Seat Belt Fastness Detection Based on Image Analysis from Vehicle In-Cabin Camera

Alexey Kashevnik[1,2], Ammar Ali[3], Igor Lashkov[1], Nikolay Shilov[1]
[1]SPIIRAS, St. Petersburg, Russia
[2]Petrozavodsk State University, Petrozavodsk, Russia
[3]ITMO University, St. Petersburg, Russia

Abstract—**Seat belt fastness detection in vehicles is an important factor due to the high protection role in case an accident occurs. Modern vehicles usually have belt fastness detection systems that can be simply tricked. There are also algorithms that can recognize seat belt fastness based on driver visual monitoring. Unfortunately, the existing algorithms are not so efficient and car manufactures do not implement them to vehicles. Most of them based on Hough, Canny, or other edge detection. In this paper, classification for driver seat belt status using a camera inside the driver cabin is proposed. The model based on YOLO neural network for detecting the driver seat belt fastness. Two steps approach was used to solve the problem: the main part of the belt detection and corner detection. These steps allow the system to recognize the situation when the seat belt is fastened behind the human body. Tiny-YOLO was used to detect the main part of the belt as the first object as well as the belt corner as a second object. The model classifies belt fastness between three cases: the belt is not fastened, the belt is fastened correctly, and the belt is fastened behind the back.**

I. Introduction

In 2014, according to the National Highway Traffic Safety Administration (NHTSA), there were 9,385 fatalities on US roads due to unbuckled passenger vehicle occupants. Between 2010 and 2014, 63,000 lives were saved in vehicle accidents, because they were using a seat belt restraint device. Drivers or passengers in a vehicle could become projectiles during an accident. If the belt is unbuckled the passengers can very easily be ejected resulting in death and statistics showed that using a belt could give a surviving probability 44% more [1].

Automatically detecting and tracking driver behavior inside the vehicle cabin is important in safety applications [2], [3] such as detecting head rotation angles, eye status, driver activities (eating, drinking, using the phone) and seat belt state as well more important since it could rescue from death in crucial accidents.

Most existing algorithms depend on edge detection that are not stable in variation environments. They are not accurate because of the huge possibility to find edges with a slope similar to the belt in the driver jacket. Especially inside the car's cabin as well they are weak against rotation and noise.

We introduce a new model based on the YOLO neural network to detect the seat belt fastness and recognize its status. We classify belt fastness between three cases: the belt is not fastened, the belt is fastened correctly, and the belt is fastened behind the back. We have collected the videos from different drivers that drive the vehicle with a fastened belt, without a fastened belt, and with a belt fastened behind the human body. We train the neural network model based on these videos and test it for different drivers. Then we identify the situations when the model works incorrectly to study it again. So, after a few interactive steps, we can conclude that the model works fine against driver rotation and noise related to the lighting conditions.

II. Related Work

The section gives an overview of related research in the topic of seat belt detection problem. A brief start of object detection which still included as one of the most important issues in the computer vision field. The variety of object detection problems solving is expanding every day. Research and development community widely use huge progress in the topic of machine learning solve these problems.

In the paper [4] authors propose a Region-based Fully Convolutional Networks (R-FCN). They use position-sensitive score maps to address a dilemma between translation-invariance of classification and object detection. The backbone of R-FCN is inspired by Residual Neural Network (ResNet).

Other types of object detection neural networks are Single Shot MultiBox Detector (SSD). This deep neural network generates scores for each object presence and produces adjustments to the predicted box that matches the object shape [5].

YOLO neural network, on the other hand, works in a different way comparing to other detectors by assuming that each cell is responsible to predict the bounding box of an object if and only if this cell is the center of the object (Fig. 1). Authors of the paper [6] presented a trained model to detect 9000 different objects. Later YOLOv3 has been released with the same accuracy of SSD but faster by three times [7]. We are using TinyYOLO architecture to detect the seat belt fastness.

Authors of the paper [8] proposed an algorithm based on extracting multi-scale features to train the detection models using convolution neural networks (CNN) and an optimization method to increase accuracy by using the detection scores to train a classification model through support vector machine (SVM).

On the other hand authors of the paper [9] proposed an algorithm to solve this problem, their algorithm based on a combination of Haar-like features and Histograms of Oriented Gradients (HOG) and choosing the most special feature by AdaBoost learning.

