UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Asmar Hasanova

Pedestrian Detection and Tracking in Urban Context Using a Mono-Camera

Master's Thesis (30 ECTS)

Supervisor(s): Amnir Hadachi, PhD

Tartu 2017

Pedestrian detection and tracking in urban context using a mono-camera

Abstract:

Pedestrian detection and tracking are one of the important aspects in Advanced Driver Assistance Systems. These systems help to avoid dangerous situations, by guiding drivers and warning them about the upcoming risks. The main ideas of pedestrian detection and tracking are to detect pedestrians, while they are in the secure zone, and predict their position and direction.

The goal of this thesis is to examine possible methods and based on these, to develop a good pedestrian detection and tracking algorithm.

The solution developed in this thesis, focuses on accurately detecting and tracking a pedestrian. In order to estimate the accuracy of the system, obtained results will be compared to the existing solutions.

Keywords: Pedestrian detection, pedestrian tracking, Histogram of Oriented Gradients, Support Vector Machine, Kalman filter, feature detection.

CERCS: P170

Jalakäijate tuvastus ja jälgimine linna kontekstis monokaameraga

Lühikokkuvõte:

Jalakäijate tuvastus ja jälgimine on üks tähtsamaid aspekte edasijõudnud sõitja abisüsteemides. Need süsteemid aitavad vältida ohtlikke olukordi, juhendades sõitjaid ja hoiatades ettetulevate riskide eest. Jalakäijate tuvastuse ja jälgimise põhiideed on tuvastada jalakäijad siis, kui nad on turvalises tsoonis ja ennustada nende asukohta ja suunda.
Selle lõputöö eesmärk on uurida võimalikke meetodeid ja arendada nende põhjal hea algoritm jalakäijate tuvastuseks ja jälgimiseks.
Selles lõputöös arendatud lahendus keskendub jalakäija täpsele tuvastamisele ja jälgimisele. Süsteemi täpsuse hindamiseks on saadud tulemusi võrreldud olemasolevate lahendustega.

Võtmesõnad: Jalakäijate tuvastus, jalakäijate jälgimine, orienteeritud kallete histogramm, abivektori masin, Kalmani filter, funktsionaalsuse tuvastamine

CERCS: P170

Acknowledgement

Foremost, I would like to thank my supervisor Amnir Hadachi, for his supportive guidance to writing this thesis, for motivation, and for being patient.
Besides my supervisor, I would also like to thank Artjom Lind, for his help.
Also, I want to thank my friends, especially Olha Shepelenko, for her help and motivation.
Finally, I want to thank my parents for their support and understanding.

Contents

Chapter 1: Introduction

1.1 General view and Background

Based on World Health Organization [1], each year, approximately 1.25 million people die from road accidents, and half of the victims are “vulnerable road users” (VRU) such as pedestrians, cyclists, and motorcyclists. More than 50 million people get various non-fatal injuries, which result in different disabilities. Currently, traffic accidents are in 10th place among [2], the leading causes of death. Despite, some driver-assist technologies, such as anti-lock brakes, electronic stability control, traction control, which have been operating in the background without driver awareness have existed already for decades, the road accidents remain one of the actual problems. By 2030, without preventive action, the number of road accidents predicted to increase, reaching the 7th place amid top reasons of death.

To mitigate this problem and decrease the number of the victims of traffic accidents, research has been conducted through applying intelligent systems of adoption Advanced Driver Assistance Systems (ADAS), in terms of providing guidance to the drivers and warnings in dangerous situations; contributing decision support; and even taking over control of the vehicle on extreme occasions. ADAS work principle depends on input information from various sources like radar, LIDAR, in-car networking, image processing, computer vision, other vehicles (Vehicle-to-vehicle, Vehicle-to-Infrastructure) etc.

By reason of, consumer interest in safety applications and demand for ADAS are expected to increase over the next decade. As an example, to provide each vehicle with forward-collision warning systems and autonomous emergency braking systems will be mandated by the European Union and the United States, by the year 2020. Customers become more interested in ADAS systems, especially those assist with parking or monitoring blind spots, as is recently proposed in McKinsey [3] research.

Currently, there are a number of ADAS offerings, for examples to the latest ones, Adaptive Cruise Control, Automatic Braking, Automatic Parking, Blind Spot Monitoring, Collision Avoidance Systems, Night Vision, Driver Drowsiness Detection, Pedestrian Detection etc.

Pedestrian Protection Systems

Pedestrian Protection Systems (PPSs) are a particular type of ADAS, which became popular research field in many developed and also developing countries, for mitigating traffic

accidents. It can be considered that the PPSs systems consist of two parts: Passive Safety Systems (based on Vehicle Design) and Active Safety Systems (based on Pedestrian Detection).

Passive Safety Systems

In case of an accident, the design of the vehicle plays a great role of the severity of the injury that a pedestrian endures. Collision with small vehicles can cause an impact of below the pedestrian's centre of gravity, with probability the pedestrian falls over the bumper, and possibly get injured from the head, by hitting the windshield or the windshield frame. For avoiding these injuries, different collision-mitigating components have been proposed, as an example, compliant bumpers, pop-up bonnet, and windscreen airbags etc. While the collision with big vehicles can cause much more vital-dangerous injuries in head or chest. In this case, getting the injuries from pelvic and leg parts considered less serious. For decreasing this kind of injuries, new designs for the front of bus and truck bodies were introduced. Another proposition was about truck bodies; it should be united with the chassis and for more effective usage of safety standards, it is suggested to move manufacturing from the local body builders to manufacturers.

Active Safety Systems

The first investigations on pedestrian detection for PPS were introduced at the end of 1990s. From that time, to design efficient pedestrian detection systems, different groups have conducted extensive researches. The objective is to detect pedestrians who are standing or moving on a roadway, while they are still in the safe zone. In order to detect pedestrians and predict the risk of collisions, PPS systems can make use of different types of sensors, cameras, computer vision algorithms. The output of pedestrian detection and prediction of possibility collisions can be utilized for warning the driver, take control of the vehicle, perform autonomous braking and so on.

The most developed systems, besides detecting pedestrians, also apply tracking to the detection process. Tracking detected pedestrians along time is efficient for avoiding false detections, predicting positions and direction of the pedestrians, inference about pedestrian behavior.

1.2 Objectives and Restrictions

The objective of this thesis is to develop a system, for automated detection and tracking of both, stationary and moving pedestrian.

Despite many year of research and spectacular practical progress, pedestrian detection and tracking still remain as difficult tasks, because of following challenges:

- Different styles of clothing
- Different shape of the body (specially height)
- Huge number of various postures, also variety of possible orientations
- Different positions, like different distances from camera (very close or far from camera)
- Self-occlusion - as pedestrians may have accessories such as backpacks, hats, suitcase, umbrella etc.
- Group Occlusion - it can happen, when two or several pedestrians, move really close or even parallelly, especially when pedestrians are located within a crowd
- Non-pedestrian objects, which similar to the person with shape and structure.
- The various environmental conditions, such as moving backgrounds, weather conditions, lighting etc.
- The unpredictability of pedestrian movements - can change speed (e.g can run or stop) or direction (e.g. turn around) without warning.
- Splitting of the track into several pieces, due to poor segmentation of current or previous frames. Or even lost tracking due to noise, or poor segmentation
- Splitting of the track into several pieces, due to poor segmentation of current or previous frames, even lose tracking as a result of noise, or poor segmentation.

1.3 Contributions and Relevance

While working on this thesis

- Many existing feature detection methods were tested
- Based on the first step and related research, the detection system was developed
- For performance improvement, many pre-processing methods, and several background subtraction methods were examined
- Developed detection method was compared with another existing detection method. Both detection methods were tested on 100 True images (people images), 100 False images (non-people images).
- Based on first and third steps, the first tracking approach was developed
- Developed detection and tracking systems, were tested by applying to the videos with different backgrounds

- By keeping the detection system, the second tracking method was developed (based on our detection system) and tested with the same videos

1.4 Road Map

This thesis consists of five main chapters:

In the first chapter, the general overview and objectives and restrictions of this thesis shortly were described. Furthermore, the chapter gives brief information about the work has been done.

The second chapter covers already existing related works in five main titles. The first and second sections are representing well-known features and classifiers, respectively. The third and fourth sections, give the information currently existing object detection and object tracking methods, consequently. The fifth section is related to the surveys toward various detection and tracking methods.

The following chapter consists of two main sections, System design and architecture, and Methodology. In the system design and architecture section, described the general design, and architecture of implemented systems. The Methodology section, consisting of two main sub-sections, which explains in details these detection and tracking systems.

The next chapter illustrates and analyses all the obtained results.

In the last chapter, presented conclusion, which consists of, a general summary of the thesis and future related work.

Chapter 2: State-of-the-art

2.1 Introduction

As mentioned above, regarding human detection and tracking issues, great number of researches have been done and many types of applications were developed.

In this chapter, popular features and classifiers, which are using in the many detection systems were covered. Following, a various image or video-based object detection methods, and object tracking methods were described. Moreover, an overview of some surveys concerning to human detection and tracking methods, has been explained.

2.2 Related work

Sliding window detection is one of the object detection systems, which mainly consist of two components: feature and classifier. The feature slides across an image or video frame, for encoding the visual appearance of the object. For each sliding window, classifier applies, for determining if the window contains needed the object or not.

2.2.1 Features

Scale Invariant Feature Transform (SIFT) published by David G. Lowe [4], in 2004. SIFT efficiently identifies potential needed points, which are invariant to scale and orientation, by utilizing a difference-of-Gaussian function. Selection of the key points is performing on the basis of stability measures of these key points. These key points assigned one or several orientations, and transformed relative to the assigned orientation, scale, and location of each feature. Moreover, around every key point, the measurement of the local image gradients was performed.

Speeded Up Robust Features was presented by Herbert Bay [5], at the European Conference on Computer Vision, in 2006. Partly inspired by the SIFT descriptor, but it works regarding repeatability, distinctiveness, robustness, and much faster than it. SURF achieves this, based on integral images: builds on the power of existing detectors (such as a Hessian matrix-based measure for the detector), distribution-based descriptors, and makes these methods simpler.

Rather of general image intensities, in 1998 C.P. Papageorgiou [6] suggested working with Haar wavelets-based features. Later, this suggestion used by P.Viola and M.Jones [7]. They developed it as Haar-like features. A Haar-like feature calculates the variation between of the sum of the pixels of horizontally or vertically neighbouring rectangles in a detection window.

In 1986, Robert K. McConnell [8], without naming, described the concepts behind Histogram of Oriented Gradients (HOG) and in 1994, Mitsubishi Electric Research Laboratories [9] utilized these notions. Though, N.Dalal and B.Triggs [10] popularized HOG features, by presenting their completed work, in 2005. This descriptor reminds the Histograms of Edge Orientation [11] , SIFT descriptors and Shape Context [12] ; however, it is measured on a dense grid of uniformly spaced cells and for improved performance, it utilizes overlapping local contrast normalizations.

In 2011, P. Kittipanya-ngam and E. H. Lung [13] described a square-shaped window detection, which eliminates the limitation of HOG, such as the human must be in the upright pose, detects variety poses of a person.

In 1994, for texture classification, Local Binary Pattern (LBP) feature was described as one of the most efficient methods. It can be determined, for every pixel, like an ordered collection of binary comparisons of pixel intensities. Because of, high complexity and lack of semantic consistency, LPB was not convenient for human detection. For adopting this feature to the human detection and tracking, Y.Mu [14] suggested two variants of the LPB, such as Semantic-LBP (S-LBP) and Fourier-LBP (F-LBP).

In 2007, Shechtman E. suggested [15] Self-Similarity (LSS) feature for noting similarities in accordance with matching structural self-similarities between images or **videos.** It captures self-similarity of colour, edges, repetitive patterns and complex textures in a single unified way. If there is a similar spatial layout, one image's textured area may match with another image's uniformly coloured area. Based on a thick grid of points in images or videos, at multiple scales, the self-similarity descriptors are estimated. Between image pairs or video sequence pairs a good match corresponds to determining an equal ensemble descriptors with related descriptor values at related geometric locations.

