Kahramanmaras Sutcu Imam University
Journal of Engineering Sciences

Geliş Tarihi : 12.09.2023
Kabul Tarihi : 24.11.2023

Received Date : 12.09.2023
Accepted Date : 24.11.2023

SDA: A NOVEL SKEWED-DEEP-ARCHITECTURE FOR VEHICLE MOTION DETECTION IN DRIVING VIDEOS

EDM: SÜRÜŞ VIDEOLARINDA ARAÇ HAREKET ALGILAMASI IÇIN YENI BIR EĞIK-DERIN-MIMARI

Tansu TEMEL[1*] (ORCID: 0000-0002-8359-1146)
Mehmet KILIÇARSLAN[1] (ORCID: 0000-0002-7212-5262)
Yaşar HOŞCAN[1] (ORCID: 0000-0003-0789-6025)

[1] Eskişehir Technical University, Department of Computer Engineering, Eskişehir, Türkiye

*Sorumlu Yazar / Corresponding Author: Tansu TEMEL, tansutemel@eskisehir.edu.tr

ABSTRACT

Collision avoidance mechanisms are important topics for studies in the field of autonomous vehicles. We could obtain prior information about the collision from the movement angles of vehicles. Therefore, it is important issue to learn the movement angles of vehicles in motion. In the study, an architectural model is developed that learns the horizontal movement angles of vehicles to form a base for collision warning systems. YOLOv3 is modified and used on motion profiles. Thanks to the learned angle values, also the bounding boxes match the traces in the motion profiles smoothly. The results obtained have a mAP value of 79% and an operating speed of 36 FPS. These results are better than when trained on motion profiles of the YOLOv3 architecture. In addition, the use of the new architecture on motion profiles and factors such as noise and bad weather in the image do not adversely affect the results. With these features, a fundamental step has been taken for anti-collision systems.

Keywords: anti-collision, deep learning, driving videos, movement angle, skew bounding boxes

ÖZET

Çarpışma önleme mekanizmaları otonom araçlar alanındaki çalışmalar için önemli bir konudur. Araçların hareket açılarından çarpışma hakkında önceden bilgi alma imkanımız olmaktadır. Bu nedenle hareket halindeki araçların hareket açılarının öğrenilmesi önemli bir konudur. Çalışmada çarpışma uyarı sistemlerine temel oluşturmak amacıyla araçların yatay hareket açılarını öğrenen bir mimari model geliştirilmiştir. Başarılı derin öğrenme mimarilerinden biri olan YOLOv3 geliştirilerek elde edilen yeni mimari hareket profilleri üzerinde kullanıldı. Öğrenilen açı değerleri sayesinde sınırlayıcı kutular da, hareket profillerindeki izlerle tam olarak eşleşmektedir. Elde edilen sonuçlar %79 mAP değerine ve 36 FPS çalışma hızına sahiptir. Bu sonuçlar, saf YOLOv3 mimarisinin hareket profilleri üzerinde eğitildiklerinde elde edilen sonuçlardan daha iyidir. Yeni mimarinin hareket profillerinde kullanılması ile görüntüdeki gürültü, kötü hava gibi etkenler sonuçlarımızı olumsuz etkilememektedir. Bu özellikleri ile çarpışma önleyici sistemler için önemli bir adım atılmıştır.

Anahtar Kelimeler: çarpışma önleme, derin öğrenme, sürüş videoları, hareket açısı, eğik sınırlayıcı kutular

ToCite: TEMEL, T., & KILIÇARSLAN, M., & HOŞCAN, Y., (2024). SDA: A NOVEL SKEWED-DEEP-ARCHITECTURE FOR VEHICLE MOTION DETECTION IN DRIVING VIDEOS. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 27(1), 92-104.

INTRODUCTION

With the advancement of autonomous systems, studies on autonomous vehicles are also increasing. At the forefront of these studies are anti-collision systems. In these systems, it is of great importance to detect the surrounding objects and their movements by using images taken from the camera inside the vehicle. Therefore, in addition to detecting moving objects, their direction of movement must be successfully detected. While doing this, it is necessary to have an architecture that produces fast results in order to be used in real-time systems. One of the challenges here is working on images with ever-changing backgrounds. Besides, it is important in terms of the durability of the method to be developed in dealing with external factors such as noise, bad weather conditions, and reflection in the image.

Collision warning mechanisms are an important issue for today’s autonomous vehicles. We can have information about whether there will be a collision by looking at the movement angles of the vehicles. To do this, we need to observe both the horizontal and vertical movements of the vehicles. In the study, an architectural model that learns the horizontal movement angles of vehicles has been developed to form a basis for collision warning systems. The probability of collision is much higher at the point where the horizontal movements of the vehicles are close to 0 degrees (Kilicarslan & Zheng, 2018). Therefore, learning the motion angles of moving vehicles is an important issue.

In general, objects are detected first and then the detected objects are tracked to make sense of the movements of moving objects. Shape-based vehicle detection systems only provide the position of the vehicle within a picture frame. For autonomous systems, successful detection of both the vehicle position and the direction of movement is a very important criterion.

