the experiment, 213 pictures are used to test the proposed method, which are captured from traffic crossroad under different weather conditions as mentioned in article. The result of proposed method is detected 61 cases for driver fasten seatbelt from 77 cases with accuracy 79.2%, and 117 cases for driver unfasten seatbelt from 136 cases with accuracy 86%. However, these results were obtained by experiment on authors images with good condition and resolution (1360 * 1024) pixel. However, the proposed method is not conducted on other benchmark or image cases include variant situations.

Yusuf Artan et al [19] propose computer vision methods for detecting vehicle occupancy, seatbelt violation, and driver cell phone usage. methods consist of two stages. First, the vehicle's front windshield and side window from captured images using the deformable part model is localized. Next, a region of interest in the localized images is defined for each violation type and perform image classification using one of the local aggregation-based image features. A data set over 4000 images including front/side view vehicle images with seatbelt and cell phone violations was collected on a public roadway and is used to perform the experiments.

The seat-belt detection techniques are based on gap of colour between driver or passenger and their background in vehicle, or the seat-belt colour and colour of human cloths. The difficult issues are represented by nearest colour of foreground (pixel colour of drivers or passengers) and background (pixel colour of chairs) to recognize seat-belt detection or missed detection, where the seat-belt is fasten by drivers or missing the seat-belt. The traffic observing is needed to intelligence and automatic detection accuracy of recognize seat-belt is fasten.

Zhang [20] proposed method is relied on three stages; first, classification model to target vehicle including the steps of extract the vehicle features such as: type, colour, size and position of vehicle on roadmap. Second, driver and passenger position in vehicles, which responses on detection accuracy of fasten seat-belt reorganization. The second stage of Zhang proposed method is based on rigid and non-rigid to find relationship between human body and vehicle body by using supervised learning model. Third stage is obtained the seat-belt features based on marshalling process of seat belt unique linear on driver's body. The sub-steps in stage three based on features of image pixels in drivers and passenger's area such as grayscale and texture, edge detection points, lines, arcs and gradient orientation direction to make decision whether driver and passenger wearing seat-belt or not.

In [21], the authors adopted AlexNet [22] classical structure and updated it by adding BN (Batch Normalization) Module. They applied their technique to seat belt classification applications. BN-AlexNet approach outperforms the classical AlexNet. When BN layer is added to the network, the distribution of the parameters became more concentrated. The resulting model obtained more powerful classification accuracy in seat belt detection applications. BN-AlexNet can be used without the help of GPU because the training speed is improved. Before the classification, the Bootstrap is applied to BN-AlexNet to check the performance and find the rejection area. By rejecting many samples from the input data-set, the false acceptance rate is reduced, and the classification accuracy is enhanced.

In [23], the authors proposed an innovative approach to detect seat belt using multi-scale Convolution Neural Network (CNN) features in conjunction with deep neural network on numerous complete connection layers. Then they used SVM classifier to carry out the learning and inferencing detection score for each detected window. The key advantage of this approach is to accurate, robust and intuitive.

However, all the above-mentioned methods depend entirely on creating their own dataset. The images are clear with high resolution. None of the above approaches consider the effect of weather conditions (night, day, rain, etc.). They made a single NN model to detect the existence/non-existence of fasten seatbelt. In this paper, we proposed S-AlexNet model to detect car seatbelt in different weather conditions. Multiple AlexNet models are trained on the dataset, on AlexNet model is trained for specific weather condition. Results showed that Using single model for each weather condition is better than general model that support all weather conditions.

3. The proposed methodology

To detect the seatbelt with very high detection accuracy, the exact weather condition must be detected before applying deep neural network classification model. The proposed S-AlexNet approach has two stages, namely training and classification. In the training process, more than one CNN classifier will be trained, one classifier will be trained for single weather condition. In the classification process, first, the sensors will detect the weather condition and then the classification is done by using the corresponding trained CNN model.

3.1. Training

Input images in specific weather condition are selected to be an input for the training phase of CNN. For example, rainy images are used to train deep neural network model. A trained neural network is obtained for the rainy images, we call it Rainy Model (R) for simplicity. That trained model will be used in the future for classification of the input images. We have mainly 4 weather conditions, as shown in Figure 1, Rain (R), Night (N), Rain at Night (RN) and Clear (C).

