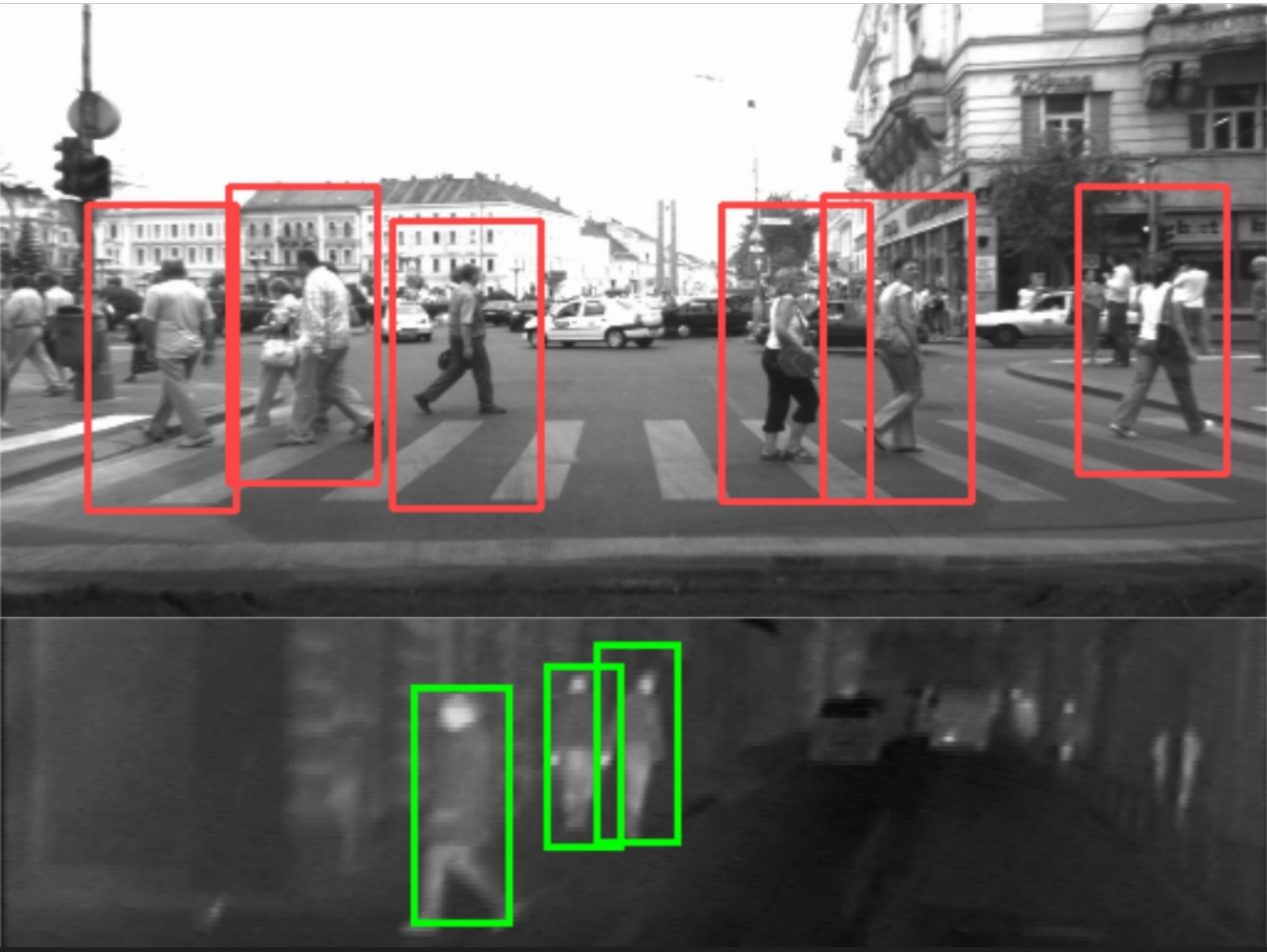
Raluca Didona BREHAR



Contributions to Feature Based Pedestrian Detection Methods in Visible and Infrared Monocular Images

UTPRESS
Cluj-Napoca, 2020
ISBN 978-606-737-461-2

Raluca Didona Brehar

Contributions to Feature Based Pedestrian Detection Methods in Visible and Infrared Monocular Images

UTPRESS
Cluj - Napoca, 2020
ISBN 978-606-737-461-2

Editura U.T.PRESS
Str. Observatorului nr. 34
C.P. 42, O.P. 2, 400775 Cluj-Napoca
Tel.:0264-401.999
e-mail: utpress@biblio.utcluj.ro
http://biblioteca.utcluj.ro/editura

Director: ing. Călin Câmpean

Recenzia: Conf. dr. ing. Camelia Lemnaru
Conf. dr. ing. Tiberiu Marița

ISBN 978-606-737-461-2

Contents

List of Figures

Preface

The detection of humans in still images and especially in traffic scenarios is an important problem for artificial vision, pattern recognition and in a broader context for autnomous vehicles. A robust solution to this problem has various applications to fields such as autonomous driving systems, video surveillance, image retrieval, vulnerable road user protection.

