Automatic decision making system with environmental and traffic data

Sezgin Ersoy[1], Tayyab Waqar[2], Ersin Toptaş[3]
Marmara University, Technology Faculty, Mechatronic Engineering, 34722, Kadıkoy, Istanbul, Turkey
[1]Corresponding author
E-mail: [1]*sersoy@marmara.ed.tr*, [2]*tayyab.waqar@arcelik.com*, [3]*etoptas@marmara.edu.tr*

Received 30 April 2021; received in revised form 13 May 2021; accepted 24 May 2021
DOI https://doi.org/10.21595/mme.2021.22020

Abstract. By using computer vision and machine learning methods, driving lane detection and tracking, the position of the vehicles in the vicinity, their speed and direction will be determined through real-time processing of images taken from the traffic camera. Processing of the collected data using artificial intelligence and fuzzy logic and to calculate the data within the scope of "game theory" and to implement the dynamic control of the vehicle in the light of calculated data is planned. In addition to that, the designed system can also function as a driver assistant for non-autonomous vehicles with an appropriate user interface. First, the positions of the vehicles and driving lanes will be detected and monitored using computer vision and machine learning methods. Then, the vehicle speeds will be calculated by taking advantage of the historical data of the vehicle positions in the surrounding area from the previous observations, and the location estimation will be made by creating probability distributions of where each vehicle will be in the future. With the position estimation and the obtained speed information, it will be ensured that the vehicle is in the safest position in the transportation process to the destination and that it travels again at the safest speed.

Keywords: autonomous driving, traffic date, automatic decision systems.

1. Introduction

In last years, we see that switching to autonomous vehicles in vehicle technology is accelerating. Autonomous vehicles are vehicles that can sense surrounding environment and travel without any human intervention. Autonomous vehicles can sense the environment with technologies like radar, LIDAR, odometer and computer vision. When we look into history of autonomous vehicles, we see that first autonomous projects appear in 1980s. First prototype came to life with navlab and ALV projects that ran by Carnegie Mellon University. Followed by project Eureka Prometheus by Mercedes-Benz and Bundeswehr partnership at 1987. After these, lots of company manufactured many more and some of these vehicles could find a place in active traffic in some countries.

It is believed that in near future, autonomous vehicles cause an unprecedented change in economic, social and environmental areas of life. Based on these results, in this study, we aimed to build a software that can autonomously extract data related to factors of vehicle environment. In this study, we build three models that can carry out some predictions about vehicle environment in a certain domain with video footage from a vehicle centered camera as input. First model is an SVM that can detect vehicle positions on image. Second model fits a linear-parabolic function for right and left lane boundaries of the vehicle. Third model approximates a bird-view version of image and it could carry out a more realistic approximation of steer angle on bird-view image, rather than original.

Today's autonomous vehicles use the locations of other vehicles in their environment to determine their routes. This is sufficient for the desired result, but it can be made even more effective with the information about where those vehicles are and where they may be. Therefore, the usage of object detection and classification plays an important role in that regard. The aim of object detection is to successfully detect and classify the objects, which are predefined, and to specify their position in an image.

The detection, thus classifications, of objects surrounding autonomous vehicles can be done by using the standard methods which are based on feature matching techniques using sets of pre-taught features. For low level object detection and recognition, techniques i.e. SIFT [1] and SURF [2] are utilized. For high level feature extraction techniques such as Haar Features [3] and HOG [4] are found to be more useful. They have shown good performances in the detection of Human [5-7] which is the most important feature in automatic decision-making system for self-driving vehicles [8, 9] or autonomous [10-13].

The development of autonomous vehicles will depend on the detection system design [14, 15] and processor speed [16, 17], which are high-tech products. In order to achieve this development, it is also important to support sensor production and logistically. These hardware features will be used more effectively with the software. In addition to security problems [18, 19], development of production capabilities in sensors and software, another problem to be solved is energy management [20].

Our study focuses on the development of software decision-making systems in automotive vehicle technology.

2. Method

In this section, the method used for the image segmentation for the application of the proposed clustering framework is discussed. In order to analyze image meaningfully with ease, image segmentation is performed. The division or partitioning of a digital image into several sub-regions with the aim to converting the image into a representation is called image segmentation. The recognition and thus classification of resultant pixel can be labelled as clustering a problem where each pixel represents a part of an object forming the original image.