3.2. Classification

For simplicity we will consider only two weather conditions, namely Rain and Night. The weather conditions will be detected using hardware sensors. Rain sensor and Night sensor. The output of the sensors will be binarized, either True or False. True means the required weather condition are met and False means the required weather condition are not detected. The output of the sensor is set to 0 (False) if the electrical output of the sensor is not passing specific threshold. If the sensor passed that threshold, so the output is considered 1 (True). This means the continuous output of the sensors are digitized by simple binarization process.

Two sensor outputs create 4 possible combinations. The first combination is no weather conditions detected, and hence the clear CNN trained model (C) will be used in classification. The second combination is night sensor only is active, and hence the trained night CNN model (N) will be used. The third combination is the Rain sensor only is active then the trained rain CNN model (R) will be considered for classification. The last combination is both sensors are active and the trained Rain-Night CNN model (RN) will be used.

The question is how to implement the above selection criteria? In other words, how to select the suitable trained CNN model according to sensor inputs? The best choice is to use 2 by 4 decoder. Generally, we can use $n \times 2^n$ decoder, n is the number of sensors used. The truth table of the required decoder is shown in Figure 2.

Why didn't we use webservices to detect the weather condition? The reason for this is because traffic authority put the camera detector in the street without internet connection. Webservices need internet connection to work. They collect the images of the violated drivers manually using laptop and the camera's SD memory. Manual collection of images is too much cheaper than using dedicated internet connection.

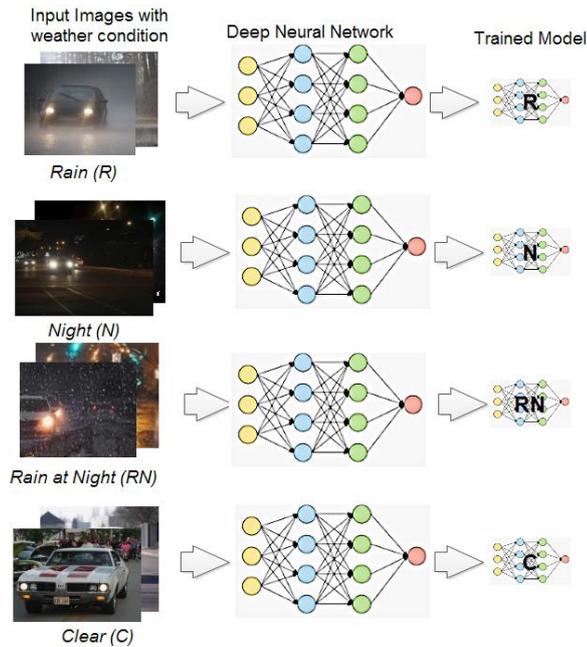


Fig. 1: Training Phase for CNN Models with Different Weather Conditions, Rain (R), Night (N), Rain at Night (RN) and Clear (C). in Each Row of the Figure, We Train Different CNN Model. So, Figure Rows are Independent. in Each Row, Input Images are Passed to CNN Model for Training. the Result Is A Trained CNN Model in Each Row.

4. Results and discussion

4.1. Collecting dataset images

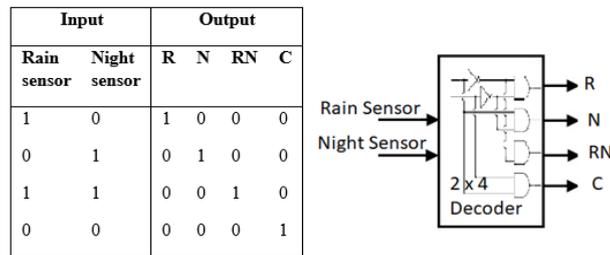


Fig. 2: Sensor Decoder Implemented in the S-Alex Net (A) Truth Table for the Decoder (B) the Logic Circuit of the Decoder.

The dataset contains 925 rain images, 1211-night images, 969 rain night images and 1321 clear images. For standardizing the experiment, 900 images are used in each weather type. A set of 450 images are positive samples and 450 images are negative samples. In addition to a hybrid dataset with random images taken from the set of all images of different weather types. The images size is 64 x 64 pixels, the color model is RGB. Some images are not in that standard, they are standardized by resizing them to the required size. Four folders are used, each folder for each weather condition. In each folder there are two sub folder, one folder for positive images and the other folder for negative images a total of 5 x 900=4500 images.

