MINISTERUL EDUCAȚIEI NAȚIONALE

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTY OF AUTOMATION AND COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTMENT

Eng. Răzvan ITU

PhD THESIS

ABSTRACT

Monocular perception system using artificial vision and intelligence

PhD supervisor:
Prof.dr.ing. Radu DĂNESCU

Str. G. Barițiu 26-28, 400027 Cluj-Napoca, România
Tel. +40-264-401220, fax +40-264-599893
www.cs.utcluj.ro

Introduction, context and motivation

The development of modern advanced driver assistance systems (ADAS) has grown in the latest years due to the recent technological advancements. ADAS have been developed even as early as the 1960s when they were very rudimentary. Sistems based on perception of the scene using computer vision have been developed since the 1980s and especially in the 1990s. The main interest of these systems is to increase the overall safety of all the participants in road trafic by reducing the number of accidents. The main active and passive safety systems for vehicles include: safety belts, trajectory or lane keeping systems, forward collision systems. These are based on different types of sensors that are used to measure and to perceive the surrounding scene: video cameras, laser or lidar sensors. ADAS using cameras have the advantage of being cost efficient and due to the dense data that cameras provide. The disadvantage is given by the need of a clear and unobstructed view of the observed scene, but perception solely on cameras can also be influenced by adverse weather conditions. However, the main advantage is that adding a video camera assistance system can be done much more easily than other sensors, meaning that existing and older vehicles can be improved. One of the easiest ways of to add a perception system to a vehicle is through the use of mobile platforms that can be equipped with additions sensors, such as mobile phones or tables.

Smart mobile devices are omnipresent and they come equipped with one or more cameras at different resolutions, but also with increasing processing power. Also, modern mobile devices feature a variety of additions sensors such as: satellite positioning sensors (GPS, GLONASS), accelerometer, gyroscope, geomagnetic sensor, compass, which means that they can be used for driver assistance systems or as systems for road traffic analysis or monitoring. However, the analysis or perception of the surrounding scene will require a calibrated camera in order to compare the features of the 3D scene of the world and the 2D features in the camera image. This calibration step is often neglected or even omitted by the average end-user.

Objectives

The main objective of the thesis is to design and implement a road traffic perception system based on monocular vision and additional sensors, that is capable of running in various lightning conditions, in motorway scenarios and especially in urban traffic scenarios.

To achieve the main goal, I have identified a number of secondary objectives:

1. creating an image and sensor data acquisition and processing system
2. obstacle detection:
    - identifying the presence of obstacles in the image space using semantic segmentation with convolutional neural networks
    - extract 3D information from the monocular image by analysing the image with the perspective effect removed (the bird's eye view image)
3. tracking and estimating obstacle position by using 3D models:
    - using occupancy grid based particle filters for tracking
    - tracking at a cuboid level
4. automatic camera calibration

- initial automatic calibration
- dynamic adjustment of the camera parameters

A modern driver assistance system requires a very good perception of the scene. I chose to use a single camera that is mounted in the vehicle behind the windscreen. The advantage of a mobile device (phone or tablet) for capturing images is that it offers good image resolution, but also a wide range of additional sensors, all integrated into a small size device that can be easily positioned, maneuvered and installed. The acquisition system consists of accessing the live image feed from the mobile device's camera and by using the available sensors (GPS, accelerometer, gyroscope, etc.) and by processing them directly on the device, or by saving the data and sending it to a faster and more efficient processing system. In order to use cameras from different manufacturers in different vehicles, a more efficient and less elaborate calibration methodology is needed. I initially developed a manual calibration system that requires minimal user intervention and input, and then I introduced various automatic camera calibration approaches where no user action is required. Thus, a system that is capable of self-calibration is more portable and can be easily installed and retro-fitted in old vehicles, which usually do not have any kind of driver assistance systems.

Semantic segmentation of a traffic scene means interpreting the image from a monocular camera based system and understanding what it contains. For this objective, I chose to use processing techniques based on artificial intelligence, more specifically I have use a convolutional neural network capable of providing predictions of the driveable road area surface from images. This approach is robust and invariant to the scale of the images and provides very good results, requiring only a step of resizing the input images (to the size with which the original network was trained). I chose to detect the road surface, the free space on which the vehicle can circulate safely, because it can be assumed that everything that is not driveable (free space) can be considered an obstacle.

Areas that are not part of the road will be analysed and tracked over time using 3D perception models based on particle filters. A scene occupancy grid is built based on a particle filter, where each obstacle in the scene is represented by a set of particles. This is a dynamic representation model because each particle has a component for speed estimate, one for the position in the occupancy map, and one for the "age" from the moment the particle is created. This information is used to predict and update the particle grid when certain particles that are possible objects are added or removed.

By integrating all components into a single project, I have succeeded in implementing a monocular perception system. The system developed and presented in this thesis is based on ideas and algorithms that have been published in journals, book chapters and specialized conferences.

Thesis structure

In this doctoral thesis I have implemented a perception and obstacle tracking system in road traffic using a monocular camera. The thesis is structured in four chapters: the first contains an introduction to the field and the need for driver assistance systems and the objectives of this thesis, the second chapter presents methods for detecting and tracking obstacles, and the third chapter describes the camera calibration techniques. The last chapter of the thesis refers to conclusions and further developments, followed by bibliography and annexes.

The first chapter describes statistic of traffic accidents in the recent years, from which results a clear need to have more intelligent and advanced vehicles on the road. I presented a brief history of intelligent vehicles: the first ones dating from the 1960s that used rudimentary techniques to maintain their lane, continuing with those of the late 1980s and 1990s that were already using artificial vision using cameras and computers. In the 2000s there were major leaps in this field that were mainly driven by the DARPA competition, and after 2010 vehicles with a higher level of automation have already been launched. The current international standards describe a total of 5 degrees of automation, and the sensors used to get traffic data are also presented in the first chapter. The proposed objectives, the structure of the thesis and the acknowledgments are presented in sections 1.2-1.4.