For obtaining distinctive invariant features from interest regions, two new methods such as Oriented Local Self-Similarities (OLSS, C) and Simplified and Oriented Local Self-Similarities (SOLSS, C), suggested by J. Liu [16], in 2012. These two approaches based on the altered versions of the popular Local Self-Similarities, such as Cartesian location grid and gradient orientation for binding and they utilize the combination of the SIFT algorithm and simplified LSS features.

For handling partial occlusion, in 2009 X. Wang [17] suggested the new feature set, of the combination of HOG and LBP features. Later, Y.Xin proposed [18] another combined feature set, by the conjunction of HOG and Haar features, which, can increase the speed and accuracy of detection. The new feature introduced by S. Walk [19], which includes motion

features, HOG features, and self-similarity features, that applied to color channels. For better detection accuracy, B.Wu presented [20] another new feature, by combining of edgelet, HOG and covariance descriptors.

Gabor features [26], constructed from Gabor filters, specifically successful in face recognition and fingerprint matching. Work principle of Gabor features is to extracting local pieces of information. These pieces of information, are combined for object recognition or finding the region of interest.

Another existing features, such as, a set of multi-scale orientation (MSO) [21] features, which containing HOG and coarse, the Integral Channel Features (ICF) [22], which efficiently compute local sums, histograms, Haar features and their different generalizations, CENsus Transform hISTogram (CENTRIST) [23] features , Shapelet [24] features , Granularity-tunable Gradients Partition (GGP) [25] features, also are using in object and human detection systems.

2.2.2 Classifiers

In 1963, V.Vapnik and A.Chervonenkis suggested the original Support Vector Machine (SVMs) algorithm and later, in 1992 another suggestion was creating nonlinear SVM classifiers. The current SVM published by C.Cortes [27], in 1995.

In machine learning, SVM is one of the efficient tools for solving classification issues. It maximizes the margin of hyperplane to achieve maximum separation between the object classes. By the use of so-called 'kernel trick', SVMs effectively is doing a non-linear classification, mapping their inputs samples, into high-dimensional spaces implicitly. Because of efficiency, the linear kernel is used widely.

In 2008, S.MAji [28] introduced Kernel Support Vector Machines (Kernel SVMs) and proved that it is more efficient that different non-linear kernels. Latent SVMs suggested by P.Felzenszwalb [29], which is similar to Conditional Random Fields (CRF) [30], leads to a non-convex training issue. Linear SVMs (T.Joachims [31]) are among the most prominent machine learning techniques for such high-dimensional and sparse data.

Y.Freund and R.Schapire [32], presented Adaptive Boosting (AdaBoost), which published in 1997. This algorithm can be combined with other learning algorithms, for making performance better. Such that, a weighted sum, which demonstrates output of the boosted classifier, join with the outputs of these algorithms.

Partial Least Squares (PLS) - an effective dimensionality reduction technique. It, suggested by W. R. Schwartz [33], preserves important discriminative information, for projecting the data onto a much lower dimensional subspace.

Naive Bayes classifier [34] widely examined starting from the 1950s, which previously presented into the text retrieval community. This classifier can be competing with advanced methods like SVM, by applying proper pre-processing.

2.2.3 Object Detection

P.Papageorgiou [35] suggested detection framework, which based on a wavelet representation of an object class and SVM classifier, This framework overcomes in-class variability problem, also reduce the number of false detections. P.Viola [36], trained using of AdaBoost for taking advantage of both motion and appearance information to detect a walking person. This detector is very efficient, even with detecting pedestrians at very small scales, and the number of false positive detections remarkably small. Multiscale fast pedestrian detection method suggested by P.Dollar [37], can approximate features, including gradient histograms at nearby scales from features computed at a single scale.

Example-based detectors presented by A.Mohan [38], which were examined for detecting four separate components of the human body, such as the head, legs, left and right arms. When a person changes pose or due to occlusion, one of the components cannot be detected, but other components are combined with suitable hierarchical classifiers, the detector can still detect the person. Component detectors utilize Haar wavelets and SVM classifier. Part based detection method suggested by P.Felzenszwalb [39], relies on new methods for discriminative training with partially labelled data. This detection method uses HOG features with latent SVM classifier. For detecting fully visible humans in videos, with a variety of poses and movement directions, new detectors trained by N.Dalal [40]. These detectors are considered the high-performance detectors. The performance of these detectors is based on the combination of motion-based and HOG appearance descriptors in a linear SVM classifier. A Pose-Invariant Descriptor presented, by Z.Lin [41]. By concurrently segmenting human shapes and poses, these detectors manage to classify if an image contains human or not. The performance is based on HOG features and Kernel SVM classifiers.

One of the simple and efficient methods of human detection is Background subtraction. It is equal to the subtraction of current frame and background frame. Nevertheless, there are several limitations with background subtraction such as video for detection should be

taken by the static camera, it cannot detect stationary objects, cannot isolate humans from the human group, or differ them from other moving objects. An example of the detection by background subtraction method is proposed by A. Elgammal [42].

The detection method, presented by B. Wu [43], detects and segments multiple objects in the images, which partly occluded. These part detection and whole-object segmentation trained on boosting shape oriented local image features.

2.2.4 Object Tracking

In the [44] tracking method is based on filtering approach, which estimates motion parameters of the human body. In the [45][45], the author has suggested tracking algorithm based on a model of affine image changes and discriminative features during tracking. The kernel-based tracking method introduced by D.Comaniciu [46], working principle relying on the similarity function attraction. In this algorithm, for optimization, the mean shift procedure has applied. S.Avidan [47] treats tracking as a binary classification problem. By utilizing AdaBoost, from a group of on-line trained weak classifiers, an author forms a strong classifier. Hence this new strong classifier is used to compute a confidence map of the next frame. Similar tracking technique to this method, described in [48].

By applying mean shift algorithm, it is possible to find the peak of the map and the new position of the objects. In this [49] case, algorithm selects the most suitable on-line features that best differentiate between object and background. In [50] given tracking system, by using Multiple Instance Learning (MIL) algorithm, allows updating an adaptive appearance model with a set of image patches.

Model-based human tracking system is suggested by O. Masoud [51]. This tracker uses a single camera and works in many difficult cases, like occlusions, ambiguities and so on. While human is visible, the system produces this pedestrian's location and velocity information.
One of the well-known tracking method the Kalman filter, published in 1960, by R.E.Kalman [52], for solving the discrete-data linear filtering problem. Multiple research has been done, and many applications have been developed toward this filter.

Tracking-by-Detection methods

Lots of methods, (for example [53], [54], [55]]) have been described for multi-target tracking-by-detection algorithms.

B. Benfold [53] presented multi-target tracking system, which is multi-threaded. With simultaneous KLT tracking, their system collects asynchronous HOG detections. This tracker uses Markov-Chain Monte-Carlo Data Association (MCMCDA), for providing high real-time tracking performance in high definition video.

In [54] one of the tracking-by-detection methods, where associating object detections with tracks is formulated as finding the maximum weight independent set (MWIS) of a graph of tracklets is demonstrated.

Another approach is developed by C.Kuo [55] for online learning of discriminative appearance models for robust multi-target tracking in a crowded scene. Within a time sliding window from tracklets, training samples are assembled online and also for combining effective descriptors and their similarity measurements, the AdaBoost algorithm is used.

2.2.5 Surveys

For a large number of detection and tracking methods, many researches [56], [57], [58], [59], [60] have been conducted.

A.Yilmaz [56] presented a survey of the state-of-the-art tracking methods, by classifying them into three categories such as methods establishing point correspondence, methods using primitive geometric models, and methods using contour evolution.

The first part of [57], includes comprises of models and components of a pedestrian detection system. The second part covers a different set of state-of-the-art systems, such as Haar wavelet-based AdaBoost cascade, the histogram of oriented gradient (HOG) features/linear support vector machine (SVM), the neural network (NN)/local receptive fields (LRF) and combined shape-texture detection.

R.Lomte [58], mainly focused on general methods and characteristics of human detectors and explained these detection techniques comprehensively.

In the article [59], analysis of 40+ detectors, which are proposed last decade is presented. As the authors indicated, their experiments show that the progress in pedestrian detection systems mostly related to an improvement of the feature. The result of this greater survey, is summarized in Appendix.

H. Parekh [60], presented a brief survey of various object detection and tracking methods, and object classification algorithms.

2.3 Conclusion

In this chapter, related works, toward detection and tracking problems have been covered. In the first part, popular features, such as SIFT, SURF, Haar, HOG, LBP, LSS etc. described. In the second part, most preferred classification methods are illustrated (SVM, Latent SVM, Kernel SVM, Linear SVM, AdaBoost, PLS). Following, multiple existing object detection and tracking methods explained. In the end, several interesting types of research related to the object detection and tracking methods have been described.

Chapter 3: Methodology and contribution

3.1 Introduction

This chapter consists of two main parts. The first part gives the general overview of the developed detection and tracking systems. The second section describes in details the methodology that has been used during this thesis work. The main contributions are resumed in proposing two algorithms, one for detecting pedestrian and the second for tracking them. These subsections cover the foremost important components that are used in the application.

3.2 System design and architecture

In a detection method, one of the popular feature detection, Histogram of Oriented Gradients (HOG) was used. Because the HOG performance is slow with video frames, linear Support Vector Machine (linear SVM) classifiers are applied for better performance.
While working on this thesis, two tracking approaches were examined:

In the first approach, tracking process does not depend on detection process. The idea is to keep track, even if the system cannot detect a person or detection is lost. The general architecture of the first approach for detection and tracking is given in Figure 1.

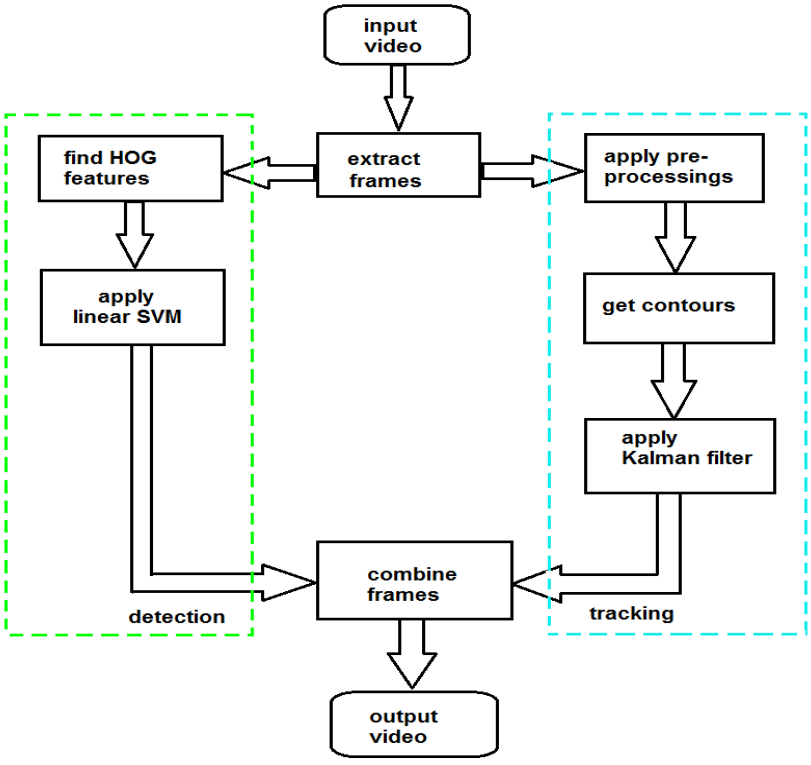


Figure 1. The system overview (the first approach)

By applying several pre-processing methods (such as Morphological Transformation, threshold, background subtraction) to the frames, the system gets contours of moving objects, which Kalman filter uses for tracking.

This approach works well, even when the system cannot detect a person or loses detection. As tracking is based on motion, the system tries to keep tracking of all moving objects, even, the shadow of pedestrian, reflection of pedestrian and so on. This tracking method is efficient for the safety aspect, but our goal is to detect and track a human.

