Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2022 SCPE. Volume 23, Issues 4, pp. 441–455, DOI 10.12694/scpe.v23i4.2027

IMAGE-BASED SEAT BELT FASTNESS DETECTION USING DEEP LEARNING

RUPAL A. KAPDI, PIMAL KHANPARA, ROHAN MODI, AND MANISH GUPTA *

Abstract. The detection of seat belts is an essential aspect of vehicle safety. It is crucial in providing protection in the event of an accident. Seat belt detection devices are installed into many automobiles, although they may be easily manipulated or disregarded. As a result, the existing approaches and algorithms for seat belt detection are insufficient. Using various external methods and algorithms, it is required to determine if the seat belt is fastened or not. This paper proposes an approach to identify seat belt fastness using the concepts of image processing and deep learning. Our proposed approach can be deployed in any organizational setup to aid the concerned authorities in identifying whether or not the drivers of the vehicles passing through the entrance have buckled their seat belts up. If a seat belt is not detected in a vehicle, the number plate recognition module records the vehicle number. The concerned authorities might use this record to take further necessary actions. This way, the organization authorities can keep track of all the vehicles entering the premises and ensure that all drivers/shotgun seat passengers are wearing seat belts.

Key words: Image Processing, Deep Learning, Number Plate Recognition, Seat Belt Detection

AMS subject classifications. 68T05

1. Introduction. When driving, it is critical to adhere to all safety regulations. It not only helps to prevent accidents but also protects drivers in the event of a crash or other adverse incidents. Seat belt detection systems are already installed in the majority of vehicles. However, many drivers just ignore the importance of using them. Many professional organizations make it a point to observe vehicle and traffic safety guidelines to be followed by their employees on their premises. Fastening seat belts is one of the basic and very important traffic and safety rules [17]. Many universities or other corporate organizations make efforts to ensure that everyone who enters their campus/office premises is aware of the fundamental traffic and vehicle safety regulations and also follows the same. However, manually keeping track of all entering vehicles and mandating drivers to wear seat belts is an extremely time-consuming and tedious task that is practically infeasible to execute in large organizations with a large number of employees coming to work in their own vehicles. Therefore, there is a need to have an automated system to keep track of entering vehicles to ensure compliance with the required safety protocols. This kind of system has obvious advantages of reducing manual intervention, improved accuracy, and overall faster execution. In this paper, an automated system to detect the fastened seat belts and number plates of the vehicles is proposed for a university campus. The proposed model is built based on the concepts of image processing and deep learning to achieve a high level of accuracy and performance.

2. Literature Review.

2.1. Existing Solutions for Number/License Plate Detection. Many techniques have been proposed by researchers for identifying the number plates/license plates of vehicles. The identification of the owner of the vehicle via number/license plate is crucial in implementing traffic safety rules either by government authorities or other private/public organizations [20]. The major problem in detecting the registered vehicle number from its number/license plate is the format and language in which the required details are printed. Not every country follows the same format/pattern and/or language for specifying vehicle registration details on the vehicle number plates. Therefore, though many solutions have been proposed in this domain, it is difficult to find a solution that can address all the above-mentioned issues related to extracting vehicle registration details. In the Indian context, it is required to devise a multi-language number plate detection technique as different states allow the

*Computer Science and Engineering Department, Institute of Technology, Nirma University, India (pimal.khanpara@nirmauni.ac.in). This work was supported by Nirma University.

Table 2.1: Summary of existing Number/License Plate Detection Approaches

Ref.	Salient Features	Accuracy	Dataset	Advantages	Limitations
[9]	Detection using image processing and Gaussian filter	Extraction - 84.28% Segmentation - 77.14% Recognition - 71.43%	70 images of Iranian vehicles in different weather conditions	Works well with low resolution noisy and low contrast images	Missing rate is 16% from 70 vehicles
[8]	YOLOv2 sensor with ResNet attribute extractor heart for NP detection; Uses YOLOv2 with ResNet50 model	Indian- 88.33% Test images 4789 (max for India)	DeebNP D2020	Can detect Multi-Language license plates	-
[2]	Multiple Pre-processing stages; License plate region extraction; characters segmentation and recognition; Uses Sobel edge detector, Median filter for noise removal	-	White images with black background with high contrast	Pixel by pixel Segmentation	Blurred images, broken number plates, similarity between some characters like O-D, 5-S, 8-B-E, 0-O
[10]	Input is taken as a video feed, split, segmented and CNN/KNN used in OCR	Accuracy (CNN): 93%	40512 images, (75/25) Test:4800 images + 50 short videos	Segmentation free because of use of RNN	Problem in KNN based prediction due to cropped images
[1]	Epochs 130k, learning rate 0.01; Uses YOLO model	Number plate detection: 100% Recognition: 91% Sorting Output: 100%	Manually collected 6500 images, tested on 640 images (90/10) split	Data augmentation to increase the number of characters extracted from less images training set	Difficulty in detecting similar looking characters such O and 0

usage of different languages for license plates. Moreover, based on our analysis of the existing approaches, it is found that the Convolutional Neural Networks would be the best choice for implementing the detection module for our proposed system. Table 2.1 describes the analysis of various existing techniques of number/license plate detection along with their advantages and limitations.