Chapter 2 describes methods for detecting and tracking obstacles. I have presented approaches for tracking obstacles using probability, where several hypotheses about the current position or status of obstacles or vehicles are maintained, and each hypothesis has a different probability. I presented the Bayes filter and then other approaches based on it: Kalman filter, expanded Kalman filter and particle filter (useful for Gaussian models). Sections 2.5 and 2.6 represent an introduction to artificial intelligence and artificial neural networks based on a model inspired by the biological neuron. I presented and exemplified the convolution operation and in Section 2.7 I described the difference between classification and regression and then in section 2.8 I presented different types of activation functions that can be used for the two types of learning and also for semantic segmentation. Section 2.9 contains a summary of the backpropagation algorithm and the "gradient descent" technique and section 2.10 describes the cost functions for neural networks. The semantic segmentation using neural networks is described in 2.11, while in Section 2.12 I introduced techniques for monocular perception. The first part (2.12.1) is a study of current solutions based on smart mobile devices. In 2.12.2 I described the techniques used to estimate depth in images captured using a single camera and then I presented a technique that I used in other papers to detect obstacles in monocular images where the perspective effect is eliminated. The main contributions are described starting with section 2.13, where I first presented the existing traffic image databases (2.13.1) that I used to develop a complex traffic perception system. My own software for acquiring new images is described in section 2.13.2. The following subchapter (2.13.3) broadly describes a neural network that I used to segment road traffic images to extract only the road area from an image. The obtained results are very good, even on my own dataset (which was not used at all during training). On the validation datasets from the other databases the results are good even if the evaluation metric is lower than other similar approaches. By analysing the images where the segmentation results were not as expected I found that there are some errors in the labelling of the existing databases, but the network that I have developed actually offered good predictions of the driveable road area. There were also the opposite situations, where the network predictions were worse than the labelling, and all of these situations are described and illustrated in the thesis. In section 2.13.4 I have described the particle filter based tracking algorithm that uses a dynamic occupancy map. This approach is not an original one and has already been published, but the way I have extended it and how

I built the measurement part of the filter is unique. The measurement map is used to generate new particles that are grouped into cells from which cuboids can be extracted. The cuboids can represent the obstacles detected and tracked in the scene. This measurement map is in fact a binary map built from the driveable road image segmented by the convoluted neural network. By having new measurements in each frame, the particles can be created, moved or destroyed, therefore the obstacles can be tracked. The uncertainty of the measurement is determined by the distance to the obstacles that can be computed in the IPM image. In the final step, the particles are grouped according to their similarities (proximity, velocity vectors) and then cuboids are extracted from them. Another contribution to the work that was published in the past is given by the refinement of the occupancy map by segmented image processing: I have developed an algorithm to join (combine) the histogram peaks that belong to an obstacle. The effect on particle formation is well illustrated and exemplified in subchapter 2.13.5 (a). Another contribution is the refining of camera parameters by generating IPM images with different pitch angles and then by comparing occupancy maps with the current particle filter. In Section 2.13.6 I have described a method to estimate the local orientation of objects and their dimensions using a convolutional neural network. This implementation is not unique, but the contribution consists in integrating it with the particle filter based perception system. Thus, the orientation and size of the cuboids extracted from the particle filter are adjusted using the CNN. Further details on the implementation of the technologies used and the architecture of the system are presented in section 2.13.7. Chapter 2 ends with conclusions and a list of published papers.

In chapter 3 I have described camera calibration techniques. In the first subchapter (2.1) I presented the camera model and then intrinsic parameters (2.2) and extrinsic parameters (2.3). The main goal behind the calibration is to determine the camera parameters in order to compute the projection matrix, which can be used to generate images with the perspective effect removed, a top view of the road scene (“bird’s eye view”). In section 3.4, I conducted a study of current of camera calibration techniques, and in section 3.5 I presented methods to automatically calculate the focal length. Section 3.6 contains IPM imaging algorithms, whereas in section 3.7 I presented an intuitive interface for assisting end users for manual calibration of extrinsic parameters. Another contribution is given by the implementation of a self-calibration algorithm to determine the translation vector parameters by analysing a sequence "offline" frame by frame. The height of the camera from the ground is adjusted so that the lane width detected in the IPM image corresponds to the pixel value corresponding to a standard width of 3.5 meters. In this section I also briefly presented how to determine the lane width from traffic images. In section 3.9 I have described techniques for determining the vanishing point from images captured in road traffic. I presented existing approaches and then my own algorithm, based on image processing which analyses the orientation and magnitude of edge points belonging road markings. Using a voting system, I create a vote map for the vanishing point. The results were good enough to create a database of images and vanishing points that I then used to train a convolutional neural network that provides predictions of the image x and y coordinates of the vanishing point (section 3.10). The method was published at an international conference. I have used several techniques to augment the database and then I have extended this approach by using a network for the prediction of the x coordinate, and the same network for the y axis coordinate. The results from the dual CNN approach are better than the initial version (section 3.11). I also implemented and tested this solution on Android mobile devices (3.11.4). In section 3.12 I described a camera self-calibration solution. From the analysis of vehicle trajectories, I determined the height of the camera from the ground and the pitch angle using an EKF filter. I also determined the vanishing point, which is used to adjust the IPM image. The whole solution was published at an international conference and implemented on both desktop and Android mobile devices. Chapter 3 ends with conclusions and contributions and a list of published papers.

Contributions

The main original contributions described in chapter two are: creating a proprietary data acquisition system using the sensors available on a smart mobile device (camera, satellite position sensors, accelerometer, gyroscope, etc.), creating my own databases and a modality to view and process them on desktop systems, and the implementation of a road traffic perception and tracking system based on a particle filter and convolutional neural networks.

The device's main camera (generally the rear facing one) is used to capture images, as these cameras provide a better sensor. The acquisition system will also store additional information during the vehicle movement: acceleration and travel speed data, position (expressed in latitude and longitude coordinates), magnetic orientation (from the magnetometer) and the local orientation angles of the device (obtained from the gyroscope). All this information is stored in the internal memory and synchronized with the images using timestamps.

I have developed a monocular camera vision based system, combining convolutional neural networks to detect objects in the scene and then using particle filters to track them. The first step is to implement an artificial network for semantic segmentation that based on the U-NET architecture [Ronneberger2015]. The modified U-Net convolutional network has the following structure: 5 encoding layers, one central layer, and 5 decoding layers. After the training, the network features excellent performance of predictions in complex traffic scenarios where there are many vehicles in the scene (figure 1). This precise delimitation of what is driveable road surface and what is not, is very useful for creating a monocular perception system where we do not have depth information or other information of the scene.