Due to this reason, the tracking technique was changed. In the second approach, we examined tracking-by-detection, by leaving detection method (HOG + linear SVM) the same. In this approach, tracking bases on the detection results, to be more precise, Kalman filter uses detected features for tracking and also for predicting the position of the pedestrian. The system overview of this method is given in Figure 2.

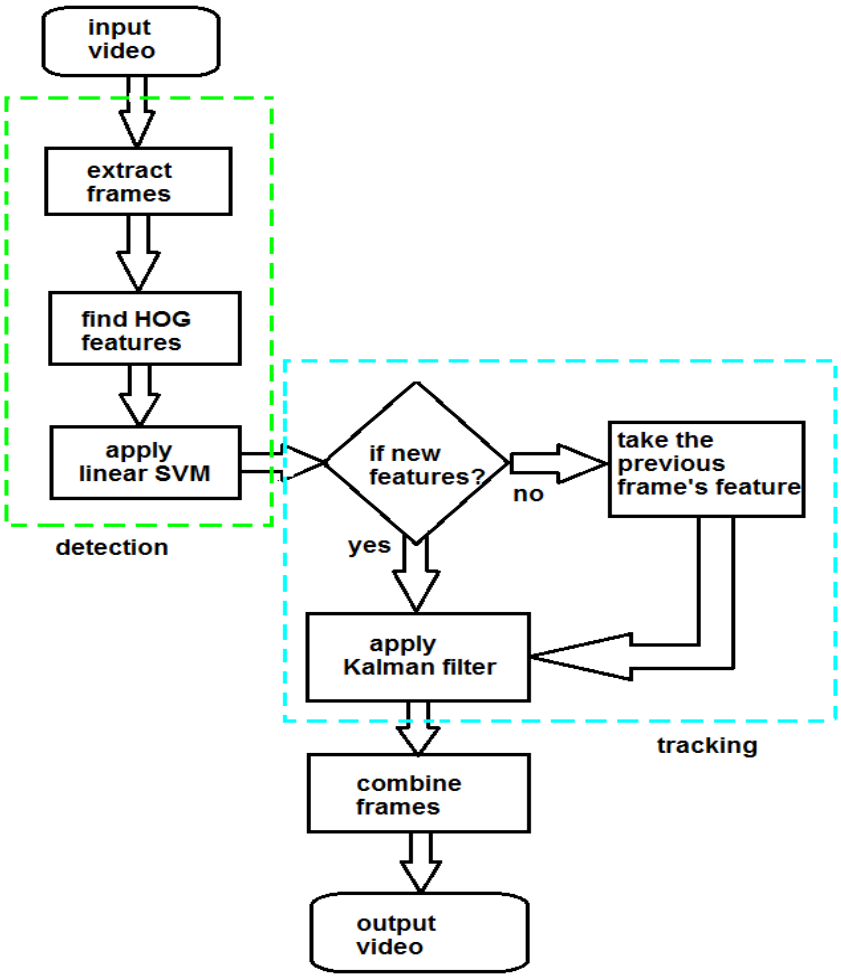


Figure 2. The system overview (the second approach)

3.3 Methodology

In this section, the methodology of out detection and tracking methods is described in details.

3.3.1 Detection

An overview to object detection chain with HOG and SVM, is given in Figure 3. The main idea is that, even without knowing the corresponding gradient or edge positions, by the distribution of local intensity gradients or edge directions, local object appearance and shape can be well characterized.

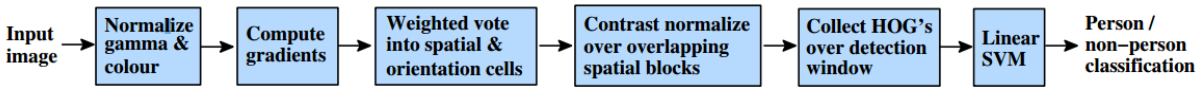


Figure 3. An overview of feature extraction and object detection method (N. Dalal and B. Triggs [10])

3.3.1.1 HOG Implementation

As illustrated in Figure 3. the HOG feature extraction process, consist of five main steps:

Normalize Gamma/Colour

In experiments N. Dalal and B.Triggs [10], evaluated some grayscale RGB, and LAB colour spaces, with power law (gamma) equalization. They point out that these normalizations have just a moderate influence on performance, possible, because of obtained results by the subsequent descriptor normalization are similar. Thus, this pre-processing step can be omitted.

Gradient Computation

For gradient computation, N. Dalal and B.Triggs tested several masks, such as uncentred [-1, 1], centred [-1, 0, 1], cubic-corrected [1, -8, 0, 8, -1], the 3 × 3 Sobel or 2 × 2 diagonal mask. The best results obtained by using simple 1-D centred masks. This method requires filtering the colour of the image (gray-scale image) with [- 1, 0, 1] and [- 1, 0, 1]^T kernels for computing gradients of image in horizontal (x) and vertical (y) directions.

Mathematical measurement of gradients demonstrated below.

Computation of pixel at coordinate (x, y), where, magnitude - m, and direction – θ computation demonstrated in (1), (2), (3), (4), correspondingly.

Horizontal gradient:

$$I_x = I(x+1, y) - I(x-1, y) \qquad (1)$$

Vertical gradient:

$$I_y = I(x, y+1) - I(x, y-1) \quad (2)$$

Magnitude:

$$m = \sqrt{I_x^2 + I_y^2} \quad (3)$$

Direction:

$$\Theta = \arctan \frac{I_y}{I_x} \quad (4)$$

Spatial / Orientation Binning

In the third step, based on the elements obtained in the gradient measurement, each pixel casts a weighted vote for an edge orientation histogram channel, within cells. The shape of the cells can be rectangular or radial. If the gradient is "unsigned" the orientation bins are equally spaced over 0^0 - 180^0, if the gradient is "signed" then the orientation bins are spread over 0^0 - 360^0.

For good performance, the fine orientation coding is important, while spatial binning can coarsely influence. In [20], demonstrated that performance significantly improves up till around 9 bins, by raising the number of orientation bins. This is for "unsigned" gradients (orientation bins are spread over 0^0 - 180^0).

Descriptor Blocks

Because, the broad range of changes in illumination and foreground-background contrast, for good performance, locally normalization turns out to be important for gradient strengths. Most of the different normalization schemes are based on grouping cells into large connected blocks and individually normalize each block. The final HOG descriptor is a vector, which consists of a combination of normalized cells, from all of the blocks regions. Typically, the blocks overlap, with the idea that, each cell contributes several components to the final HOG descriptor. Dalal and B.Triggs declare that overlapping blocks, efficiently increase the performance, about 4%.

Two block geometries: rectangular block separated into grids of rectangular spatial cells (R-HOG) and circular ones distributed into cells in log-polar fashion (C-HOG).

R-HOG blocks, very similar to SIFT descriptors, but, they are computed in dense grids at a single scale without dominant orientation alignment and used as part of a larger code vector that implicitly encodes spatial position relative to the detection window, whereas SIFTs are computed at a sparse set of scale-invariant key points, rotated to align their dominant

orientations, and used individually. [20]. The parameters of R-HOG: cells for each block, pixels for each cell, and orientation bins. 3 × 3 cell blocks of 6 × 6 pixel cells are considered the best for human detection.

C-HOG blocks remind the Shape Contexts [22], but the difference is that the shape contexts utilize a single orientation-independent edge presence count, while in C-HOG, every cell contains a stack of gradient-weighted orientation cells. Two variants of the C-HOG: those with one circular central cell and those with the central cell is split into angular sectors. The parameters of C-HOG [20]: the numbers of angular and radial bins, the radius of the central bin in pixels, and the expansion factor for subsequent radius. For obtaining the best performance, at least 4 angular bins, 2 radial bins, 4 pixels radius for the central bin are needed.

Block normalization

For block normalization, Dalal and B.Triggs [20] evaluated four methods. Let assume that, v is unnormalized descriptor vector, v_k is k-norm, where k = 1, 2, 3, .. , ε is a small constant.

The normalization methods:

$$\text{L2-norm}, f = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon^2}} \quad (5)$$

L2-Hys, clipping L2-norm (decreasing max values of v till 0.2)

$$\text{L1-norm}, f = \frac{v}{\|v\|_1 + \varepsilon} \quad (6)$$

$$\text{L1-sqrt } f = \sqrt{\frac{v}{\|v\|_1 + \varepsilon}} \quad (7)$$

Experiment [20] shows, that the performance of L2-Hys, L2-norm, and L1-sqrt equally good. But, the performance with L1-norm normalization decreases by 5% and omitting normalization, reduces by 27%.

Detection Window and Context

After, extracting features, the descriptors size should be decided: so-called minimal sliding window size. The 64 × 128 descriptor size, containing around 16 pixels, improves detection performance, in contrast, the window size 48 × 112 (about 8 pixels) reduces performance by 4%.

3.3.1.2 Support Vector Machine

Support Vector Machines are machine learning techniques that, use to analyse data for classification and regression analysis. Besides simplicity and accuracy, the SVM decreases

training set error, a bound on the empirical error and the complexity of the classifier, maximize the geometric edge of the area, easily can deal with high-dimensional data. By default, linear SVM trained with SVMLight is being used for human detection.

After feature extraction, SVMs are applied to the system, for determining, if these features belong to the human or not.

Assume, x is feature vector (in our case, the HOG feature). The mathematical computation of the SVM as below:

$$f(x) = w^T x + b \qquad (8)$$

Where b is the bias, w is the weight vector [63].

3.3.1.3 Implementation

Above described steps of detection, simple commands of Open Source Computer Vision (OpenCV / version 3.0.0), which is library of programming functions, were implemented and were mainly used in computer vision at real-time. In the application, for the HOG Descriptor, system uses parameters as below:

```
<opencv_storage>
  <hog type_id="opencv-object-detector-hog">
    <winSize>64 128</winSize>
    <blockSize>16 16</blockSize>
    <blockStride>8 8</blockStride>
    <cellSize>8 8</cellSize>
    <nbins>9</nbins>
    <derivAperture>1</derivAperture>
    <winSigma>4.</winSigma>
    <histogramNormType>0</histogramNormType>
    <L2HysThreshold>2.0000000000000001e-01</L2HysThreshold>
    <gammaCorrection>1</gammaCorrection>
    <nlevels>64</nlevels>
    <signedGradient>0</signedGradient>
  </hog>
</opencv_storage>
```

- detection window size is 64 × 128,
- block size in pixels is 16 × 16,
- block stride is 8 × 8, cell size is 8 × 8,
- number of bins is 9,
- Gaussian smoothing window parameter is 4,
- normalization method is L2-Hys (2.0000000000000001e-01),

- gammaCorrection value is 1, which means gamma correction preprocessing is not needed (if G < 1 - makes the image appear darker, if G > 1 - makes the image appear lighter, G=1 - does not effect on the input image/frame),
- 64 is maximum number of detection window (nlevels).

After setting SVM to detector, detectMultiScale method handles detecting pedestrians.