ISSN 2305-7254

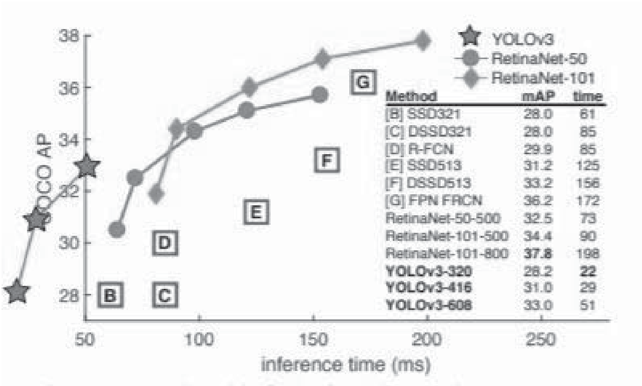

Fig. 1. Comparing YOLO with other models [7]

The paper [10] proposes a new convolutional neural network architecture (NADS-Net) proposed for both pose estimation and seat belt detection (see Fig. 2). NADS-Net inspired by the Feature Pyramid Network(FPN) backbone[11]. This network has been tested over 50 driving sessions including different illumination conditions. The seat belt algorithm applied by producing a probability density function over the image indicating the likelihood of a pixel being seat belt segmentation. Then thresholded to a binary seat belt segmentation mask to evaluate the quality of it according to sensitivity, specificity, precision, F1 score and the intersection of the union. These metrics are shown to be poor to solve this problem since sensitivity, precision and F1-score were 63.51%, 63.58%, and 63.55% respectively and the intersection of the union was only 47%.

Another algorithm proposed by [12] uses an edge detection method for extracting the edge features for constructing a salient gradient map. The authors propose using this map for incorporation into the machine learning approach. They propose Radial Basis Functions (RBF) to make a binary decision about the seat belt state.

Authors of the paper [13] include an edge detection algorithm to identify the belt state depending on the direction information measure in the HSV color space and verifying of the edges will obtain the result. Going back to neural networks [14] proposed an improved convolutional neural network BN-AlexNet. The results showed that this method achieves a 92.51% correct detection rate by rejecting 6.50% test samples. Comparing with other algorithms based on image processing this method has a higher correct detection rate.

Fig. 2. NADS-Net Output for seat belt segmentation

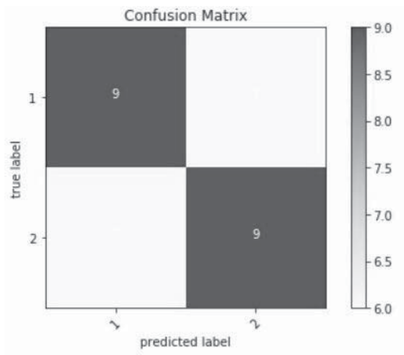

Fig. 3. Confusion Matrix ”MobileNet-V1”

Other researchers also tried to split the problem into two problems by detecting the passenger regions first using single-shot detector SSD and then applies a CNN to detect seat belt [15].

We propose to use YOLO instead of over other neural network architectures because it shows a better response for this particular task. MobileNet-v1 also has been trained on a part of the training data and obtaining accuracy over 60% taking into account that MobileNet is way faster but has less accuracy. The accuracy of the MobileNet model calculated using a test-set and visualized using a confusion matrix (see Fig. 3).

III. The Proposed Model for Seat Belt Fastness Detection

In this section, we propose a new neural network model based on YOLO for seat belt fastness detection. More specifically, the main objectives are to differentiate between the three cases: (1) seat belt placed correctly, (2) seat belt is fastened behind the back, and (3) seat belt is not placed at all.

A. Problem Overview

The driver seat belt detection is similar to any other driver's activity detection (eating, using the phone, and etc.). However, this problem is more complex since the type of features we can obtain from the belt which has a wide similarity to multiple other objects in the vehicle cabin.

Most of the belt detection models are trained based on local data, and it is hard to find such as these images for drivers inside the car cabin hence more similar projects are for detecting the seat belt from camera surveillance images. So, we gather data locally from multiple drivers and preprocess it to obtain quality training.