Sample of the images used in our dataset is shown in Figure 3. The figure shows samples of negative images (images with seatbelt) and positive images (images without seatbelt).



Fig. 3: Dataset Images, Sample of Images in Different Weather Conditions. From (A) to (D) Are Samples of Images with Seatbelt (Negative Images). from (E) to (H) are Samples of Images without the Seatbelt.

Each set of images in each weather type is divided into 630 images for training, 135 images for validation and 135 images for testing. If a model is trained with specific dataset, it will expect to classify perfectly if its performance is tested on that specific dataset. The model is said to be overfitting. To overcome this problem, the model is trained on a dataset and tested on a different dataset. The overfit is tested during training by using a validation dataset. Having 3 datasets, one for training, the other for validation and the last for testing ensure that the model will be generalized model and hence the problem of overfitting will be avoided.

4.2. Implementing the neural network

In the training process, the data is passed to the model as (64, 3, 64, 64) batches. i.e. 64 images with 3 RGB channels and size of 64 x 64. [24].

For the experiment, AlexNet is adopted. It contains 5 convolutional layers and 5 fully connected classification layers. The classification layers are organized as follow; one input layer, 3 hidden layers and one output layer. The input layer contains $64*64*3=12288$ nodes. Number 12288 represents the number of pixels in a single image which is 64 pixels width by 64 pixels height and 3 RGB channels. The hidden layer nodes are 512 nodes, 256 nodes and 128 nodes for the first, second and third hidden layers respectively. The hidden layers are activated by using ReLU activation function with dropout probability of 0.5. The output layer contains two nodes, one node represents negative images (with seatbelt) and the other node represents positive images (without seatbelt). The output is activated with softmax function. The bias is also included as a separate node in each layer.

4.3. Training the network

Before the training process, the input images are augmented randomly to introduce some uncertainty in the input data. The input images are rotated randomly by 30 degrees, mirrored, scaled and cropped. Augmentation is important to get more generalized model. The model can be generalized by seeing the same images in different sizes, locations and orientation. The input data is also normalized, normalization is done by subtracting each image channel values from the mean and dividing them by standard deviation, formally as

$$v' = \frac{v-\mu}{\sigma} \quad (1)$$

Where v' is the newly normalized value, v is the current image value, μ is the mean and σ is the standard deviation. Normalization forces values to be between 1 and -1. Normalization is very important in the backpropagation process; the network may fail in training without normalization. The weights and bias for all network layers are initialized randomly. A forward pass is done to calculate the next layer's output. The input vector is multiplied by the weights of the first layer and summed up. Then the result is passed to the ReLU function which passes its output to the next layer. The forward pass is continued the same way through all next layers. The ReLU function is defined as

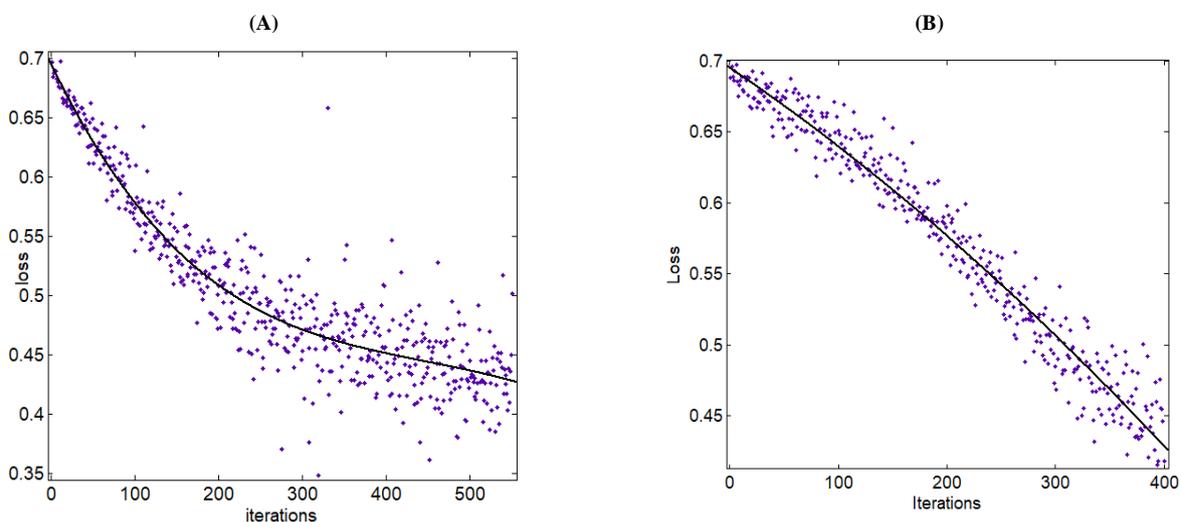
$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (2)$$

