RIGA TECHNICAL UNIVERSITY
Faculty of Electronics and Telecommunications
Institute of Radio Electronics

Roberts Kadiķis
Doctoral Student of the Study Programme “Electronics”

EFFICIENT METHODS FOR DETECTION AND CHARACTERIZATION OF MOVING OBJECTS IN VIDEO

Summary of the Doctoral Thesis

Scientific supervisor
Senior researcher Dr. sc. comp.
MODRIS GREITĀNS
Institute of Electronics and Computer Science

RTU Press
Riga 2018

Kadiķis R. Efficient Methods for Detection and
Characterization of Moving Objects in Video.
Summary of the Doctoral Thesis.
Riga: RTU press, 2018. 32 pages.

Published in accordance with the
decision of Promotion Council "RTU P-08"
of 8 January 2018, Minutes No. 42.

ISBN 978-9934-22-059-3 (print)
ISBN 978-9934-22-060-9 (epub)

DOCTORAL THESIS PROPOSED TO RIGA TECHNICAL UNIVERSITY FOR THE PROMOTION TO THE SCIENTIFIC DEGREE OF DOCTOR OF ENGINEERING SCIENCES

To be granted the scientific degree of Doctor of Engineering Sciences, the present Doctoral Thesis has been submitted for the defence at the open meeting of RTU Promotion Council on 19 April 2018 at the Faculty of Electronics and Telecommunications of Riga Technical University, 12 Azenes Street, Room 212.

OFFICIAL REVIEWERS

Professor Dr. habil. phys. Andris Ozols
Riga Technical University, Latvia

Professor Dr. sc. ing. Aleksandrs Grakovskis
Transport and Telecommunication Institute, Latvia

Professor Dr. sc. ing. Pēteris Grabusts
Rezekne Academy of Technologies, Latvia

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the Doctoral Thesis submitted for the review to Riga Technical University for the promotion to the scientific degree of Doctor of Engineering Sciences is my own. I confirm that this Doctoral Thesis had not been submitted to any other university for the promotion to a scientific degree.

Roberts Kadiķis .... (Signature)

Date: ....

The Doctoral Thesis has been written in English. It consists of an introduction, 5 chapters, Bibliography, 6 appendices, 37 figures, 4 tables; the total number of pages is 132. The Bibliography contains 111 titles.

CONTENTS

ABBREVIATIONS

2D – 2 Dimensional
3D – 3 Dimensional
BRIEF – Binary Robust Independent Elementary Features
CNN – Convolutional Neural Network
CPU – Central Processing Unit
CV – Computer Vision
DoG – Difference of Gaussians
FN – False Negative
FP – False Positive
GUI – Graphical User Interface
IoVDL – Intervals on Virtual Detection Line
ITS – Intelligent Transportation Systems
LBP – Locally Binary Patterns
LSTM – Long Short-Term Memory
MSER – Maximally Stable Extremal Regions
MTBF – Mean Time Between Failure
ORB – Oriented FAST and rotated BRIEF
RAM – Random-Access Memory
R-CNN – Regions with CNN features
RGB – Red, Green, Blue
RNN – Recurrent Neural Network
RNN-VDL – Recurrent Neural Network-based Virtual Detection Line
ROI – Region of Interest
SIFT – Scale Invariant Feature Transform
SURF – Speeded Up Robust Features
SVM – Support Vector Machine
TIRTL – The Infra-Ref Traffic Logger
TP – True Positive
YOLO – You Only Look Once

GENERAL DESCRIPTION OF THE WORK

The Urgency of Subject Matter

In this Thesis, a video is defined as a sequence of digital images (frames). The methods of interest acquire information about the real world by processing the input video. It is a task of a computer vision (CV) field, which is used in many practical applications, including:

- automation (control of industrial robots [1], assembly line supervision [2]);
- health (analysis of biomedical images for diagnosis [3], drug discovery [4]);
- security (face recognition [5], smart surveillance systems [6]);
- transportation systems (detection of vehicles [7], license plate recognition [8]).

New and more efficient algorithms can speed up the spread of computer vision by enabling the existing size-limited devices to use modern CV. **In this Thesis, the efficiency of the method relates to its computational demands.**

One of the main practical beneficiaries of the work carried out within the scope of this Thesis might be the field of Intelligent Transportation Systems (ITS). Such systems aim to effectively manage transportation, which is facing an increase of congestions, energy consumption, pollution, health risks, costs, and human demand for mobility. The target infrastructure for ITS is large, thus the need for efficient and scalable solutions. Also, the proposed methods must be adaptable since ITS have to operate in an outdoor environment, which includes different and changing weather conditions, seasons, and times of the day.

Many of the CV applications include the task of object detection, which is the primary focus of the Thesis. In this paper, an object is considered as correctly detected in a video if its presence is recognized at least once during its existence in the scene.

The Objective of the Paper

This Thesis aims to improve efficient detection of moving objects in a video by developing new video processing methods. The developed methods need to be adaptable to the changing environment and usable by devices with limited computational power. Several tasks have been defined in order to reach the aim of this paper:

- to perform a review of literature on object detection in images and videos;
- to identify efficient approaches for detection of moving objects in videos;
- to develop efficient object detection methods with improved capabilities compared to the existing efficient methods;
- to develop data acquisition methods and acquire the data needed for the development and testing of proposed methods;
- to implement and conduct experimental research on the proposed methods;
- to draw conclusions about the results of this Thesis.

The Methodology of Research

The first tasks of the Thesis are accomplished by an analytical research. Analytic methods are used to review existing literature in the field and also to propose and develop novel moving object detection algorithms and labeled data acquisition methods. The developed methods are implemented using C++ and Python languages. These implementations are experimentally tested and compared on different videos. The tests measure the computational efficiency and accuracy of the developed methods.

Scientific Novelty and Main Results

The Thesis results in novel and efficient methods for the detection and characterization of moving objects in a video. In addition, novel data labeling methods are developed for training and testing of the proposed detection methods.

IoVDL – Intervals on Virtual Detection Line is a novel computationally efficient method for detection of moving objects. Similarly to existing efficient methods, the IoVDL processes only a small part of a frame, and it has a similar detection accuracy. The advantage of the developed method is that it is more flexible in its application than the conventional efficient methods. In the vehicle detection context, the developed approach does not require the unchanging trajectories of vehicles. The IoVDL is applicable on the roads where lanes are not specified or their number changes depending on the time of the day. IoVDL is capable of detecting several objects that enter the virtual detector simultaneously. The IoVDL method is implemented on a low-cost Raspberry Pi Zero computer, and its vehicle detection accuracy is experimentally tested on videos with different weather conditions.

Extended IoVDL demonstrates an additional advantage of the proposed object detection approach. By placing several IoVDL detectors in the frame, the extended IoVDL is able to capture the object's movement through the detectors. The method characterizes the object by measuring its speed and size parameters and uses these measurements to classify the object. The method is computationally efficient compared to the existing methods that can track and characterize objects in a video. Extended IoVDL is implemented and tested on a vehicle classification task.

RNN-VDL – Recurrent Neural Network-based Virtual Detection Line combines the idea of the efficient detection line and a recurrent neural network. Similarly to IoVDL, the new method processes a single line of pixels and detects objects that cross this line. The method is implemented and compared with the IoVDL on the vehicle counting task. The use of the machine learning means that the same implementation of the RNN-VDL can be retrained for the detection of different kinds of objects, so it is also tested on a people detection task.

Manual labeling method is a GUI-based approach for labeling of video data. The user labels each frame of the video, and the method converts and saves these frames and labels so that they can be used for training object detection models.

Semi-automatic labeling method combines initial automatic labeling of video with a specific GUI that allows a human user to quickly detect and correct the mistakes made by the computer. It is a significantly faster labeling approach than the manual method, but it is usable only for detection line-based methods. The semi-automatic method is used to label training and test videos used in the experiments of this Thesis. The labeled datasets are available in [9].

Thesis Statements to Be Defended

1. The developed IoVDL method detects moving objects in a video with a similar accuracy and computational efficiency as alternative virtual detection region-based methods while being less sensitive to the differing number and trajectories of objects that cross the detection region.
2. The combination of several IoVDL detectors (extended IoVDL) tracks and measures the speed and size parameters of objects with less computational resources than conventional tracking methods that process the whole frame.
3. The combination of the virtual line detector approach with a recurrent neural network results in an adaptive virtual detector RNN-VDL that can be retrained for the detection of different kinds of objects without changes in its architecture and the handcrafted feature engineering.
4. The speed of labeling the several hour long video data for the training of recurrent neural network-based object detector RNN-VDL is increased at least ten times by the developed semi-automatic labeling method.

Practical Value and Approbation

The developed methods IoVDL, extended IoVDL, and RNN-VDL introduce novel ways of making computationally efficient vision-based moving object detectors. The IoVDL method is tested on a vehicle detection task, and the tests show the suitability of the method for ITS. In turn, the combination of RNN-VDL and the proposed data labeling methods makes the machine learning-based approach usable for a wider range of practical applications.

Doctoral thesis has been developed in connection with several projects developed at the Institute of Electronics and Computer Science, Latvia:

- European Regional Development Fund project “Multifunctional intelligent transportation system point technology” (MITS) No. 2010/0250/2DP/2.1.1.1.0/10/APIA/VIAA/086;
- European Social Fund project “Innovative technologies for acquisition and processing of biomedical images” No. 2013/0009/1DP/1.1.1.2.0/13/APIA/VIAA/014;

- National Research Programme “Cyber-physical systems, ontologies and biophotonics for safe&smart city and society.” (SOPHIS) project No. 4 “Development of technologies for secure and reliable smart-city”;
- European Regional Development Fund project “Deep neural network method for improve the accuracy of tracking and classification of vehicles registration plates.” (Dz-iNTA) No.1.2.1.1/16/A/007;
- European Regional Development Fund project “Competency centre for Latvian Electrical and optical manufacturing industry” No. 1.2.1.1/16/A/002.