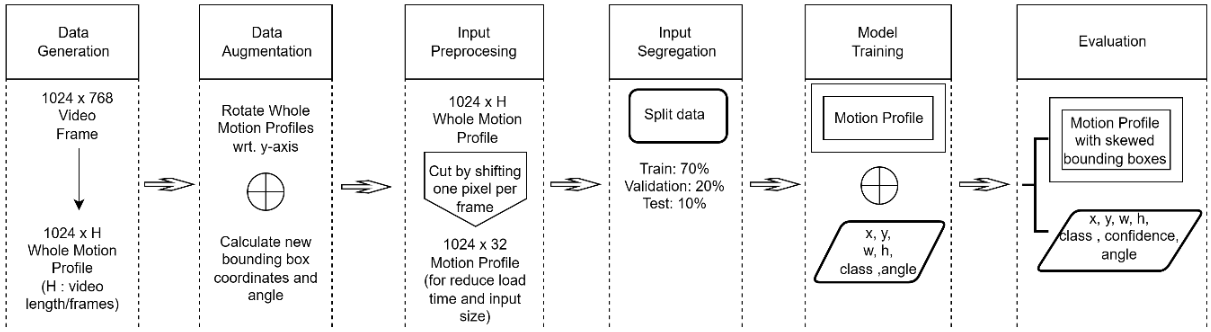
Data Generation
1024 x 768 Video Frame
1024 x H Whole Motion Profile (H : video length/frames)
Data Augmentation
Rotate Whole Motion Profiles wrt. y-axis
Calculate new bounding box coordinates and angle
Input Preprocesing
1024 x H Whole Motion Profile
Cut by shifting one pixel per frame
1024 x 32 Motion Profile (for reduce load time and input size)
Input Segregation
Split data
Train: 70% Validation: 20% Test: 10%
Model Training
Motion Profile
x, y, w, h, class ,angle
Evaluation
Motion Profile with skewed bounding boxes
x, y, w, h, class , confidence, angle

Figure 1. The Figure Illustrates the Overall Framework and Pipeline of the Method

As an alternative to other methods of identifying vehicles in each video frame and monitoring video frames, motion information is extracted and analyzed as smaller-dimensional data. Motion profiles represent picture frames that contain summary information about the motion of moving objects. These profiles have smaller sizes than normal images. In this way, the processing time is shorter. The movements of the vehicles can be seen as certain traces on the movement profiles. These tracks represent information containing relevant movements relative to the current vehicle. When looking at the relative movements of other vehicles from the current vehicle, it is seen that these movements have a certain structure. The traces left by fixed and background objects can be easily distinguished from the structure. Traditional shape-based axis-aligned deep learning architectures have a structure based on the x-axis, y-axis, width, and height values of detected objects. Considering these architectures, it is seen that the traces left by the vehicles on the movement profiles do not fully comply with these structures. Axis-aligned bounding boxes are used in common object detection methods. However, when the movement profiles of the driving videos are examined, it is seen that the traces left by the vehicles on the profile are not in line with the axis and have an inclined structure depending on the movement speed.

The process steps of the developed method from beginning to end are shown in Figure 1. In the first step, the whole motion profiles of the video sequences were created. These profiles are increased by some data augmentation methods. Then, after a preprocessing step, motion profile patches of 1024x32 dimensions were created so that they could be used in the new architecture. For the training process, the dataset was divided into training, validation, and testing at the rate of 70%, 20%, and 10%, respectively. After the model training step, the obtained model was evaluated on the test data, and the results were interpreted.

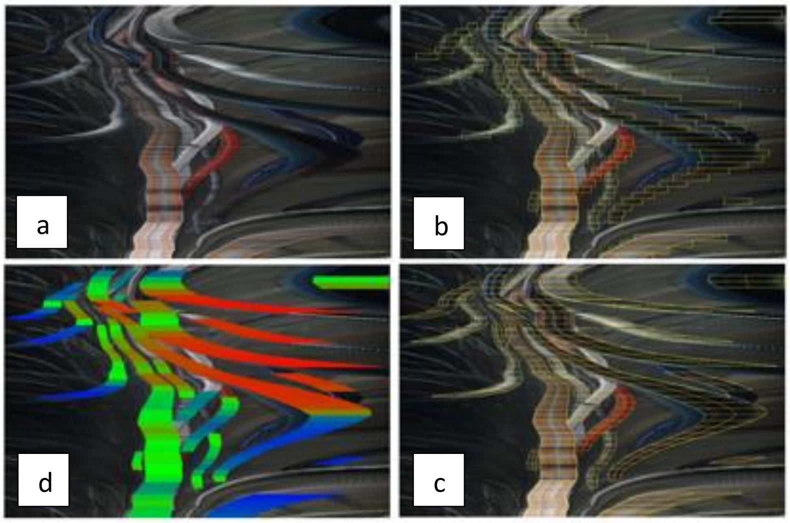

Figure 2. Different Representations of The Motion Profile of the Whole Video

Figure 2 presents the different types of representations of the whole motion profiles. There is a whole motion profile in Figure 2(a), a representation of vehicles with traditional boxes in Figure 2(b), a declaration of vehicles with skew boxes in Figure 2(c), and a colored display according to the angle of movement of vehicles in Figure 2(d).

The remainder of this paper is organized as follows: Section 2 explains the related works. Section 3 presents the proposed SDA algorithm, and Section 4 shows the experiments for a proposed method and accuracy evaluation results. Finally, Section 5 clarifies the results and informs about future works of the paper.

RELATED WORK

Processing video footage is a demanding but challenging task today. With image processing methods, meaningful information can be extracted from moving and fixed camera images. One of the most challenging aspects of working with motion camera images is the objects' location in an ever-changing environment. In addition, being able to deal with noise, reflection, scaling and similar challenges on video images is critical to developing a durable model.