The difference between pixels can be defined by many methods i.e. by using their brightness, color, texture, edge variations or their combinations. In this research, Histogram Over Gradient (HOG) is used to find similar cluster forming an image.

Like many systems that utilizes automatic decision making, machine learning techniques were used in this project, and positive and negative sample groups were needed for training. The sample data set was created by combining the GTI and the KITTI vehicle image databases (Fig. 1).

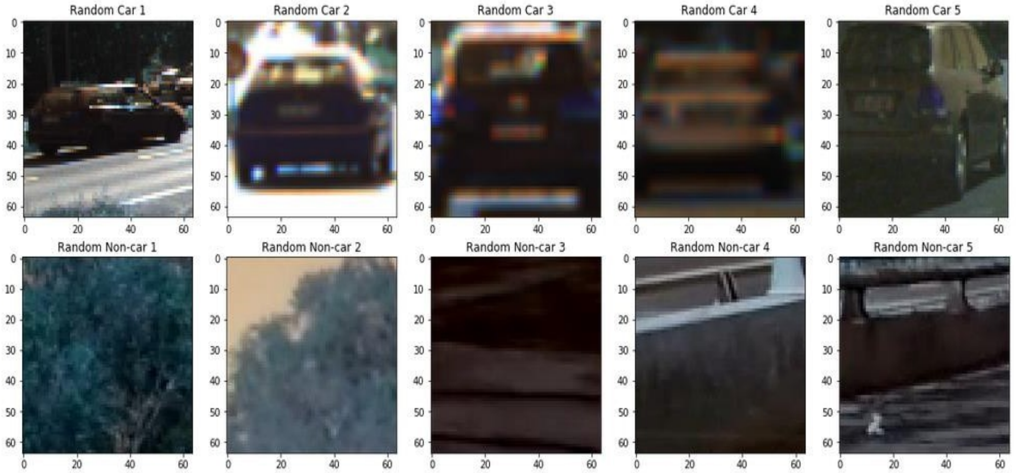


Fig. 1. Random training data samples

The data from the HOG algorithm was used in the training of the decision-making mechanism with the SVM (see 2.3.2) algorithm. The SVM model (see Fig. 2) obtained after training was subjected to standard normalization. PCA (see 2.3.3) algorithm was used as one of the ways to reduce transaction intensity. After the completion of training part, vehicle detection was performed in a 1280×720 pixel image array using the sliding window method.

In the future, it is planned to calculate the speeds of the detected vehicles and to apply the lane changing system with an appropriate lane detection system.

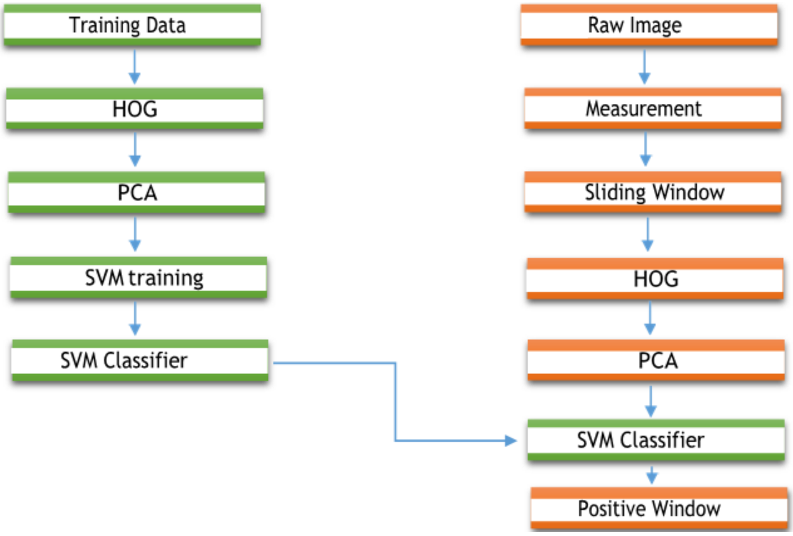


Fig. 2. SVM model

2.1. Histogram of oriented gradients (HOG)

The basic idea behind the HOG algorithm is that the object found in any region of the view can be identified by color intensity variations (derivatives) or edge detection. The image is divided into small pieces called cells, and a vector histogram is generated containing the directions of the derivatives for the pixels in each cell. To increase the accuracy, cellular histograms can be normalized by assuming a relatively larger segment of the region (see Fig. 3). In addition, normalized results give better results in shadows and reflections.