The goal of this book is to bring an overview of existing solutions for feature based pedestrian detection systems and to present the contributions brought by the author in this field. The book presents the main milestones of the pedestrian detection techniques in the context of classical machine learning methods such as Support Vector Machines, Adaptive Boosting or Bayesian Network trained on visual features extracted from monocular intensity or infrared images.

The detailed theoretical and pragmatic solutions are extracted from the author's PhD Thesis entitled "Adaptive Search Space Pruning in the Context of Multiple Attitude Pedestrian Detection Models" (2015) and are also found in the scientific papers published by the author.

This book is addressed to computer science students that are in their senior year or pursuing a Masters degree and to young researchers that want to get an introdoction to classical pedestrian detection systems. This book assumes that the reader has reasonable knowledge in the areas of image processing, computer programming, data structures and algorithms, artificial intelligence and pattern recognition systems.

Chapter 1

Introduction

Pedestrian detection is an extremely active field of scientific and technological exploration. From many years researches in all the world have tried to build systems able to detect pedestrians in images captured with various sensors setups. Monocular or sterovision systems, Lidar, Radar or Infrared based frameworks have been widely explored.

The detection of humans in still images and especially in traffic scenarios is an important problem for artificial vision and pattern recognition. A robust solution to this problem should have various applications to autonomous driving systems, video surveillance, image retrieval.

In general, the goal of pedestrian detection is to *determine the presence of humans in images and videos and return information about their position.* The problem of detecting pedestrians has a high degree of complexity because of the large intra-class variability, as pedestrians are highly deformable objects whose appearance depends on numer- ous factors like: pose, orientation, shape, attitude, occlusions, imaging conditions, background.

With respect to other identities that appear in traffic (vehicles, road, traffic signs) that have a rigid structure, pedestrians may adopt a large variety of appearances due to the actions they perform (walk, run, stand), due to the motion of different body parts, due to the clothing and accessories they wear. Hence pedestrians posses a large intra-class variability because they are highly deformable instances in a traffic scene and their appearance depends on numerous factors like: pose, orientation, shape, attitude, occlusions, imaging conditions, background. The difficulties in pedestrian detection algorithms arise exactly from this high variance in appearance.

The main components of a classical pedestrian detector are shown in Figure 1.1.

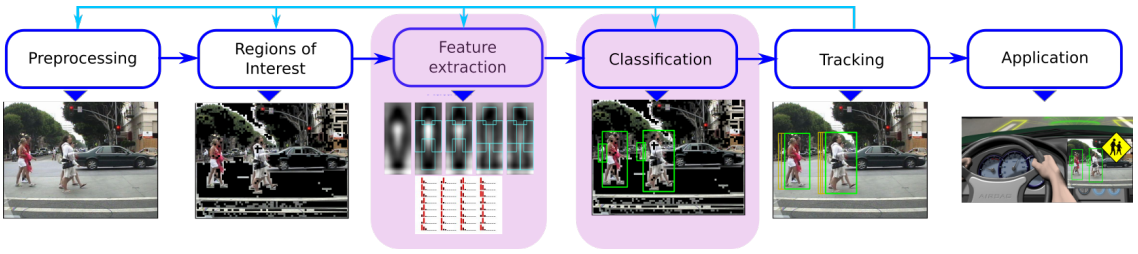


Figure 1.1: Sample processing pipeline for a classical pedestrian detector. Feature extraction and classification are detailed in this book.

A first component of a classical pedestrian detector comprise a region of interest generation

pocess. This is performed by means of sterovision or in the case of monocular cameras it is done using geometric assumptions – for example most pedestrians are localized on the ground (road) plane or if a pedestrian is far away from the ego-vehicle than its size in the aquired image will be smaller than the size of a pedestrian that is close to the vehicle. The region of interest generation process is a separate subject that is not in the context of this book.

The second component of the classical pedestrian detection system is the feature extractor that explores the relevant features which make a pedestrian distinguishable with respect to other entities that appear in a traffic scene.

The third component is the classifier or detector. Based on the design priciples of the detector, several features are extracted in the given regions of interest, and based on those features a classification model is applied having as result several detections. Each detection has associated a likelyhood that it represents a pedestrian.

Furthermore, in the detection pipeline, tracking operations can be applied in order to improve the flickering effects of a detector (that is a pedestrian is detected in the previous frames, than in the current frame it is not detected and in the next frames it is detected again).

In this book two components of a classical pedestrian detector are explored: the feature extraction and classification. The explanations provided in this book detail classical approaches for a pedestrian detection system applicable in the context of monocular infrared and monocular visible systems.