Object detection and recognition, which are essential elements of digital image processing applications, have been struggling for many years. In recent years, thanks to advances in graphics processing units (GPU) and deep learning, object detection, and identification methods have been developed with higher accuracy. Along with these developments, many studies (Zhang et al., 2017; Gordon et al., 2018; Cao et al., 2017) have been conducted on moving objects in video images. These studies are generally on object detection and tracking of detected objects (Behrendt et al., 2017). The first approach using deep learning in multi-object tracking was (Wang et al., 2014), presented in 2014. Another study (Cadieu et al., 2008), proposed a two-layer autoencoder used to improve visual features. Computation of similarities was performed using an SVM after the feature extraction step and it was shown that feature improvement soundly enhanced the model performance. However, the dataset the algorithm tested is not widely used and the results are not comparable to other methods.

In a study (Chen et al., 2018) to calculate a similarity score and a bounding box regression simultaneously, the convolutional portion of ResNet was used to construct a custom model by stacking an LSTM cell on top of convolutions. A visual displacement network was proposed that learns to predict the next position of the object based on the preceding positions of the objects and the effect of an object on other objects in the scene (Zhou et al., 2018). This network was then used to predict the position of the objects in the next frame, taking their past trajectories as input. The network was also competent in extracting visual information from predicted locations and principal detections to calculate a similarity score.

In the method developed based on deep learning to detect and monitor traffic lights in real-time (Behrendt et al., 2017), 2 different neural networks were used to detect and monitor traffic lights based on YOLO architecture. One of the neural networks was trained for traffic light detection and the other one was trained to detect cases where traffic lights are misperceived. In another study (John & Mita, 2019), an image-based semantic interpretation architecture was proposed for the autonomous vehicle. To understand the movements of neighboring vehicles, a view-based vehicle spatio-temporal prediction framework was proposed using YOLOv3 and the new multi-frame semantic segmentation architecture. A rule-based system was used to make sense of the movements of neighboring vehicles by using the estimated vehicle location and movement information.

In the study, which is aimed at real-time in-vehicle video analysis to find and follow the vehicles in front (Jazayeri et al., 2011), the features extracted from the video are continuously projected and followed on a 1D profile. The hidden Markov Model (HMM) was used to separate and track target vehicles from the background. The method is tested in daytime and night-time videos on different road types, where it is robust and effective in dealing with background and lighting changes, as well as working in real time for dash cams. In the method (Li et al., 2020) that detects the rotated objects based on the YOLO architecture, although the rotated state of the object was detected, no information about the rotation angle was accessed. A summary of the above-mentioned studies is given in Table 1.

Image and video processing methods have various application areas mentioned above. One of these application areas is autonomous vehicles and advanced driving support systems. Today, object detection from images has reached high levels of accuracy. However, these methods usually detect objects independent of the temporal dimension and then try to establish relationships between objects by following the objects using models such as LSTM (Hochreiter & Schmidhuber, 1997; Liang & Zhou, 2018). LSTM is a recurrent neural network (RNN) that remembers values at promiscuous intervals. Each sequence requires four linear layers per cell at a time step and for each row. These layers consume large amounts of memory bandwidth. Therefore, most computational units cannot be used when the system does not have sufficient memory bandwidth to feed the computational units. There may be such constraints in the use of LSTMs as it is difficult to add more memory bandwidth. This makes it difficult to use in real-time systems. Detection of vehicles in traffic, which is one of the applications that need to work in real-time, is an important problem for advanced driver support systems (ADAS) and autonomous systems. Therefore, it is necessary to detect the target vehicle and make sense of its movement, both with high accuracy and in real-time. In safe driving systems, to understand the vehicle-vehicle interaction, the direction of movement of the target vehicle relative to the vehicle must be detected instantaneously.

When previous studies were examined, no detailed study was found on learning the direction of movement of vehicles. Therefore, a study was conducted to learn the movement angles of the vehicles. Some angle values are more effective in causing an accident. The importance of this study is that angular movements can be detected instantly at continuous values. In this way, it can be used for autonomous vehicles and collision avoidance systems.

Table 1. Comparative Table of the Literature Works

Authors	Cadieu et al., 2018	Jazayeri et al., 2011	Wang et al., 2014	Behrendt et al., 2017	Chen et al., 2018	Zhou et al., 2018	John & Mita, 2019	Li et al., 2020

Methodology	Two-layer encoder and SVM	Hidden Markov Model	Multi-object tracking	Detect-track approach, two different YOLO architecture	ResNet and LSTM combination	Visual displacement network	YOLOv3, semantic segmentation, and rule-based approach	Rotated object detection using YOLO architecture

SKEWED-YOLO FOR DETECTION ANGLES AND DIRECTIONS

YOLOv3 is an efficient convolutional neural network used for object detection. This architecture splits any input image into the SxS grid system. Each grid in the input image is accountable for object detection. Grid cell estimates the number of bounding boxes of an object (Hui, 2018). There are five items for each boundary box (x, y, w, h, confidence score). (x,y), w and h are the coordinates, width, and height of the object in the input image, respectively. The confidence score is the probability that the box contains objects and how precise the bounding box is. Such algorithms are frequently used in real-time object detection. YOLOv3 has many features we need for real-time object detection by correctly classifying objects. In the study, a new architecture was created by developing the YOLOv3 architecture, which is one of the existing axis-aligned deep learning methods, on learning the angular values of the target vehicles, movement directions, and movements of the vehicles by working with in-vehicle video images.