The results of this Thesis are protected by a European patent [10]. They have been described in several papers [11]–[14] and in image processing related sections of papers [15]–[17]. These results have been promoted in the following international conferences:

- International Conference on Signal and Image Processing 2012 (ICSIP 2012), December 13–15, Coimbatore, India;
- The 6th International Conference on Machine Vision (ICMV 2013), November 16–17, London, United Kingdom;
- Advances in Information Electronic and Electrical Engineering, July 5–7, 2013, Riga, Latvia;
- Microwave and Radio Electronics Week 2015 (MAREW 2015), April 21–23 Pardubice, Czech Republic;
- The 11th International Young Scientist Conference. Developments in Optics and Communications, April 8–10, 2015, Riga, Latvia;
- New Challenges of Economic and Business Development – 2017: Digital Economy, May 18–19, Riga, Latvia;
- The 10th International Conference on Machine Vision (ICMV 2017), November 13–15, Vienna, Austria.

Structure of the Thesis

The thesis consists of 132 pages. It is divided into five main sections. Section 1 is an overview of object detection methods. Non-camera based detection methods are compared with the video processing approach, whose different methods are then described in more detail. Section 2 highlights the existing efficient video processing methods, then introduces a novel method IoVDL. The proposed method is expanded so that it not only detects moving objects but also characterizes them. Then another object detector RNN-VDL is proposed. This machine learning approach needs sufficient amount of labeled data, so Section 3 is dedicated to approaches for data labeling. In Section 4 the proposed methods are tested on vehicle and people counting tasks. The conclusions about developed methods and test results are presented in Section 5.

1. OVERVIEW OF OBJECT DETECTION METHODS

1.1. Sensors for Vehicle Detection

Video camera is not the only sensor that can be used for object detection. Different sensors have been used on the roads to detect vehicles, including intrusive sensors such as pneumatic tube detector, inductive detector loop, magnetometer, weight in motion system, as well as non-intrusive sensors, such as microwave radar, active infrared laser radar, passive infrared sensor, ultrasonic sensor, passive acoustic sensor.

Studies [18], [19] experimentally compared different sensors in 2010–2011, and the video camera was not found to be the most accurate or the most robust approach. However, there are several considerations, why the current Thesis focuses on the video based detection.

- The computer vision field has undergone the most rapid progress in the years since these tests were carried out. In the first ImageNet competition [20] in 2010, the best image classification approaches failed at 28.2 % of the test images. In 2017 the winning method achieved 2.25 % error. The human performance on this dataset is approximately 5 % [21].
- A single camera sensor can cover a wide road with many lanes, so a single camera can replace several alternative sensors.
- The camera is a versatile sensor. It can be used for detection of emergency vehicles, pedestrians, animals, and other objects.
- The same camera that is used for the detection can also supply the necessary data for other computer vision tasks, such as characterization, classification, and tracking of objects.

1.2. Detection Using Parameters of Individual Pixels

2D digital images consist of $R \times C$ pixels, where each pixel is described by some value. In the case of grayscale images, the value of each pixel is the intensity I. In simple cases of the object detection task, the objects can be distinguished from the background by analyzing the values of each pixel. These approaches may not be as robust as more sophisticated methods; however, they are a good starting point for computationally efficient methods.

Thresholding. If objects in a grayscale image are lighter than the background, they can be detected by simply finding the bright pixels in the image. Thresholding operation compares each pixel with a threshold value T. The result of thresholding is a binary image, where pixels with $I > T$ are white, while pixels with $I \leq T$ are black. The important parameter of such approach is the value T. It can be detected automatically, for example, using Otsu method [22].

Color can be used as another distinguishable parameter. In an ordinary RGB image, the color of each pixel is determined by three values. The distance between the color values of different pixels can separate differently colored objects in the image. This idea can be generalized to other color models besides RGB. It also can be used for multispectral and hyperspectral images, where each pixel consists of more than three values ($N_\mathrm{k} > 3$) so that the color of a pixel becomes its spectrum.

Background Subtraction in video processing means the comparison of the current frame with a reference frame of an empty scene. The objects of interest can be distinguished from the background using different parameters, such as intensity or color values of the pixels [23]–[25]. Edges and corners are robust parameters in changing lighting or weather conditions [26].

Since many object detection systems have to deal with the changing outdoor environment, they use an adaptive background image, which can be estimated in different ways. There are non-recursive methods where a specified number of frames are stored in a buffer. The value of the background pixels may be then selected as some statistical value of stored frames, such as mean or median. In the recursive methods, the background is updated using such methods as Kalman filter [26], Mixture of Gaussians [27], a weighted sum of the background image and the current frame [25], or a similar technique.

1.3. Detection Using the Object's Features

More robust approaches are based on object's feature detection and description.

Template Matching. If there is an image (template) of the object $\mathbf{I}_\mathrm{t}$, then this image can be compared to all regions of the input image $\mathbf{I}$ to find the areas of $\mathbf{I}$ that are similar to the object. Usually, the comparison is done in the sliding window manner. Different distance measures can be used to assess the similarity of the region and the template.

Keypoint Detectors. An object can be detected by finding distinctive points or regions that are peculiar to this object. Such points are usually called keypoint features or interest points and they can be the corners, blobs, and T-junctions of an image. Some widely used keypoint detectors incldue:

- Harris Corner Detector [28];
- Hessian Matrix Detector [29];
- DoG – Difference of Gaussian detector [30];
- MSER – Maximally Stable Extremal Regions [31];
- FAST – Features from Accelerated Segment Test [32], [33].

Hand-crafted Region Descriptors. After the discovery of the keypoints, the neighborhood of these points is described by the feature vector. Commonly used descriptors include:

- SIFT – Scale Invariant Feature Transform [34];
- SURF – Speeded Up Robust Features [35];
- LBP – Locally Binary Patterns [36];
- ORB – Oriented FAST and Rotated BRIEF [37].

Matching the Feature Points. The described feature points must be matched between the object and the image. Commonly, a distance between descriptors is measured to determine if they represent the same point. This Thesis proposes a novel feature-matching method that allows the use and combination of different kinds of feature points.

The algorithm assumes an initial position of the object in the input image, then uses the identified keypoints to find the most probable translation of this object. All possible shifts among object's and image's keypoints are measured and used to construct a new image (Fig. 1.1). The coordinates of the pixel with the greatest value correspond to the most probable translation. The proposed keypoint matching method can also be made rotation and scale-invariant.

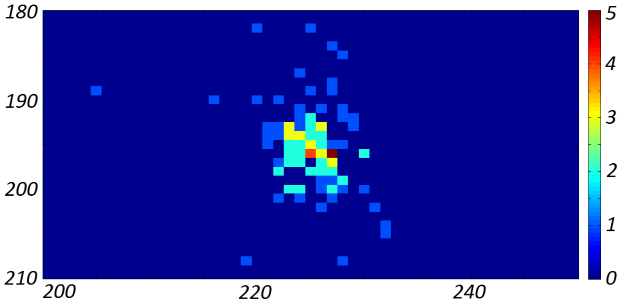


Fig. 1.1. Possible translations between the assumed and actual positions of the object.

1.4. Deep Learning Approaches

CNN. The current state-of-the-art object detectors, such as R-CNN (regions with CNN features) [38], *Faster* R-CNN [39], and YOLO (You Only Look Once) [40] are based on the artificial neural networks. In the vision tasks, the most commonly used type of network is the Convolutional Neural Network (CNN). A popular CNN architecture (AleXnet) is depicted in Fig. 1.2.

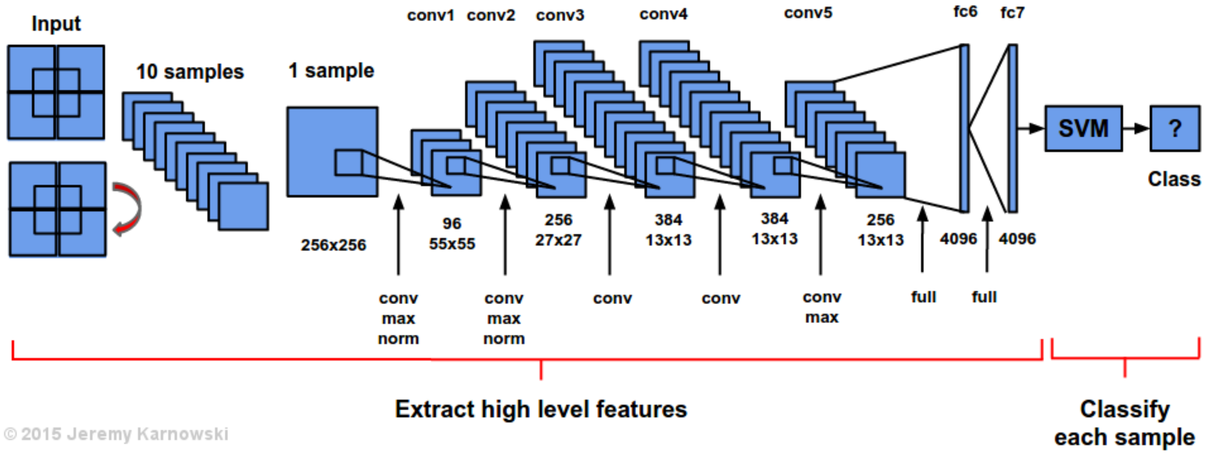


Fig. 1.2. AlexNet architecture [41].

