

Driver's Seat Belt Detection Using CNN

DS Bhupal Naik¹, G Sai Lakshmi², V Ramakrishna Sajja³, D Venkatesulu⁴, J Nageswara Rao⁵

^{1,2,3,4}Department of Computer Science & Engineering, Vignan's foundation for Science, Technology & Research

⁵Sr.Assistant.Professor, Dept of Computer Science and Engineering, Lakireddy Bali Reddy College of Engineering (Autonomous), Mylavaram, Krishna Dt, Andhra Pradesh, India, 521230

¹dsbhupal@gmail.com, ⁵nagsmit@gmail.com

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: Seat belt detection is one of the necessary task which are required in transportation system to reduce accidents due to abrupt stop or high speed accident with other vehicles. In this paper, a technique is proposed to detect whether the driver wears seat belt or not by using convolution neural networks. Convolution Neural Network is nothing but deep Neural Network. ConvNet automatically collects features using filters or kernels from images without human involvement to classify the output images. Compared to different classification algorithms, preprocessing required in ConvNet is least. In this proposed method, first ConvNet is built and trained using Seatbelt dataset of both standard and non-standard. ConvNet learns the features from the images of seat belt dataset and performed better with an accuracy of 91.4% over SVM with 87.17% and an error rate of 8.55% when compared with SVM with 12.83% in case of standard dataset.

Keywords: Convolution Neural Network, Neural Network, Maxpooling, Machine Learning, Deep learning.

1. Introduction

Passengers and drivers may also be injured with the aid of a unexpected halt or a great velocity crash in a vehicle[8][13]. To limit these damages, disaster management section need that all motorists and travelers wear seat belts, which are designed as a shield and reduce deadly injuries. If driver put on seat belt while driving a vehicle then chance of getting essential injuries are very less even though accident happen[11][12].

Seat belt detection is important and is extensively required in the discipline of transportation system[14][15]. So Seat belt detection method has been proposed. It detects whether the driver wear belt or not using Convolution neural network. Seatbelt detection dataset constructed which is given as input to ConvNet. ConvNet has mainly Convolution Layer to learn features using convolution operation between input image and filter and pooling layer which is gradually reduce the spatial dimension of the feature map to minimize the no of parameters and computation in network which leads to decrease of over fitting. These features are utilized to classify that the driver wear belt or not. Seat belt detection dataset has images of driver both belt and no belt.

2. Related Work

To tackle seat belt detection in image, good work has been accomplished till now.

[1] Proposed a method. It automatically detects the seat belt in the images which are captured by surveillance cameras, which follows mainly two procedures. In first method they identifies driver area using vehicle outline ,for vehicle outline detection first they get the vertical boundaries of driver area according to the position of license plate and then get the horizontal boundaries of windscreen for the identification of one's of the driver area. Then identifies the possible seat belt edges using an effective algorithm which is based on direction evidence measure in the HSV color space which extracts some regions as candidate regions based on potential edges and projected by using sobel operator. Finally further verification is done by using judging rules.

[2]Proposed an automatic seatbelt detection approach which follows 3 steps first it extracted edge features from an input image using edge detection method. next a salient gradient feature map is constructed by taking salient gradient features from the input image. At last the gradient feature map is given to the machine learning method for binary decision which yields the input image contains seat belt or not. Here edge detection method plays a main role to give best results, because edge detection decides the corresponding salient gradient map which is given to machine learning method to identify seat belt or belt in given input image. In proposed technique parameter selection is done experimentally. To improve accuracy results more robust edge detection and proper parameter selection needed.

[3] Proposed an algorithm for detection of seat belt. This algorithm uses a feature which is based totally on gradient orientation to discover seat belt. In this after completion of image pre processing, front window location and human face were detected .finally feature is obtained in required region and result is depends and given by computing seat belt feature in the place which is near to identified human face right side. In this method accuracy depends more on human face detection. A kind of classifier needed to complete the detection.

[4] Proposed a method for seat belt detection under the multi-detectors contextual framework which can train multi detectors and maintains the output of score map with a detector which is used as contextual information for the help of decision at the subsequent stage. Histograms of Oriented gradients features and Haar like features are combined collectively to get very powerful image representations. Most discriminative feature is chosen

automatically by using AdaBoost learning.

[5] Presented a method based on cascade adaboost classifier for detection of seat belt. In this first it learns cascade adaboost classifier to detect the vehicle window images by using many labeled training samples after that it computes gradient map. Finally Hough transform and canny edge detection are used to determine whether the driver is wearing a belt or not.