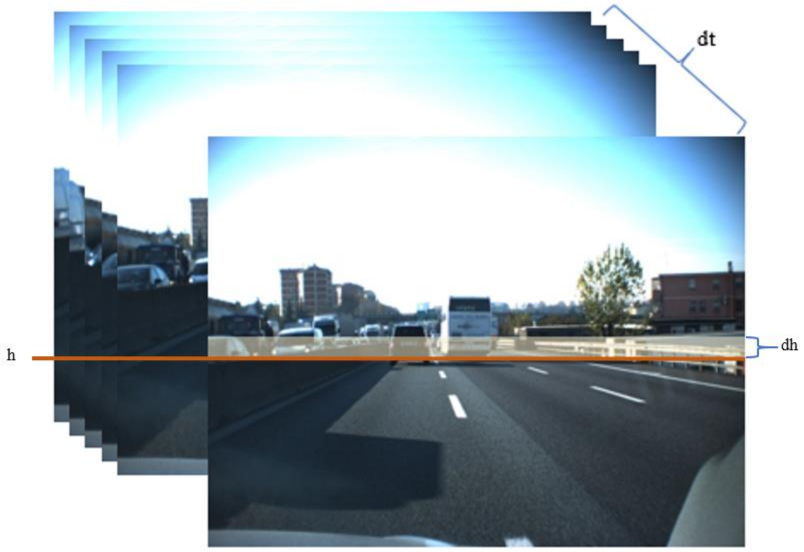

Figure 3. Creating Motion Profiles from Video Frames.

The new architecture seen in Figure 1 was trained on image fragments containing motion information of vehicles, called motion profiles, and a study was conducted to form a basis for anti-collision systems. Thanks to motion profiles, several video frames are combined into a single frame. Thus, the size of the data to be processed is significantly reduced. Since the obtained method works with the smallest size images suitable for the problem, it also improved the FPS value that YOLOv3 obtained on 416x416 images, which can be used in real-time applications. In addition, the method, without the need for an object detection-tracking paradigm, it has been transformed into a single combined image processing as in conventional shape-based object detection methods. In addition, the angle parameter, which is a newly added parameter to architecture different from others, was learned by participating in the training process.

Creation of Motion Profiles

In the training phase of the obtained architecture, picture fragments containing the motion information of the vehicles, called motion profiles, were used. While creating motion profiles, a one-dimensional array is obtained by vertically averaging the pixel values in that area along a certain height from the ground near the horizon according to the camera’s point of view. “dt” of these sequences will be combined sequentially to form the horizontal movement information of the vehicles. Here, “dt” also denotes the retrospective video frame count used to create the motion profile. The motion profile shows the relative motion relative to another vehicle. The formula (Kilicarslan & Temel, 2022) for the creation of motion profiles is as in (1).

$$MPI(x;t) = \sum_{y=h}^{h+dh} I(x,y,t) \tag{1}$$

Here “h” represents a point close to the horizon line, and “dh” represents the height of the piece to be taken from the horizon line. An example of this is shown in Figure 3. The yellow cross-section in this picture shows the area used when creating the motion profile, and the orange line shows the “h” point. Movement profiles are in the structure given in Figure 2(a). The boxes on the motion profiles represent vehicles and their motions. Since it is averaged in the “dh” range, smooth transitions and continuity of the tracks are ensured in the driving videos. Also, luminance consistency is not required like in optical flow-based methods. Therefore, interruption in intensity luminosity will not influence the understanding of motion.

Choosing the “dt” value is very important to understand the movements in the movement profiles. When very small "dt" values were selected, the angles could not be learned sufficiently because there was not enough vehicle motion information in the motion profiles. At very large "dt" values, it may contain more than one movement pattern other than the current situation. In this case, an attempt will be made to learn the angle value on a wrong pattern, and this will cause inconsistency.

Taking these considerations into account, “dt” is set to 16 frames (0.8 s) in MPI as the motion profile patch image. However, these patches are resized to 32 pixels to be compatible with YOLOv3 sublayers. In this way, the size of the image to be used as input is reduced by 24 times.

Angles of Moving Objects

After creating motion profile images, angles must be calculated. Bounding boxes in labeled videos were used while performing this process. While calculating the angles, bounding boxes between the current frame and previous dt frames are used. The midpoints of the two bounding boxes are found, then these midpoints are connected with a straight line to find the motion direction. The position of the vehicle we are in does not affect the calculation. Because the angle is calculated depending on the movement of the surrounding vehicles between the previous position of the “dt” frame and the bounding boxes in their current positions. We can test this by making use of the feature of motion profiles providing us with backward motion information.

The calculation of the formula is shown in (2) and Figure 2. While the red boxes in Figure 5 show the bounding box in the previous picture as much as the “dt” video frame, the blue ones show the bounding boxes of the same vehicle in the current video frame. We find the midpoints of both boxes on the horizontal axis and combine them with a line to determine the slope of this line as the angle of the vehicle′s movement. Colored curved lines are the line segments that show the movement angles of the vehicles.

$$\alpha = \mathrm{atan}\left(\frac{\mathrm{Xcenter(t) - Xcenter(t - dt)}}{dt}\right) \tag{2}$$

Xcenter = midpoint of the horizontal axis of the bounding box

t = current video frame

dt = number of images to be taken backward from the current video frame (time interval)

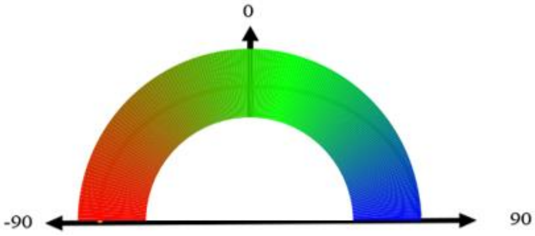

Figure 4. Colored Representation of the Created Angles on the Coordinate System.