Figure 1. Example of semantic segmentation using the artificial network.

The segmented images are integrated into a probabilistic framework based on a dynamic map of the scene in front of the vehicle. From the detection of the free space area, we can assume that everything that is not a road is a possible obstacle and it can be interpreted and tracked. Therefore, this solution is useful for detecting any type of obstacle and is not limited to just pedestrians or vehicles. To implement this functionality, I chose to use an occupancy grid based particle filter. The binary occupation map is generated from the CNN segmented image in which the perspective effect (IPM) is eliminated. This image is then analysed using an algorithm that scans the front of the vehicle using the rays originated at the point where the camera is mounted. Objects in IPM images will be feature radial distortions, therefore this property can be used.

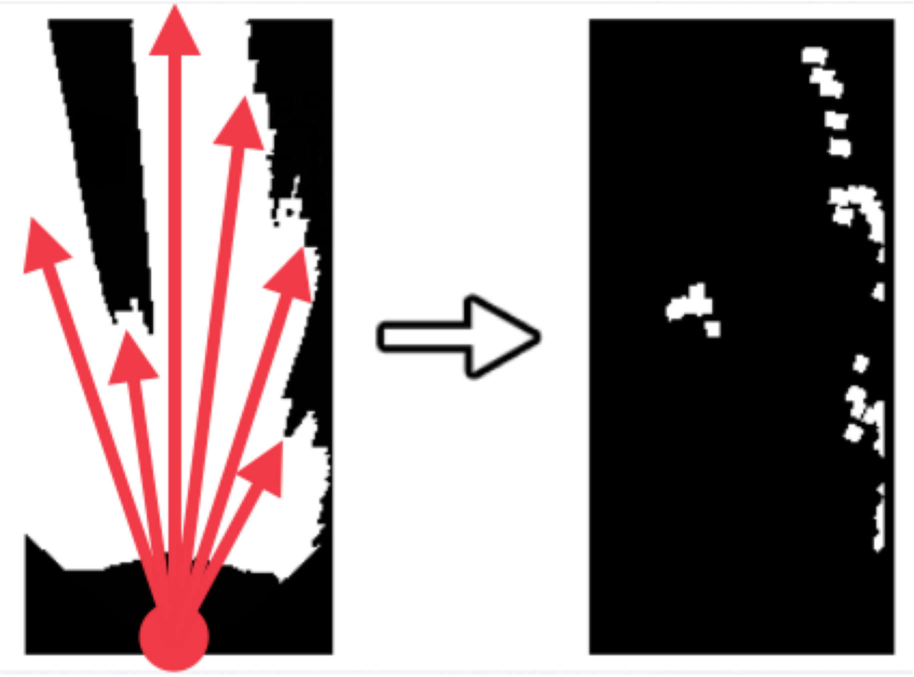

Figure 2. Binary occupancy map (right), generated from the segmented road image (left). Illustrated on the left are part of the rays along which intensity transitions are searched.

The particle filter uses a dynamic occupation map, which is not an original implementation (it was originally published in [Dănescu2011]), but it has been extended (the occupancy map is of a double size to allow tracking and obstacles leaving the visible area of scenes). I have also adapted and enhanced for the occupancy grid tracker in order to use it with monocular images. I have also refined the obstacle detection by analysing each obstacle's radial histogram and also by generating more IPM images in order to refine the camera parameters. The first step of the tracking algorithm is the prediction and is applied to each particle. The position of the particles in the set is modified according to their speed (they move in accordance with their own velocity vector) and depending on the movement parameters of the ego-vehicle (the velocity and the "yaw rate") read from the sensors of the vehicle or mobile device: GPS, accelerometer and gyroscope. The second step of the algorithm is the update process, more specifically the processing of the measured information, and is based on the binary occupancy map of the cells created from the processed IPM image. Measurement information is used to weigh the particles and then to resample them in one step. By weighting and resampling, the particles in a cell can be either multiplied or destroyed.

The final step is to calculate the speed and then estimate the probability of a cell being occupied by an obstacle.

The individual cells of the occupancy map contain particles that can be grouped according to their similarities, so that cuboids can be extracted from them. The grouping algorithm takes into account the proximity of cells, their velocity vectors, and their displacement to extract the connected regions that will represent an object. Finally, a rectangular shape is extracted from these connected areas and the 3D cuboid will be built (the cuboid height is fixed fixed with 1.5 meters for all objects).

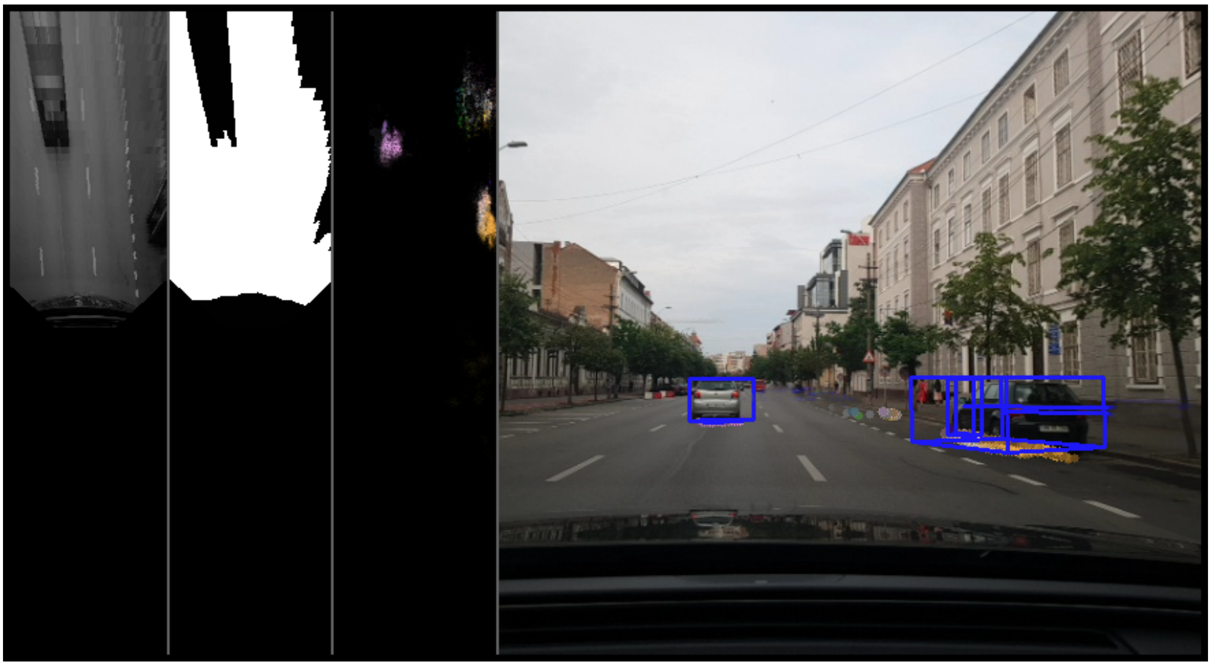

Figure 3. The scene perception probabilistic framework.