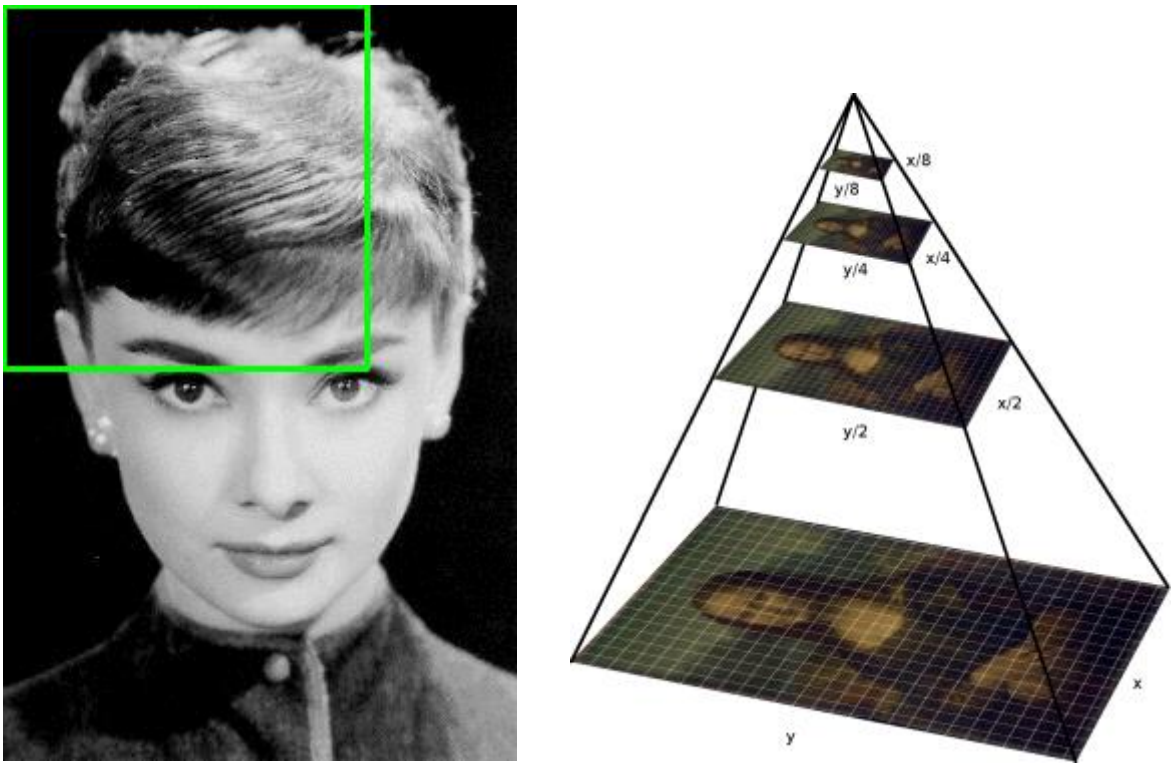
```
<opencv_storage>
  <hog2 type_id="opencv-object-detector-hog">
    <winSize>64 128</winSize>
    <blockSize>16 16</blockSize>
    <blockStride>8 8</blockStride>
    <cellSize>8 8</cellSize>
    <nbins>9</nbins>
    <derivAperture>1</derivAperture>
    <winSigma>4.</winSigma>
    <histogramNormType>0</histogramNormType>
    <L2HysThreshold>2.0000000000000001e-01</L2HysThreshold>
    <gammaCorrection>1</gammaCorrection>
    <nlevels>64</nlevels>
    <signedGradient>0</signedGradient>
    <SVMDetector>
      5.35938591e-02 -1.47214547e-01 -5.53217009e-02 5.07730693e-02 1.15470812e-01
      -4.26880382e-02 4.63583395e-02 -5.46819903e-02 8.23208392e-02 1.04240678e-01
      -2.29451805e-02 1.10851899e-02 1.37869297e-02 1.11935101e-01 1.26841804e-02
      8.52834582e-02 -6.30923882e-02 1.30546331e-01 8.10072869e-02 -5.20973913e-02
      -4.31552902e-02 9.34138373e-02 1.10350259e-01 -7.59621784e-02 -5.51751107e-02
      -4.46529612e-02 2.94733401e-02 4.55553606e-02 -3.55954492e-03 7.81895593e-02
```

The parameters of detectMultiScale function:

- Source image /frame, which we want to detect people
- hitTreshold is an optional parameter for the distance between the HOG features and SVM classifying plan
- winStride is one of the most important parameters. In the x and y positions, it determines the "step size". In computer vision, a sliding window is a rectangular area, with a fixed width and height. It "slides" across an image/frame and each of these window regions being used for applying SVM classifier to determine if the window contains a person or not
- Padding is an indicator, in the x and y directions, a number of pixels.
- Another important parameter is scale. It controls the number of levels of the image pyramid. Pyramid has the original image/frame at the bottom with original width and

height. At each layer, it resizes (until obtaining a minimum size) and optimally soothes the image/frame. Image pyramid helps to find needed objects at a different scale of an image. A combination of image pyramid with a sliding window helps to find objects in images in various locations.

- useMeanShiftGrouping is a boolean parameter, used for handle overlapping bounding boxes, which usually by default is False, and in many cases do not advisable to set it to True.



(a) (b)

Figure 4. Example to window sliding (a) and pyramid of the image (b) [61]

3.3.2 Tracking

3.3.2.1 Thresholding

Thresholding is one of the simplest methods of image segmentation, which can create binary images from a grayscale image. The purpose is to improve the quality of the image and extract pixels from image, which represent an object. The working principle of the simplest methods of thresholding is to replace pixels of the image with black or with white pixels. If an intensity of the image is smaller

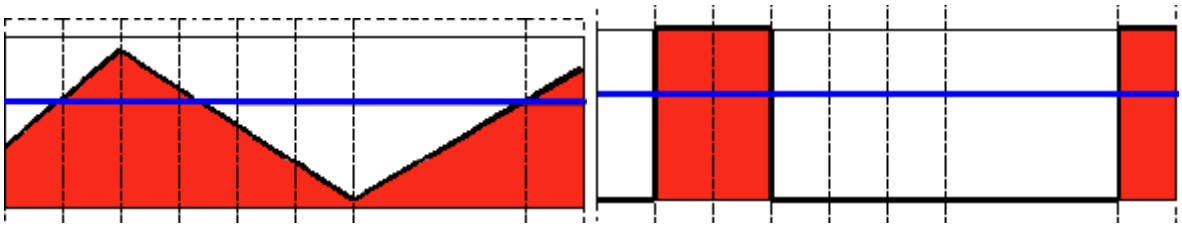
than fixed constant value, so-called intensity threshold, each pixel in an image replaced with a black pixel, if bigger that intensity threshold, then replaced with a white pixel.

Figure 5. Thresholding process

OpenCV offers the simple thresholding function. By utilizing this function, we can effectuate five existing types of Thresholding:

- Threshold Binary
- Threshold Binary, Inverted
- Threshold Truncate
- Threshold to Zero
- Threshold to Zero, Inverted

As mentioned above, while working on this thesis, two approaches of tracking were examined. In the first approach, as pre-processing, Threshold Binary was utilized.
Let assume a source image, has pixels with src(x, y) intensity values. The blue line in Figure 6. (a) is the fixed threshold value (thresh).



(a) (b)

Figure 6. Thresholding process: (a) an original image. (b) a binary image

Binary Thresholding operation can be expressed as following:

$$\mathrm{dst}(x,y) = \begin{cases} \mathrm{maxVal}\ if\ \mathrm{src}(x,y) > \mathrm{tresh} \\ 0,\ \ \mathrm{otherwise} \end{cases} \quad (9)$$

If thresh is smaller than the src(x, y) intensity value, the new pixel intensity is set to the MaxVal, otherwise, the pixels are set to 0. Figure 6. (b)

3.3.2.2 Background Subtraction

For improving productivity, BackgroundSubtractorKNN, offered by OpenCV, which performs non-parametric statistical estimation was utilized. We can consider this algorithm as a kernel density estimation type.

Assumed that, we have vector x, we can estimate the density of x, as follows:

$$\hat{p}(x|D_t, BG + FG) = \frac{1}{|D_t|}\sum_{D_t} K\left(\frac{x-x_i}{h}\right) \quad (10)$$

Here: $\hat{p}(x|D_t, BG + FG)$ is common model, K is the kernel function, h is bandwidth, d is a dimensional vector.

3.3.2.3 Dilation (Image Morphology)

Another pre-processing, which we used in the first tracking approach, is Dilation. It is one of the primary operations in mathematical morphological transformations. Dilation is widely being used in varied contexts, for instance, eliminating the noise of the image, isolating individual elements, and joining disparate elements of the image. This morphological transformation can also be utilized in finding intensity peaks in a picture, and to determine a particular form of an image gradient. For expanding the shapes in the input image, usually, dilation operation uses a structuring element.

Dilation is a convolution of the image with the kernel (usually, "solid" square kernel, or sometimes, a disk). In the kernel, each given pixel is replaced with the local maximum of all of the kernel covered pixel. Actually, the exact result depends on the kernel, but generally, dilation expands a bright region and tend to fill concavities in the image.

OpenCV library also offers a function to implement dilation. In dilation process, the value of some point p is set to the maximum values of all of the points covered by the kernel.

$$dilate(x, y) = \max_{(i,j)\in\, kernel} src\ (x + i, y + j) \quad (11)$$

3.3.2.4 Kalman filter

Kalman filter and also known as Linear quadratic estimation is a series of mathematical equations, which, by minimizing the mean of the squared error, provides an efficient computational means to determine the state of a process. This filter can estimate states of past, of the present, and of future time, even when the precise nature of the modelled system is unknown.

The Kalman filter has numerous applications in technology, such as navigation, guidance, vehicles, aircraft. Moreover, The Kalman filter is a widely used in signal processing, econometrics, and robotic motion planning and control fields.

The algorithm of this filter works in two steps, such as prediction and update. At first, the filter estimates the current state variables with their uncertainties in the prediction step. After obtaining the estimates of the next measurement, by utilizing a weighted average, these estimates are updated. The estimates with higher certainty, get more weight. As the algorithm of the filter is recursive without any additional past information, only by utilizing the present input measurements and the previously calculated state and its uncertainty matrix, it can run in real time. These filters built on linear operators, perturbed by errors, which Gaussian noise may include.

The Kalman filter estimates the state x_k of a discrete-time controlled process, by the following equation:

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1} \quad (12)$$

the measurement equation z_k:

$$z_k = Hx_k + v_k \quad (13)$$

w_k and v_k are random variables, which describe the process and measurement noise, correspondingly. They are considered as independent from each other, white, and with normal probability distributions (Q is process noise covariance and R is measurement noise covariance)

$$p(w) \sim N(0, Q) \quad (14)$$

$$p(v) \sim N(0, R) \quad (15)$$

F is the state transition model (n × n) matrix, which relates to the previous state x_{k-1}. B is the control-input model (n × l) matrix, which relates the control input u l. H is the observation model (m × n) matrix, which relates the state to the measurement z_k. F, B, and H model matrices are assumed constant.

Detailed explanation of the Kalman Filter Algorithm

The Kalman filter does process estimation by feedback control. At some time, the filter estimates the process state, then gets feedback as noisy measurements. Kalman filter works in two groups, such as time update equations and measurement update equations. For obtaining the priori estimates for the next time step, the time update equations predict the current date,

and error covariance estimates. To obtain a corrected a posteriori estimate, the measurement update equations include a new measurement into the a priori estimate. The time update equations are also known as predictor equations and the measurement update equations as corrector equations.

Figure 7. The Kalman filter cycle

Mathematical explanation of time and measurement updates (inspired from [62]):

The time update (predict) equations:

$$\hat{x}_{k|k-1} = F_k\, \hat{x}_{k-1|k-1} + B_k u_k \tag{16}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{17}$$

The measurement update (correct) equations:

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{18}$$

$$S_k = H_k P_{k|k-1} H_k^T S_k^{-1} \tag{19}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{20}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{21}$$

$$P_{k|k} = (I - K_k H_k P_{k|k-1}) \tag{22}$$

Usually, the measurement noise covariance R is measured before the operation of the filter. Generally, determination of the measurement error covariance R is practical, because, the measurement of the process is needed anyway, but the measurement of the process noise covariance Q, is more difficult because we cannot directly observe the process, which is estimating.

Prediction of state estimate and estimate covariance are measuring, with (16), (17), and Measurement and covariance, are calculating with (18), (19), respectively. Computation of Kalman gain is executing with (20). Updating estimate with measurement and the error covariance are implementing by (21) and (22), correspondingly.

3.4 Conclusion

The first part of this chapter explains the general architecture and design of the detection and tracking methods.

The second part consists of two main sections, such as Detection and Tracking. In the Detection part, information about most important components of detection, such as HOG and SVM was given. Furthermore, the implementation method was detailed.

Chapter 4: Results and analysis

4.1 Introduction

This chapter discusses the experiments that were performed in the dissertation, namely experiments based on feature detections, pre-processing, background subtraction methods. In addition, the comparison of two detection methods is shown. Moreover, final results are provided as well as their analysis.