3. Dataset

Seat belt detection Dataset includes two datasets called standard dataset containing the 2155 images which took from Yawning Detection Dataset [7][10], Non standard dataset consists of 8058 images which are downloaded from online sites including belt and no belt images of various drivers .This is divided into two parts known as train set and test set further divided into two parts and named as belt and no belt images.

Detailed explanation of dataset is shown in table 1 and table 2.

Dataset	No of train Images	No of Test images
Standard Dataset	2038	117
Non Standard Dataset	7928	130

Table 1: Dataset Description

Dataset	Belt images	Non belt images
Standard dataset(Train Set)	1600	438
Standard dataset(Test Set)	55	62
Non Standard dataset(Train Set)	3968	3960
Non Standard dataset(Test Set)	80	50

Table 2: Detailed explanation of belt and non belt in train and test sets

4. Proposed Method

Driver's Seat Belt Detection System based on Convolution neural Network. It is additionally acknowledged as CNN or ConvNet and stimulated with the aid of the biological visible cortex which has small areas of cells that are sensitive to the areas of the visible field. CNN is a multilayer neural network. It comprises of one input layer, one output layer and hidden layers as per the requirement.

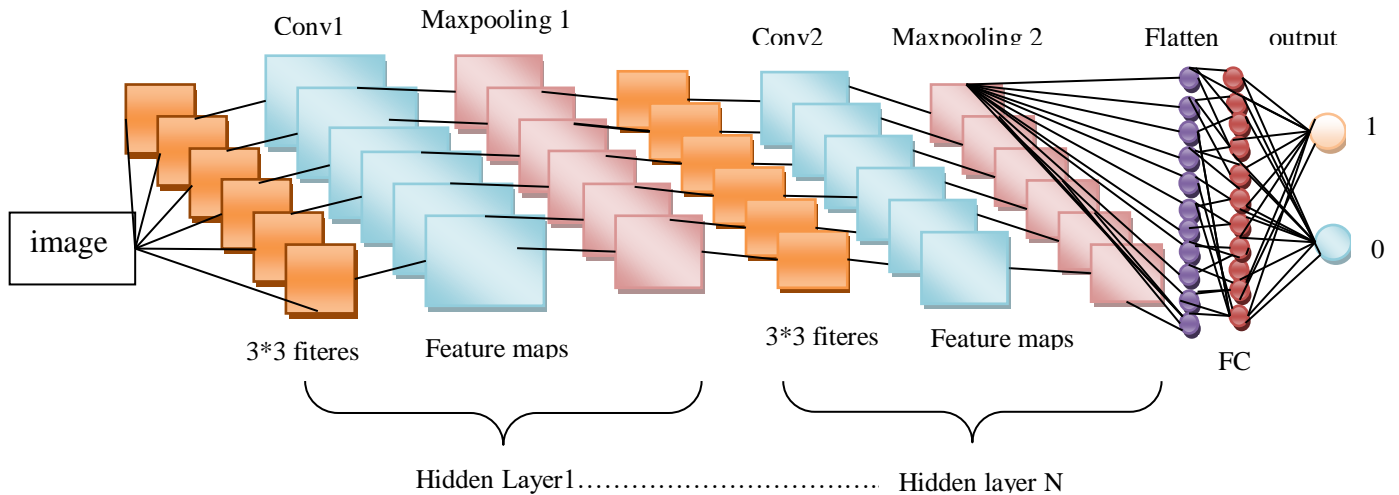
ConvNet has mainly two parts one is feature learning (Conv, relu and Pool) and Classification (Fc and softmax). Each hidden layer contains convolution layer, ReLU layer, and pooling layer. Each convolution layer will detect set of features such as lines, edges, curves etc. It uses filters for extraction of features from images. Each filter produces one output feature. We can vary the no of filters and filter size to obtain best results.

Convolution neural networks play an important function in the area of computer vision. Following are some case studies and functions which are using these neural networks. Image classification, segmentation, object detection, face recognition, Self driving motors that leverage CNN primarily based vision structures etc. It consists of generally convolution, sub sampling and full connection. Detailed explanation of each and every layer of CNN is described as follows.

CNN Architecture

Convolution operation in Convolution Layer

The convolution operation focuses on extraction of different features from input, and they are Lower layers and



higher layers for convolution operation. Low level features such as edges, lines and corners were extracted by lower-layers. And the high-level features, were extracted by higher layers. In this we took input image of size 6*6 and filter of size 3*3. In convolution operation the filter slide all over the input image and does element wise dot product between filter and input image elements. The size of the feature map is $(n - f + 1) * (n - f + 1) * c$ when the size of input image is $n * n * c$ (n is width n is height c represents the no of channels, here c is 1 because it is a gray scale image) and the size of filter is $f * f * c$. So in the below example the size of the result table is $(6-3+1)*(6-3+1) = 4*4*1$. Convolution operation in case of gray scale image is shown in figure 2.