2.2. Existing Solutions for Seat Belt Recongnition. Many solutions have been proposed by researchers for detecting seat belts. Most of these solutions consider the problem as an object detection task and hence, they suggest YOLO (You Only Look Once) [6] algorithm for implementing the solution [18]. However, in our proposed system, the problem of detecting seat belts is considered an image classification task [5]. The main reason for not using the YOLO algorithm is that it requires a dataset in form of bounding boxes and the same is not available for the given problem domain [14]. Table 2.2 presents the study and analysis of various existing solutions along with their key features and scope of deployment.

As summarized in Table 2.2, most of the existing approaches for seat belt detection either do not provide details about the dataset used for training and testing their models or generate and use their own custom dataset.

3. Overview of the Proposed System. Our proposed system captures the image of the driver sitting in the car via the windshield using the pre-installed camera. The captured image is then processed and provided to a trained deep learning-based model to predict whether or not the driver is wearing a seat belt. In the event that the driver has not secured the seat belt, the system captures the vehicle's number plate. After processing the captured number plate image, the vehicle number is extracted and recorded. This information is then forwarded to the concerned authorities to take necessary measures. The proposed system's working and performance are tested for a university campus in India but the same system can be deployed in any organizational environment by making minimal changes to incorporate the necessary regulations.

The proposed system is divided into two modules – Seat Belt Detection Module and Number Plate Recognition Module. The first step in our proposed system is to capture videos of cars entering the university campus,

Table 2.2: Analysis of the existing Seat Belt Detection Approaches

Reference	Objective(s)	Proposed Methodology	Dataset Used
[7]	Detection of seat belts using monitoring images consisting of full scene information of the moving car	Extracts driver area using vehicle outline; Detects seat belt edges in the HSV color space; Further processing of edges to obtain the final outcome	100 real-world images captured under various conditions
[12]	Classification of driver seat belt status using a camera inside the driver's cabin	Uses a YOLO Neural Network-based model; Classifies outcomes as seat belt fastened correctly, not fastened, and fastened behind the back	Videos of drivers in multiple vehicles over three months in the wintertime in addition to images of unknown drivers taken from the Internet
[15]	To detect whether a driver wears a seat belt or not using a convolutional neural network	Uses a 6-hidden layered convolutional neural network model to classify the input images	A standard dataset containing the 2155 images taken from Yawning detection dataset
[22]	To detect seat belts automatically using salient gradient	Extracts edge features using edge detection; Building the salient gradient feature map and forwarding it to a machine learning model for prediction	Not mentioned
[21]	Seat belt detection using Convolutional Neural Network (CNN)	Use of improved CNN called BN-AlexNet to enhance the classification ability; Analysis of the confidence detection results	Custom dataset
[11]	To build a fully automated seat belt detection system	Use of sensors for weather condition detection to have better accuracy; Use of a single specialized model for each weather condition; Deploys Sensor-based AlexNet for classification;	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
[3]	To propose a seat belt detection algorithm for complex road backgrounds based on multi-scale feature extraction using deep learning	Extraction of multi-scale features; Use of CNN and SVM; Use of detection scores to improve accuracy	Not mentioned

using the cameras installed at the main gate of the university. The captured videos are then processed and the required images are extracted to prepare the dataset. Once the image dataset is prepared, data augmentation is performed and the model is trained. In the last step, the model is tested and necessary improvements are made.

Based on the outcomes and understanding of the literature review process, the best approach for implementing the required solution is determined. A review of existing systems also helps to understand the advantages and limitations of the available solutions and build an improved system to overcome the limitations of existing solutions. The proposed approach and implementation steps are explained in detail in the subsequent sections. The derived results and conclusions are also presented at the end of this paper.

4. Design and Deployment of the Proposed System.

4.1. Design and Deployment of Seat Belt Detection Module. Researchers have used different approaches to propose solutions for seat belt detection. However, the design of such solutions highly depends upon the deployment environment and expected end outcomes. As an outcome of our research and literature review process, it is established that CNN-based models can be used to identify whether the driver is buckled up. If the images of vehicles entering the university premises are available, they can be used to train the CNN model. Hence, generating a dataset of images from the video recordings is the first step in the processing as the deployment environment considered for the proposed system does not have images readily available and therefore, it is necessary to extract the images from the video recordings of the main entrance of the university. The dataset generation process also includes resizing and re-scaling of the images that are divided into two groups: with and without seat belts.

A data augmentation process is to next step where the training dataset is populated with additional images generated using various augmentation techniques. Moving further, a transfer learning-based approach [16] is

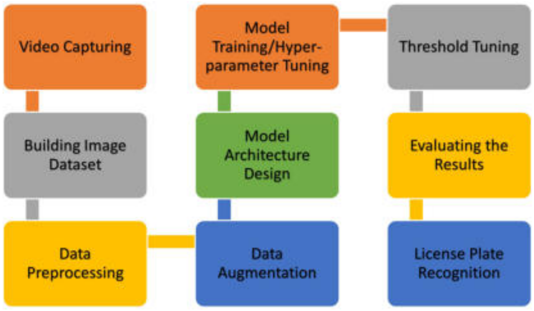


Fig. 4.1: Design Flow of the Proposed System

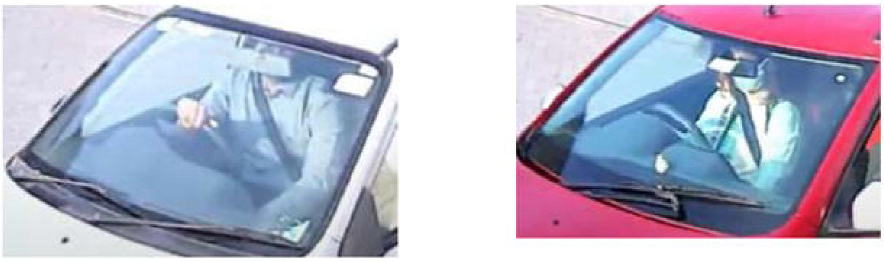


Fig. 4.2: Sample input images with a minimum requirement of pre-processing

employed to generate a CNN-based model. The model uses MobileNetV2 [4] as the base model with a few additional layers for precise classification. Following that, sample testset images are classified and results are generated. Figure 4.1 depicts the functional blocks and the flow of execution for the proposed system. The same is described in detail in the following subsections.