RNN. The conventional neural networks, as well as CNNs, are not explicitly suitable for learning events that change in time. The output of these methods depends only on the current input. In turn, the recurrent neural networks (RNN) have neurons with recurrent edges, which makes them state-of-the-art methods in several sequence-related tasks [42].

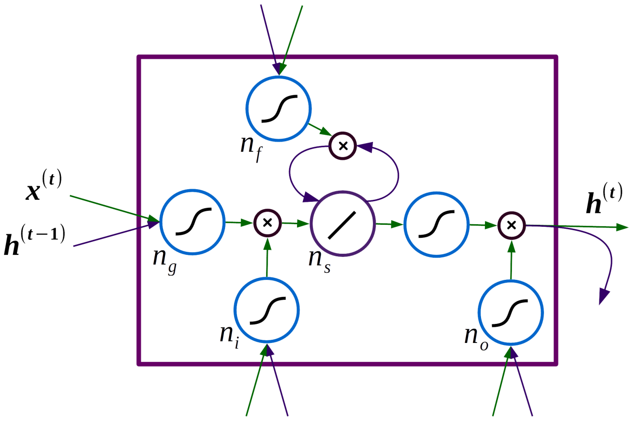


Fig. 1.3. Memory cell of LSTM network.

LSTM – Long Short-Term Memory [43] – is a successful type of RNN. A hidden layer of LSTM consists of a memory cell depicted in Fig. 1.3. The cell includes internal state node n_s and several additional nodes $n_\text{g}, n_\text{i}, n_\text{f}, n_\text{o}$. All the latter nodes receive the input signal $\mathbf{x}^{(t)}$ and the output signal formed by the LSTM cell in the previous time step $\mathbf{h}^{(t-1)}$.

The forward pass computations of the LSTM memory cell are shown in equations (1.1)–(1.6):

$$\mathbf{n}_\text{g}^{(t)} = \alpha(\mathbf{W}_\text{gx}\mathbf{x}^{(t)} + \mathbf{W}_\text{gh}\mathbf{h}^{(t-1)} + \mathbf{b}_\text{g}) \tag{1.1}$$

$$\mathbf{n}_\text{i}^{(t)} = \alpha(\mathbf{W}_\text{ix}\mathbf{x}^{(t)} + \mathbf{W}_\text{ih}\mathbf{h}^{(t-1)} + \mathbf{b}_\text{i}) \tag{1.2}$$

$$\mathbf{n}_\text{f}^{(t)} = \alpha(\mathbf{W}_\text{fx}\mathbf{x}^{(t)} + \mathbf{W}_\text{fh}\mathbf{h}^{(t-1)} + \mathbf{b}_\text{f}) \tag{1.3}$$

$$\mathbf{n}_\text{o}^{(t)} = \alpha(\mathbf{W}_\text{ox}\mathbf{x}^{(t)} + \mathbf{W}_\text{oh}\mathbf{h}^{(t-1)} + \mathbf{b}_\text{o}) \tag{1.4}$$

$$\mathbf{n}_\text{s}^{(t)} = \mathbf{n}_\text{g}^{(t)} \odot \mathbf{n}_\text{i}^{(t)} + \mathbf{n}_\text{s}^{(t-1)} \odot \mathbf{n}_\text{f}^{(t)} \tag{1.5}$$

$$\mathbf{h}^{(t)} = \alpha(\mathbf{n}_\text{s}^{(t)}) \odot \mathbf{n}_\text{o}^{(t)}. \tag{1.6}$$

where $\mathbf{W}$ is a matrix of weights and $\mathbf{b}$ is a vector of biases between different nodes; $\alpha(x)$ is some nonlinear activation function; $\odot$ stands for pointwise multiplication.

Using RNN in video processing is computationally demanding. In the following section, this Thesis researches and proposes a method of using the RNN for video processing in a computationally efficient manner.

2. EFFICIENT METHODS FOR OBJECT DETECTION

2.1. Existing Efficient Methods

One of the methods to create faster object detection algorithms is to process only a part of the image or frame. In the case of videos acquired by a static camera, a region of interest (ROI) may be defined in the frame. Only the pixels of ROI are processed, so the objects have to be either inside or eventually enter the ROI in order to be detected. This limitation still allows use of such methods in several practical computer vision applications, for example, vehicle counting on highways [44]–[46], detection of people entering buildings [47], [48], inspection of products on the conveyor belt [49], [50].

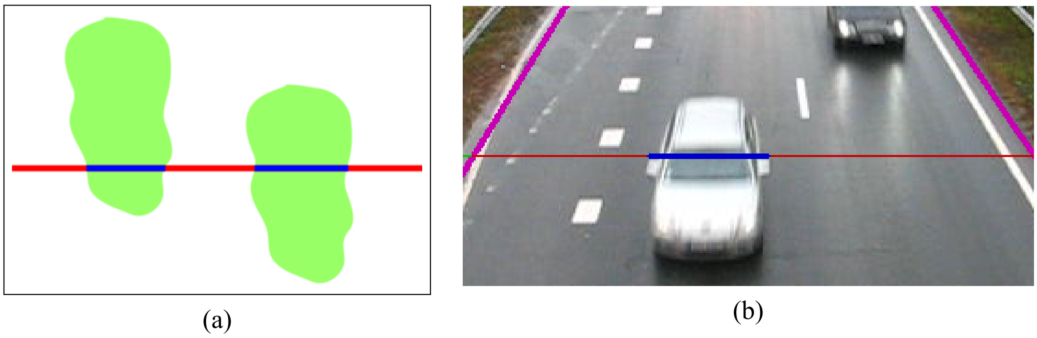
In papers such as [51]–[53] the ROI is composed of one or more lines within a frame. The lines are perpendicular to the usual movement of the objects of interest. Such lines are also called virtual-lines, detection lines or virtual loop detectors.

In the specific application of highway monitoring, some ROI based approaches (for example [53]–[55]) have to precisely define the detection regions for each lane of the road. Such approaches are susceptible to errors caused by long shadows of vehicles or large trucks that may be detected as independent vehicles by virtual detectors at adjacent road lanes. Furthermore, such methods are not usable in cases when the number and position of lanes vary over time.

2.2. Novel Detection-line Based Method

The proposed ROI based method is more flexible than the alternatives since the novel interval approach allows the detection of objects of various widths anywhere on the virtual detector.

Intervals on a virtual detection line (IoVDL). The IoVDL method detects objects that arrive on a defined line in a video frame (Fig. 2.1). This line (a row of pixels $\mathbf{l}_{\text{rgb}}$) is typically placed perpendicularly to the expected movement of the objects. All further processing is carried out on this line only, thus making the detection algorithm computationally efficient.


(a)

(b)

Fig. 2.1. The detection line, objects, and corresponding intervals in a frame.

In the detection process, intervals are created on the detection line denoting the segments covered by the objects. Each interval is updated with the information acquired from the following frames to reflect the changing overlap of the corresponding object with the detection line. When the object leaves the detection line, its interval is closed, and the object's parameters are acquired. The main steps of the IoVDL method are depicted in Algorithm 1.

Algorithm 1. Detecting Objects on the Detection Line

```
 1: procedure DETECT-OBJECTS(line from color frame $\mathbf{l}_{\text{rgb}}$, frame number $t$)
 2:     line $\mathbf{l}_{\text{g}}^{(t)} \leftarrow$ convert $\mathbf{l}_{\text{rgb}}$ to grayscale
 3:     line $\mathbf{l}_{\text{d}} \leftarrow$ medianFilter($|\mathbf{l}_{\text{g}}^{(t)} - \mathbf{l}_{\text{g}}^{(t-1)}|$)
 4:     line $\mathbf{l}_{\text{t}} \leftarrow$ threshold $\mathbf{l}_{\text{d}}$ with threshold value $T$
 5:     temporary buffer of intervals $\mathbb{B}_{\text{i\_temp}} \leftarrow$ FIND-INTERVALS($\mathbf{l}_{\text{t}}$)
 6:     add contents of $\mathbb{B}_{\text{i\_temp}}$ to $\mathbb{B}_{\text{i}}$
 7:     for each (interval $i$ and $j$ in $\mathbb{B}_{\text{i}}$) do
 8:         if $i$ and $j$ overlap then
 9:             merge $i$ and $j$
10:         end if
11:     end for
12:     line $\mathbf{l}_{\text{i\_t}} \leftarrow$ INTENSITY-BACKGROUND($\mathbb{B}_{\text{i}}, \mathbf{l}_{\text{g}}^{(t)}$)
13:     line $\mathbf{l}_{\text{e\_t}} \leftarrow$ EDGE-BACKGROUND($\mathbb{B}_{\text{i}}, \mathbf{l}_{\text{g}}^{(t)}$)
14:     line $\mathbf{l}_{\text{rgb\_t}} \leftarrow$ COLOR-BACKGROUND($\mathbb{B}_{\text{i}}, \mathbf{l}_{\text{rgb}}$)
15:     for each (interval $i$ in $\mathbb{B}_{\text{i}}$) do
16:         int $Z_{\text{i}} \leftarrow$ the number of white pixels of $\mathbf{l}_{\text{i\_t}}$ covered by $i$
17:         int $Z_{\text{e}} \leftarrow$ the number of white pixels of $\mathbf{l}_{\text{e\_t}}$ covered by $i$
18:         int $Z_{\text{rgb}} \leftarrow$ the number of white pixels of $\mathbf{l}_{\text{rgb\_t}}$ covered by $i$
19:         if $Z_{\text{i}} + Z_{\text{e}} + Z_{\text{rgb}} < closingConstant$ and $i$ was not updated for delay frames then
20:             if $minWidth < width(i) <$ maxWidth and $duration(i) >$ minDuration then
21:                 report a detected object
22:             end if
23:             remove interval $i$ from $\mathbb{B}_{\text{i}}$
24:         end if
25:     end for
26: end procedure
```

The method FIND-INTERVALS($\mathbf{l}_{\text{t}}$) in Algorithm 1 detects white regions on a thresholded line and creates corresponding intervals. Intervals need to correspond to manually chosen size parameters in order to be saved. In the example of highway monitoring, these parameters will determine if motorcycles and bikes will be detected along with vehicles.