B. Reference Model

The reference model of the proposed seat belt fastness detection system is presented in Fig. 4. The input image from the camera is analyzed by the YOLO-based neural network model. Based on detected objects there are three cases are

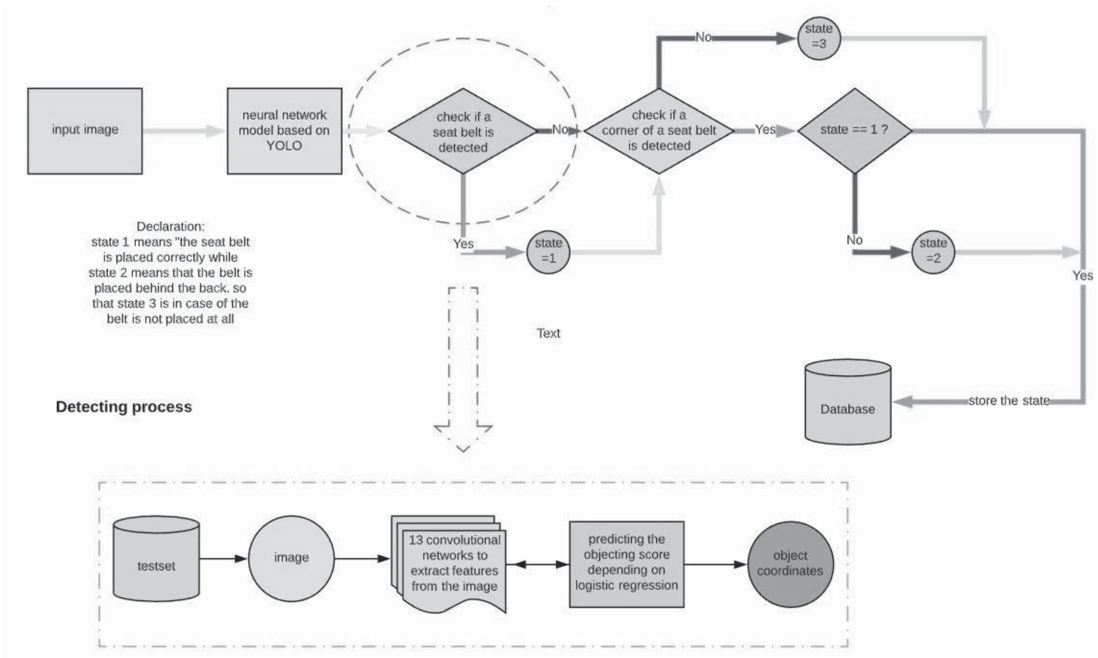

Fig. 4. Reference model of the seat belt fastness detection system

considered: the belt is fastened correctly, the belt is not fastened, and the belt is fastened behind the human body.

It is necessary to mention that all images from the camera are preprocessed before passing to the neural network.

The checking process depends on the neural network outputs. Since the neural network gives us information about the belt state by showing the belt coordinates if it is detected. Furthermore, the outputs of the model are coordinates and two possible labels (0 or 1) depending on the belt part detection: the main part and belt corner (see Fig. 5).

C. Image preprocessing

The driving environment has many illumination varieties. We propose to make the belt more visual. We use the IR camera that provides possibilities to get high quality pictures at night time. Since the camera gives a visual appearance of the driver and the belt, but as well it could receive the same waves length to two different similar colors. Then in some lighting conditions, it is really hard to differentiate between the belt and background.

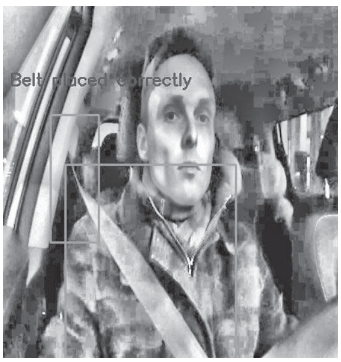

Fig. 5. Main part and belt corner

We implement image transformation from colored to gray images. Then a histogram equalization is applied to decrease the contrast between the daylight and the cabin backgrounds. Followed by a transformation from grayscale to Lab coordinates where L holds for lightness and a,b for the compressed values of the pixels. An adaptive algorithm for contrast enhancement [16] is applied.