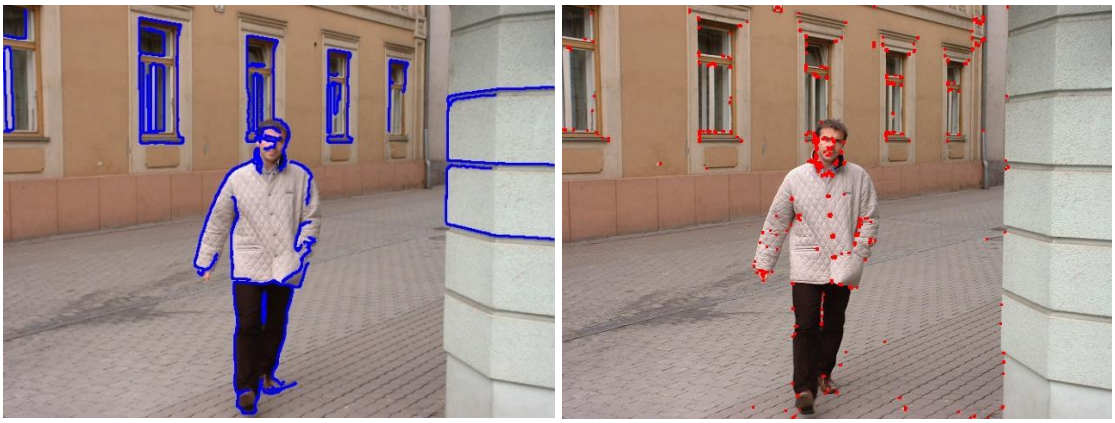
All images, used for the experiment were taken from INRIA [64] dataset and the videos were taken by Carl Zeiss Tessar HD 1080p Logitech camera.

4.2 Features Detections

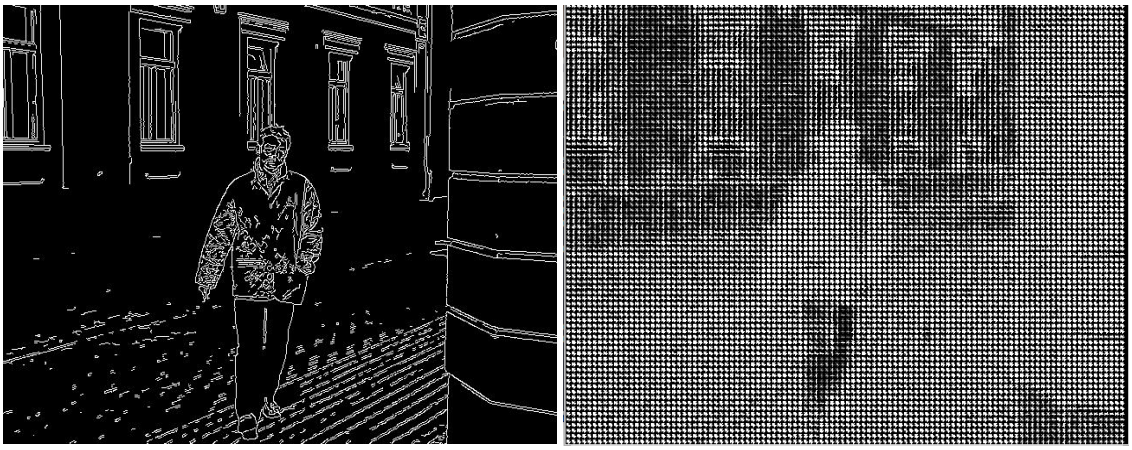
Multiple feature detection methods, such as SIFT, SURF, LoG, Harries operation, Canny, HOG, Contour features, Line Segment Detector and etc., were tested. Some of the obtained results are illustrated in Figure 8. the HOG feature description, Contour features, were chosen for developing the detection and tracking systems.

The reason choosing these methods is, they are more productive to detect the features of the object rather than the background. From Figure 8. clearly can be seen that HOG features (f) and Contour features (g), emphasize the background less, than other experimented feature detectors.

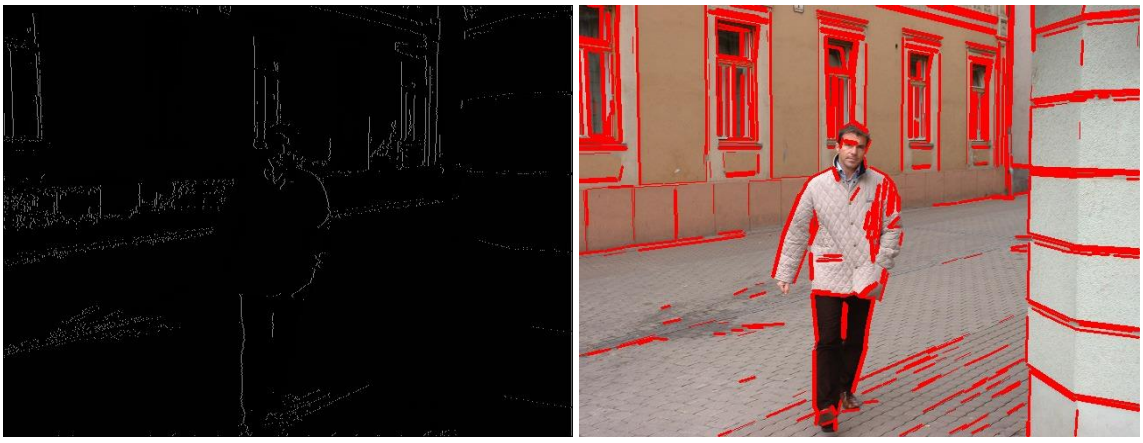
(a) (b)



(c) (d)



(e) (f)

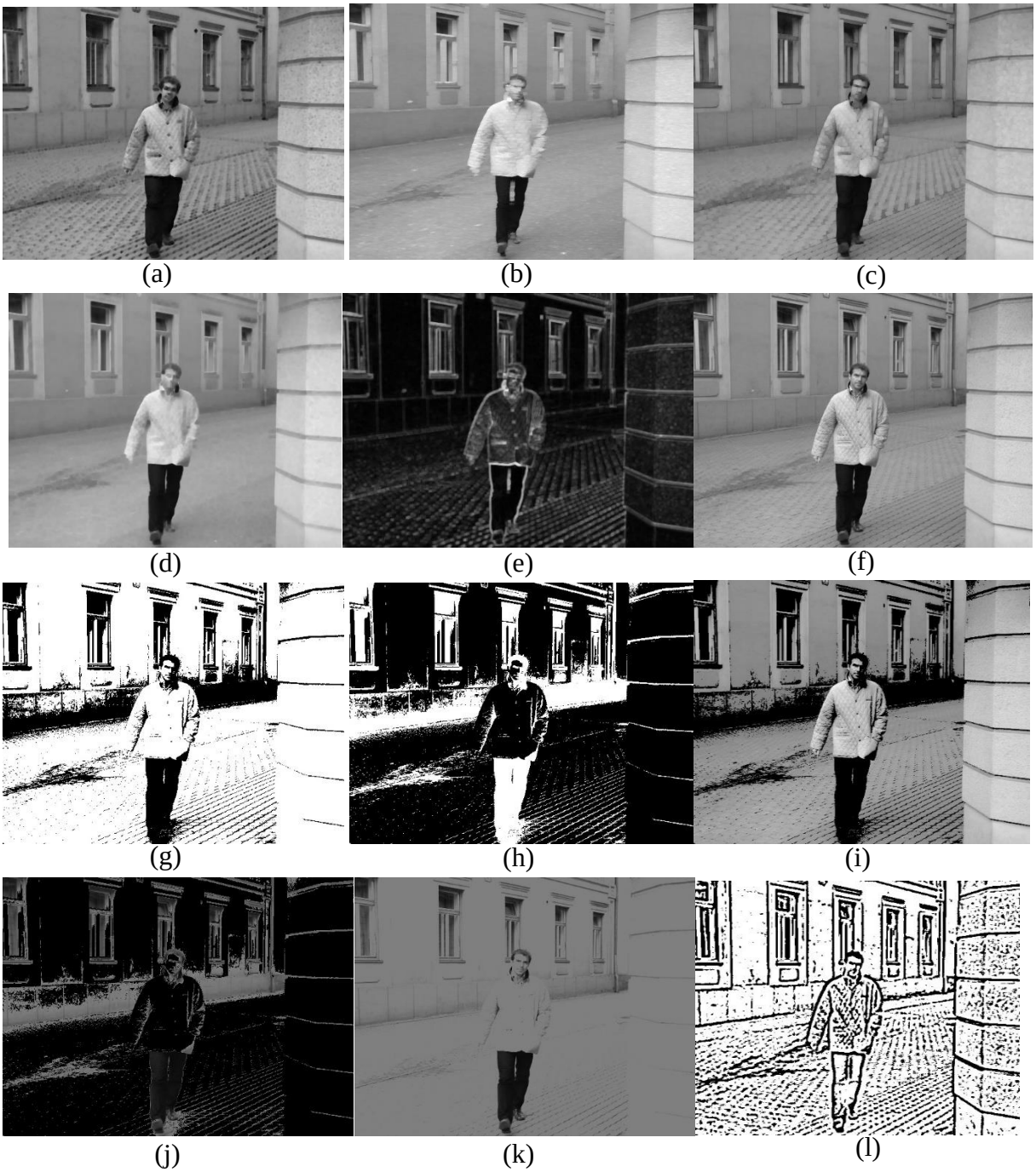


(g) (h)

Figure 8. Feature detection methods: (a) SIFT; (b) SURF; (c) LoG (Laplacian of Guassian); (d) Harris operation, (e) Canny, (f) HOG, (g) Contour feratures, (h) Line Segment Detector

4.3 Pre-processing

For the improvement of the detection and tracking performance, multiple pre-processing methods were examined. Some of the obtained results are illustrated in Figure 9



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 9. Preprocessing methods: (a) Erosion; (b) Dilation; (c) Opening; (d) Closing; (e) Morphological Gradient; (f) Normalization; (g) Threshold Binary; (h) Treshold Binary Inv; (i) Threshold Trunk; (j) Threshold to Zero; (k) Threshold to Zero Inv; (l) Adaptive Mean Thresholding; (m) Adaptive Gaussian Thresholding; (n) Otsu's Thresholding; (o) Otsu's thresholding after Gaussian filtering

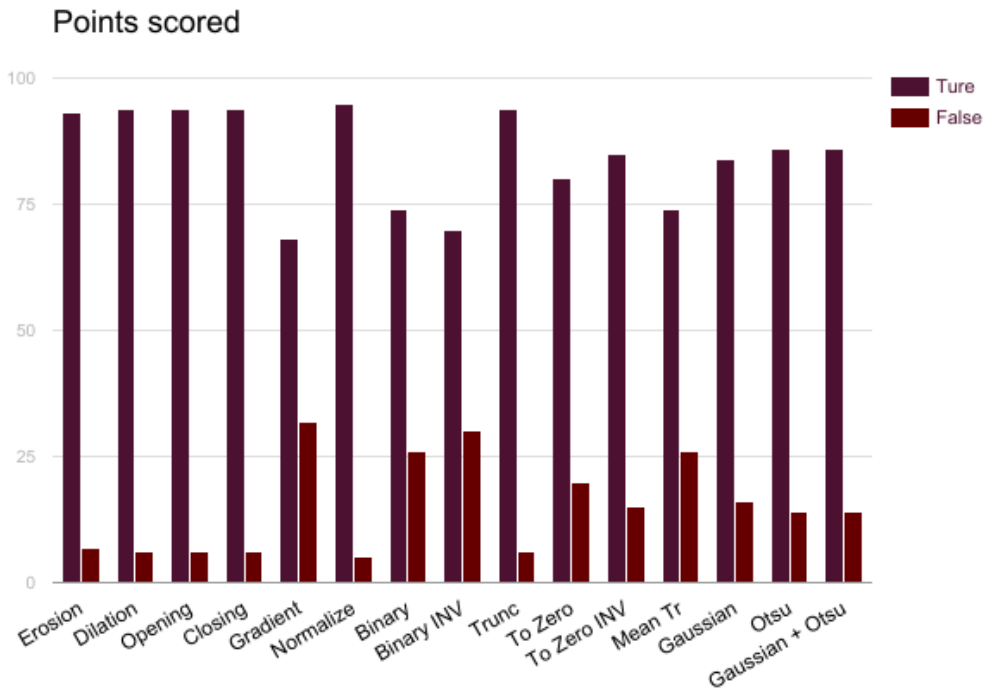


Figure 10. Detection results with pre-processing methods on True images

Above illustrated pre-processing methods have been applied to the detection algorithm and each of them separately tested.