Chapter 2

Visual Fetures

Choosing the features implied in the design of a classifier represents a very important step. A robust set of attributes must be used in order to recognize the humanoid shape in a cluttered background and under difficult illumination conditions.

Existing approaches exploit descriptors based on first order partial derivatives computed in four directions, histogram of gradient orientations, anisotropic Gaussians, Gabor wavelets, Haar wavelets, Local Binary Patterns.

2.1 First Order Partial Derivatives

The magnitude of the first order partial derivatives is used because the sign of the magnitude of the first order partial derivative is uninformative due to varying clothing and background colors. In order to decrease the influence of small spatial shifts in the detection window, a local average of the first order partial derivatives in each direction by convolving their responses with a 2D averaging filter is performed. For each image $I(x)$ in the training set the following operation is performed:

$$GI_d(x) = |I(x) * G_d)| * B \tag{2.1}$$

where $*$ denotes the convolution, G_d is the derivative kernel ($[-1, 0, 1]$ or $[-1, 0, 1]^T$) used for obtaining the derivatives in direction $d \in D$, B is a 2D averaging filter and GI_d is the result image that captures the amount of first order partial derivative information at every pixel location, in direction d. The set $D = \{0°, 45°, 90°, 135°\}$ represents the directions in which the partial derivatives have been computed. Figure 2.1 shows the resulting features for the four directions. Each feature is characterized by direction, absolute value of the first order partial difference and position in the image. We divide each image GI_d into several blocks of different dimensions. The features in each block are normalized. For each block we perform a correlation feature selection (CFS) algorithm. The selected features from all the blocks form a vector of descriptors that represent the input to a classification algorithm.

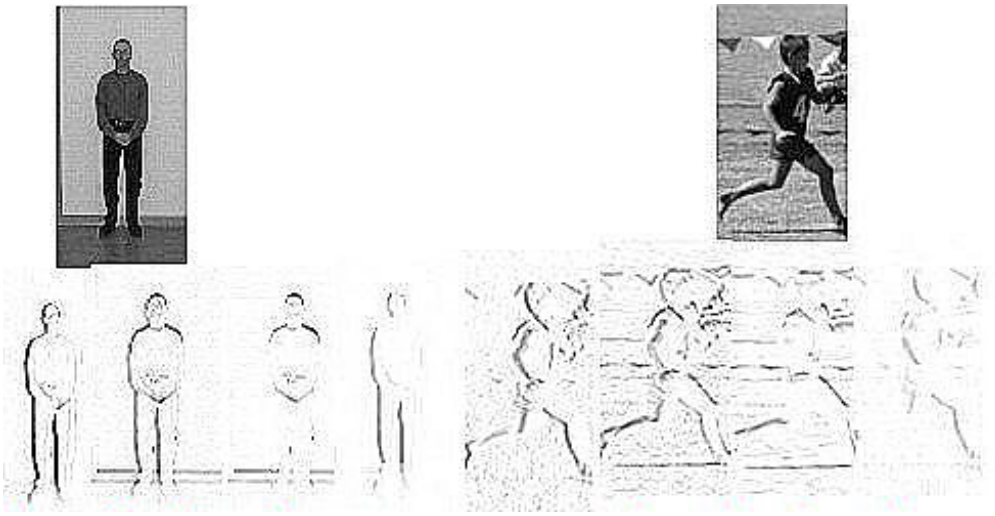


Figure 2.1: Directional derivatives

2.2 Histogram of Oriented Gradients

History and usage of HOG features

HOG features have become extremely popular in pedestrian and object detection in general since their introduction in 2005. The HOG features are employed by [116], [114], [99], [26], [117] , [98], [52], [22], [11], [4], [115], [56], [118].

Several variations and improvements of the HOG feature model have been proposed. For example [8] propose a PCA algorithm to reduce the number of features per cell (will be referred as FHOG), [98] describe a co-occurrence semantic HOG, [74] analyze the influence on accuracy of different ways of computing the histograms in the HOG blocks. FHOG are used by all papers that refer to the deformable part model like [24], [77], [64].

LiteHog designed to make use of the spare computational resources and LiteHog+ which is more balanced in terms of memory and processor bandwidth are proposed by [119].

Another extension of HOG based on Comparison of Granules (HOG-CoG) is proposed by [120] and extended in the form of histograms of oriented gradient of granules (HOGG) by [121]. Instead of collecting gradient information at each pixel, the histograms of gradients in small regions are computed. HOGG with different granularity can describe the contour while ignoring the noisy edges. They prove that with the help of the integral image technique, the evaluation of HOGG can be efficient.

A data driven feature transformation that improves the performance of gradient histogram based features is proposed by [122]. They replace the gradient orientations with general filters that preserve the unit norm and 0-mean properties. A modified spherical k-means algorithm that uses sample medoids is employed for defining a vocabulary of image patches that are used for constructing the block descriptors.