4.1.1. Dataset Generation. As there is no proper dataset available for seat belt detection, it is required to generate one as the first step of the proposed solution. The deployment environment considered in this work is a university organization and the cameras are installed at the entry gates to capture the vehicles entering the university premises. Therefore, it is required to extract images from the captured videos for the preparation of the dataset. There are two categories in which the images are stored: with seat belts and without seat belts. In our experiment, a total of 263 images were stored in the dataset, out of which 233 were included in the "with seat belt" category and the remaining 32 images were included in the "without seat belt" category. However, the angles of the installed cameras and quality of the video recordings lack adequacy for generating the dataset which leads to the further processing of the images.

4.1.2. Data pre-processing. In the data pre-processing phase, the images are processed in such a way that they become well-suited for training the model. There can be different types of images in the dataset and each type requires a different level of pre-processing to resolve specific types of problems. For example, the images shown in Figure 4.2 are easy to use and do not require much processing.

On the other hand, some images are either too bright or too dark and that makes the identification of a seat belt difficult. Moreover, sometimes, the color of the seat belt is also a problem if it blends with the color of the clothes the driver is wearing, making it really difficult to detect. Such erroneous sample images having these problems are shown in Figure 4.3 for which our proposed system could not detect the seat belt.

Another problem with the input images is the presence of some obstacles that block the view of the seat belts. Such obstacles can be any objects such as hands, windshields, etc. Such objects obviously make the detection of seat belts very difficult. An example of such an input image is shown in Figure 4.4.

Fig. 4.3: Sample input images with a maximum requirement of pre-processing

Fig. 4.4: Sample input image with Seat belt view blocked

Before applying any scaling or processing method to the input images, it is required to take care of the above-mentioned practical issues. Many of the above issues can be resolved if the brightness or contrast levels are modified in the images. In our experiments, the following approaches are used for the basic pre-processing of all the images included in the dataset:

- Cropping the image so only the front view of the driver is visible
- Resizing the image to 128 x 128
- Blurring the image for smoothness. Here Gaussian blur is used with the kernel size (1,1)
- Applying the built-in pre-process_input() function from the keras.applications.mobilenet_v2 package. This function directly pre-processes the image according to what MobileNetV2 expects the input to be. It scales input pixels between -1 and 1 and other basic transformations

Following that, the whole dataset is split into a 90% training set and a 10% validation set. Class labels are also kept stratified into both training and validation sets. The model is trained on the training set after applying data augmentation and then validated on the validation set after each epoch.

4.1.3. Data Augmentation. Data augmentation is important to perform when there is a limited number of images available for the data set. The proposed system uses a CNN-based model and it has quite a large number of parameters. Data augmentation is necessary to prevent over-fitting. Another advantage of data augmentation is that it makes the trained model more robust for classifying unseen images.

In the Keras library [13], the ImageDataGenerator class is used for implementing various augmentation transformations. The following transformations are applied with their respective parameters:

- Random rotation (range = 20)
- Random zoom (range = 0.15)
- Random width shift (range = 0.2)
- Random height shift (range = 0.2)
- Random brightness (range = 0.8 to 1.2)

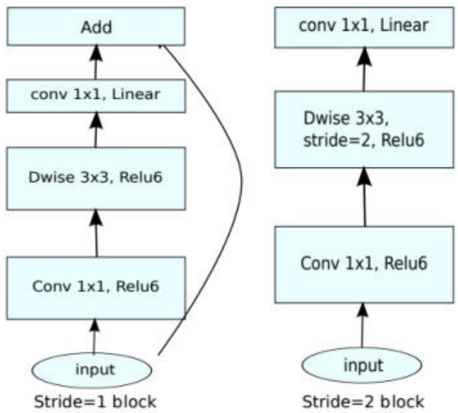


Fig. 4.5: MobileNetV2 convolutional block [19].

Table 4.1: Comparison of base models

Base model	**Total parameters**	**Training accuracy**
Xception	21,386,538	51.85%
VGG16	14,846,530	88.89%
ResNet50	24,112,770	88.89%
InceptionV3	22,327,842	74.37%
MobileNetV2	2,586,434	88.89%

- Horizontal flip

4.1.4. Model Architecture Design. It is required to have a model which performs binary image classification on the dataset. CNN is proven to be best suited for problems related to image classification. Therefore, the proposed system uses a CNN-based model. Since there are many reference models available for image classification problems, they can be directly utilized. Most of these base models are trained on the ImageNet dataset hence they can be utilized for performing preliminary image localization and feature extraction. In addition, the concept of transfer learning is applied in the proposed work. The traditional supervised learning paradigm breaks down in the absence of sufficient labeled data for the domain in which there is a need to train a reliable model. This may lead to a deterioration in performance. Hence, transfer learning helps to deal with this problem by leveraging already existing labeled data of some related domain. Therefore, instead of training a neural network from scratch, the state-of-the-art lightweight MobileNetV2 [19] is used as the base model. Further, instead of training weights, ImageNet weights are directly used. This base model acts as a generic feature detector. The architectural details of MobileNetV2 are as follows:

- There are two types of 3 layers convolutional blocks. One is a residual block with stride 1 and the other is a block with stride 2 used for down-sampling.
- It outperforms both MobileNetV1 and ShuffleNet which have almost similar model size and computational cost.