To maintain intervals when no movement is detected, the IoVDL method incorporates the background subtraction approach. Background lines of several parameters are used: intensity, edges, and color. The output of the corresponding functions (Steps 12–14 of Algorithm 1) are thresholded lines $\mathbf{l}_{\text{i\_t}}$, $\mathbf{l}_{\text{e\_t}}$, $\mathbf{l}_{\text{rgb\_t}}$ where white pixels correspond to the foreground objects.

When an interval is closed, its parameters, such as width and duration, are examined in order to discard too small or too short-lived intervals. The interval's parameters also may be used to distinguish different types of objects if such exist.

Adaptive background subtraction. Algorithm 2 shows the background subtraction process of the proposed method on the example of function EDGE-BACKGROUND($\mathbb{B}_\mathrm{i}$, $\mathbf{l}_\mathrm{g}$).

Algorithm 2. Background Subtraction Using Edge Lines

```
function EDGE-BACKGROUND(interval buffer $\mathbb{B}_\mathrm{i}$, grayscale line $\mathbf{l}_\mathrm{g}$)
    line $\mathbf{l}_\mathrm{g} \leftarrow$ blur line $\mathbf{l}_\mathrm{g}$ with a Gaussian low-pass filter
    $\mathbf{l}_\mathrm{e} \leftarrow$ find edges on $\mathbf{l}_\mathrm{g}$
    if $\mathbb{B}_\mathrm{i}$ is empty then
        update the whole background edge line $\mathbf{l}_\mathrm{e\_bg}$ according to (2.1)
    else
        update the regions of $\mathbf{l}_\mathrm{e\_bg}$ outside intervals according to (2.1)
        append $\mathbf{l}_\mathrm{e}$ to $\mathbb{B}_\mathrm{e}$
    end if
    $\mathbf{l}_\mathrm{e} \leftarrow |\mathbf{l}_\mathrm{e} - \mathbf{l}_\mathrm{e\_bg}|$
    int $T_\mathrm{e} \leftarrow$ FIND-THRESHOLD($\mathbb{B}_\mathrm{e}$)
    line $\mathbf{l}_\mathrm{e\_t} \leftarrow$ threshold $\mathbf{l}_\mathrm{e}$ with threshold $T_\mathrm{e}$
    return (thresholded edge line $\mathbf{l}_\mathrm{e\_t}$)
end function
```

The edges on the intensity line are found by computing the absolute differences between adjacent pixels (Step 3). When algorithm recursively updates the pixels of background line $\mathbf{l}_\mathrm{e\_bg}$, it uses following equation:

$$I_{\mathrm{bg}}^{(t)} = E \cdot I_{\mathrm{bg}}^{(t-1)} + (1 - E) \cdot I^{(t)}, \tag{2.1}$$

where $I_{\mathrm{bg}}^{(t)}$ is intensity of the background pixel in the current frame; $I_{\mathrm{bg}}^{(t-1)}$ is intensity of the background pixel in the previous frame; $I^{(t)}$ is intensity value of the pixel in the current frame; and E is update rate.

Function FIND-THRESHOLD($\mathbb{B}_\mathrm{e}$) uses the lines from buffer $\mathbb{B}_\mathrm{e}$ to form image $\mathbf{I}_\mathrm{e}$. The image is used by the Otsu method to automatically determine threshold T_e. $\mathbf{I}_\mathrm{e}$ is updated in time, so the T_e also adapts to the changing environment.

Improvements of the IoVDL method. Until now, the proposed IoVDL method was described in its most efficient form. This Thesis proposes optional, application-specific improvements, which expand the capabilities of IoVDL while reducing its computational efficiency.

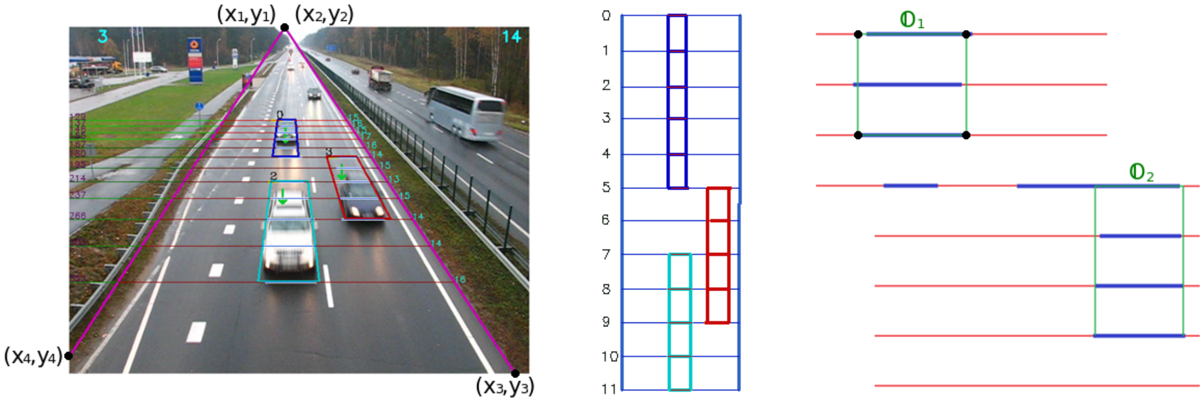
- **Occlusion detection** improves the accuracy of object detection by analyzing the changes in the interval's width during its existence.
- **Detection of shadows** inside existing intervals improves accuracy when pronounced shadows of objects incorrectly widen the corresponding intervals.

- **Vehicle detection at night** on poorly lit roads is challenging for any video-based detector. If one uses a camera with a fast shutter (1/24500 seconds), the IoVDL can detect the headlights of vehicles.
- **Detection of rapid lighting changes** allows IoVDL method to quickly change otherwise slowly adaptive background.
- **Increased motion sensitivity** is accomplished by combining several adjacent rows of pixels to construct the detector. This approach improves the detection of objects that are similar to the background.

2.3. Extension of the IoVDL Method for Object Characterization

Combining intervals into objects. The visual field of IoVDL is limited to a small region in the frame. An extension of IoVDL is proposed, which obtains information from a much broader area of the frame, while still processing only few detection lines. Extended IoVDL can track the objects and measure their speed and dimensions. The positive features of the interval-based approach are maintained, and the computing power is significantly reduced compared to other object tracking methods that process all pixels in the tracking area.

Several detection lines are placed in the frame (Fig. 2.2 a) so that each line is perpendicular to the movement of objects. The detection lines are parallel and equidistant to each other on the actual world plane (Fig. 2.2 b).



(a) (b) (c)

Fig. 2.2. Combining intervals from several virtual detectors.

Different detection lines may contain intervals that are created by the same moving object. Such intervals are found and combined into a rectangular virtual object ($\mathbb{O}_1$ and $\mathbb{O}_2$ in Fig. 2.2 c). Adding new and deleting old intervals tracks the object's movement through the detectors.

The extended IoVLD method uses projective transform to analyze objects in world coordinates. At each frame, only the intervals have to be transformed. Only two coordinates are needed to describe an interval, so the transformation step of the method is computationally efficient.

Object characterization. In the world coordinates, all existing virtual objects are analyzed in order to obtain the following parameters.

- **Direction** of the object's movement is inferred from the position of newly added intervals.
- **Speed** is determined by how quickly the object has reached the different detection lines.
- **Size**. The length of the object is acquired by multiplying its speed with the lifespan of its intervals. Width and height are found by analyzing the length of object's intervals.
- **Class** of some objects can be determined by using the already measured parameters. In the example of ITS, one can use the size to distinguish compact vehicles and trucks.

2.4. RNN Based Virtual Detection Line

This subsection develops RNN-VLD method. It combines the detection line approach with a recurrent neural network, resulting in a computationally efficient deep learning-based object detector.

Architecture. The detector is a line which is perpendicular to the movement of objects. This line is forwarder trough a recurrent neural network. The reduced architecture of the proposed artificial neural network is depicted in Fig. 2.3.

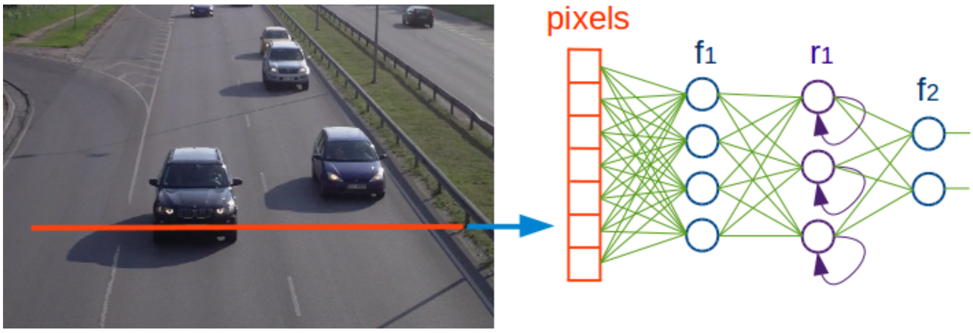


Fig. 2.3. RNN based virtual detection line (RNN-VDL).