Figure 3 illustrates the processing steps within the probabilistic framework. In the left image we can see the processing steps and the creation of the dynamic occupancy map: the first image represents the scene with the perspective effect eliminated (IPM), the second image is the IPM segmented image (obtained from the convolutional neural network), the third image illustrates the particles with their color coded by the velocity vectors (as in [Dănescu2011]: the hue of the color represents the direction of travel, the saturation is the magnitude of the velocity and the intensity encodes the occupancy probability). The image on the right shows the result of the processing and grouping of particle cells in obstacles. The particles are also projected and illustrated in the initial perspective image of the scene.

A contribution and improvement to the work published in [Danescu2016] is the pre-processing of the measurement image. To improve the vehicle detection in the segmented image I have used a technique based on some ideas from the [Bertozzi1998] paper.

Obstacles only touch the asphalt with the wheels and this is amplified in IPM images, resulting in a false perception of the distance to obstacles (especially in the distant ones). I have solved this problem by analysing the radial histogram of each vehicle and by joining the peaks by filling the empty area between the wheels of a vehicle. The histogram peaks are identified by calculating local maxima and then they are processed to identify pairs of lines describing the edges of an object. Having these pairs of lines corresponding to an object, the next step is to fill the area between the peaks. This operation has the role of significantly improving the measurement process in the particle filter and the occupancy map, because determining the distance from the ego-vehicle to the other vehicles is important. The effect of these operations is illustrated in the following figure (Figure 4), where a significant improvement in how the particles are generated in the occupation map (they will form an object similar to the real one) can be observed.

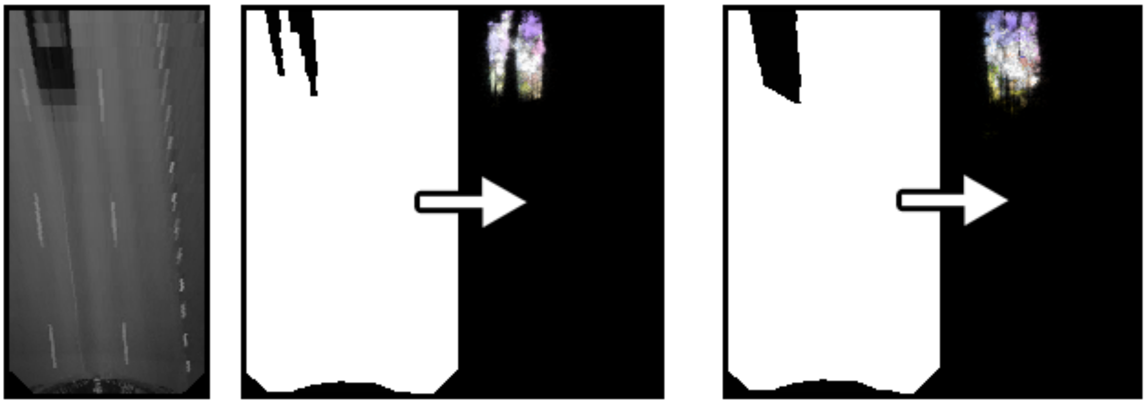

Figure 4. Improvement of the dynamic occupancy grid.

Another contribution is the refinement of the camera parameters. Extrinsic parameters relating to pitch and yaw angle can be corrected and refined during the processing of each image. For each image in a sequence, a total number of *N* perspective images (IPMs) are generated, each having a different pitch angle. Similarly, I also generate *N* IPM images having different *N* yaw angles. This step is applied only after the camera has been self-calibrated. In the monocular perception system, we generated *N=10* IPM images using variations of 0.1 degrees around the pitch angle that was calculated using the extended Kalman filter (section 3.12.2). In the 10 IPM images generated, the obstacles in the scene are then detected using the algorithm described in section 3.6. These binary images are compared to the dynamic binary occupancy grid (generated from CNN segmented image) and then I compute the percentage of pixels that overlap between them. The image with the highest percentage larger than a fixed threshold will represent a better fit of the data from the observed scene, which means that the pitch angle can be adjusted to increase the accuracy and robustness of the system. The same technique was applied for the yaw angle. The particle filter occupancy grid is used to validate and adjust the parameters as it will estimate the movement of the vehicles in the scene as close as possible to the reality and will be less influenced by the camera's tremors and small movements while driving.

I've also implemented my own version of an existing [Mousavian2017] network to provide predictions about the size of the objects detected in the scene and also predict their orientation. This information is merged with the particle filter to improve the robustness and accuracy of the system and is an important contribution to the monocular perception system. To determine the size and orientation of vehicles, I chose to use a network similar to [Mousavian2017] with small changes to the cost function presented in the article. The convolutional neural network has the first 5 layers the same as the VGG16 [Simonyan2014] network. I used the transfer learning technique to initialize the weights of the first 5 layers with those of the VGG16 network trained to classify objects in the images. The output of the last convolution is then used for regression of dimensions, orientation, and probability of orientation. Estimating 3D dimensions and object orientation is accomplished by expanding an object detection network that provides rectangular bounding boxes for each detected obstacle. The main constraint is that the three-dimensional object in the scene is inside the 2D rectangle in the image, so its 3D projection will be within the same bounding box obtained from the object detector. The CNNs used for object detection is outlined in Section 3.12.1 (SSD MobileNet [Wei2016], [Howard2017]). Figure 5 shows the vehicle detection results using the MobileNet SSD, whereas Figure 6 presents the orientation and size prediction for each vehicle.