Feature Interaction Descriptor (FIND) is employed by [68]. It is based on HOG features and it is computed in two phases: (1) constructing the localized oriented gradients histogram and calculating the interaction of adjacent histogram bins using a histogram-similarity function.

Irregular HOG patterns are learned in a discriminative fashion in the work of [123].

The opponent colors (OPP) space is exploited by [124] and [108] as a biologically inspired alternative for human detection. They feed the OPP space in the baseline framework of [116] and obtain better detection performance than by using RGB space.

HOG is improved by the usage of a segmentation based weighting scheme instead of gradient magnitude based weighting in the work of [125]. The gradient at a certain pixel location only encodes differences between adjacent pixels in intensity or color, while [125] incorporate in the HOG descriptor differences based on a wider spatial support. The differences are provided as weights resulting from a mean-shift segmentation algorithm applied to the CIE Luv color space.

The process of constructing the histograms of oriented gradients comprises the following steps:

- Gradient computation.
- Spatial/orientation binning.
- Normalization and descriptor blocks.

Each step will be described in detail in the next paragraphs.

Gradient computation

For each point of an image I the gradient magnitude, M and orientation θ are computed as follows:

$$GI_x = (I * B) * G_x \tag{2.2}$$

$$GI_y = (I * B) * G_y \tag{2.3}$$

$$M = \sqrt{(GI_x)^2 + (GI_y)^2} \tag{2.4}$$

$$\theta = \arctan \frac{GI_y}{GI_x} \tag{2.5}$$

where B is a Gaussian smoothing kernel, $G_x = [-1, 0, 1]^T$, $G_y = [-1, 0, 1]$.

Spatial/orientation binning

In the next step each pixel provides a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centered on it, and the votes are accumulated into orientation bins over local spatial regions that are called *cells* as shown in2.2. Rectangular cells were considered. The orientation bins are evenly spaced over 0 °– 360 °("signed" gradient).

To reduce aliasing, votes are interpolated bi-linearly between the neighboring bin centers in both orientation and position. The vote is a function of the gradient magnitude at the pixel.

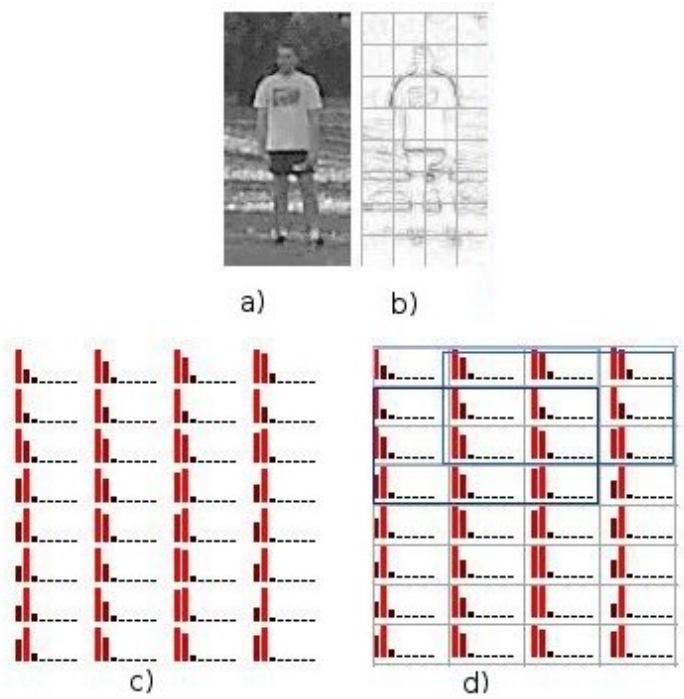


Figure 2.2: a)Original Image; b)Gradient magnitude and cell division; c)Histogram of Oriented Gradients computed on each cell; d)Cell grouping into blocks within which normalization is made.

Normalization and descriptor blocks

Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast, so effective local contrast normalization turns out to be essential for good performance. The normalization scheme groups the cells into larger spatial blocks and contrast normalizes each block separately. The final feature descriptor is the vector of all components of the normalized cell responses from all of the blocks in the image. The blocks are overlapped, so that each scalar cell response contributes to several components of the final descriptor vector, each normalized with respect to a different block. This may seem redundant but good normalization is critical and including overlap significantly improves the performance.

Normalization scheme

As it turns out from the work of [116], effective local contrast normalization is essential for a good performance of pedestrian detectors based on gradient attributes. For both feature types L_2-norm normalization algorithm applied on image blocks is used. Suppose that within a block there is a vector of k features denoted by f_d. The value of a feature $f_d(i)$ can either be the magnitude of the first order partial derivatives computed in four directions or a histogram

value (for HOG attributes). The normalization equation is:

$$f_d(i) = \frac{f_d(i)}{\sqrt{\sum_{i=1}^{k}(f_d(i))^2 + \epsilon}} \tag{2.6}$$

where ϵ is a small constant.