Contrast Limited Adaptive Histogram Equalization (CLAHE) is an image processing technique used to manipulate the contrast in an image. The difference between it and the regular histogram equalization that, CLAHE computes more than one histograms corresponding to the sections of an image and redistributes the lightness value of the image.

We use this algorithm to enhance the definition of edges which will make the belt more visualized even if the lightness conditions were bad (see Fig. 6 and Fig. 7 for IR images). Also, it will act well in case of using IR cameras.

D. Data collection

We collected videos of drivers in multiple vehicles over three months in the wintertime as well as find some images from the Internet for unknown drivers. The environment around the cars chose randomly with variety for each driver more than one trip was provided. It also included some static gesture sessions where the driver asked to perform different activities such as drinking, eating, using a cell phone, yawing, sneezing and some particular hands motion. Also, the provided videos have been recorded at night and during the day as well during the driving. During the data collection the camera has

Fig. 6. Driver image example processed by CLAHE

TABLE I: INFORMATION ABOUT TRAINING DATA

Origin	Number of people
White	31
Black	13
Asian	5
N/A	11

been placed in different places inside the car cabin. We also use both: RGB cameras and IR ones.

Training data includes 41 males and 19 females people participating in the video collection. Table 1 shows the race of the people who participated in the video collection.

For the test process, we used pictures of new drivers with fasten and unfasten seat belt as well as we test the model in real environment inside the vehicle during the driving. Real environ,emt testing includes multiple trips for more than ten people, at different periods and lightening conditions. Some trips hold for hours including different activities asked from them. To fasten the belt in the middle of the trip or unfasten it multiple times. Since the system will record videos for any critical situations and the unplaced belt will be considered as a critical situation. The model showed high accuracy response of IR videos and regular videos in the day time. But unfortunately, for regular cameras at night time, there was small confidence in belt existence so some false detection

Fig. 7. Driver IR image example processed by CLAHE

```
net.optimized_memory = 0
batch = 32, time_steps = 1, train = 1
   layer   filters  size/strd(dil)      input                output
   0 conv     16       3 x 3/ 1    416 x 416 x   3 ->  416 x 416 x  16 0.150 BF
   1 max               2x 2/ 2    416 x 416 x  16 ->  208 x 208 x  16 0.003 BF
   2 conv     32       3 x 3/ 1    208 x 208 x  16 ->  208 x 208 x  32 0.399 BF
   3 max               2x 2/ 2    208 x 208 x  32 ->  104 x 104 x  32 0.001 BF
   4 conv     64       3 x 3/ 1    104 x 104 x  32 ->  104 x 104 x  64 0.399 BF
   5 max               2x 2/ 2    104 x 104 x  64 ->   52 x  52 x  64 0.001 BF
   6 conv    128       3 x 3/ 1     52 x  52 x  64 ->   52 x  52 x 128 0.399 BF
   7 max               2x 2/ 2     52 x  52 x 128 ->   26 x  26 x 128 0.000 BF
   8 conv    256       3 x 3/ 1     26 x  26 x 128 ->   26 x  26 x 256 0.399 BF
   9 max               2x 2/ 2     26 x  26 x 256 ->   13 x  13 x 256 0.000 BF
  10 conv    512       3 x 3/ 1     13 x  13 x 256 ->   13 x  13 x 512 0.399 BF
  11 max               2x 2/ 1     13 x  13 x 512 ->   13 x  13 x 512 0.000 BF
  12 conv   1024       3 x 3/ 1     13 x  13 x 512 ->   13 x  13 x1024 1.595 BF
  13 conv    256       1 x 1/ 1     13 x  13 x1024 ->   13 x  13 x 256 0.089 BF
  14 conv    512       3 x 3/ 1     13 x  13 x 256 ->   13 x  13 x 512 0.399 BF
  15 conv     21       1 x 1/ 1     13 x  13 x 512 ->   13 x  13 x  21 0.004 BF
  16 yolo
[yolo] params: iou loss: mse (2), iou_norm: 0.75, cls_norm: 1.00, scale_x_y: 1.00
  17 route  13                                    ->   13 x  13 x 256
  18 conv    128       1 x 1/ 1     13 x  13 x 256 ->   13 x  13 x 128 0.011 BF
  19 upsample                 2x    13 x  13 x 128 ->   26 x  26 x 128
  20 route  19 8                                  ->   26 x  26 x 384
  21 conv    256       3 x 3/ 1     26 x  26 x 384 ->   26 x  26 x 256 1.196 BF
  22 conv     21       1 x 1/ 1     26 x  26 x 256 ->   26 x  26 x  21 0.007 BF
  23 yolo
[yolo] params: iou loss: mse (2), iou_norm: 0.75, cls_norm: 1.00, scale_x_y: 1.00
```

Fig. 8. The proposed neural network model

considered as the belt was not placed while it was.