In a complicated background, distance background objects may have similar traces to the target vehicles, and empty regions in between two vehicles cause ambiguities. Hence, angles less than minus 85 degrees and greater than 85 degrees were considered as background objects. The visual representation of the coloring of the skewed bounding boxes depending on the angles in the test phase is as in Figure 4.

New IOU

While the results are being obtained, the IOU metric is different from the one used in general object detection, and the angle difference between the learned angles and the real angles is also obtained by participating in the calculation, following the problem. The mentioned formula is shown in (3). In the new architecture created, besides the x-coordinate, y-coordinate, width, and height information of the object, the angle is also included in the calculation as a parameter to be learned.

$$IOU_{new} = \left(\frac{intersectionArea}{box_1Area + \ box_2Area - intersectionArea}\right) \\ x\cos(box_1\alpha - box_2\alpha) \qquad (3)$$

Loss Function Calculation

In addition to the total loss function used in YOLOv3, the loss function of the angles has also been added. The mean square error method was used while calculating the loss function of the angles and width-height of the boxes. For loss calculation of confidence, class, and x-y coordinates of boxes binary cross entropy was operated. The formula for the calculation is as in (4). Besides, the weight value of $\mathcal{L}_x$ is set larger than the other loss functions to increase the learning rate of the angles.

$$ss = \mathcal{L}_{xy} + \mathcal{L}_{wh} + \mathcal{L}_{confidence} + \mathcal{L}_{class} + \mathcal{L}_{angle} \qquad (4)$$

Table 2. Model Training Parameters

	YOLOv3	SDA-YOLOv3
Learning Rate	0.001	0.0001
Batch Size	32	32
Epoch	51	101
Optimizer	Adam	Adam
Anchors (w x h)	[40 x 24]	[10x32] [16x32] [28x32]
	[120 x 24]	[42x32] [62x32] [96x32]
	[350 x 24]	[116x32] [156x32] [337x32]

Image size (w x h)	1024 x 64	1024 x 32

EXPERIMENTS

In the study, the publicly available Toyota Motor Europe Motorway Dataset (TME) (Caraffi et al., 2012) dataset was used. This dataset consists of image frames with a resolution of 768 x 1024. The number of frames per second is 20. TME dataset consists of a total of 28 videos containing more than 30000 images. 20 of these videos covering different conditions were selected. The TME dataset consists of videos at different times of the day. In this way, data with different lighting conditions are available. Data is taken only from the highway. It includes all possible traffic conditions on the highway. The videos were converted into images and 20600 images were obtained. Images obtained from the videos were increased using the data augmentation method.

The SDA model is developed in Python using Tensorflow Keras API and trained on the Nvidia Tesla T4 graphics card with 16GB memory and with Cuda v12 dependencies for GPU support. Also, 12 GB of memory is enough to handle the model creation and testing task. Experiments and inferences are made on an Ubuntu machine with Intel(R) Xeon(R) CPU 2.20GHz and Tesla T4 GPU hardware. For all the training experiments, the default YOLOv3 parameters are used. different from the default settings, anchors are set to different scales in Table 2. To learn the angles better, the loss values of the angle were multiplied by a larger weight value.

During the training phase, 70% of the dataset was used, while 20% and 10% were used for validation and testing, respectively. The bounding boxes in the existing labeling have been changed to cover only the rear of the vehicles. In addition, vehicles are not labeled in cases where there is no rear view of the vehicles and some parts of them are covered by different vehicles. In addition, the oncoming vehicles were excluded from the data set, since the movements of oncoming vehicles left traces in the motion profiles similar to the background image, and the outgoing and incoming vehicles in the data set were clearly separated by a barrier. Data distributions according to the angles are given in Figure 7. As seen in Figure 7(a), the data in the dataset is not evenly distributed according to the angles. This creates an obstacle to learning the angles that do not have enough numbers. Therefore, the data augmentation process has become mandatory. While increasing the data, the number of data was doubled, and the pictures were rotated to be symmetrical with respect to the y-axis. In addition, the coordinates and angles of the vehicles were recalculated according to the formula (5) and added to the available data, and the training process was repeated under the same conditions.

$$\begin{aligned}
&\mathrm{Xcenter}_{\mathrm{augmented}} = W_i - \mathrm{Xcenter}\\
&\mathrm{Ycenter}_{\mathrm{augmented}} = Y\mathrm{center}\\
&Width = Width\\
&Height = Heigth\\
&\alpha_{\mathrm{augmented}} = (-1) * \alpha
\end{aligned} \tag{5}$$