4	9	2	5	8	3
5	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
1	5	4	8	4	4
2	3	9	1	6	8

$$(4*1)+(9*0)+(2*(-1))+(5*1)+(6*0)+(2*(-1))+(2*1)+(4*0)+(5*(-1))=2$$

4	9	2	5	8	3
5	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
1	5	4	8	4	4
2	3	9	1	6	8

1	0	-1
* 1	0	-1
1	0	-1

$$(9*1)+(2*0)+(5*(-1))+(6*1)+(2*0)+(4*(-1))+(4*1)+(5*0)+(4*(-1))=6$$

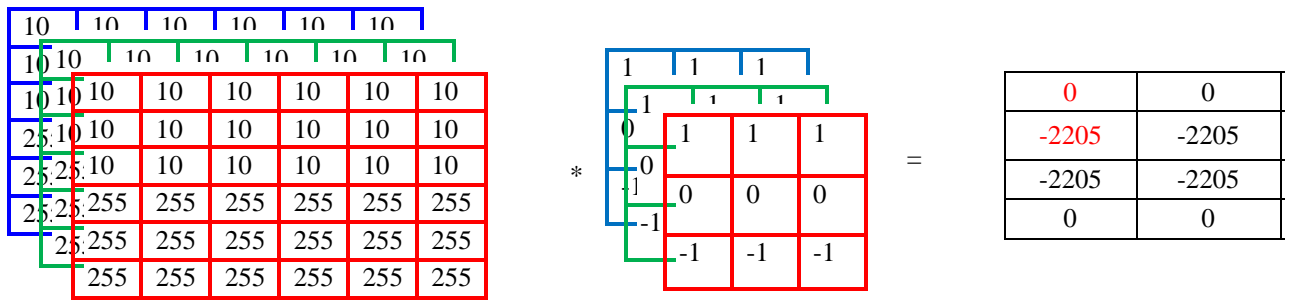
2	6	-4	

4	9	2	5	8	3
5	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
1	5	4	8	4	4
2	3	9	1	6	8

$$(2*1)+(5*0)+(8*(-1))+(2*1)+(4*0)+(0*(-1))+(5*1)+(4*0)+(5*(-1))=-4$$

Figure 2: Convolution operation in case of gray scale image

The below examples shown how convolution operation done in case of color images. The input image contains three channels like red, green, blue so the filter also should have 3 channels. We can use different filters instead of same filter that is for red channel vertical edge filter, for green channel horizontal edges and for blue channel diagonal edges .if we want to detect features only in red channel and we don't want to detect features in green, blue channels in the input image then keep all zeros in green and blue channel of filter.



$(10*1)+(10*1)+(10*1)+(10*0)+(10*0)+(10*0)+(10*(-1))+(10*(-1))+(10*(-1))=0$ for Red channel
 $(10*1)+(10*0)+(10*(-1))+(10*1)+(10*0)+(10*(-1))+(10*1)+(10*0)+(10*(-1))=0$ for Green channel
 $(10*1)+(10*0)+(10*(-1))+(10*1)+(10*0)+(10*(-1))+(10*1)+(10*0)+(10*(-1))=0$ for Blue Channel
 $0+0+0=0$ -> first pixel value in result table
 $(10*1)+(10*1)+(10*1)+(10*0)+(10*0)+(10*0)+(255*(-1))+255*(-1))+255*(-1))=-735$ for Red channel
 $(10*1)+(10*1)+(10*1)+(10*0)+(10*0)+(10*0)+(255*(-1))+255*(-1))+255*(-1))=-735$ for Green channel
 $(10*1)+(10*1)+(10*1)+(10*0)+(10*0)+(10*0)+(255*(-1))+255*(-1))+255*(-1))=-735$ for Blue channel
 $(-735)+(-735)+(-735)=-2205$ -> fifth pixel value in the result table

Figure 3: Convolution operation in case of colour images

Convolution operation using more than one filter is shown in Figure 4. Finally all feature maps corresponding to all filters are stacked.