There are a few features that positively distinguish the HOG algorithm from the others. Because the algorithm operates at the cellular level, it is unregistered against geometric and photometric changes except object orientation. That is, the performance of the observer is successful in almost every case where there is no major change in the overall appearance.

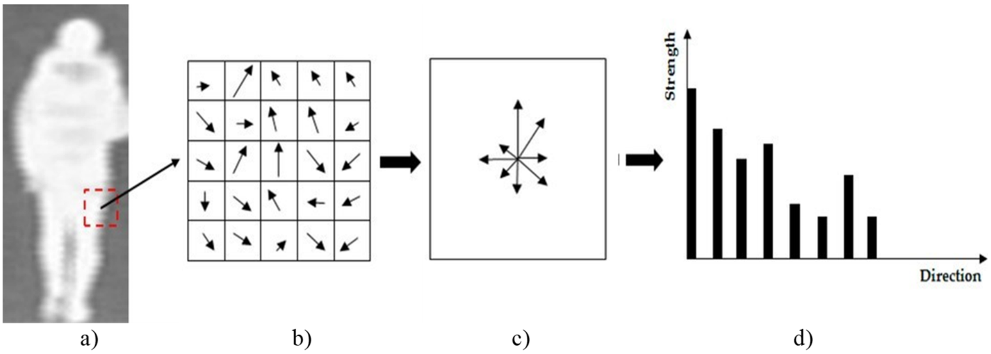


Fig. 3. A demonstration of the HOG feature extraction method: a) the input image; b) gradient map with gradient strength and direction of a sub-block of the input image; c) accumulated gradient orientation; and d) histogram of oriented gradients [14]

n the derivative calculation, the kernel that shows the most success in most applications is one-dimensional vector mask Eq. (1) and (2):

$$[-1,0,1][-1,0,1]T, \tag{1}$$

$$f(x,y) = A\exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2}\right)\right). \tag{2}$$

In addition, the Gaussian kernel (see Fig. 4) applied before the derivation calculation process provides a slight increase in performance in practice.

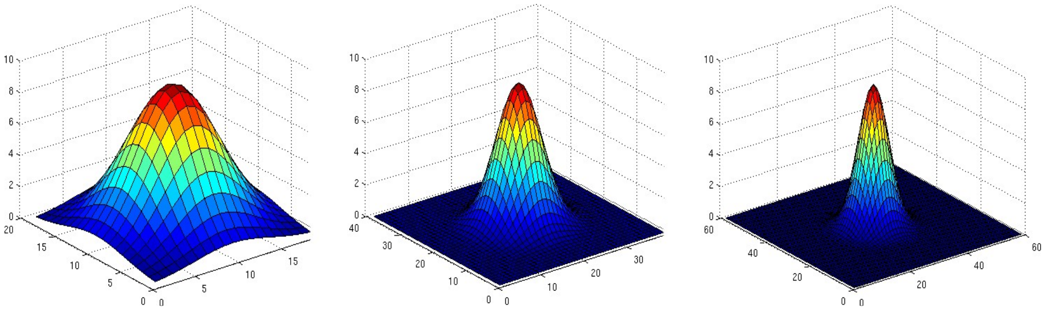


Fig. 4. Gaussian kernel visualization

The derived image is divided into cells at the next step of the application. Each pixel in the cells uses a weighted game that specifies the direction of the derivative on the histogram divided by 0-180 or 0-360. The voting weight itself can be a derivative or a function of a derivative, but usually the derivative at Eq. (3):

$$\frac{X-\mu}{\sigma}. \tag{3}$$

The blocks are then typically formed into rectangular or circular blocks. Rectangular blocks are preferred in this project. Blocks can be on top of each other, so a cell can provide data to more than one block. The data passes through a separate normalization process after being collected in blocks. L1-norm is preferred for this application (see Eq. (4)):