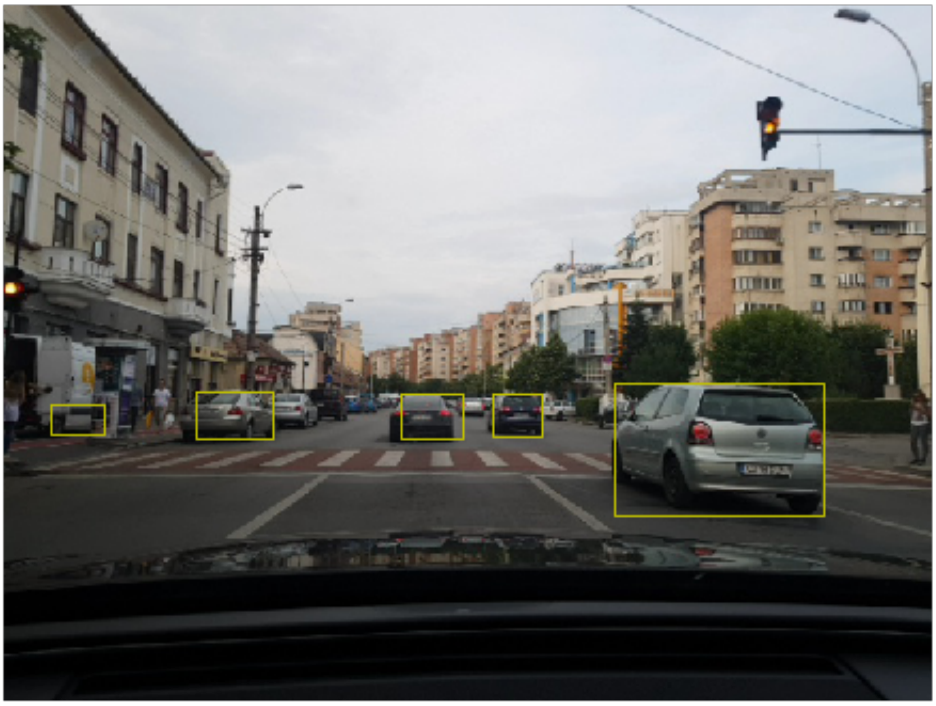

Figure 5. The input image with the detected obstacle bounding boxes using SSD MobileNet.

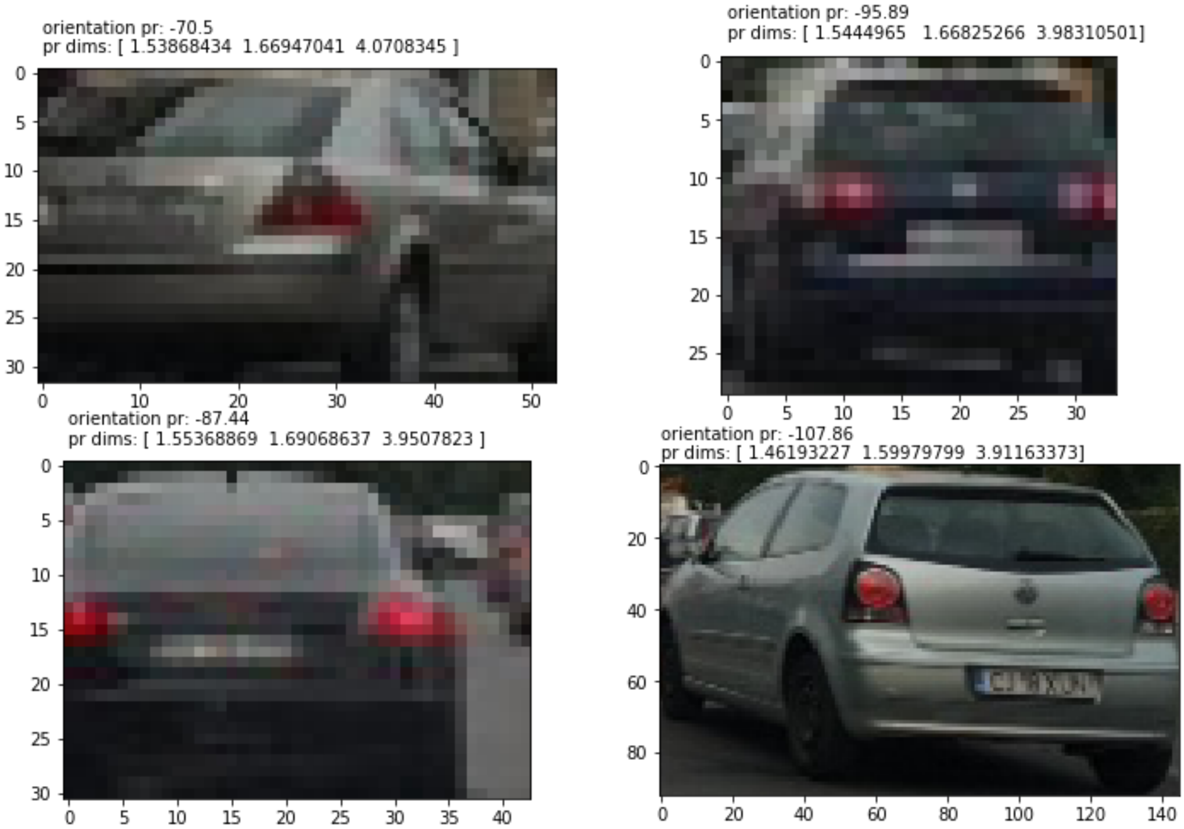
orientation pr: -70.5
pr dims: [1.53868434 1.66947041 4.0708345]
orientation pr: -95.89
pr dims: [1.5444965   1.66825266 3.98310501]
orientation pr: -87.44
pr dims: [1.55368869 1.69068637 3.9507823]
orientation pr: -107.86
pr dims: [1.46193227 1.59979799 3.91163373]

Figure 6. The predictions for vehicle orientation and dimensions.

The local size and orientation of a vehicle is integrated into the particle filter, more specifically in the particle grouping algorithm that extracts cuboids from the particles in similar cells. The correspondence between the vehicles detected by CNN and those tracked by the particle filter is made by calculating the Sorensen Dice (IoU) coefficient. I computed the IoU between the front and back sides of the cuboid from the particle filter and the bounding boxes predicted by the SSD MobileNet convolutional neural network. Situations with an IoU score higher than 0.5 are considered valid and the dimensions and local orientation of the cuboid are changed to the values from the CNN. Therefore, I obtain the data fusion between the convolutional neural network and the particle filter tracking.