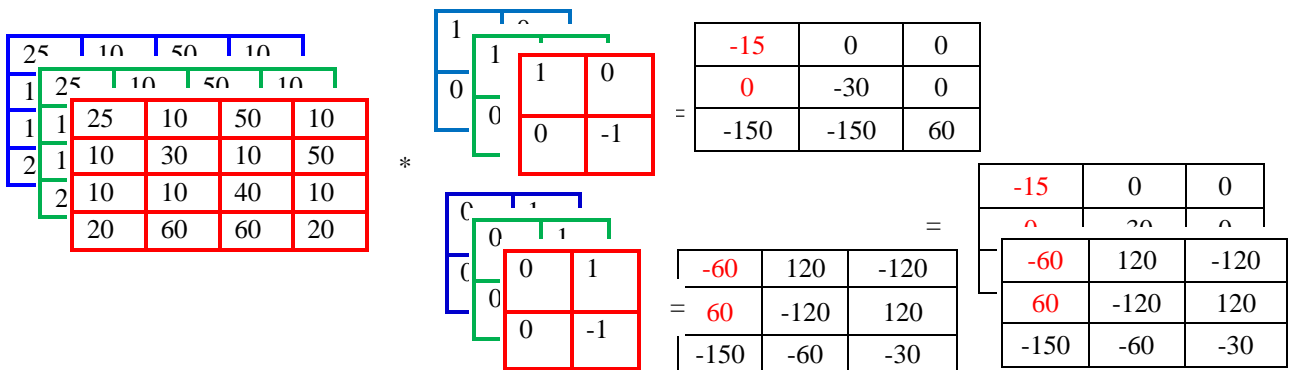


Figure 4: Convolution operation using more than one filter.

Convolution operation in case of padding is shown in figure 5. Padding is of two types one is same and the other is valid. Valid means no padding and same means pad so that the output is similar as input size i.e. $n + 2p - f + 1 = n$. Zero-Padding is used in the example. Zero-padding is nothing but the technique of adding zeroes uniformly to the input matrix. It is a generally used variation that permits the input size regulate to our requirement. Layers when the dimensions of the input volume need to be preserved in the output volume. In this Size of feature map will be $(n + 2p - f + 1) * (n + 2p - f + 1) * c$.

Here input image size is 4*4 and padding is one because we add one row to above and below of the table and one column to left and right of the table. Then the size of the feature map is $(4+2(1)-3+1) * (4+2(1)-3+1) * 1=4 * 4 * 1$.

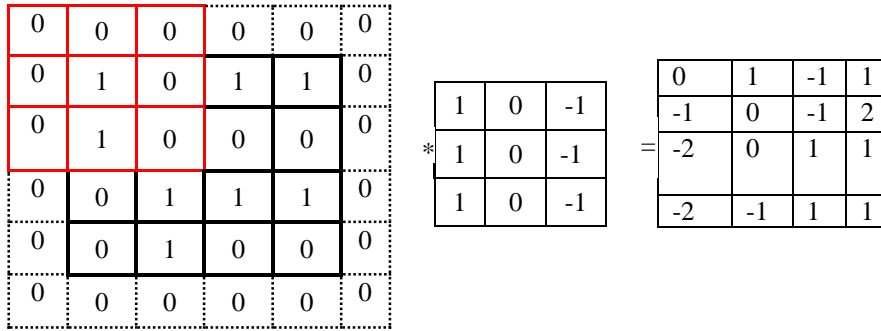


Figure 5: Convolution operation in case of padding

Striding plays an important role in convolution operation. In this filter convolves all around the input volume by shifting one unit at a time. Stride is nothing but the amount through which the filter shifts. Then, the stride was implicitly set at 1. Stride is usually set in a way that output volume is an integer and not a fraction. size of feature map is $\frac{n+2p-f}{s} + 1 * \frac{n+2p-f}{s} + 1 * c$. Convolution operation in case of striding is shown in Figure 6. Here stride is 2. So $[(6+2(0)-2)/2+1] * [(6+2(0)-2)/2+1] * 1 = 3 * 3 * 1$.

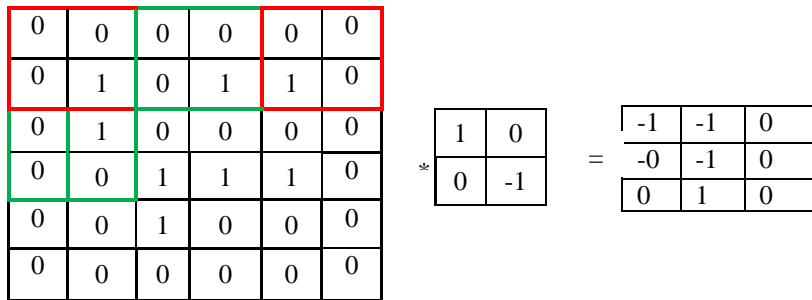


Figure 6: Convolution operation in case of striding

ReLU Function

ReLU stands for rectified linear unit and is an activation function. In this layer it removes every negative value from the filtered image and replaces it with zeros as shown in figure7. The output of a ReLU layer is the same size as whatever is put into it, just with all the negative values removed. Activation function used in the example is shown below.