$$\text{L1-norm:} f = \frac{v}{(\|\vartheta\|_1 + e)}. \tag{4}$$

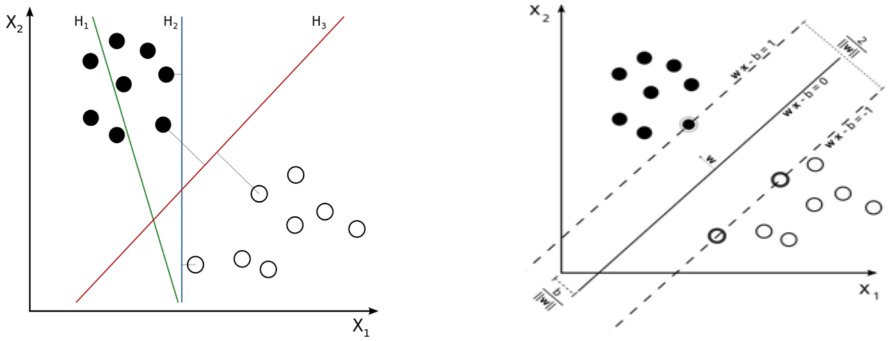
2.2. Support vector machines (SVM)

The SVM core function is very similar to the logistic regression, that is to divide the data into groups, dividing a p-dimensional data group with a $(p-1)$-dimensional hyperplane. But the most important feature that differentiates the SVM from the logistic regression and puts it ahead of practice in performance is that SVM tries to maximize the gap between different data groups (see Fig. 5(a)). This approach offers a much more distinctive distinction between data groups.

Once collected, the histograms separated by the greatest difference from the others should be locally normed so that they do not affect the final result alone. It is better to use standard deviation instead of arithmetic average for normalization itself is the most accurate result.

SVM minimizes the error function shown below to ensure that we obtain the necessary parameters and obtain a hyperplane that separates the data into groups (see Fig. 5(b)).

However, with this method, points that are far from where it should be at hyperplanes can affect the end result. To avoid this, a one-dimensional Gaussian kernel centered on the hyperplane is used. Weights are obtained for each data point and the effect of distant points is reduced compared to the nearest point.

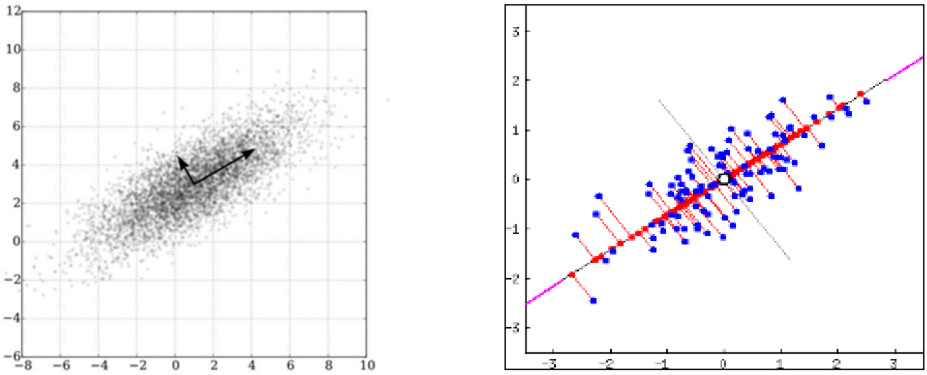


a) b)

Fig. 5. a) SVM convergence, b) maximum margin and support vectors

2.3. Principal component analysis (PCA)

PCA can be thought of as fitting n-dimensional ellipsoid into data. Each axis length of ellipsoid represents a principal component. If an axis length of ellipsoid is short, the data point distribution along that axis is too small, and if the short axis is removed from the data we lose very little data (see Fig. 6(a)).



a) b)

Fig. 6. a) PCA of a multivariate Gaussian distribution, b) PCA implementation

Following the PCA implementation, the data points are reduced to projection onto the existing self-vectors (see Fig. 6(b)). In the end, variables that have almost no effect are removed from the data group.

In order to find the ellipsoid axis lengths, we first have to calculate the arithmetic mean of each variable from the data set. As a result of this process, the data group will be positioned around the origin. Then, the covariance matrix of the data is found, and the eigenvectors of the covariance matrix and the eigenvectors corresponding to the eigenvalues are computed. Then the eigen-vectors are normalized to be reduced by the unit vectors.

As a result, the data diversity that each eigen-vector represents can be calculated by dividing the corresponding eigen-value by the sum of all eigen-values.

2.4. Deep feedforward network

A different and more complex artificial neural network approach has also been dealt with because the optimization algorithm described at uses a small amount of function space that can be transformed if the data is used without pre-processing and the pre-processing takes a long time.