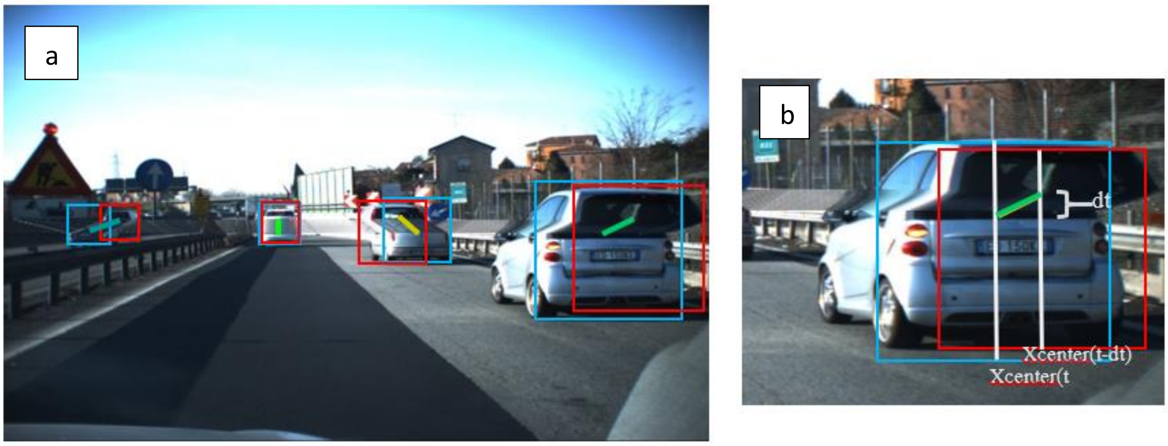

Figure 5. a. Motion Directions are Color Lines, Red Boxes are Current Frame, Blue Boxes are dt Frame Previous **b.** Visualization of Determining the Direction of Movement.

In the study, progress was observed in learning the angle as a new parameter on the YOLOv3 (Muehlemann, 2019) deep learning architecture. Generally, each picture is given separately to the training process. Instead, a single motion profile picture of the video was created, and this whole profile was cut piece by piece in the code (Figure 6) and participated in the training process. In this way, the time taken to read the picture at each step is shortened. The training process has been expedited. Model training time is important for practical implementations. The average training time for SDA-YOLOv3 is 9 hours, this training time with YOLOv3 is approximately 10 hours. While the model size obtained using the YOLOv3 architecture is 246 MB, this value is 236 MB in the SDA-YOLOv3 architecture. While calculating the average precision value, the values produced for each of the test images were compared with the ground truth values. Boxes with an intersection-over-union (3) value above 0.5 were considered TP. The values that should not be in the picture but we found are accepted as FP, and the values that should be but we could not find are accepted as FN. When the results were interpreted, a 79% average precision value, which is better than the average precision value obtained on black padded motion profiles (Kilicarslan & Temel, 2022) of the YOLOv3 architecture used for object detection, was found.

Mean average precision (mAP) is used to evaluate object detection models. mAP compares the ground truth bounding box with the detected box and returns a score. The higher the score, the more accurate the model's determinations. When a model has high recall but low precision, the model correctly classifies most positive examples but has many false positives. When a model has high precision but low recall, the model is accurate when it classifies an example as positive but can classify only a fraction of the positive examples.

Since it is known that there is a general trade-off between precision and recall, a balanced precision-recall graph is required to obtain the best mAP results. Since the mAP metric is used in the testing phase of general object detection algorithms, this metric was used as the accuracy metric for YOLOv3 and SDA-YOLOv3. Thanks to the high mAP value, we can make more accurate determinations. In the proposed method, the 3% difference in mAP value is a significant difference for object detection models.

The angles obtained in the test results were compared with the ground truth data. It can find angles with an average difference of 3 degrees. When the angle differences between ground truth data and predicted data are examined, it is seen that the angle differences of the data, which are very few in number in ground truth data, are much higher. This is clearly seen in Figure 8. Angles can be found with a difference of about 2 degrees in the parts where the data is sufficient.

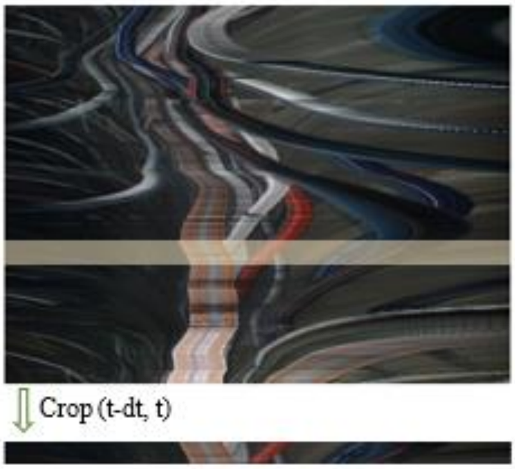

Figure 6. The Transition from Whole Profile to Motion Profile for the Training Process.

In the test part, the necessary development has been made for the drawing of angular bounding boxes, unlike the normal bounding boxes. In this way, bounding boxes can be drawn on the motion profiles that fully adapt. This shows that the angle can be learned together with the coordinate information of the boxes. This is shown in Figure 9.

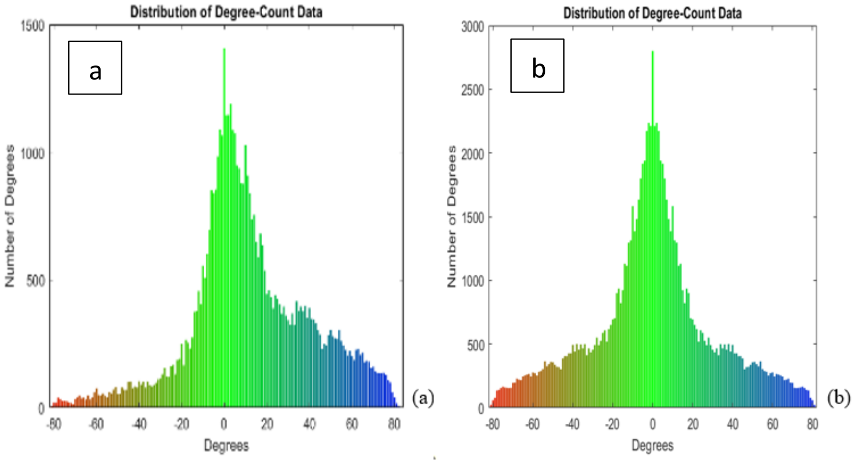

Figure 7. Chart Showing the Distribution of the Data by Angles. (a) Normal Data, (b) Augmented Data)

RESULTS AND FUTURE WORKS

Looking at the results in Table 3, better results were obtained with the normal YOLOv3. In addition, motion angles could be detected with high accuracy. Due to the high number of FP's in both YOLOv3 architectures, the mAP rate is low. The reason for this is that the background images and vehicle reflections have similar images to the vehicle movement, so even if there is no vehicle there, it can produce results as a vehicle. Therefore, not only the motion information is sufficient, but also the shape information must be added.