The first and last layers of the proposed network are fully connected layers. The hidden layer consists of the LSTM cell, which considers $S = 40$ previous frames to compute its output. All activation functions in the network are linear rectifiers $\rho(x) = max(0, x)$. The goal of training such model is to learn all the weights, so that the network returns a preferred class for each sequence of input vectors $\mathbf{x}^{(t-S)}, \mathbf{x}^{(t-S+1)}, \mathbf{x}^{(t-S+2)} ... \mathbf{x}^{(t)}$.

Data labeling modes. The output class labels of RNN-VDL may represent several possible events on the detection line. In this paper, three labeling modes are proposed. In the first mode, the output label corresponds to the current number of objects on the detection line. The labels of the second mode show how many objects have left the line at the current frame. In mode 3, the label of any sequence is determined by counting how many objects have left the line in the last $S/2$ frames.

3. TRAINING DATASET ACQUISITION METHODS

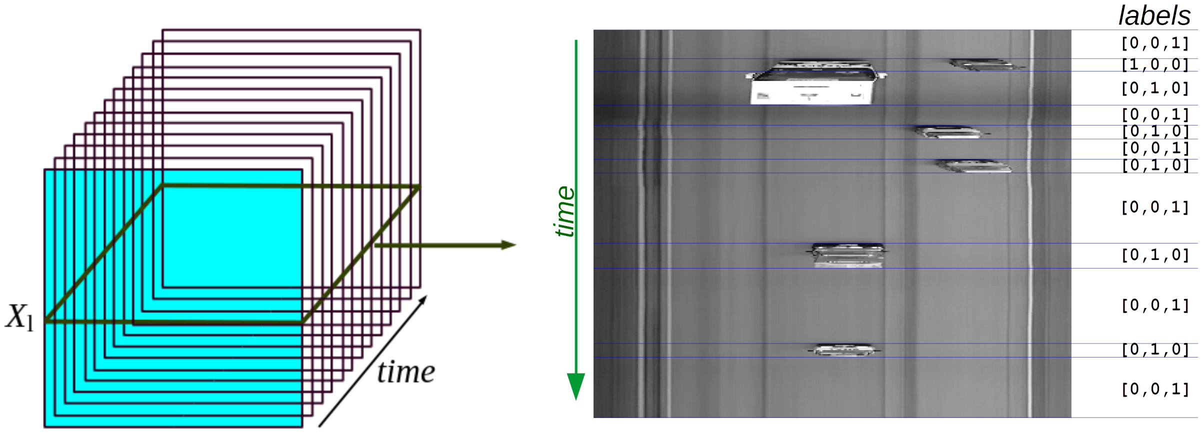
The practical use of the proposed machine learning-based object detector RNN-VDL depends on the availability of labeled training data. It is important to acquire new labeled data quickly and cheaply, so this section proposes three methods for the acquisition of such data, beginning with manual labeling and ending with a fully automatic approach.

3.1. Format of the Labeled Data for RNN-VDL

The required labeled data for the proposed RNN-VDL method consists of two matrices – a spatio-temporal training image and a matrix of one-hot encoded labels.

A row of the spatio-temporal image corresponds to the pixels on the detection line at a specific frame. It is shown in Fig. 3.1 a, where the video is depicted as a box in x, y, and time dimensions. The training image is marked as a plane that slices the video box at the row X_1 and is parallel to the x-time plane.

An example of the training image and its one-hot encoded labels are shown in Fig. 3.1 b. In the example, regions with different labels are separated by thin lines. The first labeling mode is used, so the label of each row encodes the number of moving objects on the detection line in the corresponding frame. While training the RNN model, this image is split into S frame long overlapping sequences. Each sequence has an appropriate label that indicates the situation on the detection line at the last frame of the sequence.



(a)

(b)

Fig. 3.1. The desired result of labeling.

(a) a slice of the input video that corresponds to a spatio-temporal training image; (b) a spatio-temporal image of vehicles that have crossed the detection line and the corresponding labels of its rows.

3.2. Proposed GUIs for Data Labeling

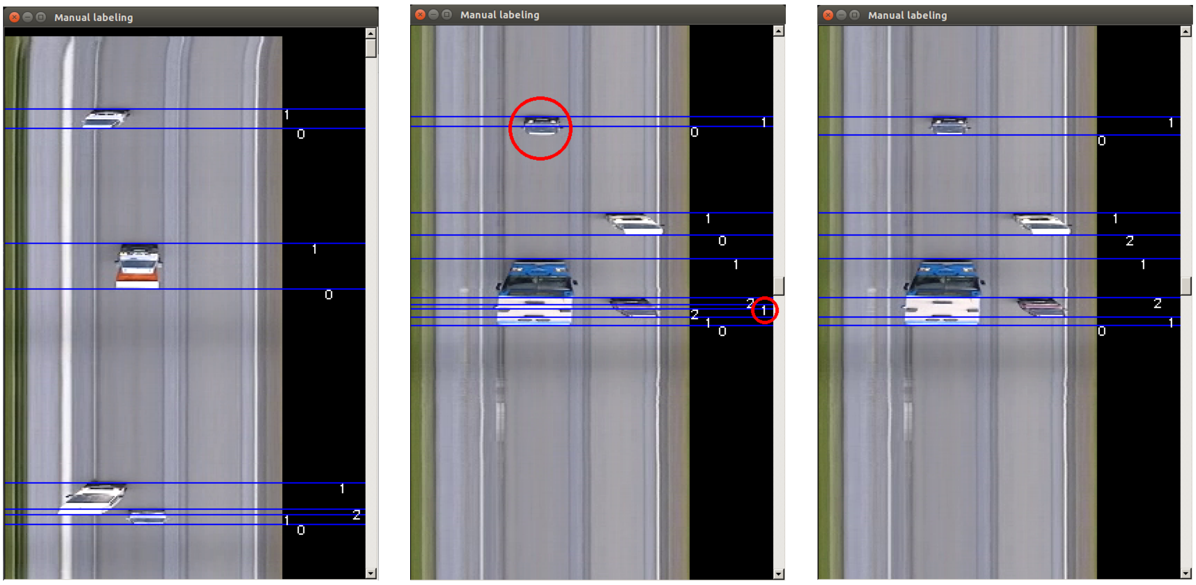
The consecutive frames in a video are often similar and might often have the same label. So a good interface would allow the user to quickly determine and assign the same label to the consecutive frames. This Thesis develops new graphical user interface-based methods that facilitate the creation of labeled data. Both GUI-based labeling approaches are implemented in Python 3 language. They use *OpenCV*, *numpy*, and *PyQt4* libraries.

Manual labeling uses a specific graphical user interface to make it easy for the user to go through the videos and manually annotate each frame. At each frame, the user has to inspect the detection line and press a number on the keyboard. A label corresponding to this number is converted to a one-hot vector, which is appended to the matrix of labels. The current detection line becomes a new row in the spatio-temporal training image. After the keypress, the GUI displays the next frame of a video and awaits new user input.

This same method can be used for non-detection line based labeling. In that case, the user has to count all the objects in the frame, not only the ones on the detection line.

Semi-automatic labeling approach takes advantage of the specifics of the RNN-VDL object detector and makes the labeling of data even faster than the manual approach.

First, it tries to label the data automatically, by using a background subtraction approach. The resulting thresholded spatio-temporal image undergoes further processing (morphological closing, followed by opening) in order to reduce the noise. The number of continuous white regions at each row of the resulting image becomes the label for this row (labeling mode 1).



(a) (b) (c)

Fig. 3.2. GUI for inspection and correction of errors of automatic labeling.

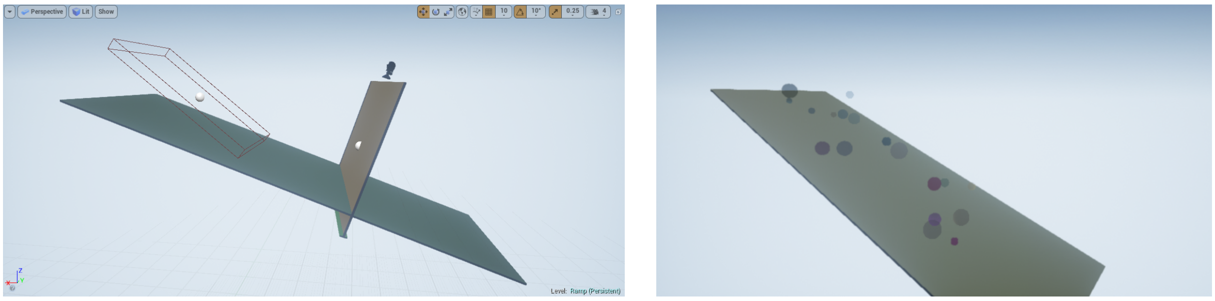
The next labeling stage consists of GUI (Fig. 3.2) that allows fast manual detection and correction of the errors made in the automatic stage. Fig. 3.2 a depicts a region of the spatio-temporal image which was successfully labeled by the automatic process. Each blue line in Fig. 3.2 depicts a frame at which the label on the detection line changes. The number on the right of each dividing line is the corresponding label of the rows under this line.