2.3 Haar Filters

The used Haar filters are reminiscent of Haar basis functions which have been used by [153] and [154]. They operate on gray level images and their value is represented by the difference of sums computed over rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent. The three kinds of features proposed by [154]: two-rectangular, three rectangular and four-rectangular features and a nine rectangular feature, as shown in Figure 2.3 have been explored. The value of a two-rectangular feature is the

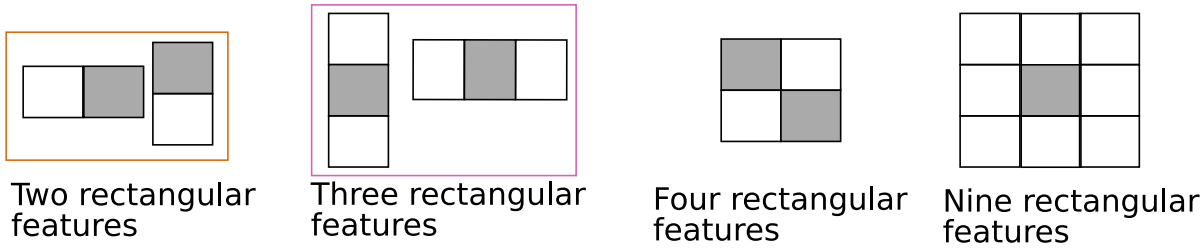


Figure 2.3: Haar features

difference between the sums of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent. A three-rectangular feature computes the sum within two outside rectangles subtracted from the sum in center rectangle. A four-rectangular feature computes the difference between diagonal pairs of rectangles. Two other types of features can easily be generated by rotating the first two types by 90°. A nine-rectangular feature is computed in a rectangular region divided into nine equal blocks. The value is given by the difference between the sum of the pixels in the middle rectangle and the sum of the pixels in the other eight rectangles.

Integral Image

Rectangle features can be computed very rapidly using an intermediate representation for the image called the *integral image*[154]. The integral image at location x, y contains the sum of the pixels above and to the left of x, y, inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{2.7}$$

where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image. The integral image can be computed in one pass using the following pair of recurrences:

$$s(x, y) = s(x, y - 1) + i(x, y) \tag{2.8}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \tag{2.9}$$

where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$ and $ii(-1, y) = 0$.

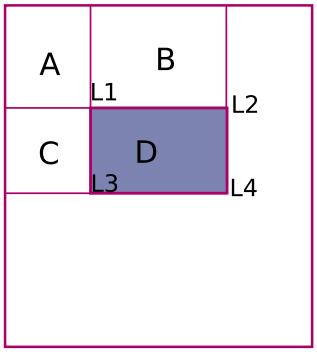


Figure 2.4: Integral Image representation

As shown in Figure 2.4 the sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location L1 is the sum of the pixels in rectangle A. The value at location L2 is $A + B$, at location L3 is $A + C$, and at location L4 is $A + B + C + D$. The sum within D can be computed as $L4 + L1 - (L2 + L3)$. Hence, using the integral image, any rectangular sum can be computed in four array references. Clearly, the difference between two rectangular sums can be computed in eight references. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, nine for four-rectangle features and five for the nine-rectangle features.

2.4 Local Binary Patterns

Local Binary Patterns (LBP) have been introduced by [155]. Originally they have been used for texture recognition and more recent approaches included them in the field of pedestrian detection. The LBP local histogram was used for face detection by [156] and this idea was extended to work with pedestrian detection.

LBP is used by [157] for detecting pedestrians at night or in a dark environment setting that needs to overcome problems of low contrast, image blur and noise. They propose three types of LBP based features: weighted LBP, Multi-resolution LBP, and Multi-scale LBP. Their experimental results show that the proposed method improves upon the basic LBP significantly and outperforms benchmarks such as histogram of oriented gradients and co-occurrence histograms of gradient orientations (CoHOG).

Another LBP-based feature, termed pyramid center-symmetric local binary/ternary patterns (pyramid CS-LBP/LTP), for pedestrian detection is introduced by [158]. The proposed CS-LBP captures the gradient information, it is easy to implement and computationally efficient, which is desirable for real-time applications.

For each point of an image I the LBP operator generates a binary code considering a threshold-ed difference of intensity values between the pixel and some points in its local

neighborhood. The threshold value is zero. The value of the LBP code for a pixel (xc; yc) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \times 2^p \tag{2.10}$$

where P is the number of neighbors to be analyzed on a circle of radius R centered at xc, yc; $s(x) = 0$ if $x \geq 0$ and $s(x) = 1$ otherwise; g_p is the intensity of neighbor p and g_c is the intensity of the current (center) pixel. The number of features extracted by the LBP operator can be reduced by using the so called uniform patterns [159]. These patterns are used to reduce the length of the feature vector and also implement a simple rotation-invariant descriptor. A local binary pattern is called uniform if the binary pattern contains at most two bit-wise transitions from 0 to 1 or vice verse when the bit pattern is traversed circularly [160]. In the computation of the LBP labels, uniform patterns are used so that there is a separate label for each uniform pattern and all the non-uniform patterns are labeled with a single label. For example if a neighborhood of 8 pixels on a circle of radius 1 is used the total number of patterns is 256 and for them 59 different labels are obtained out of which 58 are uniform and the last is used for the non-uniform patterns.