A neuron (unit/element) of an artificial neural network is basically composed of two functions. The functions should be selected as linear in the first and non-linear (activation) in the second. A neuron mathematically can be written as follows, where h is the output value of the neuron, g is the nonlinear function, x is the input value of the neuron, W is the matrix of the weight matrix and b is the matrix of constants. The formula is seen at Eq. (5):

$$h = g(W^T x + b). \tag{5}$$

One of the most important features that distinguish artificial neural networks from classical machine learning methods is that the second part of neurons, that is, the activation parts, can be transformed into elements of a large function space as the model increases in depth. Without the activation function, the model is reduced to a series of linear functions, and the weight matrices of these inline linear functions can be written as a single matrix with the appropriate multiplication, indicating that the entire model can be reduced to a single linear combination without activation functions.

In real world applications, it is rarely possible for a given task to have only linear connections between input and output values, or a small number of nonlinear relations. Because of this and the constant increase in the power of data processing, deep learning methods and artificial neural networks are finding themselves in the spotlight day by day. And a deep learning model has also been added to this project, although results obtained cannot be labelled as successful.

The choice of non-linear functions in the applications is the biggest factor affecting the learning speed and the overall performance of the model. The ReLU (rectified linear unit) function (see Fig. 7) was chosen as the activation function in the hidden layers of the model designed in this project and the sigmoid activation function was used because the output layer showed the desired Bernoulli distribution.

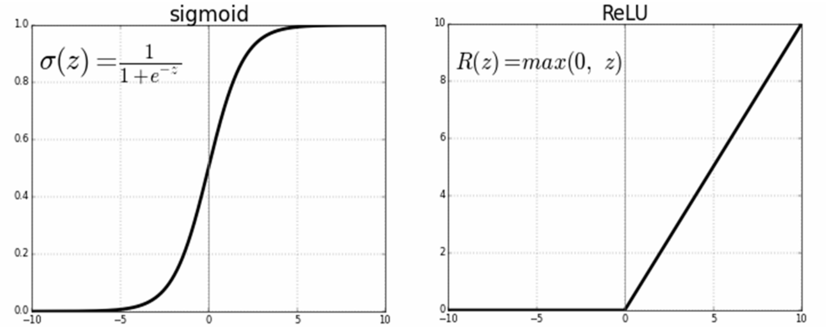


Fig. 7. Sigmoid and ReLU functions

In order to optimize the model, forward propagation and back propagation algorithms should be performed. Forward propagation is the calculation of the output value and the loss function of the model according to the input values from the beginning to the end of the model at Eq. (6):

$$L(y,\dot{y}) = -y\log\dot{y}' - (1-y)\log(1-\dot{y}), \tag{6}$$

as described above. "Cross-entropy loss" is used as the loss function.

The backpropagation algorithm is simply a derivation of the loss function with respect to all weight matrices. Contrary to forward propagation, back propagation is performed from the output units to the input units. Because the chain of calculus requires it.

After the derivative calculation, each element of the weight matrices is updated in proportion to its derivative at Eq. (7):

$$W_{ij} = W_{ij} + \Delta W_{ij}, \quad \Delta W_{ij} = -\eta \frac{\partial E}{\partial \Delta W_{ij}}. \tag{7}$$

This process is performed one or more times at each pass over the data set. As a result, the extreme point of the loss function is determined.

3. Results

The general representation of the methods used up till now in this project has been made up to this division. The solutions to the following problems will be reviewed.

Firstly, SVM was trained with various color spaces in order to select the color space that is most compatible with the HOG algorithm and data set. 90 % of the data group for training and 10 % for the lead test were allocated. In the pioneer tests, the color space was "YCbCr" with accuracy between 94-97 %, which provides the highest accuracy (see Fig. 8). However, in the tests performed on the 1280×720-pixel image array, the actual power required for the real-time operation of the application was reached as the result of exceeding the capacity of the test device. In this phase, the PCA algorithm was included in the project. PCA was used in all channels of color space with 5292 variable numbers indexed to 3000. A natural increase in the training period and a 15-20 % decrease in the test duration were observed. However, despite the decrease in test time, vehicle detection was not even close to real-time. Processes performed at 3.1-3.4 s/frame prior to PCA could only reach 2.5-2.8 s/frame later. The SVM was abandoned for accuracy and the grayscale was selected as a color space with a range of 91-95 %. The data expressed in three different channels with the greyscale color space is reduced to a single intensity value. In 64×64 pixel images, the number of variables went down to 1764. It was then pulled to 1000 by applying PCA and processed at 1.15-1.35 s/frame. 5×5 to further reduce the time. According to Gaussian kernel, the entire data set was blurred prior to training and the SVM model was re-trained. An additional 5-10 % efficiency increase in processing speed was achieved without a statistically significant decrease in test accuracy. As a result, the processing speed has been increased to 1.0-1.3 s/frame.