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x > 0 \end{cases}$$

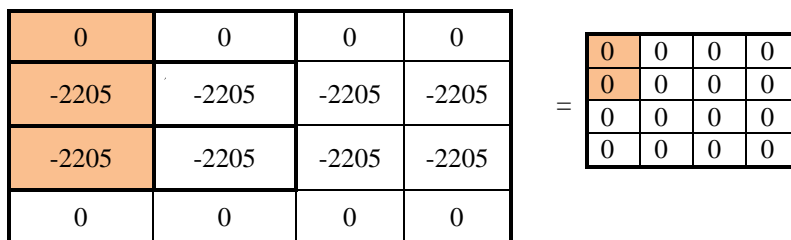


Figure 7: ReLU operation

The main idea of sub sampling is reducing the dimensions of input representation which helps in reducing over fitting. Techniques of sub sampling are maxpooling, minpooling, and average pooling. In max pooling it selects the highest pixel value and in min pooling it selects the lowest pixel value and in average pooling it calculates average value of all pixels from a region. maxpooling is used in Figure 8.

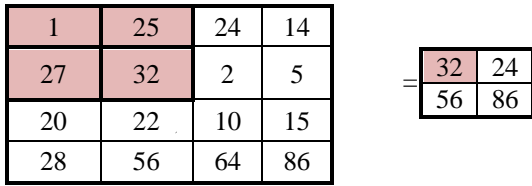


Figure 8: Maxpooling operation

Flatten layer

Flattening is the technique of converting all the resultant 2 dimensional arrays into a single lengthy continuous linear vector. The identification of this step implies that, we are going to flatten our pooled function map into a column like in the figure 9.

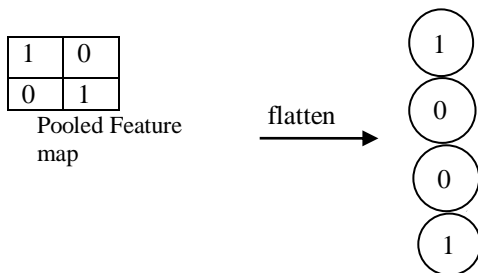


Figure 9: Flatten operation

Fully Connected Layer

The Fully Connected layer is constructed as the way its name implies: it is entirely related to the output of the preceding layer nothing but flatten layer. Fully connected layers are classically used in final stages of the CNN to connect to the output layer and assemble the favoured range of outputs. Example shown in figure 10.

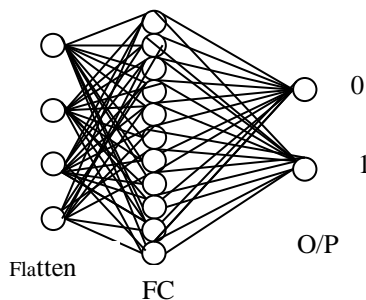


Figure 10: Fully Connected Layer

The ConvNet used in the proposed method uses the sequential model and contains one input layer, output layer, Conv layers followed by ReLu and maxpooling, flatten layer and fully connected layer. To compile the model Adamoptimizer is used. Binary cross entropy is used as loss function. ConvNet is trained with Seatbelt detection dataset. In Conv layer convolution is done between input images and filters. Then these results are gone through the ReLU and maxpolling layers. Finally output of first Conv layer is fed as input to the Second Conv layer likewise till final hidden layer. The output of final hidden layer is flattened into one layer and is given to fully connected layer and then given to sigmoid function. It will display the output i.e. whether the driver wears belt or not. Parameter Sizes in Each layer using standard dataset and Non standard dataset is shown in table 3 and table 4. Architectures are shown in figure 11.