In the proposed system, as depicted in Figure 4.5, MobileNetV2 is selected as the best choice for the base model as per its comparison with other state-of-the-art CNNs for the given problem of seat belt detection. The training accuracy of various models is shown in Table 4.1.

MobileNetV2 is comparatively faster and has fewer parameters as shown in Table 4.1. These model advantages are best suited for the task which has fewer images for training. However, in some scenarios, MobileNetV2 has lesser accuracy compared to other models but the accuracy is not a major concern in the deployment scenario

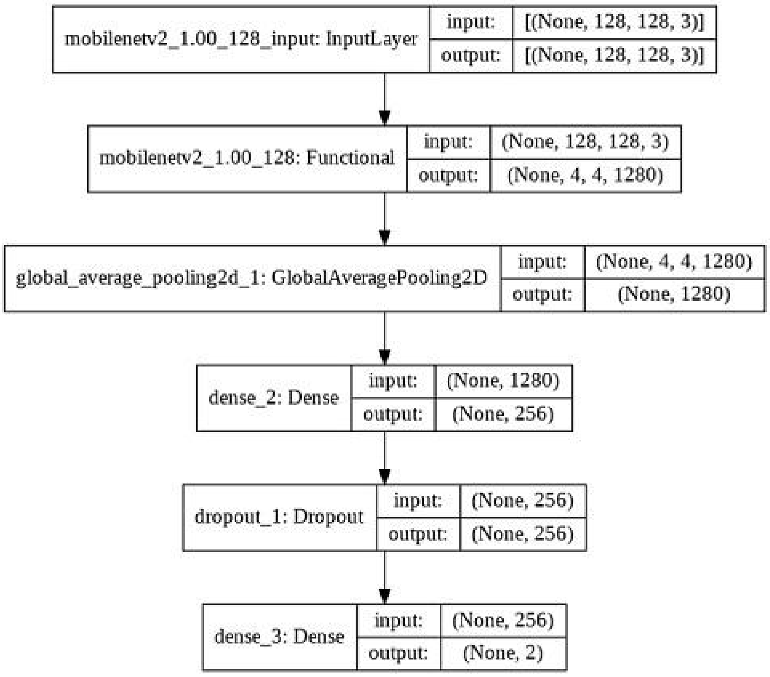


Fig. 4.6: Proposed Model Architecture

of the proposed system. During the deployment phase, some architectural changes are done in MobileNetV2, where the topmost layer is removed from the base model and a few extra layers, as described below, are added to perform the classification:

- Global average pooling layer aggressively summarizes the features from the last convolutional layer
- Fully connected layer with 256 neurons and ReLU as activation function
- Dropout layer with a dropout rate of 0.5 to prevent over-fitting
- Final classification layer with 2 neurons and softmax as the activation function
- Since it is a binary classification problem, Binary Cross Entropy is used as a loss function. Again, Adam is used as an optimizer since it performs the best in this case. Accuracy is chosen as a metric for model training. The model architecture for the proposed system is shown in Figure 4.6.

4.1.5. Model Training. For the model selected for the proposed system, there is a total of 2,586,434 parameters out of which 328,450 are the trainable parameters. The initial learning rate is fixed to 1e-3. The model is trained for 8 epochs with a batch size of 16. Here, the number of epochs is very less. The model starts over-fitting the training data as the number of epochs increases. The graph as shown in Figure 4.7 depicts this fact. The training data is passed after performing all the augmentations as described previously but the validation set is not augmented. The best combination of the parameters is determined after the model is trained multiple times.

4.2. Design and Deployment of Number Plate Recognition Module. Upon identifying a car-driver without a seat belt, the proposed system invokes the number plate recognition module. This module scans the number plate of the identified car so that the owner of the car can be identified and further disciplinary actions can be taken. The process of number plate detection consists of two phases: i) pre-processing the captured image and ii) recognizing characters from the number plate.