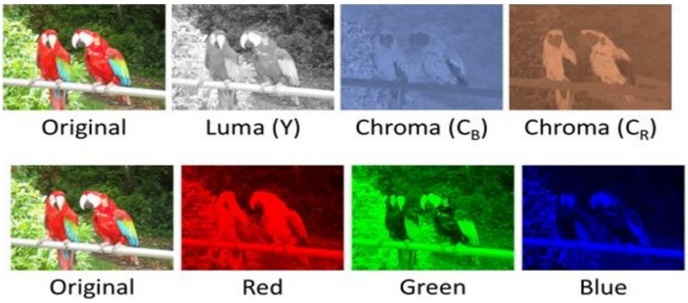


Fig. 8. Comparison of different color spaces

It is noted that in order to achieve the speed required for real-time study, the use of different architecture at processor level as well as by increasing process power the researches will be able to find solutions that will be much more economical. Different combinations of directions and cell numbers per block were tried to determine how the HOG algorithm's parameters affected the test accuracy. The most successful results were obtained with 9 directions and 4 cell per block.

4. Discussion

One of the biggest challenges in the lane detection and tracking model is that the model is vulnerable to temporary or instantaneous changes in the image. Accordingly, while determining the area of interest for the next frame, the model's resistance to changes was increased by adding a temporal filter application for the area of interest pixels. However, when the tests showed that the model resistance was not sufficient, the same application was applied to the central strip boundary orientations in addition to the temporal filter added to the pixels of interest and the changes that occurred more than 2 degrees per frame in a 25 fps image flow were ignored. In the tests performed after these two applications, it was found that the dynamic structure of the model was weakened, and it made more stable and consistent predictions. Another thing about the same model is that it is affected by color changes. Changes in the color of model road asphalt are affected by the shadows of trees and other vehicles (especially long and wide) on the side of the road. This effect affects the parabolic part of the approached part function rather than the linear part. In the model which makes the rotation angle estimation, a problem similar to the lane detection and tracking model is encountered. Considering the similarity of the techniques used by the models this can be said to be expected. The model is affected by road asphalt color changes and vehicle and tree shadows just like the other one. If the same solution is applied for this model, the error rate can be reduced in the dynamic regions of the road. Another aspect of this model is the increase in the error rate in the parts where the road slope changes.

The model uses the reverse perspective mapping method, which takes camera parameters as input. Although the position of the camera in space coordinates does not change, one of its parameters, lost point position, causes distortion in model estimation with the change of slope of the path. In order to prevent this situation, road slope changes can be estimated or determined by the image processing methods that we are interested in during our study, or with the help of various sensors that can be placed on the vehicle. After slope information, a separate model can be developed that can adapt to road slope changes for lost point estimation and the error rate can be greatly reduced.

5. Conclusions

The study is an autonomous driving support for improving artificial intelligence assisted driving behaviors and reducing human-related error parameters in general frame. This system consists of a decision-making algorithm based on game theory. The learned machine can be developed more rationally and reliably with logic and improved software and hardware support. With a communicative database to be created with this system, experience transfer of other tools can be provided and commercialized.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2, pp. 1150–1157, 1999, https://doi.org/10.1109/iccv.1999.790410

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science*, pp. 404–417, 2006, https://doi.org/10.1007/11744023_32

[3] D. G. Low, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, Nov. 2004, https://doi.org/10.1023/b:visi.0000029664.99615.94

[4] Z. Huijuan and H. Qiong, "Fast image matching based-on improved SURF algorithm," in *2011 International Conference on Electronics, Communications and Control (ICECC)*, pp. 1460–1463, Sep. 2011, https://doi.org/10.1109/icecc.2011.6066546

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, https://doi.org/10.1109/cvpr.2001.990517

[6] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 886–893, 2005, https://doi.org/10.1109/cvpr.2005.177

[7] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian Detection using Infrared images and Histograms of Oriented Gradients," in *2006 IEEE Intelligent Vehicles Symposium*, pp. 206–212, 2006, https://doi.org/10.1109/ivs.2006.1689629