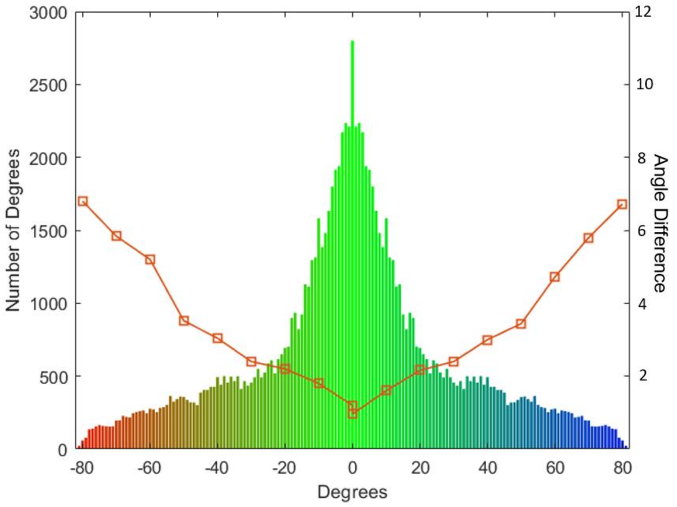

Figure 8. The Average Difference of the Angles Learned According to the Degree Distribution of the Data in the Training Data

The main purpose of the study is to present a new architecture that learns the angle parameter rather than increasing the accuracy of a found method. Therefore, although the mAP value seems low, the results are sufficient to have better results than normal YOLOv3. Learning the movement angles and directions of the vehicles is important in terms of making sense of the movements of the vehicles and thus using them in anti-collision systems. The test results of the presented method and the test results of the YOLOv3 architecture are given in Figure 10. The results obtained by learning the movement angles are more suitable for oblique patterns. The green skewed boxes represent the results obtained from the SDA-YOLOv3 architecture. The yellow bounding boxes are the test images obtained from the YOLOv3 architecture.

In the method offered, bad weather conditions, noise in the image, etc. factors do not adversely affect the results. Thus, we have obtained a more consistent and durable system.

Table 3. Overall Results and Comparison

	YOLOv3 (Kilicarslan & Temel, 2022)	**SDA-YOLOv3 (Proposed)**
mAP	73%	79%
FPS	35	36
Angle Difference	No Angle Information	~3 degrees

In such applications, motion information is mostly obtained by using the optical flow method (Liu et al., 2013; Yun et al., 2022). Then, working on sequential data such as LSTM for object tracking recurrent neural networks are used. Since the method we presented does not use methods such as optical flow and LSTM, we have obtained a real-time application that works faster. In addition, by improving on the current deep learning architecture used in object detection, we were able to find the direction and angle of movement of the vehicle on a single image without using any tracking algorithms. In this way, a study that can be the basis of anti-collision systems has been revealed.

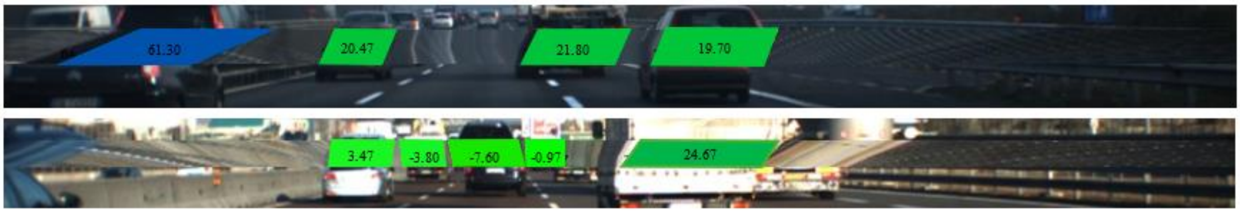

Figure 9. Images of Test Results on Motion Profiles (Motion Angles and Colorized Representation)

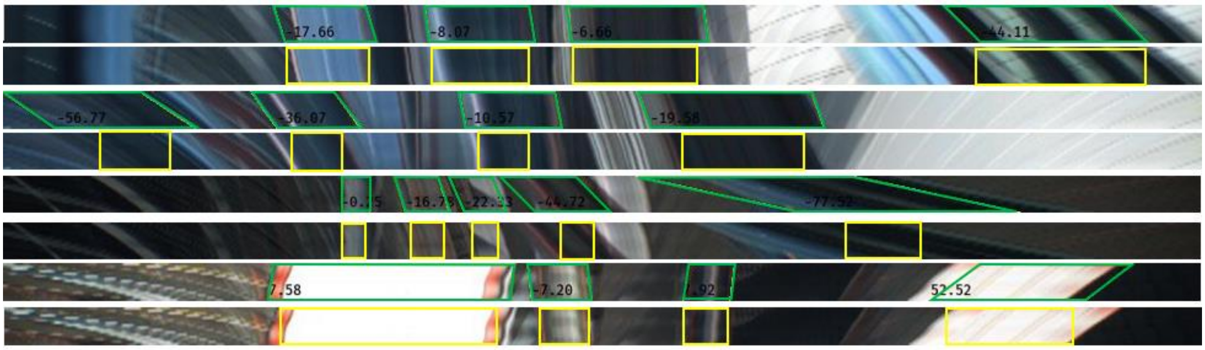

Figure 10. Comparative results of YOLOv3 and SDA-YOLOv3 architectures on visual example.