The output softmax function is used for finding the probability of each class. For example, if we pass an input image without seatbelt (positive), softmax will give the output nodes two probability values. One value for each class. It is expected to have higher probability for the positive image in its corresponding output node. If x is an input vector of numbers, then the i^{th} element value in softmax is calculated as

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (3)$$

The sum of the denominator is 1 and the component values are turned all to positive values by using the exponential. Adam optimizer [25] is used in the gradient descent optimization. Adam optimizer uses momentum in its calculations and hence it trains faster than other optimizers. Momentum adds accuracy to the learning process and reduces the number of iterations needed for the training process to converge [26], Figure 4 shows the training process for R, N, RN, and C models.

The output of the first forward pass is compared to the expected output. The weights are then recalculated by using backpropagation pass.



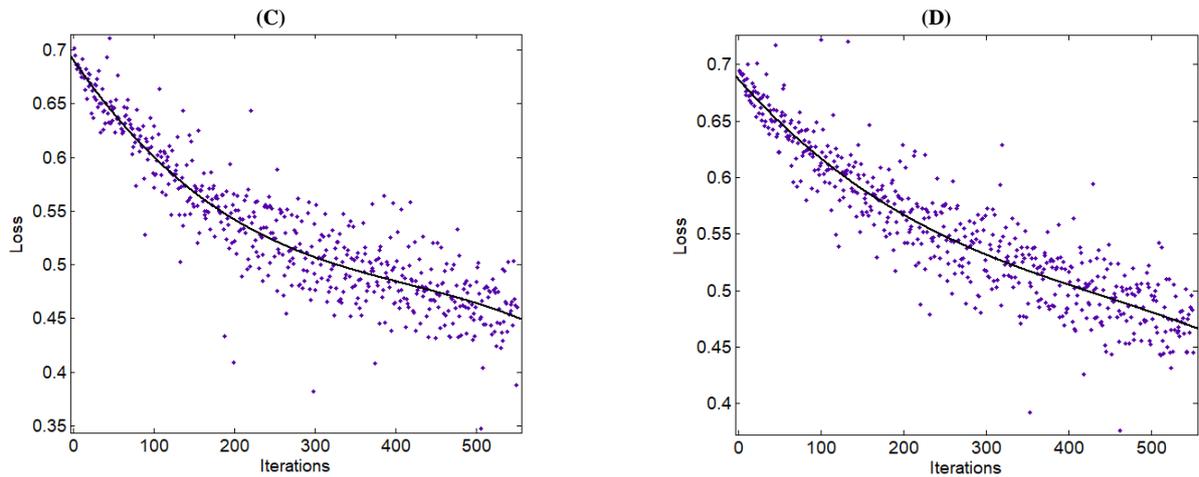


Fig 4. The Training Loss of the Four Neural Network Models, (A) Rain R Model (B) Night N Model (C) Rain Night RN Model (D) Clear C Weather Model.

The comparison between the expected output and the actual output of the neural network is done by using the classification loss. The classification loss is calculated by the following formula

$$\text{Loss}(y, y') = - \sum y \log y' \tag{4}$$

Where y is the actual output and y' is the predicted output. Which represents the cross entropy between the actual output of the neural network and the predicted output.