[8] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5636–5643, May 2014, https://doi.org/10.1109/icra.2014.6907688

[9] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pp. 3838–3843, Sep. 2011, https://doi.org/10.1109/iros.2011.6095074

[10] M. Fu, J. Ni, X. Li, and J. Hu, "Path Tracking for Autonomous Race Car Based on G-G Diagram," *International Journal of Automotive Technology*, Vol. 19, No. 4, pp. 659–668, Jun. 2018, https://doi.org/10.1007/s12239-018-0063-7

[11] K. Y. Lee, G. Y. Song, J. M. Park, and J. W. Lee, "Stereo vision enabling fast estimation of free space on traffic roads for autonomous navigation," *International Journal of Automotive Technology*, Vol. 16, No. 1, pp. 107–115, Feb. 2015, https://doi.org/10.1007/s12239-015-0012-7

[12] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Engineering Applications of Artificial Intelligence*, Vol. 60, pp. 97–116, Apr. 2017, https://doi.org/10.1016/j.engappai.2017.01.013

[13] A. Vuckovic, D. Popovic, and V. Radivojevic, "Artificial neural network for detecting drowsiness from EEG recordings," in *2002 6th Seminar on Neural Network Applications in Electrical Engineering. NEUREL 2002*, pp. 155-158, 2002, https://doi.org/10.1109/neurel.2002.1057990

[14] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transportation Research Part C: Emerging Technologies*, Vol. 89, pp. 384–406, Apr. 2018, https://doi.org/10.1016/j.trc.2018.02.012

[15] İ. Ertuğrul and O. Ülkir, "Analysis of MEMS-IMU Navigation System Used in Autonomous Vehicle," *Autonomous Vehicle and Smart Traffic*, Intechopen Pub., 2020.

[16] O. Ulkir, M. Akın, and S. Ersoy, "Development of vehicle tracking system with low-cost wireless method," *Journal of Mechatronics and Artificial Intelligence in Engineering*, Vol. 1, No. 1, pp. 8–13, Jun. 2020, https://doi.org/10.21595/jmai.2020.21471

[17] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Communications*, Vol. 12, No. 1, Feb. 2021, https://doi.org/10.1038/s41467-021-21007-8

[18] G. Li et al., "Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios," *Transportation Research Part C: Emerging Technologies*, Vol. 122, p. 102820, Jan. 2021, https://doi.org/10.1016/j.trc.2020.102820

[19] G. S. Nair and C. R. Bhat, "Sharing the road with autonomous vehicles: Perceived safety and regulatory preferences," *Transportation Research Part C: Emerging Technologies*, Vol. 122, p. 102885, Jan. 2021, https://doi.org/10.1016/j.trc.2020.102885

[20] C. Akay, U. Bavuk, A. Tunçdamar, and M. Özer, "Coilgun design and evaluation without capacitor," *Journal of Mechatronics and Artificial Intelligence in Engineering*, Vol. 1, No. 2, pp. 53–62, Nov. 2020, https://doi.org/10.21595/jmai.2020.21627

[21] D. Nguyen and K. Park, "Enhanced Gender Recognition System Using an Improved Histogram of Oriented Gradient (HOG) Feature from Quality Assessment of Visible Light and Thermal Images of the Human Body," *Sensors*, Vol. 16, No. 7, p. 1134, Jul. 2016, https://doi.org/10.3390/s16071134

Sezgin Ersoy is an Associate Professor of Mechatronics Engineering and Material Science. After graduating from Marmara University, he became a faculty member at the same university. His publications include a variety of efforts to understand changes in automotive mechatronics, polymer science and biomedical technologies. He is the editor and author of books. He has two science national awards and is an Editorial Member of several scientific journals.

Tayyab Waqar is currently pursuing his Ph.D. at Marmara University, Turkey, in the Mechatronics Engineering Department. His Ph.D. work is based on the designing, characterization, and fabrication studies of surface acoustic wave-based sensors. At the same time, he is also working as an R&D specialist at Arçelik/Beko group in the Sensor Technologies/Mechatronics R&D Department.

Ersin Toptas received Ph.D. degree in Institute of Pure and Applied Sciences from Marmara University, Istanbul, Turkey, in 2015. Now he is with the Faculty of Technology, Marmara University. His current research interests are mechanical design, computer aided analysis, SMA and composites.