We apply for different transformations of random images from the training data like rotation, increasing/decreasing contrast, adding noise, applying different filters (Gabor, High/low pass filters, and etc.). We propose to use grayscale images so that the colors of clothes or belt didn't affect the training process. As well we used a neural network to detect the faces from a set of images and remove these faces to make the data more generic.

E. Trained Neural Network Model

We train a model that has one head that generates two rectangles indices for the main part of the belt and the corner belt detection. This output processes using a simple decision tree similar to the decision tree of a NAND operator assuming that detecting the belt is similar to one and detecting the belt corner is similar to zero. Then if the main part of the belt detected we assume that the belt is placed correctly but if it is not detected then we go through a branch to a new question includes if the belt corner detected. If it is then the belt is placed behind the back otherwise the belt is not placed at all.

The model architecture is similar to Tiny-YOLO architecture. It consists of convolutional, maxpools, route and Yolo layers. The loss function as well is taking 4 errors into account and the error then calculated using a weighted average. The main parameters in the loss function are the center of the detected box, the height, and width of the rectangle, the Intersection Over Union (IOU) and the probability of an object (see Fig. 8).

F. Implementation

The proposed YOLO-Net was trained in Google Colab [17]. The provided GPU used to train the model. The model is initialized with original YOLO weights. Only the output layers were pre-trained. Additionally, random image augmentation used to train data.

The training process (see Fig. 9) includes passing images and values to the model. These values are for each input image

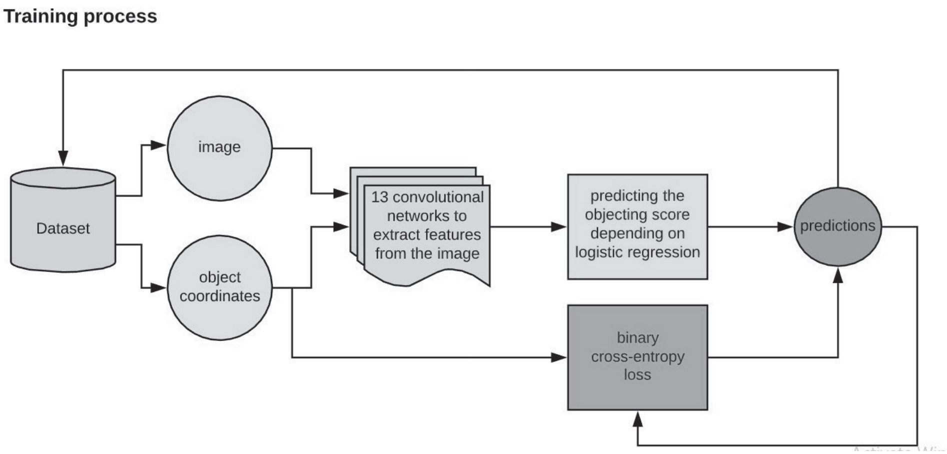

Fig. 9. Neural network model training process

that includes belt parts (main part, corner, or both). Then the coordinates of these parts and the labels of the state are provided but in case that there is no belt in the image no values were provided. The weights updated according to a weighted average function that takes into account the coordinates as well the class type and calculates each error using mean squared error (MSE). Also, it is important to mention that confidence is a part of the output vector which is calculated according to the intersection over union) since we have the coordinates as a ground truth.

All output values are averaged in one value using a weighted average and that represents the loss function which has been used to update the weights while the training process.

To enhance the edges on the image we tried different approaches, using a Canny edge detector to detect the edges and make them more visualized was a good approach but it does have a disadvantage of enhancing undesirable edges. It could be included in some images and didn't in others according to the inner environment of the cabin and the lighting issues. That will affect the training to make the model more sensitive.