Another region depicted in Fig. 3.2 b contains two errors made in the automatic labeling step. The developed GUI allows users to scroll through the whole labeled spatio-temporal image and correct the errors with the mouse and keyboard. The users see the spatio-temporal image, therefore they can inspect many frames simultaneously. In this way, the speed of the manual part of the labeling is significantly increased compared to the methods where each frame is labeled separately.

3.3. Generation of Labeled Data Using a 3D Game Engine

Recent literature [56]–[59] indicates another potential solution for dealing with the lack of labeled data, which is to train the DL models on synthetic data. To see, if this approach is applicable for the RNN-VDL method, a simulation environment was created in the Unreal Engine 4. This environment creates data for a simple task – the detection of moving spheres.

The setup of this simulation is shown in Fig. 3.3 a. It consists of an inclined base plate, sphere generator object, detection plate, lighting element, and camera. When simulation is running (Fig. 3.3 b), spheres are generated at random moments in time and at random points inside the generator's volume. The new spheres fall to the base plate and roll towards the detection plate. This plate allows the spheres to pass through, but it detects the collisions. This information is used to label each frame. The labels are saved along with the frames from the camera.



(a) (b)

Fig. 3.3. Generation of training data with a 3D game engine.

This proof of concept shows that training data for RNN-VDL can be generated automatically. However, specific human skills and time are required to create realistic environments for real-world applications. Since the computer graphics are getting more and more lifelike, in the near future this approach may have an increased role in the machine learning-based computer vision.

4. APPLICATION AND EXPERIMENTAL RESULTS

This section aims to test if the methods developed in Section 2 are computationally efficient and accurate in the practical ITS related applications.

4.1. Efficiency of Developed Methods

IoVDL is implemented in C++ programming language. It uses image processing library *OpenCV* for video input/output and common image processing tasks. In order to prove the efficiency of the proposed IoVDL algorithm, it is implemented on a Raspberry Pi Zero single-board computer. The Zero is the smallest (65 × 30 × 5 mm) and the cheapest (∼5 EUR) Raspberry Pi model, with a 1 GHz single-core CPU and 512 MB RAM.

Figure 4.1 shows how many frames per second the IoVDL method can process while running on Raspberry Pi Zero. During the test, the frames are captured in real time by a Raspberry camera module V2. Frame acquisition takes a significant portion of the running time, which depends on the frame resolution. Therefore, Fig. 4.1 shows the speed measurements at three different resolutions.

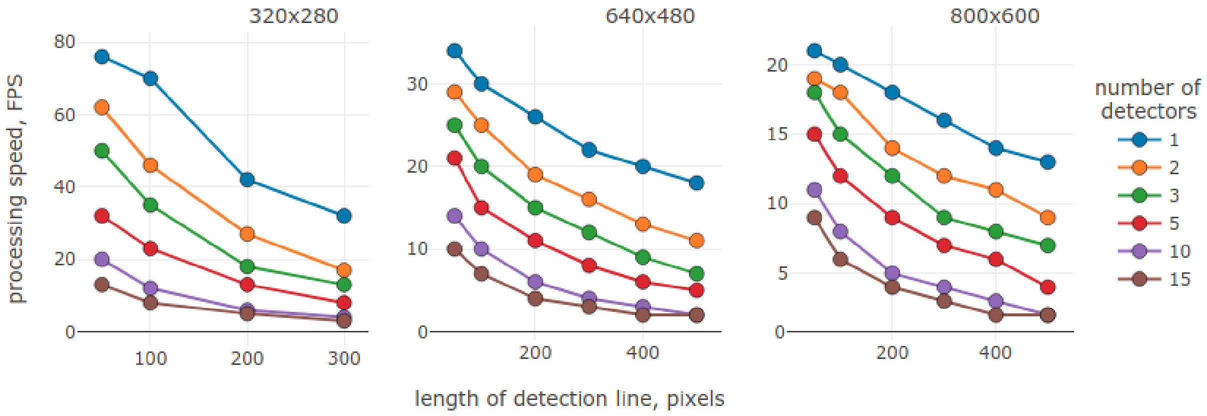


Fig. 4.1. IoVDL processing speed on Raspberry Pi Zero.

RNN-VDL is implemented in Python 3, using the TensorFlow framework. The size parameters of the network, such as the number of layers and the depth in time S, are the essential speed affecting parameters. The measurements show a linear dependency between S and the time needed to process a single frame F. With $S = 50$, $F \approx 0.004$ seconds, when $S = 100$, $F \approx 0.006$ s, with $S = 150$, $F \approx 0.008$ s. The choice of S depends on the specific task. Slow moving objects and high frame rate of the input video may require RNNs with deeper memory.

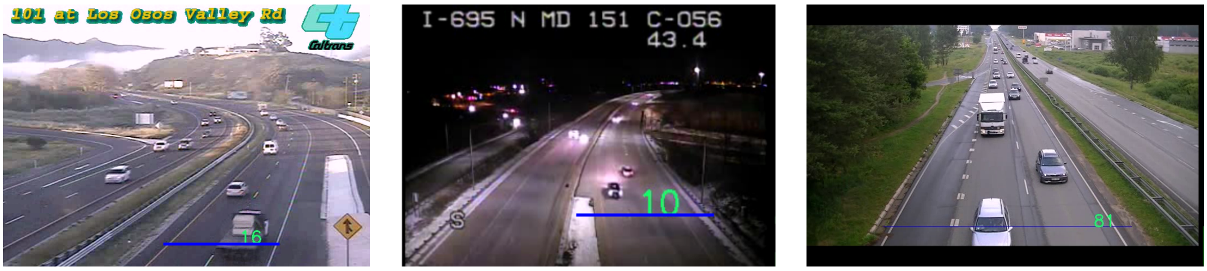
In order to compare the inference speed of RNN-VDL and that of the state-of-the-art object detector YOLO, both methods were tested on the same single processor. The [60] implementation of YOLO's fast version *tiny-yolo* processed one image in 0.34 seconds, while the proposed

RNN-VDL method needed 0.0035 seconds (when $S = 40$).

RNN-VDL is also successfully implemented and used on Raspberry Pi 3 Model B computer, which has Quad Core 1.2 GHz Broadcom BCM2837 64 bit CPU and 1 GB of RAM.

4.2. Object Counting Accuracy

Vehicle counting accuracy of the RNN-VDL and IoVDL methods was tested on the same videos (example frames are shown in Fig. 4.2).



(a) (b) (c)

Fig. 4.2. Videos used for testing the RNN-VDL and IoVDL methods.

Test results are depicted in Table 4.1, which shows the precision ($TP/(TP + FP)$), the recall ($TP/(TP + FN)$), and the F_1 score for each test:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{4.1}$$

The test videos in column *Vehicles 1* were acquired by the same highway cameras that recorded the training videos for the RNN-VDL (Fig. 4.2 a,b). According to the F_1 score, the RNN-VDL method performs better on these videos than the IoVDL.

When the IoVDL algorithm was developed, it was mostly designed to work on videos that were similar in resolutions and quality to the one used in the test *Vehicles 2* (Fig. 4.2 c). Unsurprisingly, IoVDL performs well on this test.

On the contrary, the video *Vehicles 2* is significantly different from videos used to train RNN-VDL. Nevertheless, results show that the neural network has learned some generalization of vehicle detection. As with many deep learning approaches, this capability of the RNN-VDL model might be improved by extending the training dataset with videos from different sources.

As was discussed in Subsection 2.1, there are other efficient but less flexible methods that use a small detection region to count objects in a video. The papers proposing these methods report similar test results. In [54], the counting accuracy for daytime videos varies from 70.31 % to 98.39 %. The method in [53] has the accuracy of 97.73 %. In [51], the precision of the method is 86 %–96 %, while the recall is 89 %–96 %. Since different videos are used to evaluate all the methods above, the accuracy rates are indicative but not conclusively comparable.

Table 4.1

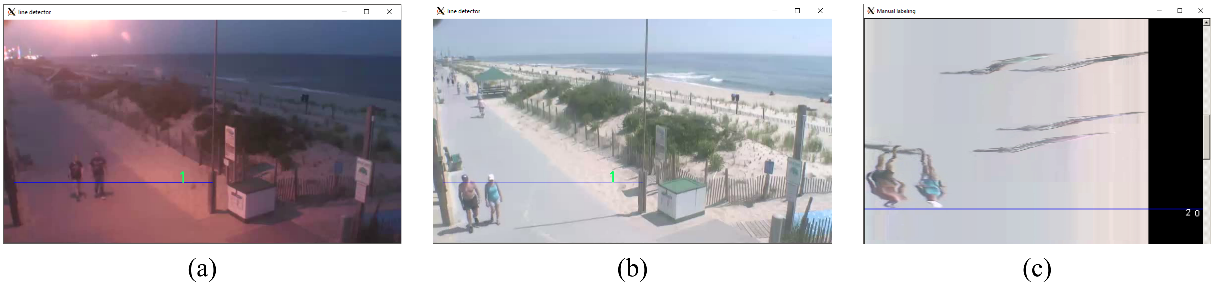
Object Detection Results

Camera	Vehicles 1		Vehicles 2		People
	RNN-VDL	IoVDL	RNN-VDL	IoVDL	RNN-VDL
Ground truth count	4238	4238	121	121	285
Predicted count	4278	3693	108	120	245
True Positives	4059	3623	103	119	209
False Positives	219	70	5	1	36
False Negatives	179	615	18	2	76
Precision	0.949	0.981	0.954	0.992	0.853
Recall	0.958	0.855	0.851	0.983	0.733
F_1 score	0.953	0.914	0.900	0.987	0.788

Additional tests of IoVDL on different videos indicate the following properties of the method:

- it is robust to changes in lighting;
- it does not count the shadows of clouds and trees as vehicles;
- it is not significantly affected by heavy rain;
- its counting accuracy is influenced by the placement of the camera;
- it works on roads with different number and direction of lanes (or no lanes at all).