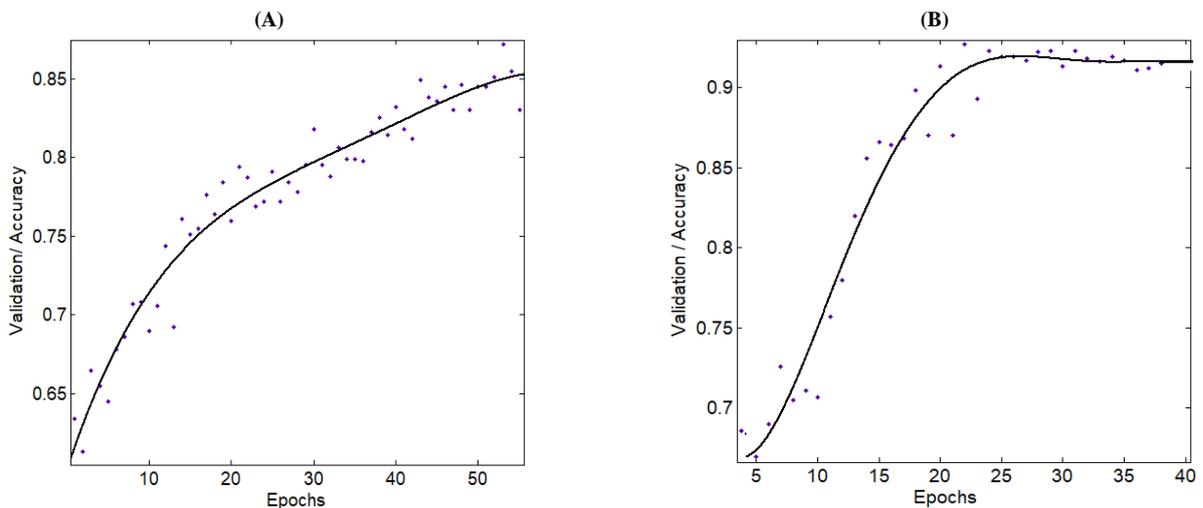
The partial derivatives are calculated with respect to network weights. The change in weights is calculated and the new weights are calculated accordingly. A learning rate of 0.006 is considered. The learning process is continued until a low loss is encountered.

Figure 4 shows the training loss of R, N, RN and C models. The loss is sharply degraded when learning N model as shown in Figure 4 (b). It is also shown in Figure 4 (a, c) that R and RN is approximately similar in their performance. However, adding night to R in RN model degrades the model. That is apparent in R curve which reduces loss sharper than RN model. The clear model is shown in Figure 4 (d).

4.4. Validating the network

The accuracy of the trained network is calculated by the following method. Suppose that we have an arbitrary 6 images. Their labels are [1, 0, 1, 0, 0, 1], 0 represents seatbelt image and 1 represents no seatbelt image. The images are passed one by one to the trained model. The outputs of the model are six pairs of probabilities. Single pair of probabilities for each image. The probabilities are (0.2, 0.8), (0.9, 0.1), (0.7, 0.3), (0.3,0.7), (0.9, 0.1), and (0.1, 0.9). The max function is applied on each pair, if the first probability is the max, 0 is returned. If the second probability is the max, 1 is returned. For example, max (0.2, 0.8) returns 1 that's because the second probability is greater than the first probability. After applying max function, the six pairs of probabilities would result in [1, 0, 0, 1, 0, 1].

The labels [1, 0, 1, 0, 0, 1] are called the actual labels. The labels [1, 0, 0, 1, 0, 1] are called the predicted labels. The two sets of labels are compared to check their equality. Then the accuracy is the mean of the result of the equality comparison.