2.5 Anisotropic Gaussians

Anisotropic Gaussian features were introduced by [161]. They are constructed from base functions of an over complete basis. The expansion of any image in the base is not unique. The generative function $\phi(x, y) : \mathbb{R}^2 \to \mathbb{R}$ is described by the equation $\phi(x, y) = xe^{-(|x|+y^2)}$. It is made of a combination of a Gaussian and its first derivative. This presents the ability of approximating efficiently contour singularities with a smooth low resolution function in the direction of the contour and it approximates the edge transition in the orthogonal direction.

Figure 2.5 shows some Anisotropic Gaussian kernels with different scaling, bending, rotating and translating parameters. Figure 2.6 displays some features computed on a pedestrian image.

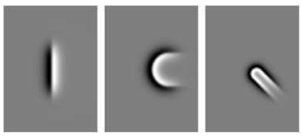


Figure 2.5: Examples of Anisotropic Gaussian kernels

Different transformations can be applied to this generative function:

- Translation by (x_0, y_0): $\mathcal{T}_{x_0,y_0}\phi(x, y) = \phi(x - y_0, y - y_0)$
- Rotation by θ: $\mathcal{R}_\theta\phi(x, y) = \phi(xcos\theta - ysin\theta, xsin\theta + ysin\theta)$.
- Bending by r:

$$\mathcal{B}_r\phi(x, y) = \begin{cases} \phi(r - \sqrt{(x-r)^2 + y^2}, r \arctan(\frac{y}{r-x})) & \text{, if } x < r \\ \phi(r - |y|, x - r + r \times \frac{\pi}{2}) & \text{, if } x \geq r \end{cases}$$

Figure 2.6: Original image and two Anisotropic Gaussians computed on it

- Anisotropic scaling by (s_x, s_y): $\mathcal{S}_{s_x,s_y}\phi(x,y) = \phi(\frac{x}{s_x}, \frac{y}{s_y})$

By combining these four basic transformations, a large collection $\mathcal{D}$ of $\psi_{s_x,s_y,\theta,r,x_0,y_0}$ as defined by equations bellow is obtained:

$$\psi_i(x,y) = \psi_{s_x,s_y,\theta,r,x_0,y_0}(x,y) = \mathcal{T}_{x_0,y_0}\mathcal{R}_\theta\mathcal{B}_r\mathcal{S}_{s_x,s_y}\phi(x,y)$$

The obtained anisotropic features were normalized and as their number was very large a random selection of 2640 features was applied. Parameters θ_j and p_j are chosen using Bayes decision rule.

2.6 Gabor Wavelets

The Gabor's theory is based on the failure of Fourier transform. The Fourier transform is a linear combination of trigonometric functions, where the scalars or coefficients are given as inner products between the original signal and each trigonometric function. Each coefficient reveals how much energy the signal contains of that particular (corresponding) trigonometric function. A trigonometric function is the same as a particular frequency Thus; a Fourier transform determines the frequency content of a signal. The Fourier transform has unique mapping of a time domain representation in frequency domain. This is called as one to one mapping. It is advantageous due to following points:

- Uniqueness.
- It is very much specific to period and scale.
- Fourier analysis is fast using Fast Fourier Transform (FFT).
- Relevant for quantification of stationary signals.

But it is has some disadvantages like:

- FFT requires the size of the image to be of the power of 2.

- Problem with boundary condition – in other words after segmentation region can be very well identified but boundary conditions are not defined.
- Time domain and frequency domain description of a signal are inversely related.

In 1946, Dennis Gabor [162], the inventor of the hologram, proposed the expansion of a wave in terms of Gaussian wave packets. An example of such a wave packet is a sine wave multiplied by a Gaussian function. If a signal is modulated (multiplied) by a Gaussian window of a certain width and central time, then a Fourier expansion of the modulated signal gives a measure of the local spectrum. Clearly such a spectrum is not unique since the width of the Gaussian is arbitrary; but nevertheless, such local spectra are extremely useful. If a collection of local spectra is computed for a suite of window positions, the result is a time-frequency decomposition called a Gabor transform. Furthermore, if the signal can be reconstructed from this decomposition, then a non-stationary filter can be achieved by modifying the decomposition before reconstruction. Gabor observed that there should be a presentation or representation which is local both in time and frequency domain and such a local time and frequency representation should be discrete so that it is better adapted to various applications. Gabor proposed to expand a function into a series of elementary functions, which are constructed from a single building block by translation and modulation.