The result of the data fusion is illustrated in Figure 7 where a top view of the scene in which the cuboid (from the particle filter) is displayed next to the adjusted and improved cuboid (from the CNN).

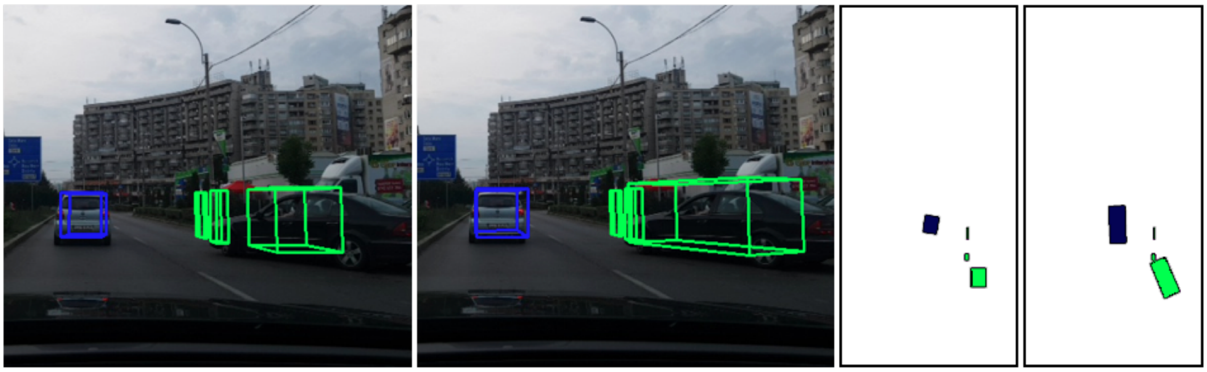

Figure 7. Using a CNN for adjusting the orientation and dimension of the vehicles. The first image represents the cuboids extracted from the particle filter, in center is illustrated the same image with the adjusted cuboids (from the CNN). The image on the right represents a top view of the scene with the same objects (cuboids from the particle filter vs. cuboids adjusted by CNN).

The purpose of the perception system is to use both artificial vision by image processing and convolutional neural networks, and to integrate them and fusing the data from these processing modules.

The original contributions that result from the solutions proposed in chapter three are: automatic determination of focal lengths on mobile devices, creation of a manual and intuitive calibration system on mobile devices where the user can adjust extrinsic camera parameters. Another contribution is to create an "offline" calibration system by analysing traffic images and determining the width of a traffic lane. The lane width is used as the correlation between the 3D world scene and 2D image space. This method determines the height of the camera above the ground (which is part of the translation vector of the camera).

One of the main contributions refers to the development of a new algorithm for calculating the vanishing point from road traffic images. These images have a perspective effect in which the lane lines will intersect at the vanishing point. The proposed algorithm exploits all points in the images that have a certain magnitude and orientation and uses them to build a vote map that is analysed and validated using sliding windows. The point in the window with the most votes will be the vanishing point. Figure 8 illustrates the vote map and the validation windows used (on the left), and the final result of the algorithm with the calculated vanishing point (the image on the right).

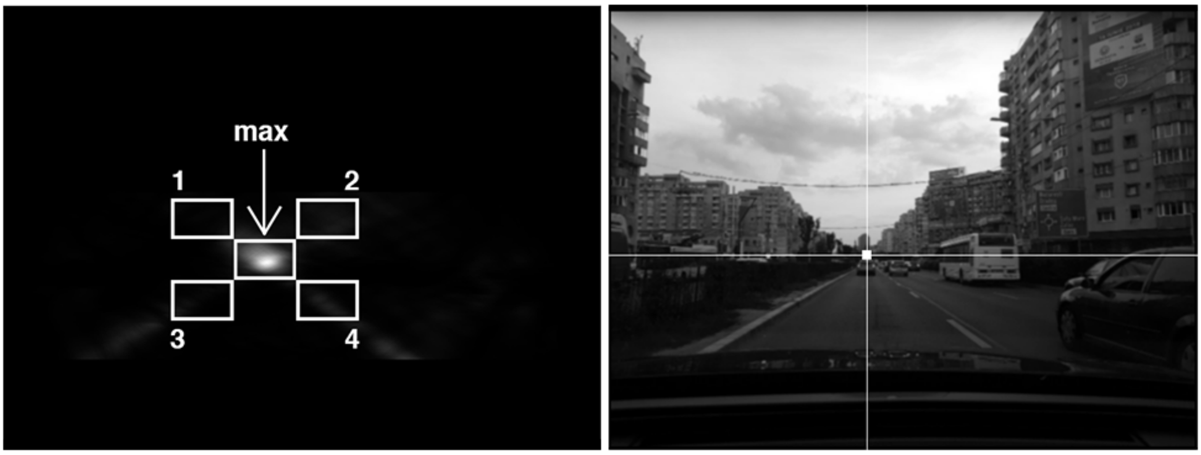

Figure 8. Left: the window with the most votes (max) from the vote map along with the validation windows that are used (numbered 1 to 4). Right: The final result of the algorithm with the computed vanishing point.

Having this method, I built a database of images and vanishing points that I then used to train a convolutional neural network in order to have predictions about the location of a vanishing point. The network has the resized scene image as input, whereas the output will consist of the two coordinates of the vanishing point. The network structure is as follows: 5 convolutional layers with a different number of filters and having a variable size kernel, and the last levels are three fully-connected levels. The last fully connected layer contains two neurons that actually represent the x and y coordinates of the predicted vanishing point. An example of the vanishing point prediction is illustrated in the following figure:

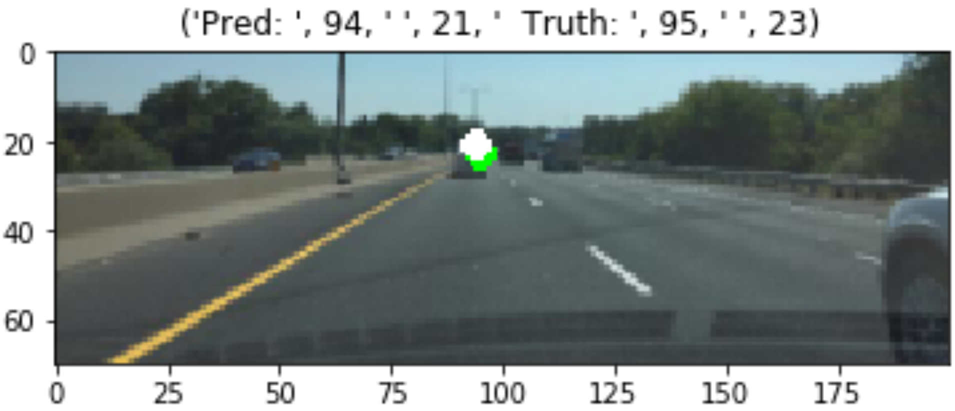

Figure 9. Example where the vanishing point calculated by CNN (white circle) is better than that obtained by traditional algorithms (green circle).