4.2.1. Pre-processing.

1. Edge Detection

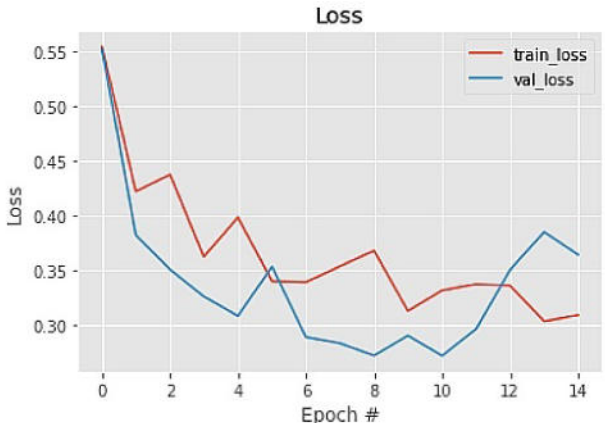


Fig. 4.7: Over-fitting of the model after certain epochs

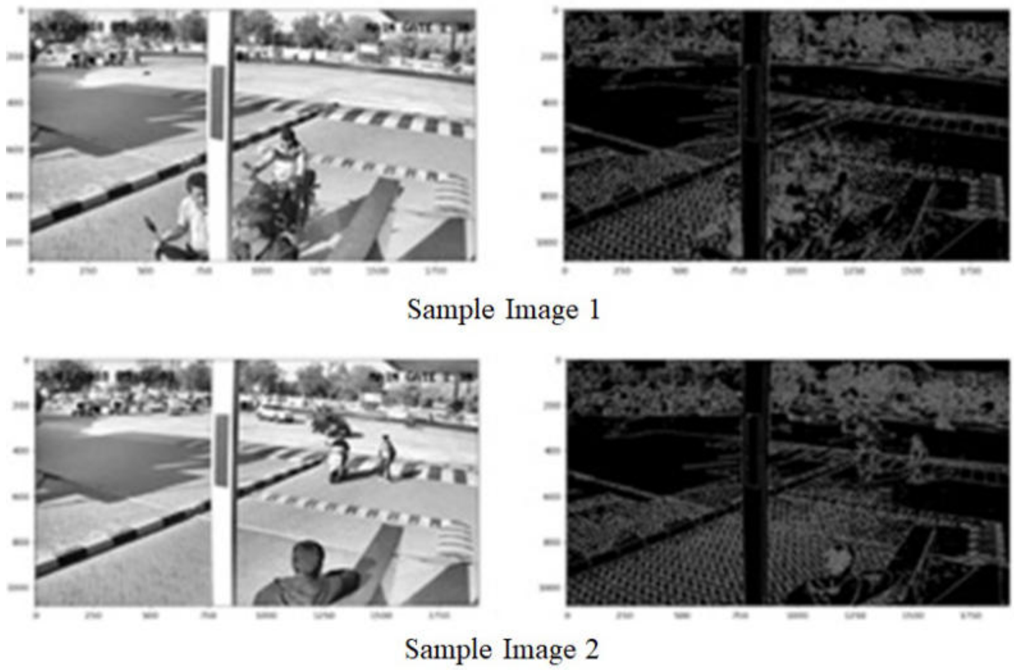
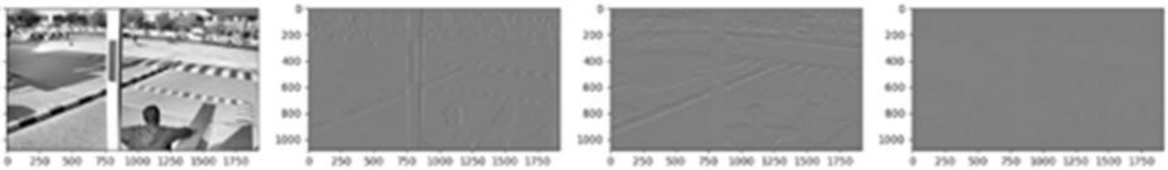


Fig. 4.8: Output of Canny Edge Detection Algorithm for Sample Images

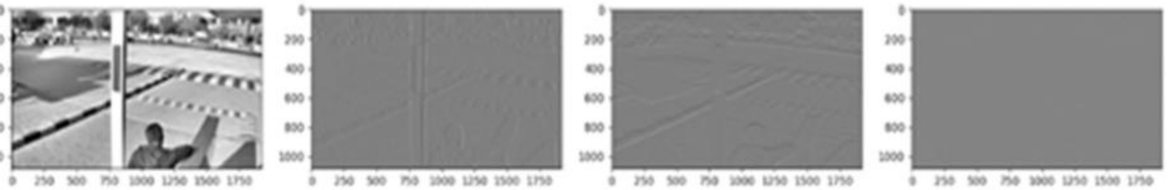
A sudden change of discontinuities forms horizontal edges, vertical edges, and diagonal edges. Most of the shape information of an image is enclosed in the edges. Hence, it is first required to detect these edges in an image using filters and enhance the image area which contains edges to improve the edge sharpness. Canny edge detector and Sobel edge detector are two widely used edge detection algorithms. The outputs of the Canny and Sobel edge detection algorithms on some of the sample images are shown in Figure 4.8 and Figure 4.9 respectively.

2. Color Space Transformations
A color space is a specific organization of colors. In combination with color profiling supported by various physical devices, it supports reproducible and device-independent color representations - whether

[htb]



(a) Sobel edge Detection using x-direction filter



(b) Sobel edge Detection using y-direction filter

Fig. 4.9: Output of Sobel Edge Detection Algorithm for Sample Images

such representation entails an analog or a digital representation. The input images are tested on three different types of space transformations, i.e. HSV, LAB, and YCrCb as shown in Figure 4.10.

4.2.2. Character Recognition from Number Plate. Number plate recognition is an important application in the area of computer vision. It is already being used in various real-life scenarios to detect and recognize license plates at junction points in urban areas. In our proposed system, the objective of the license plate detection module is to identify the vehicles through their registered license plates which are not following the vehicle driving rules set by the government/organization. The license plate recognition system can be decomposed into two major sub-components which are: i) Detect and localize a license plate in an input image/frame, ii) Extract the characters from the license plate. However, due to the complexity of the deployment scenario, the images are required to be pre-processed further before giving them as input to the proposed model. This pre-processing majorly involves the cropping of images from the scene to maximize the ROI of the license plate.