The two-dimensional Gabor functions that we have used were introduced by [163]. A two-dimensional Gabor function consists of a sinusoidal plane wave of some frequency and orientation, modulated by a two-dimensional Gaussian. The Gabor filter in the spatial domain is given by:

$$g_{\lambda,\theta,\sigma,\gamma}(x,y) = \exp\left(-\frac{x'^2+\gamma^2+y'^2}{2\sigma^2}\right)(\cos 2\pi\frac{x'}{\lambda}+\psi) \tag{2.11}$$

where:

$$x' = x\cos(\theta) + y\sin(\theta),$$
$$y' = y\cos(\theta) - x\sin(\theta)$$

The terms in equation 2.11 are:

- λ – wavelength of the cosine factor
- θ – orientation of the normal to the parallel stripes of a Gabor function in degrees
- ψ – phase offset in degrees
- γ – spatial aspect ratio (spatial width) and specifies the elliptically of the support of the Gabor function and
- σ – standard deviation of the Gaussian and determines the (linear) size of the receptive field.

The parameter λ is the wavelength and $f = 1/\lambda$ is the spatial frequency of the cosine factor. The ratio σ/λ determines the spatial frequency bandwidth of simple cells and thus the number of parallel excitatory and inhibitory stripe zones which can be observed in their receptive fields.

Figure 2.7 depicts some examples of Gabor filters with different combinations of spatial width, frequency and orientation.

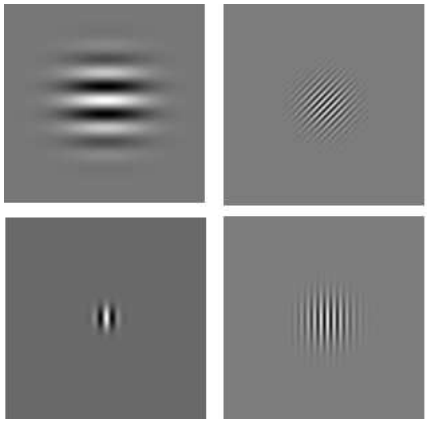


Figure 2.7: Gabor filters with different combinations of spatial width, frequency and orientation

2.7 Mixture of Features

This category comprises approaches that use a several visual features for the pedestrian detection task. These features can be computed on a single modality – for example the grayscale image or on multiple modalities like the grayscale image, density image, optical flow image or on multiple channels like RGB, LUV.

Multi-cue features

A combination of two or more features is employed by several authors. For example [117] introduce the self-similarity features on color channels and combine them with HOG features and motion features derived from optical flow.

HOG features and Local Binary Patterns (LBP) are computed for several cues (intensity, optical flow, depth) in the work of [23]. Different combinations of HOG with Local Oriented Pattern (LOP), Color Self-Similarity (CSS), and Texture Self-Similarity (TSS) fed to a Support Vector Machine are exploied by [129]. Shape and texture features are combined in the work of [51], [23], [130]. The shape-based detection involves a coarse-to-fine matching of an exemplar-based shape hierarchy to the image data. The texture representation is given by local adaptive receptive field features or HOG features. Haar wavelets and edge orientation histograms are employed by [131], [132]. HOG and LBP are combined in the work of [133], [134].

Channel features

The idea beyond channel features is derived from integral images and it strives to generate and compute features efficiently using integral images over multiple registered image channels. Integral Channel Features (ICF) [15] represent simple rectangular features that perform a sum operation over a given image region. For the particular task of pedestrian detection [15] show that good results are obtained when using 6 quantized orientations, 1 gradient magnitude and 3 LUV color channels. Integral channel features are also used by [30], [135],

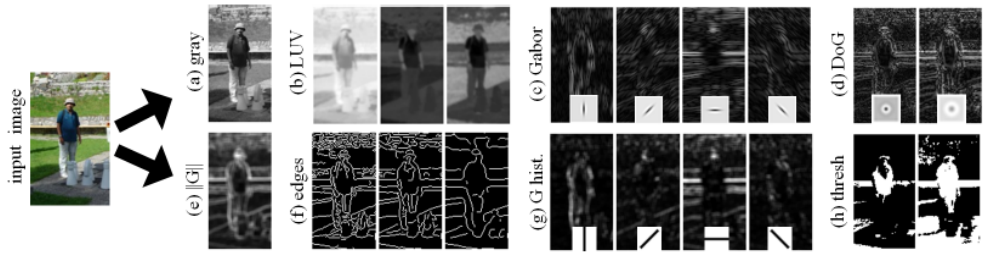


Figure 2.8: Integral channel features used by [15]

[123].

Sketch tokens computed on integral channel features are employed by [136] for improving pedestrian detection. Sketch tokens are mid-level features that capture local edge structure. Their appearance ranges from straight lines and junctions to curves and sets of parallel lines. Clustering is used to form sketch token classes and a random forest classifier is employed for efficient detection of sketch tokens in novel images.