From Figure 10. it is obvious that the best detection results on True images, obtained by detection with normalization pre-processing, about 95%. Also, the smallest number of the obtained false detection results on the True images, achieved by normalization pre-processing.

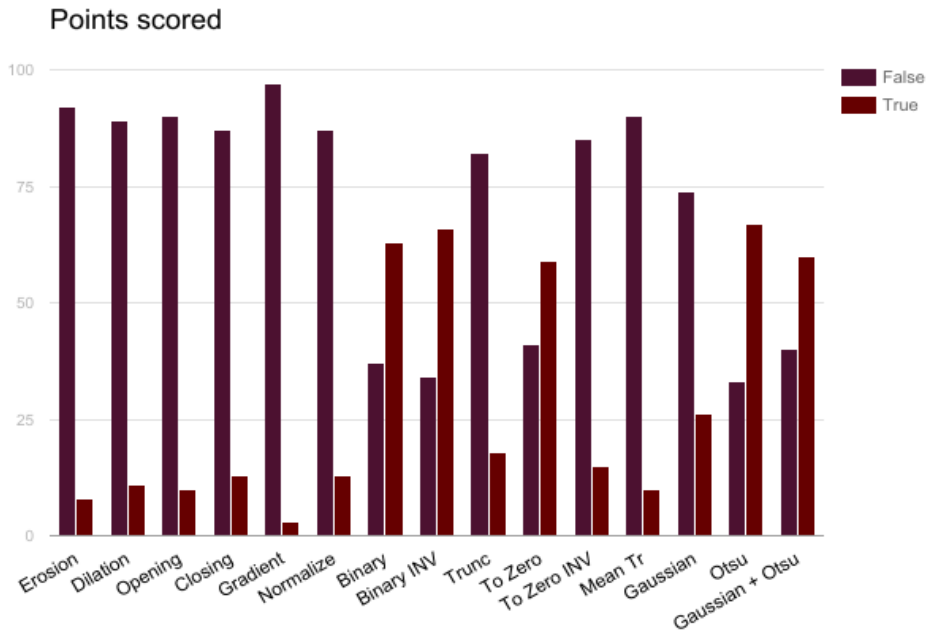


Figure 11. Detection results with pre-processing methods on False images

By applying the same pre-processing methods to the False images, the best result obtained with Morphological Gradient, which detected only 3% of False images. Figure 11.

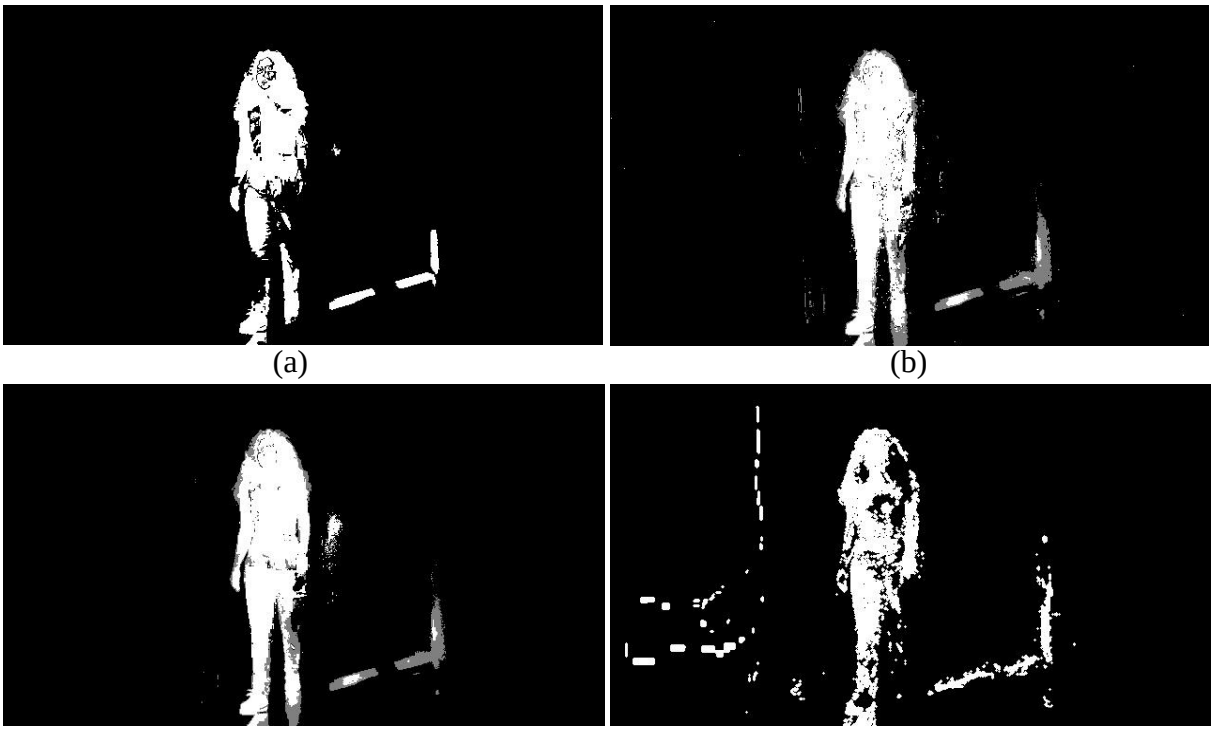
Obtained percentage of true detections on the true image with normalization pre-processing, is the same with original algorithm.

These methods have been applied to the video frames for improving the performance of the tracking system. The Dilation pre-processing was the most efficient for this system, gave the same percentage of the detection method, about 94%.

4.4 Background Subtraction

Four different background subtraction methods of OpenCV were tested, by applying them to the video examples. The results of these background subtraction methods, are given in Figure 13.

Figure 12. Original frame



(a) (b)

(c) (d)

Figure 13. OpenCV Backgraund Subtraction methods: (a) BackgroundSubtractorMOG; (b) BackgroundSubtractorMOG2; (c) BackgroundSubtractorKNN; (d) BackgroundSubtractor-GMG;

Each of these background subtraction techniques tested separately and the best performance was achieved by BackgroundSubtractorKNN method.

4.6 HOG-based detection and a Haar-based detection

After developing it, the detection system and already existing Haar-based human detection system on 100 Ture (Images containing people), and 100 False (Images without people) Images were examined. Examples of these detection outputs given in Figure 14 and Figure 15.

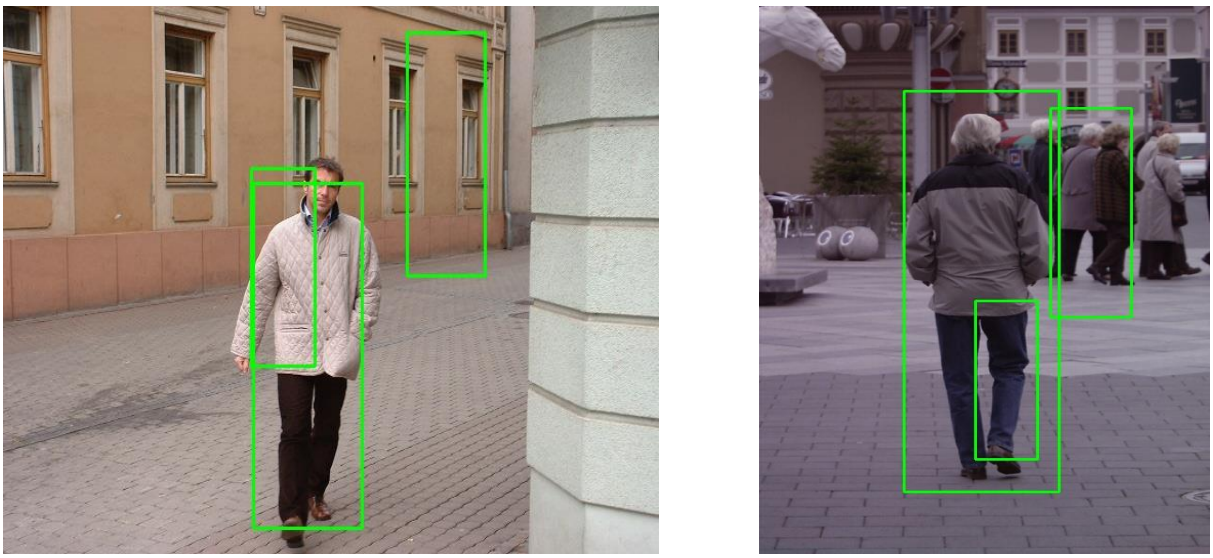


Figure 14. Haar-based detection outputs

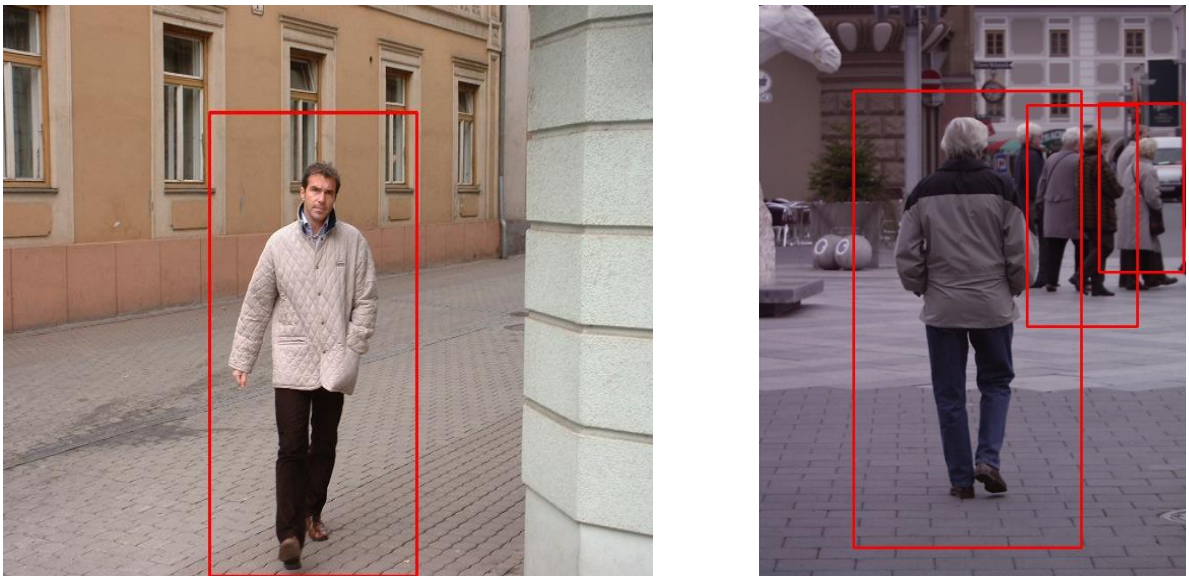


Figure 15. HOG-based detection outputs

The result of comparisons of these two detection methods, given in Figure 14 and Figure 15.