The results of this approach were very good and then the entire method was extended to use multiple training images and two different convolutional neural networks: one to predict the position on the horizontal coordinate and the other to predict the vertical coordinate of the vanishing point. I have presented an evaluation of the CNN based methods, and the use of two networks has been more effective and has significant improvements. This solution has also been implemented on mobile devices, and Figure 10 shows how to test and develop the algorithm in a controlled environment where the mobile device is located at a fixed point and oriented towards a screen where a test sequence is displayed. Determining the vanishing point from road traffic imagery using convolutional neural networks represents an original contribution.

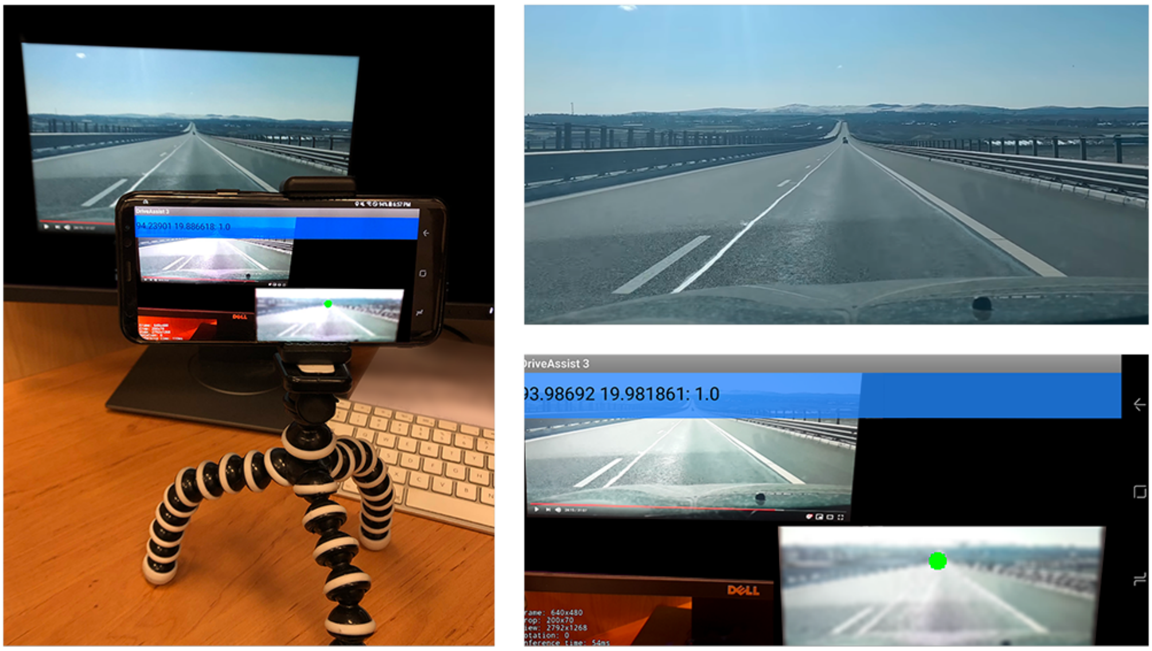

Figure 10. Determining the vanishing point directly on a mobile device. At the top right is the source image, and on the bottom right is a screen capture from the mobile device.

The last section represents an implementation of a self-calibration solution based on the trajectories of the vehicles. Vehicle detection was accomplished by using a convolutional neural network, more specifically the SSD and MobileNet, which provide a good detection precision and execution speed ratio. The result of the network will be a list of rectangles (bounding boxes) for each vehicle in the scene. These rectangles are tracked from one frame to another, and only a certain number of detected vehicles are required for calibration. From the coordinates of the detected rectangles, the widths of the objects in the scene and their coordinate on the vertical axis (y-axis of the image) are extracted. The next step is to determine the linear dependency between the widths and the vertical coordinate. The linear relationship can be computed using RANSAC, but I actually used a modified Hough algorithm. The algorithm for estimating the height of the camera and the pitch angle uses an extended Kalman filter, and the measuring vector Z is formed by width w_1 and w_2 of the vehicles at the horizontal coordinate (x-axis) $v_1 = 310$ and $v_2 = 360$, determined from the linear relation calculated in the previous step. We chose 2 points in the 3D space to represent a theoretical vehicle with a fixed width (standard) of 1750 mm at a distance of $Z_1 = 7000$ mm and another set of 3D points for a vehicle at a distance $Z_2 = 20000$ mm . These 4 points describing 2 vehicles at 2 different distances are projected into the 2D space of the image, and the pairs of lateral coordinates will form two lines that will intersect at the vanishing point. In the next step, the coordinates of the points at $v_1 = 310$ and $v_2 = 360$ will be calculated. Basically, the EKF algorithm will attempt to match these 4 theoretical points of the two vehicles and the 4 vehicle points in the measurement vector. The algorithm will adjust the camera height and pitch angle until the result is stable and the best match is reached (usually after 10 EKF iterations).

The algorithm based on vehicle detection results from a convolutional neural network is able to estimate the extrinsic parameters of a camera relative to a reference frame based on a standard vehicle. The system works precisely and robustly when there are long sequences in various traffic situations and when sufficient data is available. Since vehicle detection influences the estimation of parameters, in the future I intend to improve this aspect.

By integrating all the solutions, I have succeeded in implementing a monocular perception system and the processing flow is illustrated in Figure 11.