Layer	Filter Size	No of Filters	Pool Size	Activation Function
First Layer	3*3	32	2*2	ReLu
Second Layer	3*3	32	2*2	ReLu
Third Layer	3*3	32	2*2	ReLu

Fourth Layer	3*3	32	2*2	ReLu
First Fully connected	-	-	-	ReLu
First Fully connected	-	-	-	sigmoid

Table 3: Parameters Sizes in Each layer using standard dataset

Layer	Filter Size	No of Filters	Pool Size	Activation Function
First Layer	3*3	32	2*2	ReLu
Second Layer	3*3	32	2*2	ReLu
Third Layer	4*4	32	2*2	ReLu

Table 4: Parameters Sizes in Each layer using Non Standard dataset

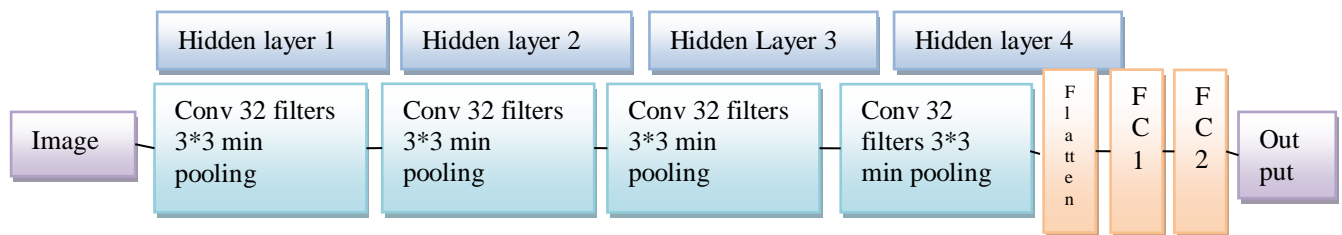


Figure 11: Proposed Architectures for Standard/Non-Standard dataset

5. Results

The proposed approach for seat belt detection using convolution neural network uses both standard and non standard data sets with 2155 images and 8058 images respectively. ConvNet is trained and validated using Seatbelt detection dataset. A total of 2038 images are used for training and 117 images are used for testing for standard dataset. In case of non-standard data set, 7928 images were used for training and 130 images for testing. The results are shown below.

Standard dataset(117)	Correctly classified	Incorrectly classified
Belt (55)	53	2
Non belt (62)	54	8
Accuracy	$((53+54)/117)=91.45$	

Table 5: Description of Accuracy Results with Standard Data Set

Nonstandard dataset(130)	Correctly classified	Incorrectly classified
Belt (80)	64	16
Non belt (50)	34	16
Accuracy	$((64+34)/130)=75.38$	

Table 6: Description of Accuracy Results with Non Standard Data Set

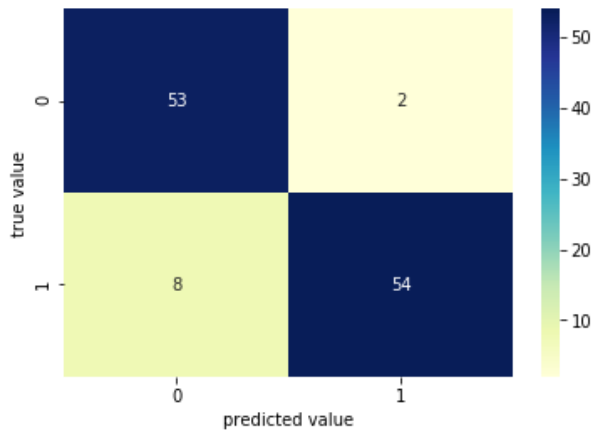


Figure 12: Confusion Matrix for Standard dataset

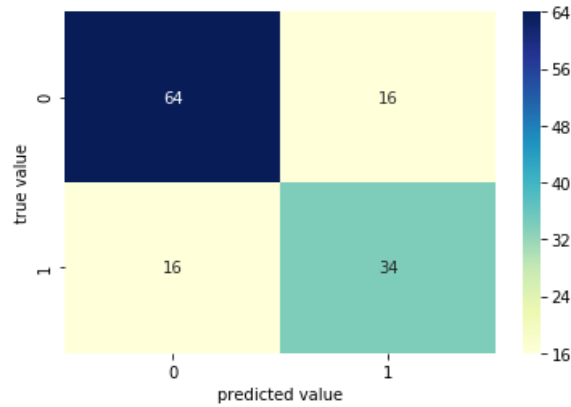


Figure 13: Confusion Matrix for Non Standard dataset

Proposed method CNN gives more accuracy compared to SVM shown in Table 7 and Table 8.