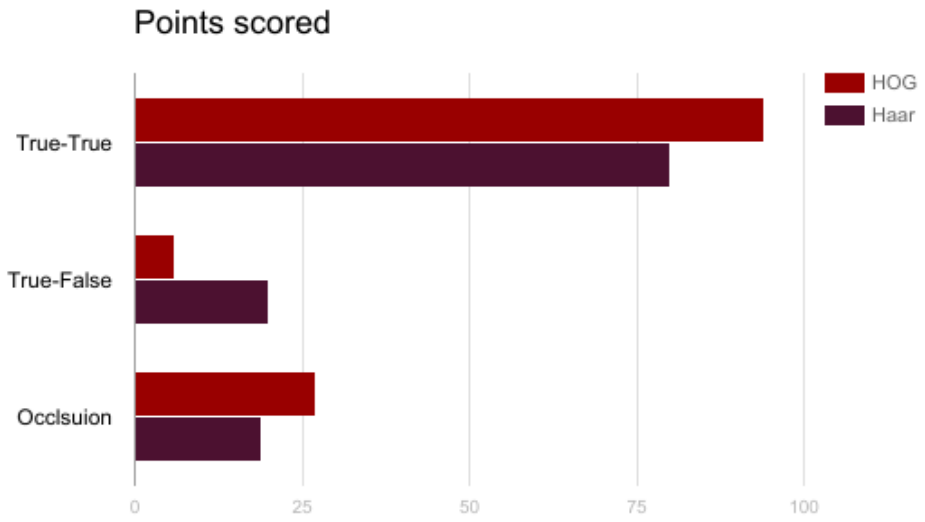


Figure 16. Comparison of HOG and Haar based human detection methods on 100 True images True-True is a True detection on True images (human images); True-False is a False detection on True images (human images).

From Figure 16. it is obvious that the number of the true detections, received by HOG based human detection method is higher than Haar-based human detection methods. The HOG detector, correctly detected about 94% and only 6% was false detection, while Haar base detector, managed to get approximately 80% true detection and 20% false detection.

With the HOG detector, obtained True detection has 29% occlusion cases, but with Haar- base detector, the percentage of occlusions was 24. Despite occlusion, based on achieved results from this experiment, it can be reported that the HOG based detection method gives three times more accurate results, rather than Haar-base detection method.

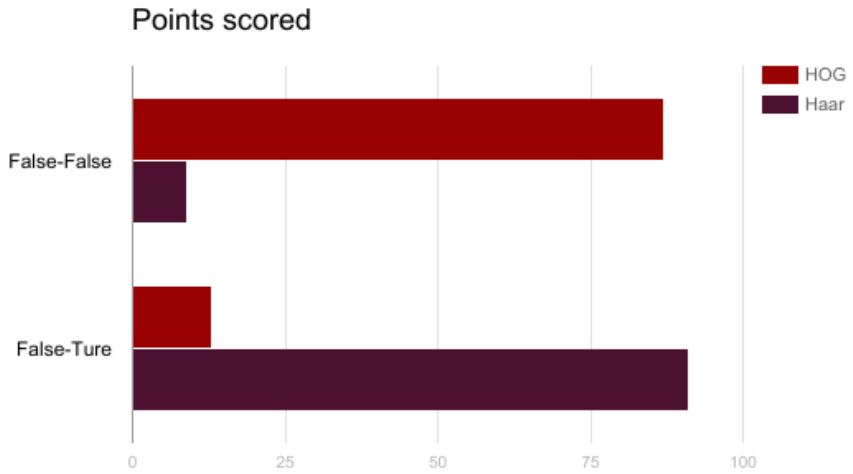


Figure 17. Comparison of HOG and Haar based human detection methods on 100 False images. False-False - where False images (non-human images) do not have detection; False-True - when False images (nonhuman images) are detected.

In an experiment with 100 False images (non-human images), the HOG based system only detected 13% of images, although Haar-based system detected roughly 91% of the images.

4.7 Detection and Tracking results

The final obtained results from our detection and tracking systems, are illustrated in

Figure 18, Figure 19, Figure 20.

Figure 18. The first pedestrian detection and tracking approach. Blue circle is a tracker, green rectangle is a detector.

As mentioned above, in this first approach, detection and tracking methods do not depend on each other. As demonstrated in the first part of the figure, the system can track, even if can not detect the pedestrian. The third part shows when the pedestrian increases the running speed and the system loses detection but still can keep track.

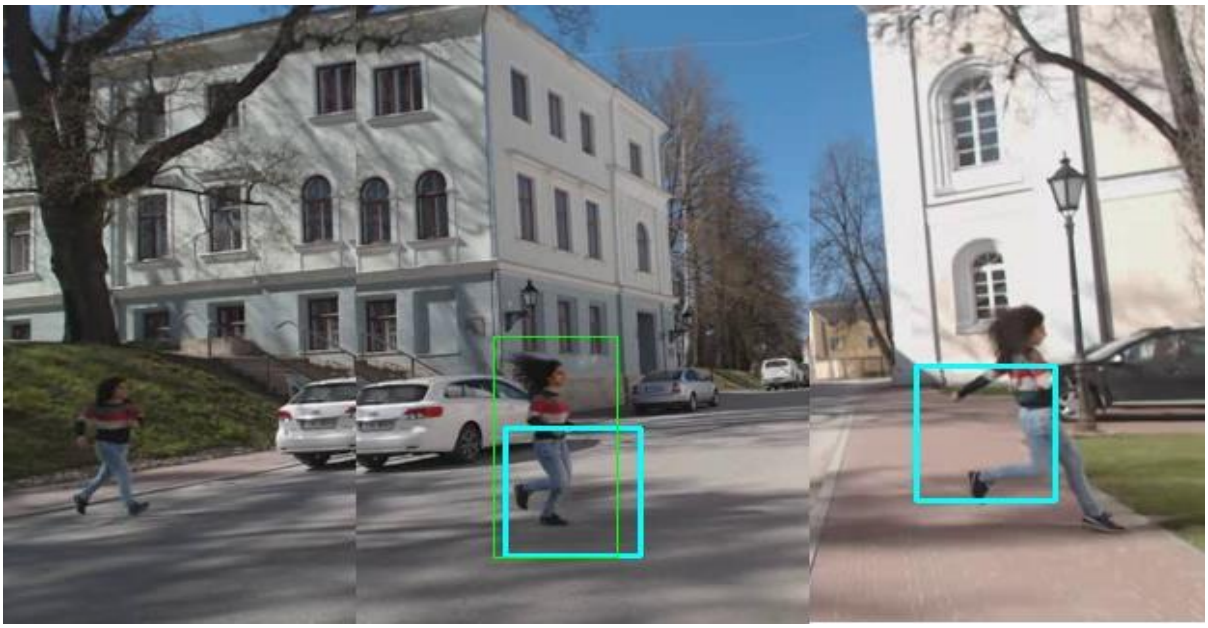


Figure 19. The second pedestrian detection and tracking approach. Here the blue rectangle is a tracker, green rectangle is a detector.

As work principle of the second tracking method depends on the output of the detection process, the system can not track a pedestrian without detecting. But the system can still keep track when it loses the detection, as demonstrated in the third part of Figure 19.

The first tracking approach is based motion, and the system tries to track, all moving objects in the frame. In Figure 18. (a), the system tries to track the reflection of a person. The second tracking system only tracks the person (Figure 18. (b)), identified by the detector.

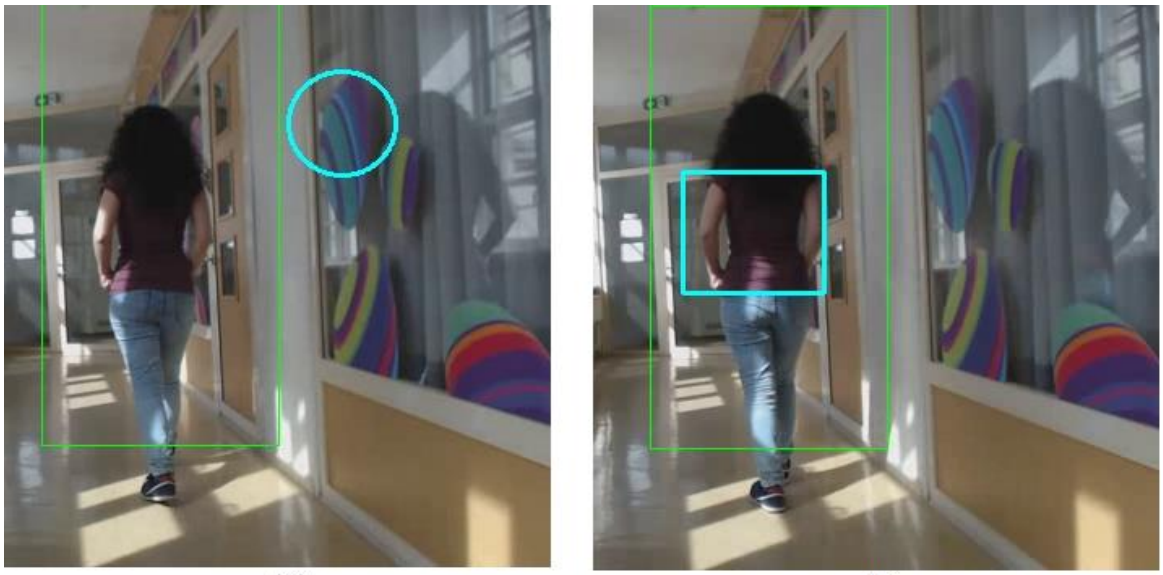


Figure 20. Detection and tracking result: (a) first approach; (b) the second approach

4.8 Conclusion

In this chapter, we demonstrated obtained outputs of the processes, have been done while working on this application.

In the first part, SIFT, SURF, LoG, Harries operation, Canny, HOG, Contour features, Line Segment Detector and so on feature detection methods have been examined. HOG and Contour feature has been selected for the detection and tracking systems development. Next, variety pre-processing methods, such as Erosion, Dilation, Opening, Closing, Morphological Gradient, Normalization, Threshold Binary, Adaptive Mean Thresholding, Adaptive Gaussian Thresholding, Threshold Binary Inv, Threshold Trunk, Threshold to Zero, Otsu's Thresholding, Threshold to Zero Inv, Otsu's thresholding following the Gaussian filtering, and four background subtraction methods, such as BackgroundSubtractorMOG, BackgroundSubtractorMOG2, BackgroundSubtractorKNN, BackgroundSubtractorGMG have been examined.

Dilation, Threshold Binary, BackgroundSubtractorKNN process have been chosen for improvement of the productivity.

In the fouth section, HOG-based detection method was tested against Haar-based detection method. Additionally, graphs of the detection achievement of these two detections method are illustrated.

In the final section, the output results of our detection, the first and second tracking methods are described.

Chapter 5: Conclusion

5.1 Conclusion

In this thesis, pedestrian detection and two different tracking methods were introduced.

In the first chapter, the objectives of the thesis and necessity of pedestrian detection and tracking systems in our life were described. Additionally, detailed the restrictions of this type systems.

In the second chapter, explained different already existing detection and tracking systems, the main components of these systems, and some researches toward to this topic.

The general architecture and most important components of the systems, which developed for this thesis, covered in the third chapter. Also, implementation methods, explained at the same chapter.

All obtained results were discussed while working on this thesis, in the fourth chapter.

5.2 Limitations and Future perspectives

From the final outputs of our pedestrian detection and tracking systems, it can be inferred that the systems have some limitations.

Already mentioned above, the first approach to tracking is motion based. Despite, the system tracks perfectly, and is efficient for security aspect, the goal of our thesis was to detect and track a human.

The second tracking approach, which is more relevant for our approach, is not as efficient as the first one. The limitation that the system can work only after detector initialization, as, it is detection-based tracking method is visible.

Even with limitations, these methods are considered as a good starting, towards to tracking issues, but for the future work, merging these two tracking methods, by taking their good features is aimed to be pursued.

Bibliography

[1] World Health Organization. 2017. Road traffic injuries. http://www.who.int/mediacentre/factsheets/fs358/en/ (Accessed 2017-05-14)

[2] World Health Organization. 2017. The top 10 causes of death http://www.who.int/mediacentre/factsheets/fs310/en/ (Accessed 2017-05-15)

[3] McKinsey & Company. Advanced driver-assistance systems: Challenges and opportunities ahead. 2017 http://www.mckinsey.com/industries/semiconductors/our-insights/advanced-driver-assistance-systems-challenges-and-opportunities-ahead (Accessed 2017-05-15)

[4] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60 (2) (2004): pp. 91-110.

[5] Bay H, Ess A, Tuytelaars T, Van L. Speeded-up robust features (SURF), Computer Vision and Image Understanding, 110 (3) (2008), pp. 346–359.

[6] C.P. Papageorgiou, M. Oren and T. Poggio. A general framework for object detection. International Conference on Computer Vision, 1998.

[7] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001.