Step 1: Detect and localize a license plate in an input image

The cropped image is further processed to obtain the exact number plate. Such scanned and cropped image areas are examined to check for existence of alphabets in them. The availability of alphabets is checked with help of contours in the targeted cropped area. K-Nearest Neighbors (KNN) classifier helps to identify contours in the targeted area. The KNN works as follows:

- Prepare the training dataset.
- Find the Euclidean distance for the test dataset record with all the data records in the training dataset.
- Find the top K distance using sorting of the test dataset record with the training dataset.
- Find out the neighbors that happen maximum times.
- Classify the character of that neighbor.

KNN finds all the possible number plates in the scene and stores their dimensions in an array. Further, the dimensions are pixel-wise checked to be a number plate. Upon successful retrieval of the number plate region,

[htb]

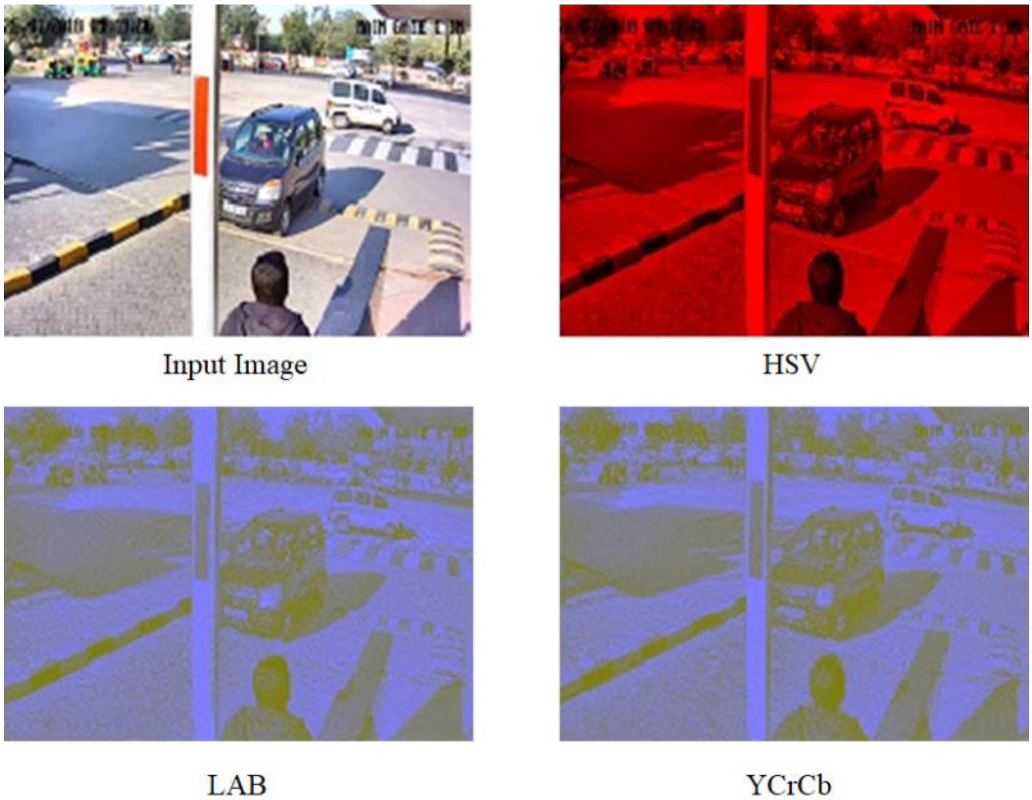


Fig. 4.10: Outcome of Color Space Transformation

it is further processed for character recognition.

Step 2: Extract the characters from the license plate

The ROI is used to extract the characters from the number plate and given to KNN classifier for further classification. The KNN classifies the characters in a similar manner as explained in step 1. The recognized characters generate the sequence of alphabets for number plate output.

5. Result Analysis.

5.1. Testing the trained model. During the training time, the model is found to achieve maximum accuracy of 89.19% and a loss of 0.3314. On the validation set also, the model performs almost similarly. It produces a validation accuracy of 88.89% and the validation loss reduces to 0.2497. The trained model is used to make predictions on an unseen set of data. This data consists of a mixed class of labels. After making predictions on this data, it is compared with the ground truth labels and the classification report is generated, as shown in Table 5.1. Figure 5.1 and Figure 5.2 present the accuracy and loss of the training and validation process after each epoch.

5.2. Thresholding for imbalanced dataset. The dataset generated and used for the proposed system is highly imbalanced. In the dataset, there is around 90% of the data that represents scenarios with seat belts and only 10% of the remaining data depicts the cases without seat belts. Hence, the dataset is said to be highly biased towards making predictions for the category, "with seat belt". This fact is clearly visible from the classification reports. It is observed that there is 0 recall and F1-score for without seatbelt class. There are many techniques to handle this issue of imbalanced classification. The simplest approach is to change the default

Table 5.1: Classification report

	Precision	**Recall**	**F1-Score**
With Seat Belt	0.89	1.00	0.94
Without Seat Belt	1.00	0.00	0.00
Accuracy	0.89	0.89	0.89
Macro Average	0.94	0.50	0.47
Weighted Average	0.90	0.89	0.84