Metrics	SVM	CNN
Accuracy	87.17	91.45
Error Rate	12.83	8.55
Precision	83.33	86.88
Recall	90.90	96.36
F1score	86.94	80.0

Table 7:Comparitive study on Evaulation Measures on Standard Dataset

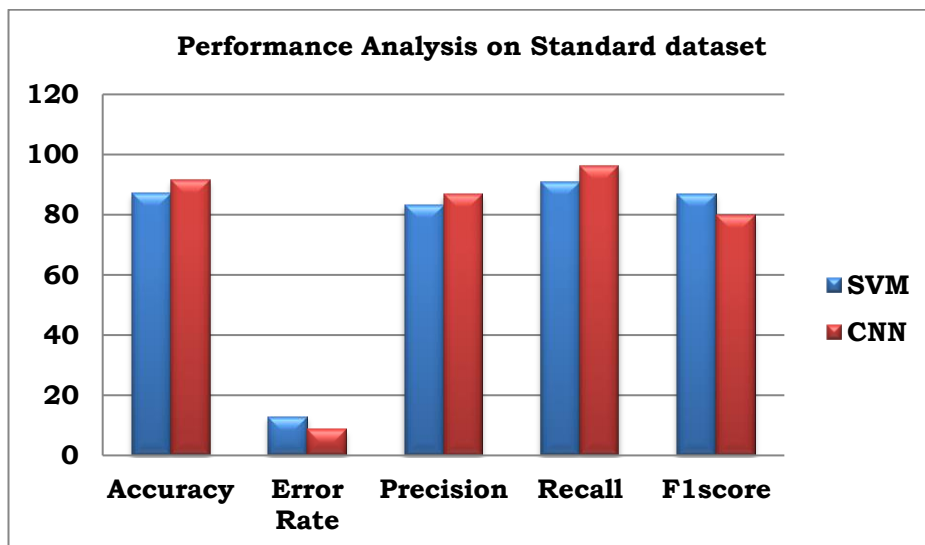


Figure14: Performace analysis of Standard dataset

Metrics	SVM	CNN
Accuracy	70.76	75.38
Error Rate	29.24	24.62
Precision	76.92	80.0
Recall	75.00	80.0
F1score	75.94	80.0