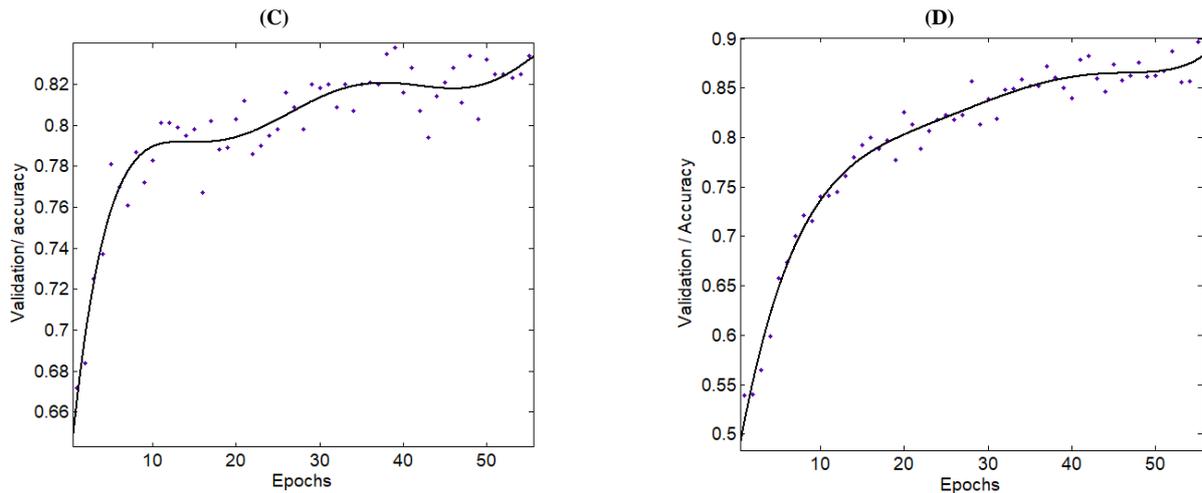


Fig. 5 The Accuracy of the Four Neural Network Models, (A) Rain R Model (B) Night N Model (C) Rain Night RN Model (D) Clear C Weather Model.

The accuracy of R, N, RN and C models is shown in Figure 5. From the figure it is shown that the best performance is introduced by N model. Figure 5 (b) shows that the accuracy curve reaches more than 0.90 in epoch 20. After epoch 20 the model keeps its performance at high level. That’s because the violation detection camera works well with spotlights at night compared to other weather conditions. C model accuracy reaches more 0.97, C model is shown in Figure 5 (d). The accuracy of the N model is better than C, R and RN. R and RN models are shown in Figure 5 (a, c). It is also apparent that R model has been learned slowly compared to other models. The learning process is very slow due to the variations found in the images due to rain weather.

The worst performance is introduced by RN model. The accuracy curve shows fluctuations in the learning process after epoch 10. That supports our hypothesis, “Having single model for each weather condition is better than having general model that supports all weather conditions”.

4.5. Predicting with network models

Five AlexNet models are trained. R, N, RN, C and Hybrid. The models R, N, RN and C are trained on Rain, Night, Rain and Night, and Clear datasets respectively. The Hybrid model is trained on the entire dataset that contains samples from R, N, RN and C images.

There are two types of errors in the prediction process, False Positive (FP) and False Negative (FN). False positive occurs when the model detects a driver not wearing seatbelt when he is actually wearing it. False negative occurs when the camera detects a driver wearing a seatbelt when he is actually not wearing it. It is obvious that FP error is more problematic than FN. It is unfair to calculate violation fee for an innocent driver that is committed to the traffic rules and wears the seatbelt. However, it is not a big issue if the system can’t detect violation (False negative). The system must be accurate in its violation detection.

True positive rate (TPR) or recall is calculated

$$TPR = \frac{TP}{TP+FN} \tag{5}$$

Where TP refers to True Positive, or a driver is violating, and the system detects that violation. False positive rate or precisions is calculated by

$$FPR = \frac{FP}{FP+TN} \tag{6}$$

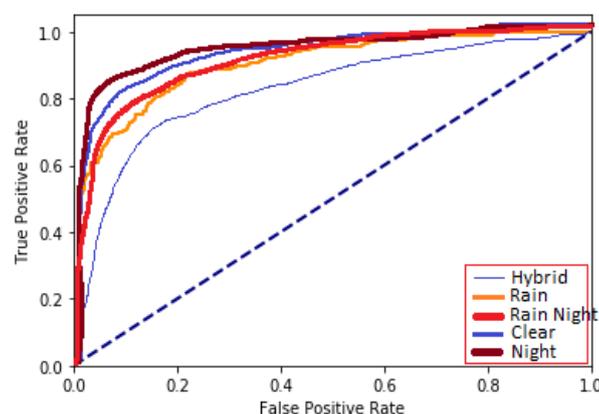


Fig 6. ROC Curve for Neural Network Models Trained on Different Weather Condition. Best Model Is the Night Model and Worst Model Is the Hybrid Model.

Where TN is the True Negative or the driver is wearing the seat belt and system detects the same. Plotting TPR versus FPR results in the receiver operating characteristic (ROC) curve as shown in Figure 6. From the figure, we see that Night model is the model with the highest accuracy. The next model with an accuracy compared to Night model is the Clear model. The lowest model in accuracy is the Hybrid

model. The Hybrid model is an AlexNet model trained with a collection of images from Night, Rain, Rain Night and Clear weather dataset. The Hybrid model is a general model that works in any weather. ROC results support the hypothesis that “Using single model for each weather condition is better than using one model for all weather conditions”.