In the next study, the sigma parameter, which gives us the approach-divergence ratio of the vehicles, can be added to the training process. In another study, both horizontal and vertical movement information can be obtained by using vertical and horizontal movement profiles together. In this way, both studies can create a method that can be used in collision avoidance systems.

SUPPLEMENTARY MATERIAL

Sample video of test results can be found at the following link: https://drive.google.com/file/d/1deo4Ct_CKQuNQfgBsX7ljLMEyHa6qkM9/view?usp=sharing

ACKNOWLEDGEMENTS

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with project numbered 122E586.

REFERENCES

Behrendt, K., Novak, L., & Botros, R. (2017, May). A deep learning approach to traffic lights: Detection, tracking, and classification. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1370-1377). IEEE. https://doi.org/10.1109/icra.2017.7989163

Cadieu, C., & Olshausen, B. (2008). Learning transformational invariants from natural movies. *Advances in neural information processing systems*, *21*.

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291-7299). https://doi.org/10.1109/cvpr.2017.143

Caraffi, C., Vojíř, T., Trefný, J., Šochman, J., & Matas, J. (2012, September). A system for real-time detection and tracking of vehicles from a single car-mounted camera. In *2012 15th international IEEE conference on intelligent transportation systems* (pp. 975-982). IEEE. https://doi.org/10.1109/itsc.2012.6338748

Chen, L., Peng, X., & Ren, M. (2018). Recurrent metric networks and batch multiple hypothesis for multi-object tracking. *IEEE Access*, *7*, 3093-3105. https://doi.org/10.1109/access.2018.2889187

Gordon, D., Farhadi, A., & Fox, D. (2018). Re3: Real-time recurrent regression networks for visual tracking of generic objects. *IEEE Robotics and Automation Letters*, *3*(2), 788-795. https://doi.org/10.1109/lra.2018.2792152

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hui, J. (2018). Real-time object detection with yolo, yolov2, and now yolov3. *Available online: medium. com/@ jonathan_hui/real-time-object-detection-with-YOLO-YOLOv2-28b1b93e2088 (accessed on 24 February 2019)*. https://doi.org/10.22214/ijraset.2021.39044

Jazayeri, A., Cai, H., Zheng, J. Y., & Tuceryan, M. (2011). Vehicle detection and tracking in-car video based on motion model. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 583-595. https://doi.org/10.1109/tits.2011.2113340

John, V., & Mita, S. (2019). Vehicle semantic understanding for automated driving in multiple-lane urban roads using deep vision-based features. In *International Joint Conferences on Artificial Intelligence; Macao, China* (pp. 1-7).

Kilicarslan, M., & Temel, T. (2022). Motion-aware vehicle detection in driving videos. *Turkish Journal of Electrical Engineering and Computer Sciences*, *30*(1), 63-78. https://doi.org/10.3906/elk-2101-93

Kilicarslan, M., & Zheng, J. Y. (2018). Predict vehicle collision by TTC from motion using a single video camera. *IEEE Transactions on Intelligent Transportation Systems*, *20*(2), 522-533. https://doi.org/10.1109/tits.2018.2819827

Li, L., Zhou, Z., Wang, B., Miao, L., & Zong, H. (2020). A novel CNN-based method for accurate ship detection in HR optical remote sensing images via rotated bounding box. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(1), 686-699. https://doi.org/10.1109/tgrs.2020.2995477

Liang, Y., & Zhou, Y. (2018, October). LSTM multiple object tracker combining multiple cues. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 2351-2355). IEEE. https://doi.org/10.1109/icip.2018.8451739

Liu, Y., Lu, Y., Shi, Q., & Ding, J. (2013, December). Optical flow-based urban road vehicle tracking. In *2013 ninth international conference on computational intelligence and security* (pp. 391-395). IEEE. https://doi.org/10.1109/cis.2013.89

Muehlemann, A. (2019). TrainYourOwnYOLO: Building a Custom Object Detector from Scratch. *Disponible on-line: https://github. com/AntonMu/TrainYourOwnYOLO (Accedido Diciembre 2020)*. https://doi.org/10.5281/zenodo.5112375

Wang, L., Pham, N. T., Ng, T. T., Wang, G., Chan, K. L., & Leman, K. (2014, October). Learning deep features for multiple object tracking by using a multi-task learning strategy. In *2014 IEEE International Conference on Image Processing (ICIP)* (pp. 838-842). IEEE. https://doi.org/10.1109/icip.2014.7025168

Yun, W. J., Park, S., Kim, J., & Mohaisen, D. (2022). Self-Configurable Stabilized Real-Time Detection Learning for Autonomous Driving Applications. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/tits.2022.3211326

Zhang, D., Maei, H., Wang, X., & Wang, Y. F. (2017). Deep reinforcement learning for visual object tracking in videos. *arXiv preprint arXiv:1701.08936*. https://doi.org/10.48550/arXiv.1701.08936

Zhou, H., Ouyang, W., Cheng, J., Wang, X., & Li, H. (2018). Deep continuous conditional random fields with asymmetric inter-object constraints for online multi-object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(4), 1011-1022. https://doi.org/10.1109/tcsvt.2018.2825679