Table 7:Comparitive study on Evaluation Measures on Non Standard Dataset

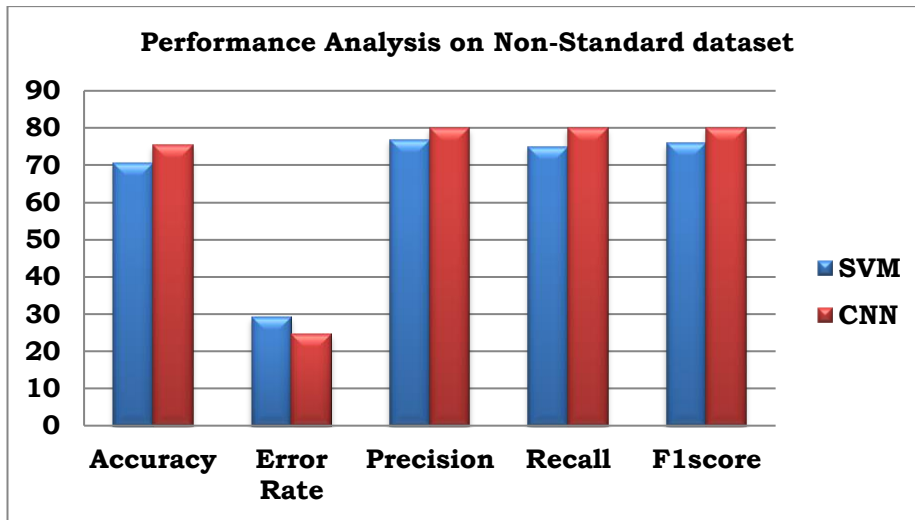


Figure15: Performace analysis of Non-Standard dataset

Loss and Accuracy graphs

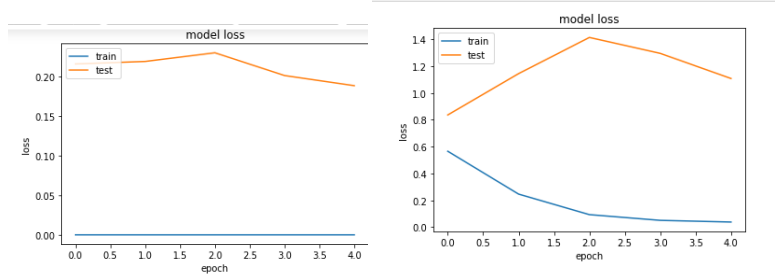


Figure16: Loss and accuracy Curves for Non Standard data set

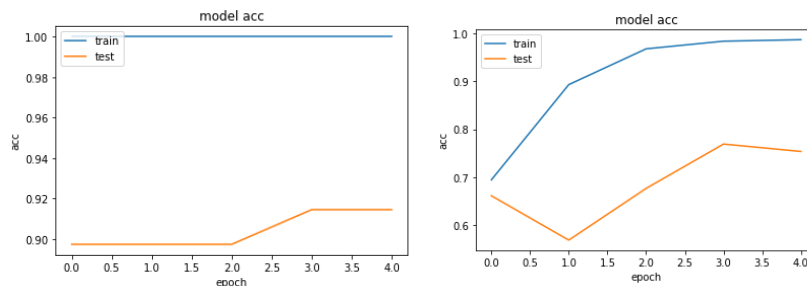


Figure17: Loss and accuracy Curves for Standard data set

6. Conclusion

earing seat belt is such an effective measure of active safety that enough attention should be paid to it. This paper proposed an approach for seat belt detection using convolution neural network. ConvNet is trained and validated using Seatbelt standard and non-standard datasets. The accuracy of 91.45% and 75.38% was achieved

in case of standard and non-standard datasets respectively over SVM with 87.17% and 70.76% respectively. Similarly, an error rate of 8.55% was achieved using CNN over SVM using 12.83%.

References

1. Guo, H., Lin, H., Zhang, S., & Li, S. "Image-based seat belt detection", In *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety*, July, 2011, pp. 161-164.
2. Zhou, B., Chen, L., Tian, J., & Peng, Z. "Learning-based seat belt detection in image using salient gradient" In *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, June, 2017, pp. 547-550.
3. Yu, D., Zheng, H., & Liu, C., "Driver's seat belt detection in crossroad based on gradient orientation", In *2013 International Conference on Information Science and Cloud Computing Companion*, December, 2013, pp. 618-622.
4. Qin, X. H., Cheng, C., Li, G., & Zhou, X., "Efficient seat belt detection in a vehicle surveillance application" In *2014 9th IEEE Conference on Industrial Electronics and Applications*, June, 2014, pp. 1247-1250.
5. Li, W., Lu, J., Li, Y., Zhang, Y., Wang, J., & Li, H., "Seatbelt detection based on cascade adaboost classifier", In *2013 6th International Congress on Image and Signal Processing (CISP)*, Vol. 2, December, 2013, pp. 783-787.
6. Yi, Z., "Evaluation and Implementation of Convolutional Neural Networks in Image Recognition", In *Journal of Physics: Conference Series* (Vol. 1087, No. 6, September, 2018, p. 062018).
7. [7] Shabnam Abtahi, Mona Omidyeganeh, Shervin Shirmohammadi, Behnoosh Hariri, "YawDD: a yawning detection dataset", In *MMSys'14 Proceedings of the 5th ACM Multimedia Systems Conference*, March, 2014.
8. G K. Srivastava, R. Verma, R. Mahrishi and S. Rajesh, "A Novel Wavelet Edge Detection Algorithm for Noisy Images," ICUMT '09, " *International Conference on Ultra Modern Telecommunications & Workshops*", St. Petersburg, USA, 12-14 Oct, 2009, pp. 1-8,
9. R. Ge and M. A. Hu, "A method of recognizing seat-belt wear based on gray scale integral projection," *Automotive Engineering*, Vol. 34, No. 9, 20012, pp. 787-790.
10. W. J. Kuo and C. C. Lin, Two-stage road sign detection and recognition, " *Proc. IEEE Int. Conf. on Multimedia and Expo*", Beijing, China, Jul. 2007, pp. 1427-1430.
11. C. W. Hsu and C. J. Lin, A comparison on methods for multi-class support vector machines, " *IEEE Trans. on Neural Networks*", Vol. 13, No. 2, Mar. 2002, pp. 415-425.
12. G.Q. Liu, X.S. Zheng, X.B. Zhang; Vehicle Plate Location based on Image Texture Feature[J], " *Journal of Image and Graphics*", 2005, Vol. 10, 1419-1422.
13. Z.H. Sun, G. Bebis, R. Miller, On-road vehicle detection: a review[J], " *IEEE Transactions ON Pattern Analysis AND Machine Intelligence*", vol. 28, no. 5, 2006, pp.694-711
14. D. Park, D. Ramanan, and C. Fowlkes, Multi-resolution models for object detection, " in *ECCV*", 2010.
15. Huiwen Guo; Coll. Of Electr. & Inf. Eng., Hunan Univ., Changsha, China; Hui Lin; Shaohua Zhang; Shutao Li, Image-based seat belt detection [C], " *2011 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*", 2011, pp. 161-164.