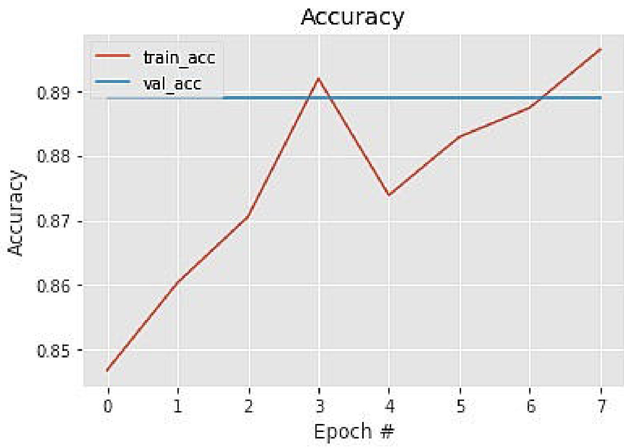


Fig. 5.1: Epoch-wise Training and Validation Accuracy

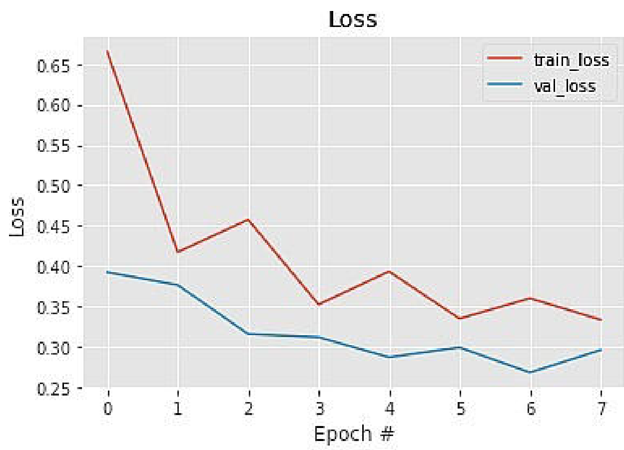


Fig. 5.2: Epoch-wise Training and Validation Loss

decision threshold. Hence, tuning or shifting the decision threshold to accommodate broader requirements of the classification problem is called thresholding.

The sigmoid function used in the experiment produces the final result between 0 and 1. The Keras library takes 0.5 as the default decision threshold and classifies the test instances accordingly. But it is not appropriate to 0.5 as the threshold and there is a need to determine an optimal threshold value. The basic approach is to

Table 5.2: Classification report based on thresholds

Threshold	**Precision**	**Recall**	**F1-Score**
0.001	0.41	0.4	0.40
0.002	0.6	0.5	0.54
0.003	0.6	0.5	0.54
0.004	0.65	0.55	0.60
0.005	0.65	0.55	0.60
0.006	0.51	0.5	0.50

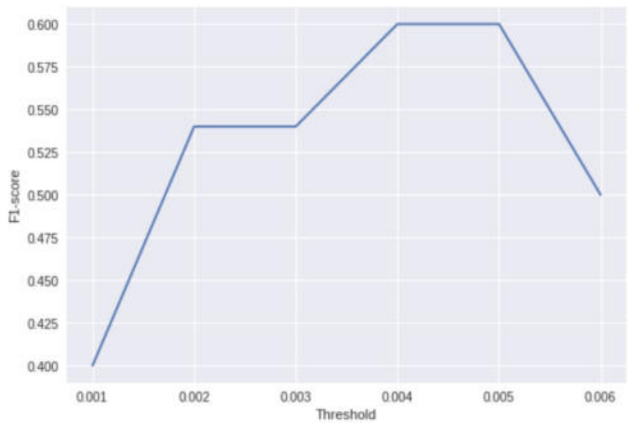


Fig. 5.3: Threshold vs. F1-score

define a set of thresholds and evaluate the predicted probabilities under each case to determine the optimal threshold. This procedure is summarized below:

- It is observed that the model achieved the f1-score of 0.84 on testing data. The same model is used but instead of directly predicting the class labels, class probabilities are predicted.
- Probabilities are required only for the minority class i.e., without a seat belt. Hence, a set of thresholds for evaluating probabilities are defined. For this experiment, thresholds considered are from 0.0 to 1.0 in step size of 0.001 i.e., we will test 0.000, 0.001, 0.002, 0.003 and so on up to 0.999.
- For each threshold, the predicted labels are evaluated using the f1-score. The one which has the largest F1-score is considered the optimal threshold. F1-scores for a few threshold values are shown in Table 5.2. The graphical analysis of the threshold and F1-scores is shown in Figure 5.3.

5.3. Outcome of classification. The example images presented in this section depict the sample output generated by the trained model. The pre-processing steps mentioned in the previous sections are applied to each input image. The model returns whether the seat belt is fastened or not along with the class probability of "with seat belt" cases. Figure 5.4 presents the sample outcome of prediction for positive class (with seat belts). Figure 5.5 shows the results of the prediction where the drivers are not wearing their seat belts.

5.4. Number Plate Recognition result analysis. The correct output for the above steps is shown in Figure 5.6(a). Some of the characters are not properly classified as shown in Figure 5.6(b). The two-step process for number plate output generation gives varied accuracy as shown in Table 5.3.

6. Conclusion. Fastening seat belts is considered one of the most important vehicle safety practices. In the proposed system, seat belt detection is done through the front view of the car by applying deep learning and image processing concepts. A model is designed that can predict whether the seat belt is fastened or not from the input image. The dataset for this model is prepared by extracting the images for the video recordings.