S-AlexNet system contains four trained AlexNet models, one for each weather condition. It is impractical to train the model every time when it is needed in classification process. After training and validating the models, they are saved to the hard drive. The model file is mainly a collection of matrices for weights and biases for each layer of the trained model network. Two sensors are connected to the experiment PC to detect rain and night weather conditions. When specific weather is detected, for example rain condition is detected, the corresponding rain model file is loaded and used in the prediction process.

The accuracy of S-AlexNet is measured by the help of true negative rate (TNR) or the precision of true samples, and the true rate (TR) or the precision of all samples. TNR and TR are defined as

$$TNR = \frac{TN}{FP+TN} \quad (7)$$

$$TR = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

In Table 1 a comparison between Standard AlexNet [22], VGGNet-16 [27], BN-AlexNet [21] and the proposed S-AlexNet. The training is done on the hybrid dataset (which is collection of images from different weather conditions). The proposed S-AlexNet is trained on the same hybrid dataset but separated into different weather conditions. The internal AlexNets inside S-AlexNets are trained separately on the corresponding weather images. The proposed S-AlexNet approach showed accuracy over 90% which is higher compared to other approaches.

Table 1: Comparison of the Proposed S-Alexnet and Other Related Convolution Neural Network Structures

Network Name	TPR	TNR	TR
Standard AlexNet	83%	82%	82%
VGGNet-16	90%	83%	89%
BN-AlexNet	89%	85%	87%
S-AlexNet	92%	90%	91%

5. Conclusion

Neural networks are very powerful in classification. Convolutional Neural networks are used in seatbelt violation detection. The objective of the paper is to use single model for detecting seatbelt in each specific weather condition. Two sensors are used to detect rain and night weather conditions. Two sensors can detect a combination of 4 weather conditions. Four NN models are trained on the images from the four weather conditions. Results showed that models trained on specific weather condition is more accurate than a model trained with images from more than one weather condition. The proposed hypothesis that “Using single model for each weather condition is better than using one model for all weather conditions” is strongly accepted.

Acknowledgement

Special thanks to Dr. Moustafa El Hosseini for his valuable review comments. Also appreciating introduced to Dr. Hani Abdelrhman, Dr. Essam Othman for valuable comments and reviews.

References

- [1] Guo, H., Lin, H., Zhang, S., & Li, S. (2011, July). Image-based seat belt detection. In Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference on (pp. 161-164). IEEE. <https://doi.org/10.1109/ICVES.2011.5983807>.
- [2] G.Y. Song, K.Y. Lee, J.W. Lee, Vehicle detection by edge-based candidate generation and appearance-based classification, in: Intelligent Vehicles Symposium, IEEE, 2008, pp. 428–433. <https://doi.org/10.1109/IVS.2008.4621139>.
- [3] B. Sun, S. Li, moving cast shadow detection of vehicle using combined color models, in: Pattern Recognition, IEEE, 2010, pp. 1–5 <https://doi.org/10.1109/CCPR.2010.5659321>.
- [4] BOERNSTEIN W. S. (1955). Classification of the human senses. The Yale journal of biology and medicine, 28(3-4), 208–215.
- [5] Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., & Muller, U. (2017). Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911.
- [6] What is backpropagation really doing? | Deep learning, chapter 3, 3BLUE1BROWN SERIES S3 E3 (2019) <https://www.youtube.com/watch?v=Ilg3gGewQ5U>.
- [7] Rio-Alvarez, A., de Andres-Suarez, J., Gonzalez-Rodriguez, M., Fernandez-Lanvin, D., & López Pérez, B. (2019). Effects of Challenging Weather and Illumination on Learning-Based License Plate Detection in Noncontrolled Environments. Scientific Programming, 2019. <https://doi.org/10.1155/2019/6897345>.
- [8] El-Melegy, M. T., El-Magd, K. M. A., Ali, S. A., Hussain, K. F., & Mahdy, Y. B. (2019, February). Ensemble of Multiple Classifiers for Automatic Multimodal Brain Tumor Segmentation. In 2019 International Conference on Innovative Trends in Computer Engineering (ITCE) (pp. 58-63). IEEE. <https://doi.org/10.1109/ITCE.2019.8646431>.
- [9] Sun, W., Du, H., Nie, S., & He, X. (2019). Traffic Sign Recognition Method Integrating Multi-Layer Features and Kernel Extreme Learning Machine Classifier. <https://doi.org/10.32604/cmc.2019.03581>.
- [10] Ershadi, N. Y., Menéndez, J. M., & Jimenez, D. (2018). Robust vehicle detection in different weather conditions: Using MIPM. PloS one, 13(3), e0191355. <https://doi.org/10.1371/journal.pone.0191355>.
- [11] Pavel Surmenok, “ResNet for Traffic Sign Classification with PyTorch”. Towards data science, Nov 2018.
- [12] A. Khammari, F. Nashashibi, Y. Abramson, C. Laugeau, Vehicle detection combining gradient analysis and Adaboost classification, in: Intelligent Transportation Systems Conference, IEEE, 2005, pp. 66–71
- [13] Li, W., Lu, J., Li, Y., Zhang, Y., Wang, J., & Li, H. (2013, December). Seatbelt detection based on cascade adaboost classifier. In 6th IEEE International Congress on Image and Signal Processing (CISP2013), (Vol. 2, pp. 783-787) <https://doi.org/10.1109/CISP.2013.6745271>.