People counting test demonstrates the versatility of RNN-VDL. The same network architecture as was used in vehicle detection is retrained on the videos with people (Fig. 4.3 a,b).



(a) (b) (c)

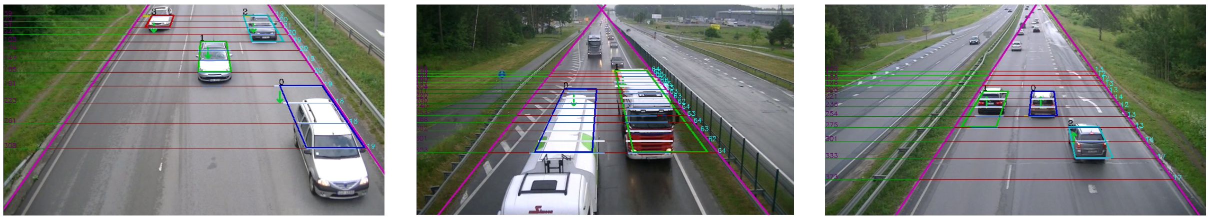
Fig. 4.3. Detection of people using RNN.

In order to show the additional benefits of using the recurrent network, the people counting task is modified so that only people going in one direction are counted. The corresponding labeling can be seen in Fig. 4.3 c. This example is a snapshot of the semi-automatic labeling process proposed in Subsection 3.2.

The people counting results are depicted in column *People* of Table 4.1. In the used videos, the intensity of the flow of people is similar in both directions. Therefore, if the detection line could not differentiate between the directions, there would be a lot of false positive detection instances.

4.3. Vehicle Detection and Classification with extended IoVDL

The developed extended IoVDL method can detect object’s parameters and classify the object. This classification capability is tested on several videos with different weather conditions and different camera positions. Example frames from the tests are shown in Fig. 4.4.



(a) (b) (c)

Fig. 4.4. Tests of extended IoVDL.

Table 4.2 compares the vehicle detection performance of a single IoVDL detector and the extended IoVDL consisting of 12–16 detection lines. The middle detection line of the extended method is chosen as a single line detector. Additionally, Table 4.2 depicts the counting result of a more straightforward combination of all detection lines, where the final count of vehicles is the mean value of the counts returned by the separate detectors. The comparison of results shows that the proposed extension of IoVDL returns closer counts to ground-truth than a single IoVDL detector.

Table 4.2 also depicts the classification accuracy of the algorithm. In the test videos, the algorithm correctly classifies 88 % of the trucks (TP), while the number of false positives is small.

Table 4.2

Traffic Monitoring with Extended IoVDL

Video	Ground-truth		Detection			Classification	
	all vehicles	trucks	Single IoVDL	IoVDL average	Extended IoVDL	TP trucks	FP trucks
a)	32	4	30	30	32	4	0
b)	122	8	121	124	122	8	0
c)	159	8	168	164	158	7	1
d)	43	0	45	44	44	0	0
e)	221	12	232	235	231	7	0
f)	208	5	209	209	209	5	1
g)	264	12	233	239	239	12	3
h)	117	4	118	118	116	4	1
i)	88	7	92	93	88	6	0

5. CONCLUSION

This Thesis researched and developed video processing methods for moving object detection. One of the target applications of such methods is vehicle detection by intelligent transportation systems. The analysis of different road sensors in Subsection 2.1 indicated that in the road setting the practical applicability of different detection methods depends on their ability to adapt to the changing conditions and their cost. Accordingly, this Thesis aimed to improve the computationally efficient detection of moving objects in a video. Six tasks were defined to achieve the set aim.

1. To perform a review of literature about object detection in images and videos. This task was accomplished in Section 1. The literature review acknowledged that state-of-the-art object detection in images is currently achieved by deep learning methods. The recurrent neural networks were identified as having great potential in video processing. However, current deep learning-based methods require significant computational resources.

2. To identify efficient approaches for detection of moving objects in videos. It was accomplished in Subsection 2.1, which showed that a common trait of efficient methods is a significant reduction of the pixels that are processed in each frame. The identified drawback of these methods is that objects have to enter the virtual detector in order to be detected, so these methods are applicable when the objects of interest do not deviate from their usual trajectories.

3. To develop efficient object detection methods with improved capabilities compared to the existing efficient methods. The IoVDL method was developed in Subsection 2.2. In the vehicle detection context, the interval approach does not require unchanging trajectories of vehicles. The IoVDL is applicable on the roads where lanes are not specified or their number changes depending on the time of the day. IoVDL is also capable of separately detecting several objects that enter the virtual detector simultaneously.

The extended IoVDL method is able to capture the object's movement through several detectors. Additionally, the method characterizes the object by measuring its speed and size parameters and uses these measurements to classify the object. The description of the extended IoVDL showed that it is still computationally efficient, since only the pixels of detection lines are processed.

Another novel video processing approach for moving object detection was proposed in Subsection 2.4. The RNN-VDL method combines the idea of the efficient detection line and a recurrent neural network.

The use of the powerful machine learning technique indicated that the RNN-VDL method could be used in even more versatile applications than IoVDL. However, this approach requires a significant amount of labeled data for training.

4. To develop data acquisition methods and acquire the data needed for the development and testing of proposed methods. This was accomplished in Section 3. A GUI based-approach was developed to speed up the manual labeling of each frame. It was concluded that this approach could be used to label data for different kinds of detection methods, not only the detection line-based ones.

Another – semi-automatic – method was developed. Its practical merit was verified by creating labeled datasets that were used for training and testing the RNN-VDL method. The use of a spatio-temporal image to detect and mark different events on the detection line turned out to be a much faster labeling approach than analyzing each frame separately, which confirms Statement 4 of the Thesis.

5. To implement and conduct experimental research on the proposed methods. All the proposed detection and labeling methods were implemented and tested on real videos (Section 4).

The tests showed that vehicle detection accuracy of IoVDL is comparable to those of the existing efficient methods. Test results in combination with the description of the method in Subsection 2.2 prove Statement 1.

The extended IoVDL method turned out to be a more precise vehicle counter than a single IoVDL detector. The object characterization capability of extended IoVDL was verified by vehicle classification test. The results and the fact that extended IoVDL is able to track and characterize objects in a wide area of the video while processing much fewer pixels than conventional tracking methods prove Statement 2.

The RNN-VDL method is even more broadly applicable than IoVDL. After training RNN-VDL with different data, the same architecture could be used for vehicle and people detection. The processing of a single line in the frame could even differentiate between people crossing the line in different directions, which confirms Statement 3.

6. To draw conclusions about the results of this Thesis The main conclusion about object detection methods developed in this paper is that their properties and test results demonstrate that the aim of this Thesis is successfully achieved – the developed methods improve the efficient detection of moving objects in a video. In addition, the accomplished tasks prove four statements defined at the beginning of this Thesis.

The contributions of this Thesis might be of interest to researchers as well as practitioners in the field of computer vision. The developed data labeling approaches contribute to the field of deep learning, where the lack of labeled data is one of the greatest current challenges. The developed IoVDL and RNN-VDL methods introduce a novel way of making computationally efficient vision-based moving object detectors. These approaches were mostly tested and used on the roads; however, the flexibility of the methods, especially that of the RNN-VDL, opens up the possibilities for their use in other practical applications.

BIBLIOGRAPHY

[1] Saxena, A., Driemeyer, J., Ng, A. Y. Robotic grasping of novel objects using vision. In: *The International Journal of Robotics Research*, 2008, 27(2), 157–173.

[2] Brosnan, T., Sun, D.-W. Improving quality inspection of food products by computer vision – a review. In: *Journal of food engineering*, 2004, 61(1), 3–16.

[3] Maglogiannis, I., Doukas, C. N. Overview of advanced computer vision systems for skin lesions characterization. In: *IEEE transactions on information technology in biomedicine*, 2009, 13(5), 721–733.

[4] Kang, J., Hsu, C.-H., Wu, Q., Liu, S., Coster, A. D., Posner, B. A., Altschuler, S. J., Wu, L. F. Improving drug discovery with high-content phenotypic screens by systematic selection of reporter cell lines. In: *Nature biotechnology*, 2016, 34(1), 70–77.

[5] Parkhi, O. M., Vedaldi, A., Zisserman, A. Deep Face Recognition. In: *BMVC 1(3).* 2015, p. 6.

[6] Junior, J. C. S. J., Musse, S. R., Jung, C. R. Crowd analysis using computer vision techniques. In: *IEEE Signal Processing Magazine*, 2010, 27(5), 66–77.

[7] Coifman, B., Beymer, D., McLauchlan, P., Malik, J. A real-time computer vision system for vehicle tracking and traffic surveillance. In: *Transportation Research Part C: Emerging Technologies*, 1998, 6(4), 271–288.

[8] Li, H., Shen, C. Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs. In: *arXiv preprint arXiv:1601.05610*, 2016, p. 17.

[9] **Kadiķis**, R. *Labeled dataset for training and testing virtual detection-line based object detectors* [online]. Riga: EDI, 2017 [viewed 28 November 2017]. Available from: http://vault.edi.lv/index.php/s/yxGPtI2Jw2yWhLR

[10] Freivalds, K., **Kadiķis**, R., Greitāns, M. System and method for video-based vehicle detection. Patent (EP 2700054 B1), Apr. 2015.