Fig. 5.4: Sample Outcome for Positive Class

Fig. 5.5: Sample Outcome for Negative Class

Table 5.3: Accuracy for two-step number plate output process

Steps	**# input images**	**# output images**	**Accuracy**
Step 1	263	250	95%
Step 2	250	225	90%

Input images are also pre-processed so that they can be used for training the CNN-based model using transfer learning. The results obtained during the testing phase of the proposed model are found to be satisfactory from the accuracy point of view. However, to improve the existing percentage of accuracy which is 89%, the

Fig. 5.6: Outcome of Color Space Transformation

dataset can be made balanced and have more images of higher quality. The proposed system can be used by any organization with minor modifications to ensure the safety of its stakeholders. After some improvements, the proposed model can be made more robust to handle diverse weather conditions and still offer a good level of accuracy.

Acknowledgement. The authors wish to thank Nirma University, Ahmedabad, India for providing all the necessary financial and technical support for carrying out this work.

REFERENCES

[1] R. N. Babu, V. Sowmya, and K. Soman, *Indian car number plate recognition using deep learning*, in 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), vol. 1, IEEE, 2019, pp. 1269–1272.

[2] G. N. Balaji and D. Rajesh, *Smart vehicle number plate detection system for different countries using an improved segmentation method*, vol, 3 (2017), p. 7.

[3] Y. Chen, G. Tao, H. Ren, X. Lin, and L. Zhang, *Accurate seat belt detection in road surveillance images based on cnn and svm*, Neurocomputing, 274 (2018), pp. 80–87.

[4] K. Dong, C. Zhou, Y. Ruan, and Y. Li, *Mobilenetv2 model for image classification*, in 2020 2nd International Conference on Information Technology and Computer Application (ITCA), IEEE, 2020, pp. 476–480.

[5] P. Druzhkov and V. Kustikova, *A survey of deep learning methods and software tools for image classification and object detection*, Pattern Recognition and Image Analysis, 26 (2016), pp. 9–15.

[6] J. Du, *Understanding of object detection based on cnn family and yolo*, in Journal of Physics: Conference Series, vol. 1004, IOP Publishing, 2018, p. 012029.

[7] H. Guo, H. Lin, S. Zhang, and S. Li, *Image-based seat belt detection*, in Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety, IEEE, 2011, pp. 161–164.

[8] J. Gupta, V. Saini, and K. Garg, *Multilanguage number plate detection using convolutional neural networks*, arXiv preprint arXiv:2008.08023, (2020).
[9] P. S. Ha and M. Shakeri, *License plate automatic recognition based on edge detection*, in 2016 Artificial Intelligence and Robotics (IRANOPEN), IEEE, 2016, pp. 170–174.
[10] S. N. Hashmi, K. Kumar, S. Khandelwal, D. Lochan, and S. Mittal, *Real time license plate recognition from video streams using deep learning*, International Journal of Information Retrieval Research (IJIRR), 9 (2019), pp. 65–87.
[11] O. Hosam, *Deep learning-based car seatbelt classifier resilient to weather conditions*, Int. J. Eng. Technol., 9 (2020), pp. 229–237.
[12] A. Kashevnik, A. Ali, I. Lashkov, and N. Shilov, *Seat belt fastness detection based on image analysis from vehicle in-abin camera*, in 2020 26th Conference of Open Innovations Association (FRUCT), IEEE, 2020, pp. 143–150.
[13] N. Ketkar, *Introduction to keras*, in Deep learning with Python, Springer, 2017, pp. 97–111.
[14] M. Maity, S. Banerjee, and S. S. Chaudhuri, *Faster r-cnn and yolo based vehicle detection: A survey*, in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), IEEE, 2021, pp. 1442–1447.
[15] D. B. Naik, G. S. Lakshmi, V. R. Sajja, D. Venkatesulu, and J. N. Rao, *Driver's seat belt detection using cnn*, Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12 (2021), pp. 776–785.
[16] S. J. Pan and Q. Yang, *A survey on transfer learning*, IEEE Transactions on knowledge and data engineering, 22 (2009), pp. 1345–1359.
[17] D. Saini, V. Arundekar, K. Priya, and D. J. D'Souza, *Identification of helmets on motorcyclists and seatbelt on four-wheeler drivers*, in Recent Advances in Artificial Intelligence and Data Engineering, Springer, 2022, pp. 99–107.
[18] A. Ş. Şener, I. F. Ince, H. B. Baydargil, I. Garip, and O. Ozturk, *Deep learning based automatic vertical height adjustment of incorrectly fastened seat belts for driver and passenger safety in fleet vehicles*, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 236 (2022), pp. 639–654.
[19] S.-H. Tsang, *Review: Mobilenetv2—light weight model (image classification)*, Towards Data Science, Svibanj, (2019).
[20] Z. Wang and Y. Ma, *Detection and recognition of stationary vehicles and seat belts in intelligent internet of things traffic management system*, Neural Computing and Applications, 34 (2022), pp. 3513–3522.
[21] B. Zhou, D. Chen, and X. Wang, *Seat belt detection using convolutional neural network bn-alexnet*, in International Conference on Intelligent Computing, Springer, 2017, pp. 384–395.
[22] B. Zhou, L. Chen, J. Tian, and Z. Peng, *Learning-based seat belt detection in image using salient gradient*, in 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2017, pp. 547–550.

Edited by: Katarzyna Wasielewska
Received: Jun 28, 2022
Accepted: Nov 2, 2022