So, Gabor-filter was chosen to analyzes whether there is any specific frequency content in the image. It showed a different response according to the environment. for each set of images, it needs different attributes to work fine. So, it was used to some images in the training set to be included in the training process to generalize the data. We also take into consideration multiple noise models (Gaussian noise, Rayleigh noise, gamma noise, exponential, and uniform noise and impulse, periodic noise). To make the data more generic for training. Further, a manipulation process with the lighting also holds using CLAHE algorithm.

IV. Results

We build several models and enhanced them iterative based on the evaluation. The first model has been built with the original data without any processing, transformations, or filters applied to them. And it was shown that it has a good quality for similar trips but it had problems with illumination and the training process that was weakly affected by the image colors (see Fig. 10).

Fig. 10. First trained model output for gray scale images

As it is shown in Fig. 10 for the same driver there are two predictions for similar situations and that caused because of changing the colored image into grayscale before passing to the model. And actually, we can define that as an overfitting problem since it was affected by changing the colored image into a gray one. So, we proposed to generalize data more. And to be effective against the colors it was better to move everything to the grayscale plane. On the other hand, passing the same picture in an RGB format will give a correct prediction as seen in Fig. 11.

So, retraining the model with more generalized data by applying different operations randomly on the original data. Transformation for all data to grayscale, applying histogram equalization for them, changing the contrast randomly applying filters (bandwidth filters, Gabor filter, median and mean filters, Gaussian filter) as well as adding noise (random, frequent, Gaussian). Then the accuracy of the model has been improved. As shown in Fig. 12, the same previous image passed to the model and detected correctly.

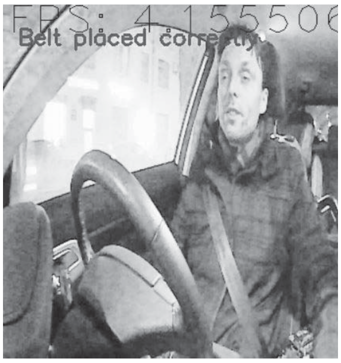

Fig. 11. First trained model output on an RGB images

Fig. 12. Second trained model output on both RGB and gray scale images

The accuracy of this model is high. For more than one thousand images its precision was over 80%. But it does have low accuracy in the case of rotated images (see Fig. 13). It had bad results in that case.

So, we proposed to retrain the data by rotating them from different angles (90, 180, 270) and rebuilt a new model. The third model showed more increased accuracy comparing to the second one. It was over 95% for normal images. But for rotated images, it didn't get high accuracy. It was around 80%. And actually, it was a more general model so in some particular cases it could detect a non-belt object like a belt so we stopped generalizing the data. Fig. 14 shows the third model response to the rotation case.

Fig. 13. Second trained model against rotation

We passed more than 100 images of the testing set to three models to approximate the accuracy of each. For the same testing data and the number of epochs, we obtained the results shown in Fig. 15. It shows how the accuracy increased from 70% to over 80% and finally over 95% for the last model. We can also assume that the accuracy of the third model even more because we checked some of the images which have been incorrectly classified (see Fig. 16).