[11] **Kadiķis**, R., Freivalds, K. Efficient video processing method for traffic monitoring combining motion detection and background subtraction. In: *Proceedings of the Fourth International Conference on Signal and Image Processing 2012 (ICSIP 2012).* Springer, 2013, 131–141.

[12] **Kadiķis**, R., Freivalds, K. Vehicle classification in video using virtual detection lines. In: *Sixth International Conference on Machine Vision (ICMV 13).* International Society for Optics and Photonics 2013, 90670Y–90670Y.

[13] **Kadiķis**, R. Registration method for multispectral skin images. In: *Radioelektronika (RADIOELEKTRONIKA), 2015 25th International Conference.* IEEE 2015, 232–235.

[14] Nauris, D., **Kadiķis**, R., Nesenbergs, K. Vehicle type and license plate localisation and segmentation using FCN and LSTM. In: *New Challenges of Economic and Business Development 2017, Proceedings of Reports.* University of Latvia 2017, 132–140.

[15] Tamošiūnas, M., Jakovels, D., Ļihačovs, A., Kilikevičius, A., Baltušnikas, J., **Kadiķis**, R., Šatkauskas, S. Application of fluorescence spectroscopy and multispectral imaging for non-invasive estimation of GFP transfection efficiency. In: *8th International Conference on Advanced Optical Materials and Devices.* International Society for Optics and Photonics 2014, 94210M–94210M.

[16] Jakovels, D., Saknite, I., Bliznuks, D., Spigulis, J., **Kadiķis**, R. Benign-A typical nevi discrimination using diffuse reflectance and fluorescence multispectral imaging system. In: *International Conference on BioPhotonics (BioPhotonics).* IEEE 2015, 1–4.

[17] Tamošiūnas, M., **Kadiķis**, R., Saknīte, I., Baltušnikas, J., Kilikevičius, A., Lihachev, A., Petrovska, R., Jakovels, D., Šatkauskas, S. Noninvasive optical diagnostics of enhanced green fluorescent protein expression in skeletal muscle for comparison of electroporation and sonoporation efficiencies. In: *Journal of biomedical optics*, 2016, 21(4), 045003–045003.

[18] Bellucci, P., Cipriani, E. Data accuracy on automatic traffic counting: the SMART project results. In: *European transport research review*, 2010, 2(4), 175–187.

[19] Yu, X., Sulijoadikusumo, G., Li, H., Prevedouros, P. Reliability of Automatic Traffic Monitoring with Non-Intrusive Sensors. In: *ICCTP 2011: Towards Sustainable Transportation Systems* 2011, 4157–4169.

[20] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. In: *International Journal of Computer Vision (IJCV)*, 2015, 115(3), 211–252.

[21] Karpathy, A. *What I learned from competing against a ConvNet on ImageNet* [online]. [viewed 2 August 2017]. Available from: http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

[22] Otsu, N. A threshold selection method from gray-level histograms. In: *IEEE transactions on systems, man, and cybernetics*, 1979, 9(1), 62–66.

[23] Rodríguez, T., García, N. An adaptive, real-time, traffic monitoring system. In: *Machine Vision and Applications*, 2010, 21(4), 555–576.

[24] Ji, X., Wei, Z., Feng, Y. Effective vehicle detection technique for traffic surveillance systems. In: *Journal of Visual Communication and Image Representation*, 2006, 17(3), 647–658.

[25] Gupte, S., Masoud, O., Martin, R. F., Papanikolopoulos, N. P. Detection and classification of vehicles. In: *IEEE Transactions on intelligent transportation systems*, 2002, 3(1), 37–47.

[26] Cheung, S.-C. S., Kamath, C. Robust background subtraction with foreground validation for urban traffic video. In: *EURASIP Journal on Advances in Signal Processing*, 2005, 2005(14), 2330–2340.

[27] Stauffer, C., Grimson, W. E. L. Learning patterns of activity using real-time tracking. In: *IEEE Transactions on pattern analysis and machine intelligence*, 2000, 22(8), 747–757.

[28] Harris, C., Stephens, M. A combined corner and edge detector. In: *Alvey vision conference.* 15 (50), Manchester, UK 1988, 147–152.

[29] Mikolajczyk, K., Schmid, C. An affine invariant interest point detector. In: *Computer Vision—ECCV 2002*, 2002, 128–142.

[30] Lowe, D. G. Distinctive image features from scale-invariant keypoints. In: *International journal of computer vision*, 2004, 60(2), 91–110.

[31] Matas, J., Chum, O., Urban, M., Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. In: *Image and vision computing*, 2004, 22(10), 761–767.

[32] Rosten, E., Drummond, T. Fusing points and lines for high performance tracking. In: *ICCV 2005. 10th IEEE International Conference on Computer Vision.* IEEE 2005, 1508–1515.

[33] Rosten, E., Drummond, T. Machine learning for high-speed corner detection. In: *Computer Vision–ECCV 2006*, 2006, 430–443.

[34] Lowe, D. G. Distinctive image features from scale-invariant keypoints. In: *International journal of computer vision*, 2004, 60(2), 91–110.

[35] Bay, H., Tuytelaars, T., Van Gool, L. Surf: Speeded up robust features. In: *Computer vision–ECCV 2006*, 2006, 404–417.

[36] Ojala, T., Pietikäinen, M., Harwood, D. A comparative study of texture measures with classification based on featured distributions. In: *Pattern recognition*, 1996, 29(1), 51–59.

[37] Rublee, E., Rabaud, V., Konolige, K., Bradski, G. ORB: An efficient alternative to SIFT or SURF. In: *ICCV 2011. IEEE International Conference on Computer Vision.* IEEE 2011, 2564–2571.

[38] Girshick, R., Donahue, J., Darrell, T., Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014, 580–587.

[39] Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems.* 2015, 91–99.

[40] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016, 779–788.

[41] Karnowski, J. *AlexNet + SVM* [online]. [viewed 5 December 2017]. Available from: https://jeremykarnowski.files.wordpress.com/2015/07/alexnet2.png

[42] Lipton, Z. C., Berkowitz, J., Elkan, C. A critical review of recurrent neural networks for sequence learning. In: *arXiv preprint arXiv:1506.00019*, 2015, p. 38.

[43] Hochreiter, S., Schmidhuber, J. Long short-term memory. In: *Neural computation*, 1997, 9(8), 1735–1780.

[44] Kastrinaki, V, Zervakis, M., Kalaitzakis, K. A survey of video processing techniques for traffic applications. In: *Image and vision computing*, 2003, 21(4), 359–381.

[45] Yue, Y. A traffic-flow parameters evaluation approach based on urban road video. In: *Int. J. Intell. Eng. Syst*, 2009, 2(1), 33–39.

[46] Mithun, N. C., Rashid, N. U., Rahman, S. M. Detection and classification of vehicles from video using multiple time-spatial images. In: *IEEE Transactions on Intelligent Transportation Systems*, 2012, 13(3), 1215–1225.

[47] Lee, G.-G., Kim, B.-s., Kim, W.-Y. Automatic estimation of pedestrian flow. In: *ICDSC 2007. First ACM/IEEE International Conference on Distributed Smart Cameras.* IEEE 2007, 291–296.

[48] Kim, J.-W., Choi, K.-S., Choi, B.-D., Ko, S.-J. Real-time vision-based people counting system for the security door. In: *International Technical Conference on Circuits/Systems Computers and Communications.* 2002, 1416–1419.

[49] Li, L., Han, S., Asama, H., Duan, F. An automatic parts detection system based on computer vision. In: *35th Chinese Control Conference (CCC).* IEEE 2016, 9493–9498.

[50] Nie, Z., Hung, M.-H., Huang, J. A Novel Algorithm of Rebar Counting on Conveyor Belt Based on Machine Vision. In: *Journal of Information Hiding and Multimedia Signal Processing*, 2016, 7(2), 425–437.

[51] Anan, L. Video vehicle detection algorithm based on virtual-line group. In: *APCCAS 2006. IEEE Asia Pacific Conference on Circuits and Systems.* IEEE 2006, 1148–1151.

[52] Michalopoulos, P. G. Vehicle detection video through image processing: the autoscope system. In: *IEEE Transactions on vehicular technology*, 1991, 40(1), 21–29.

[53] Zhang, G., Avery, R., Wang, Y. Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2007, (1993), 138–147.

[54] Lei, M., Lefloch, D., Gouton, P., Madani, K. A video-based real-time vehicle counting system using adaptive background method. In: *SITIS'08. IEEE International Conference on Signal Image Technology and Internet Based Systems.* IEEE 2008, 523–528.

[55] Ha, D., Lee, J.-M., Kim, Y.-D. Neural-edge-based vehicle detection and traffic parameter extraction. In: *Image and vision computing*, 2004, 22(11), 899–907.

[56] Lerer, A., Gross, S., Fergus, R. Learning Physical Intuition of Block Towers by Example. In: *arXiv preprint arXiv:1603.01312*, 2016, p. 11.

[57] Li, W., Azimi, S., Leonardis, A., Fritz, M. To Fall Or Not To Fall: A Visual Approach to Physical Stability Prediction. In: *arXiv preprint arXiv:1604.00066*, 2016, p. 20.

[58] Gaidon, A., Wang, Q., Cabon, Y., Vig, E. Virtual worlds as proxy for multi-object tracking analysis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016, 4340–4349.

[59] Sadeghi, F., Levine, S. CAD2RL: Real single-image flight without a single real image. In: *arXiv preprint arXiv:1611.04201*, 2016, p. 12.

[60] *Darkflow* [online]. [viewed 6 July 2017]. Available from: https://github.com/thtrieu/darkflow