- [14] Xu, J., & Song, K. (2015, October). Study on Automatic Detection Method of Automobile Safety Belt Based on the Improvement of Adaboost Algorithm. In 2015 3rd International Conference on Mechatronics and Industrial Informatics (ICMII 2015). Atlantis Press. <https://doi.org/10.2991/icmii-15.2015.185>.
- [15] J. Arróspe, L. Salgado, M. Nieto, "Video analysis based vehicle detection and tracking using an MCMC sampling framework", EURASIP Journal on Advances in Signal Processing, vol. 2012, Article ID 2012:2, Jan. 2012 <https://doi.org/10.1186/1687-6180-2012-2>.
- [16] H. Grabner, C. Beleznai, H. Bischof, Improving Adaboost detection rate by wob-ble and mean shift, in: Proceedings of Computer Vision Winter Workshop, vol. 5, 2010, pp. 23–32.
- [17] Y. Gao, F. Gao, Edited Adaboost by weighted kNN, Neurocomputing 73 (16–18) (2010) 3079–3088. <https://doi.org/10.1016/j.neucom.2010.06.024>.
- [18] Yu D, Zheng H, Liu C. (2013). Driver's Seat Belt Detection in Crossroad Based on Gradient Orientation. In Information Science and Cloud Computing Companion (ISCC-C), International Conference on 2013 Dec 7 (pp. 618-622). IEEE. <https://doi.org/10.1109/ISCC-C.2013.65>.
- [19] Yusuf Artan et al, "Passenger Compartment Violation Detection in HOV/HOT Lanes", IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 17, NO. 2, pp: 395: 405 FEBRUARY 2016 <https://doi.org/10.1109/TITS.2015.2475721>.
- [20] Zhang, D. (2018). Analysis and research on the images of drivers and passengers wearing seat belt in traffic inspection. Cluster Computing, 1-7. <https://doi.org/10.1007/s10586-018-2070-x>.
- [21] Zhou, B., Chen, D., & Wang, X. (2017, August). Seat Belt Detection Using Convolutional Neural Network BN-AlexNet. In International Conference on Intelligent Computing (pp. 384-395). Springer, Cham. https://doi.org/10.1007/978-3-319-63309-1_36.
- [22] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
- [23] Yanxiang Chen, Gang Tao, Hongmei Ren, Xinyu Lin, Luming Zhang, Accurate seat belt detection in road surveillance images on CNN and SVM, Journal Neurocomputing Volume 274 Issue C, pp. 80-87, 2018 <https://doi.org/10.1016/j.neucom.2016.06.098>.
- [24] Mat Leonard, Code for the Deep Learning with PyTorch lesson, https://github.com/udacity/DL_PyTorch GitHub, May 2018
- [25] Kingma, D. P., & Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, 1–13
- [26] Goh, "Why Momentum Really Works", Distill, 2017. <https://doi.org/10.23915/distill.00006>.
- [27] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).