Fig. 14. Third trained model against rotation

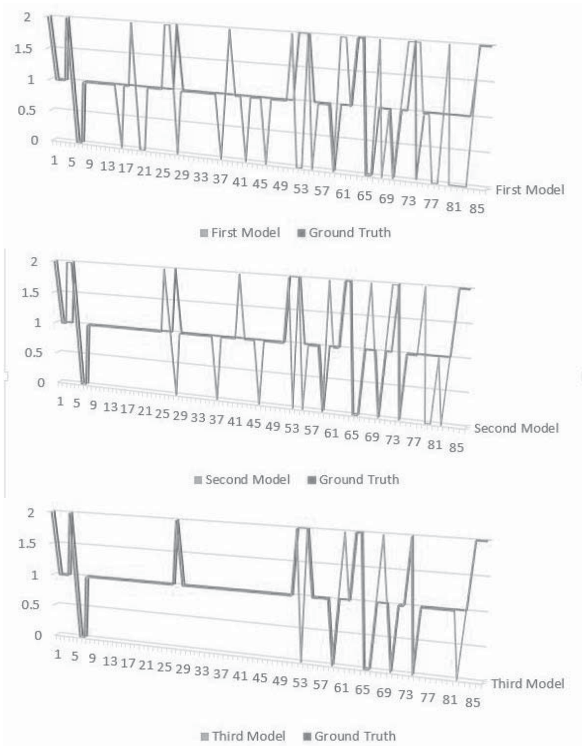

Fig. 15. Models response for testing Data

TABLE II: COMPARISON BETWEEN MODEL RESPONSE AND THE GROUND TRUTH

Predictions	Video-1	Video-2	Video-3	Video-4	Video-5
BPC	0	307	0	0	56
BC	143	307	0	2	0
BND	0	0	224	503	39
Reality	Video-1	Video-2	Video-3	Video-4	Video-5
BPC	0	307	0	0	89
BC	143	307	0	0	0
BND	0	0	224	505	0

Table II shows comparison between the model response and the ground truth when belt is placed correctly (BPC), belt corner detected (BC) and belt is not detected (BND).

This detection is close to being classified as the belt is placed behind the back even more than being classified as placed correctly. These boundary situations show the instability of the model. So, some similar situations will be classified as the belt placed correctly and others will be classified as the belt placed behind the back.

Third model has been integrated and evaluated to the Drive Safely application [18]. We evaluate the model on more than 100 videos captured in real driving conditions.

The accuracy showed to be 93.2% for belt detection and 94.5% for belt corner detection. The following table shows the model response for multiple videos included in the testing. We test the model for different images in different environments and conditions downloaded from the Internet or taken using the different types of cameras (see example in Fig. 17).

Fig. 16. Example of wrong detection

V. CONCLUSION

In this paper, we proposed a new model to detect the driver seat belt fastness based on YOLO neural network architecture. This model detects seat belt fastness in different environmental conditions. And it differentiates between three different cases: the seat belt is fastened correctly, the seat belt is fastened behind the back, and the seat belt is not placed at all. Comparing considered in the related work section methods the proposed one have the best accuracy. Comparing with BN_alexnet[14] which has an accuracy of over 92% the proposed model achieved accuracy over 93%. Also, The used image processing techniques gave the model points against lightning changes. These results may provide practical insights into future researches and industrial development.

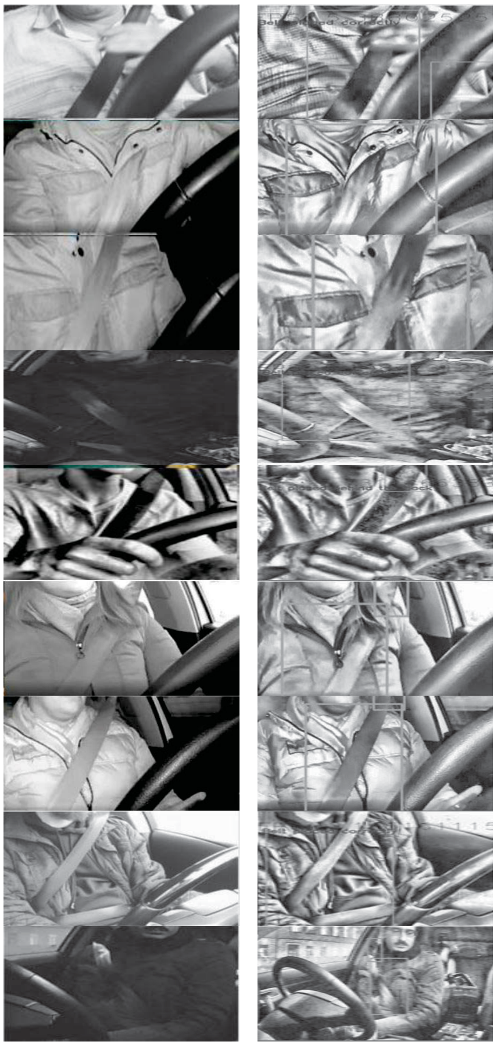

Fig. 17. Arbitrary input testing images and The model response

We use Drive Safely system to test the belt detection model. Drive Safely is an application for driver monitoring, analysis, and recommendations according to the driving behavior for accident prevention using a smartphone. It takes advantage of smartphone sensors (camera, accelerometer, gyroscope, GPS, and microphone) to analyze the driving process [18].