[8] McConnell, Robert K. Method Of And Apparatus For Pattern Recognition. 1st ed. 1986.

[9] William T. Freeman, Michal Roth. Orientation Histograms for Hand Gesture Recognition. Mitsubishi Electric Research Laboratories, 1994.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.

[11] W.T. Freeman, K. Tanaka, J. Ohta, K. Kyuma. Computer vision for computer games. Automatic Face and Gesture Recognition, 1996.

[12] S. Belongie, J.Malik, J. Puzicha. Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (4) (2002): pp. 509-522.

[13] Kittipanya-ngam P., Lung E.H. (2011) HOG-Based Descriptors on Rotation Invariant Human Detection. In: Koch R., Huang F. (eds) Computer Vision – ACCV

2010 Workshops. ACCV 2010. Lecture Notes in Computer Science, vol 6468. Springer, Berlin, Heidelberg.

[14] Yadong Mu, Shuicheng Yan, Yi Liu, Thomas Huang, Bingfeng Zhou. Discriminative local binary patterns for human detection in personal album. Computer Vision and Pattern Recognition, 2008.

[15] Eli Shechtman, Michal Irani. Matching Local Self-Similarities across Images and Videos. Computer Vision and Pattern Recognition, 2007.

[16] Liu J, Zeng G. Description of interest regions with oriented local self-similarity. Optics Communications, 2012, 285 (10): pp. 2549–2557.

[17] Wang X, Han TX, Yan S. An HOG-LBP human detector with partial occlusion handling, in: Proceedings of the IEEE 12th International Conference on Computer Vision, pp. 32–39. 2009.

[18] Xin Y, Xiaosen S, Li S. A Combined Pedestrian Detection Method Based on Haar-like Features and HOG Feature, in: Proceedings of the 3rd International Workshop on Intelligent Systems and Applications (ISA), pp.1–4, 2011.

[19] Walk S, Majer N, Schindler K, Bernt S. New features and insights for pedestrian detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1030–1037, 2010.

[20] Wu B, Nevatia R. Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, 2008.

[21] Ye, Q., Jiao, J., and Zhang, B. (2010). Fast pedestrian detection with multi-scale orientation features and two-stage classifiers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , pp 881--884.

[22] Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: BMVC. (2009)

[23] Jianxin Wu, J. M. Rehg. CENTRIST: A Visual Descriptor for Scene Categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (8): pp. 1489-1501, 2011

[24] Sabzmeydani Payam, and Greg Mori. "Detecting pedestrians by learning shapelet features." Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007.

[25] Liu, Yazhou, et al. "Granularity-tunable gradients partition (GGP) descriptors for human detection." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[26] Kamarainen, Joni-Kristian. "Gabor features in image analysis." Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on. IEEE, 2012.

[27] Cortes, C. and Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3):273--297.

[28] Maji S, Berg AC, Malik J. Classification using intersection kernel support vector machines is efficient, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, 2008.

[29] Felzenszwalb, P. F., McAllester, D., and Ramanan, D. (2008). A discriminatively trained,multiscale, deformable part model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1--8.

[30] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. PAMI , 29(10):p 1848 – 1852, October 2007.

[31] Joachims T. Training linear SVMs in linear time. InProceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining: pp. 217-226, 2006.

[32] Freund, Y. and Schapire, R. V. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55: pp. 119--139.

[33] Schwartz WR, Kembhavi A, Harwood D, Davis LS. Human detection using partial least squares analysis. In Computer vision, 2009 IEEE 12th international conference in 2009, pp. 24-31.

[34] Russell, S., Norvig, P. and Artificial Intelligence: A modern approach. Prentice-Hall, Egnlewood Cliffs, 25, pp. 27, 1995

[35] Papageorgiou, C.P., Oren, M. and Poggio, T., 1998, January. A general framework for object detection. In Computer vision, 1998. sixth international conference on (pp. 555-562). IEEE.

[36] Viola, P., Jones, M.J. and Snow, D., 2003, October. Detecting pedestrians using patterns of motion and appearance. In null (p. 734). IEEE.

[37] Dollár, P., Belongie, S.J. and Perona, P. August. The Fastest Pedestrian Detector in the West. In BMVC, 2 (3), p. 7, 2010

[38] Stauffer, C. and Grimson, W.E.L., 1999. Adaptive background mixture models for real-time tracking. In Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. (Vol. 2, pp. 246-252). IEEE.

[39] Felzenszwalb, P.F., Girshick, R.B., McAllester, D. and Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence, 32(9), pp.1627-1645.

[40] Dalal, N., Triggs, B. and Schmid, C., 2006, May. Human detection using oriented histograms of flow and appearance. In European conference on computer vision (pp. 428-441). Springer Berlin Heidelberg.

[41] Lin, Z. and Davis, L., 2008. A pose-invariant descriptor for human detection and segmentation. Computer Vision–ECCV 2008, pp.423-436.

[42] Elgammal, A., Harwood, D. and Davis, L., 2000. Non-parametric model for background subtraction. Computer Vision—ECCV 2000, pp.751-767.

[43] Wu, B., Nevatia, R. and Li, Y., 2008, June. Segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (pp. 1-8). IEEE.

[44] Broida, T.J. and Chellappa, R., 1986. Estimation of object motion parameters from noisy images. IEEE transactions on pattern analysis and machine intelligence, (1), pp.90-99.

[45] Shi, J., 1994, June. Good features to track. In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference: pp. 593-600.

[46] Comaniciu, D., Ramesh, V. and Meer, P., 2003. Kernel-based object tracking. IEEE Transactions on pattern analysis and machine intelligence, 25(5), pp.564-577.

[47] Avidan S. Ensemble Tracking. Transactions on pattern analysis and machine intelligence, 20 (2), pp 261-271, 2007

[48] Grabner, H., Grabner, M. and Bischof, H., 2006, September. Real-time tracking via on-line boosting. In Bmvc, 1 (5): p. 6.

[49] Collins, R.T., Liu, Y. and Leordeanu, M., 2005. Online selection of discriminative tracking features. IEEE transactions on pattern analysis and machine intelligence, 27(10), pp.1631-1643.

[50] Babenko, B., Yang, M.H. and Belongie, S., 2009, June. Visual tracking with online multiple instance learning. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 983-990).

[51] Masoud, O. and Papanikolopoulos, N.P., 2001. A novel method for tracking and counting pedestrians in real-time using a single camera. IEEE transactions on vehicular technology, 50(5), pp.1267-1278.

[52] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report, University of North Carolina at Chapel Hill, Department of Computer Science, 2002.

[53] Benfold, B. and Reid, I., 2011, June. Stable multi-target tracking in real-time surveillance video. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on (pp. 3457-3464).

[54] Brendel, W., Amer, M. and Todorovic, S., 2011, June. Multiobject tracking as maximum weight independent set. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on (pp. 1273-1280).

[55] Kuo, C.H., Huang, C. and Nevatia, R., 2010, June. Multi-target tracking by on-line learned discriminative appearance models. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 685-692).

[56] Yilmaz, A., Javed, O. and Shah, M., 2006. Object tracking: A survey. Acm computing surveys (CSUR), 38(4), p.13.

[57] Enzweiler, M. and Gavrila, D.M., 2009. Monocular pedestrian detection: Survey and experiments. IEEE transactions on pattern analysis and machine intelligence, 31(12), pp.2179-2195.

[58] Lomte, R.S. and Malpe, K., REVIEW OF HUMAN DETECTION TECHNIQUES. threshold, 12, p.9.

[59] Benenson, R., Omran, M., Hosang, J. and Schiele, B., 2014. Ten years of pedestrian detection, what have we learned? arXiv preprint arXiv:1411.4304.

[60] Parekh, H.S., Thakore, D.G. and Jaliya, U.K., 2014. A survey on object detection and tracking methods. International Journal of Innovative Research in Computer and Communication Engineering, 2(2), pp.2970-2979.

[61] Adrian Rosebrock. 2015. HOG detectMultiScale parameters explained. http://www.pyimagesearch.com/2015/11/16/hog-detectmultiscale-parameters-explained/ (Accessed 2017-05-14)

[62] Welch, Greg, and Gary Bishop. "An introduction to the Kalman filter." (1995).

[63] Kurzynski, M., Puchala, E., Woźniak, M. and Zolnierek, A. eds., 2007. *Computer Recognition Systems 2* (Vol. 45). Springer Science & Business Media.

[64] INRIA Person Dataset; GRAZ-01 dataset; http://www-old.emt.tugraz.at/~pinz/data/

Appendix

Method	MR	Family	Features	Classifier	Context	Deep	Parts	M-Scales	More data	Feat. type	Training
VJ [9]	94.73%	DF	✓	✓						Haar	I
Shapelet [10]	91.37%	-	✓							Gradients	I
PoseInv [11]	86.32%	-					✓			HOG	I+
LatSvm-V1 [12]	79.78%	DPM					✓			HOG	P
ConvNet [13]	77.20%	DN				✓				Pixels	I
FtrMine [14]	74.42%	DF	✓							HOG+Color	I
HikSvm [15]	73.39%	-		✓						HOG	I
HOG [1]	68.46%	-	✓	✓						HOG	I
MultiFtr [16]	68.26%	DF	✓	✓						HOG+Haar	I
HogLbp [17]	67.77%	-	✓							HOG+LBP	I
AFS+Geo [18]	66.76%	-			✓					Custom	I
AFS [18]	65.38%	-								Custom	I
LatSvm-V2 [19]	63.26%	DPM		✓			✓			HOG	I
Pls [20]	62.10%	-	✓	✓						Custom	I
MLS [21]	61.03%	DF	✓							HOG	I
MultiFtr+CSS [22]	60.89%	DF	✓							Many	T
FeatSynth [23]	60.16%	-	✓	✓						Custom	I
pAUCBoost [24]	59.66%	DF	✓	✓						HOG+COV	I
FPDW [25]	57.40%	DF								HOG+LUV	I
ChnFtrs [26]	56.34%	DF	✓	✓						HOG+LUV	I
CrossTalk [27]	53.88%	DF			✓					HOG+LUV	I
DBN-Isol [28]	53.14%	DN					✓			HOG	I
ACF [29]	51.36%	DF	✓							HOG+LUV	I
RandForest [30]	51.17%	DF		✓						HOG+LBP	I&C
MultiFtr+Motion [22]	50.88%	DF	✓						✓	Many+Flow	T
SquaresChnFtrs [31]	50.17%	DF	✓							HOG+LUV	I
Franken [32]	48.68%	DF		✓						HOG+LUV	I
MultiResC [33]	48.45%	DPM			✓		✓	✓		HOG	C
Roerei [31]	48.35%	DF	✓					✓		HOG+LUV	I
DBN-Mut [34]	48.22%	DN			✓		✓			HOG	C
MF+Motion+2Ped [35]	46.44%	DF			✓				✓	Many+Flow	I+
MDCO [36]	45.53%	-	✓		✓					HOG+LBP	C
MultiSDP [37]	45.39%	DN	✓		✓	✓				HOG+CSS	C
ACF-Caltech [29]	44.22%	DF	✓							HOG+LUV	C
MultiResC+2Ped [35]	43.42%	DPM			✓		✓	✓		HOG	C+
WordChannels [38]	42.30%	DF	✓							Many	C
MT-DPM [39]	40.54%	DPM					✓	✓		HOG	C
JointDeep [40]	39.32%	DN			✓					Color+Gradient	C
SDN [41]	37.87%	DN				✓	✓			Pixels	C
MT-DPM+Context [39]	37.64%	DPM			✓		✓	✓		HOG	C+
ACF+SDt [42]	37.34%	DF	✓						✓	ACF+Flow	C+
SquaresChnFtrs [31]	34.81%	DF	✓							HOG+LUV	C
InformedHaar [43]	34.60%	DF	✓							HOG+LUV	C
Katamari-v1	22.49%	DF	✓		✓				✓	HOG+Flow	C+

(taken from [59])

License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Asmar Hasanova** (date of birth: 24.08.1992),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Pedestrian Detection and Tracking in Urban Context Using a Mono-Camera

supervised by Amnir Hadachi, PhD

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **25.05.2017**