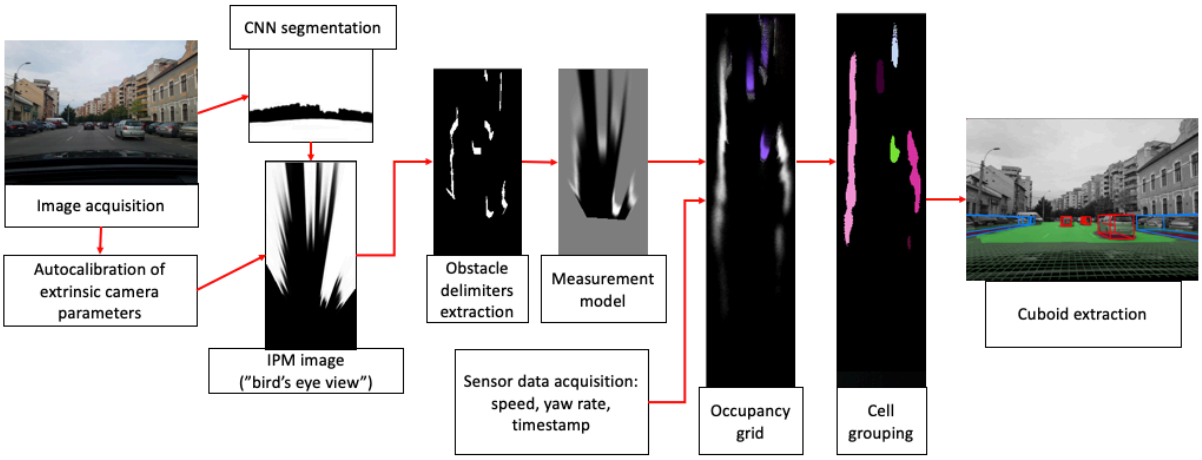
CNN segmentation
Image acquisition
Autocalibration of extrinsic camera parameters
IPM image ("bird's eye view")
Obstacle delimiters extraction
Measurement model
Sensor data acquisition: speed, yaw rate, timestamp
Occupancy grid
Cell grouping
Cuboid extraction

Figure 11. Processing flow of the monocular perception system.

The first step is to acquire images and auto-calibrate the camera's extrinsic parameters, which are used to generate the image with the perspective effect eliminated (from the image segmented by the neural network). The next step is to detect the obstacle delimitations and this image is used to generate a measurement model that is used to create particles in an occupancy grid. In the last step, the cells that are similar are grouped and cuboids are extracted from them and they are displayed in the original image of the road scene.

In conclusion, the thesis introduces contributions in the field of robotics and autonomous vehicles in which the perception is made using a single camera. Perception algorithms are based on image processing techniques, but also on artificial intelligence by using convolutional neural networks. The thesis presents a complete solution for the perception of the road traffic scene, starting with the image acquisition and ending with the data processing and displaying the results. The monocular system is capable of calibrating itself after a certain time and the processed data can be used and integrated into advanced driver assistance systems, such as collision prevention systems that can act for the driver or that can only provide acoustic or visual warnings. The solutions presented in the thesis have been implemented both on mobile systems (smartphones or tablets) and on PCs (desktop or laptop). The methods described in this paper can bring a major benefit by improving road traffic safety, with the advantage that such systems can be easily integrated in any existing vehicle at a low cost (using a single video camera for scene analysis). In conclusion, in the future I wish to implement together with Professor Dănescu a complete system that is easy to integrate into any vehicle.

Publications

The following publications have resulted from the methods proposed in this PhD thesis:

1. **Razvan Itu**, Radu Danescu, "An Efficient Obstacle Awareness Application for Android Mobile Devices", International Conference on Intelligent Computer Communication and Processing, pp. 157-163, 2014.

2. Radu Danescu, **Razvan Itu**, Andra Petrovai, "Sensing the Driving Environment with Smart Mobile Devices", IEEE International Conference on Intelligent Computer Communication and Processing, pp. 271-278, 2015.

3. Radu Danescu, **Razvan Itu**, Andra Petrovai, "Generic Dynamic Environment Perception Using Smart Mobile Devices", Sensors, Vol. 16, No. 10, 2016, Art. No. 1721.

4. Radu Danescu, Andra Petrovai, **Razvan Itu**, Sergiu Nedevschi, "Generic Obstacle Detection for Mobile Devices Using a Dynamic Intermediate Representation", Advances in Intelligent Systems and Computing, vol. 427, 2016, pp. 629-639.

5. **Razvan Itu**, Diana Borza, Radu Danescu, "Automatic extrinsic camera parameters calibration using Convolutional Neural Networks", 2017 IEEE 13th International Conference on Intelligent Computer Communication and Processing (ICCP 2017), pp. 273-278.

6. **Razvan Itu**, Radu Danescu, "Machine Learning Based Automatic Extrinsic Calibration of an Onboard Monocular Camera for Driving Assistance Applications on Smart Mobile Devices", 2018 International Forum on Advanced Microsystems for Automotive Applications, pp. 16-28.

7. Radu Danescu, **Razvan Itu**, ”Camera Calibration for CNN based Generic Obstacle Detection”, 2019 Portuguese Conference on Artificial Intelligence (EPIA), accepted june 2019.

Citations

Google Scholar:
30 citations
H-index 4

ISI web of science:
12 citations
H-index 2

Bibliography

[Bertozzi1998] - M. Bertozzi, A. Broggi, “GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection”, IEEE Transactions on Image Processing, vol. 7, no. 1, pp. 62-81, 1998.

[Danescu2011] - R. Danescu, F. Oniga, S. Nedevschi, “Modeling and Tracking the Driving Environment with a Particle-Based Occupancy Grid”, IEEE Transactions on Intelligent Transportation Systems, volume 12, no. 4, pp. 1331-1342, 2011.

[Danescu2016] - R. Danescu, R. Itu, A. Petrovai, "Generic Dynamic Environment Perception Using Smart Mobile Devices", Sensors, vol. 16, no. 10, art. no. 1721, 2016.

[Howard2017] - A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, arXiv:1704.04861, 2017.

[Ronneberger2015] - O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, Vol. 9351, pp. 234-241, 2015.

[Simonyan2014] - K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, arXiv:1409.1556, 2014.

[Wei2016] - L. Wei, et. al., “SSD: Single Shot MultiBox Detector”, European Conference on Computer Vision, pp. 21-37, 2016.