For future work we are going to enhance run-time complexity. The proposed model handles about 12 frames per second on Core i5-8265U CPU for input image size (480, 480) which are not so far from real-time systems but still need improvements. An optimization process over YOLO architecture for this particular problem should be considered to reduce the parameters and improve the performance. Also, the model has wrong detection for the cases when a small higher part and a small down part of the belt are shown and the whole other part is hidden. An improvement in these cases will be considered in future researches. Further, the current method of the work goes through a frame-by-frame detection and it will be better to take the optimal flow into account [19].

ACKNOWLEDGMENT

The presented results are part of the research carried out within the project funded by grant ## 17-29-03284 of the Russian Foundation for Basic Research. Experiments have been conducted in scope of the Russian State Research 0073-2019-0005. The neural network model training has been supported by Government of Russian Federation, Grant 08-08.

REFERENCES

[1] Virtual Drive official website,: https://www.virtualdriveoftexas.com/seat-belt.

[2] A. Kashevnik, I. Lashkov, A. Gurtov "Methodology and Mobile Application for Driver Behavior Analysis and Accident Prevention", *in IEEE Transactions on Intelligent Transportation Systems*. 2019. pp. 1–10.

[3] A. Kashevnik, I. Lashkov, A. Ponomarev, N. Teslya, A. Gurtov "Cloud-Based Driver Monitoring System Using Smartphone", *in IEEE Sensors*. 2020. pp. 1–1.

[4] J. Dai, Y. Li, K. He, J. Sun. "R-FCN: Object Detection via Region-based Fully Convolutional Networks", *the Neural Information Processing Systems Conference*,2016. pp. 379–387.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, Al. Berg. "SSD: Single Shot MultiBox Detector", *ECCV*,2016.pp. 21-37.

[6] J. Redmon, A. Farhadi. "YOLO9000: Better, Faster, Stronger", *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263-7271.

[7] J. Redmon, A. Farhadi. "YOLOv3: An Incremental Improvement", *ArXiv 2018*.

[8] Y. Chen, G. Tao, H. Ren, X. Lin, L. Zhang, "Accurate seat belt detection in road surveillance images based on CNN and SVM", *Elsevier*, 2016.pp. 80-87.

[9] X. Qin, C. Cheng, G. Li, X. Zhou. "Efficient seat belt detection in a vehicle surveillance application", *9th IEEE Conference on Industrial Electronics and Applications, Hangzhou*, 2014, pp. 1247-1250.

[10] S. Chun, N. Ghalehjegh, J. Choi, C. Schwarz, J. Gaspar, D. McGehee, S. Baek. "A Nimble Architecture for Driver and Seat Belt Detection via Convolutional Neural Networks", *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[11] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie. "Feature Pyramid Networks for Object Detectio", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*,2017.pp. 2117-2125.

[12] B. Zhou, L. Chen, J. Tian, Z. Peng. "Learning-based seat belt detection in image using salient gradient", *12th IEEE Conference on Industrial Electronics and Applications (ICIEA), Siem Reap*, 2017, pp. 547-550.

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

[14] D. Chen, X. Wan. "Seat Belt Detection Using Convolutional Neural Network BN-AlexNet", *ICIC: International Conference on Intelligent Computing*,2017.pp 384-395

[15] A.Elihos, B. Alkan, B. Balci, Y. Artan. "Comparison of Image Classification and Object Detection for Passenger Seat Belt Violation Detection Using NIR & RGB Surveillance Camera Images", *15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand*, 2018, pp. 1-6.

[16] J. Majumdar, S. Kumar. "Modified CLAHE: An Adaptive Algorithm for Contrast Enhancement Of Aerial, Medical and Underwater Images", *IJCET*, 2014, pp. 32-47.

[17] Google Colab official website: https://colab.research.google.com/notebooks.

[18] Google Play official website, Drive Safely: https://play.google.com/store/apps/details?id=ru.igla.drivesafely.

[19] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. "3D human pose estimation in video with temporal convolutions and semi-supervised training", *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7753-7762.

[20] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollar. "Focal Loss for Dense Object Detection", *The IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980-2988.