Aggregated Channel Features (ACF) introduced by [86] and available in the framework of [87] are based on integral channel features. The idea of ACF is that given an input image I, several channels $C = \Omega(I)$ are computed, then every block of 4×4 pixels in C are summed and the resulting lower resolution channels are smoothed.

Channel features are also used by [76] and are applied directly on the scan windows that remain after a pruning process. The first three channels are obtained by resizing the bounding box centered on the pedestrian with three different scales and the Y-channels from the YUV color spaces are extracted. The next three channels are Sobel edge maps of the three Y-channels. Hence [76] learn features with multiple scales and boundary cues that are given by Gabor filters.

Integral World Channels and Boosting are used for detecting pedestrians in the work of [137]. They build the integral world channels using HOG, LBP and LUV features. A visual codebook is build on top of these features. The pedestrians are detected by scanning the fixed size image with sliding windows at different scales and using the same classifier because different code words would be activated for near and far pedestrians.

Multi-modal multi-channel Haar like features

These features are Haar like templates applied on several channels like LUV, gradient magnitude, histogram of oriented gradients proposed by [16]. Haar templates are generated by sliding rectangular windows of pre-defined sizes over a pre-defined pedestrian shape model. The template can be binary – traditional Haar features with weights +1 and -1 and ternary (shown as white, black, and red areas) which are given the weights of +1, -1, and 0, respectively.

Multi-channel Local Binary Patterns

Local Binary Patterns are computed by [21] on each component of the YCbCr space and on the low pass and high pass images of the Y channel. Several quantization schemes are

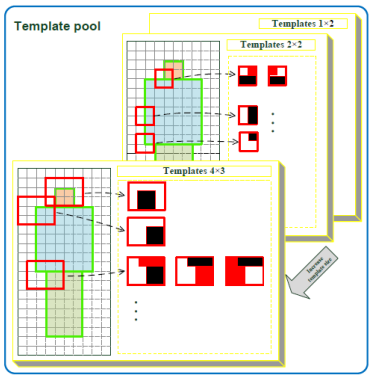


Figure 2.9: Haar templates used by [16]

explored in their work.

Binary Visual Elements

By means of random projections and adaptive thresholding [138] transform histogram based image representations like HOG, LBP and BOW into binary item transactions. Then using data mining algorithms like Jumping Emerging Patterns they build histograms of pattern sets that are used for the classification task.

Context Cues

The context is a high level information that can be added to the detection framework. Context has been explored at level feature or at pattern classification level.

For context features we should mention the work of [17] that combine the local scan windows with neighborhood windows in order to construct a multi-scale image context descriptor. They use multi-scale HOG features and local difference patterns (LDP) for constructing the feature vector. These features are iteratively learned by ContextBoost method (see section

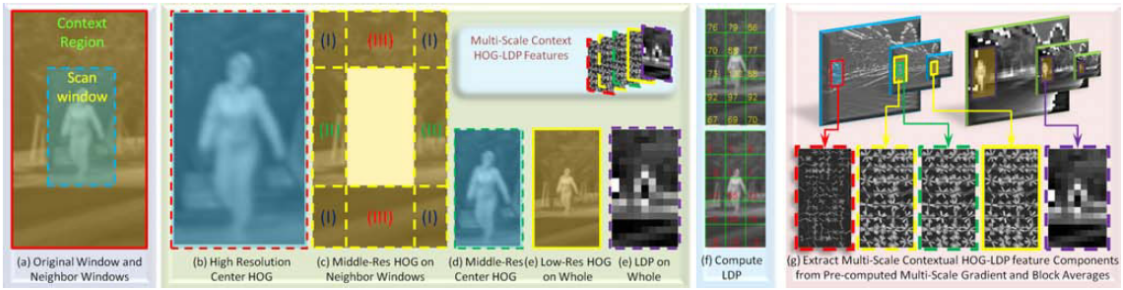


Figure 2.10: Multi context image features proposed by [17]

??). we develop an iterative classification algorithm called contextual boost. At each iteration, the classifier responses from the previous iteration across the neighborhood and multiple image scales, called classification context, are incorporated as additional features to learn a new classifier.

Correlation-based feature selection

The purpose of correlation-based feature selection (CFS) scheme is to eliminate redundant attributes as well as irrelevant ones. As presented by [164] and [165], CFS tries to select good feature subsets that contain attributes highly correlated with the class, yet uncorrelated with each other. The correlation between two attributes A and B can be measured using the symmetric uncertainty [165]:

$$U(A,B) = 2 \times \frac{H(A) + H(B) - H(A,B)}{H(A) + H(B)} \tag{2.12}$$

where H is the entropy function :

$$H(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots p_n \log p_n \tag{2.13}$$

The symmetric uncertainty always lies between 0 and 1. Correlation-based feature selection determines the goodness of a set of attributes using:

$$\frac{\sum_j U(A_j, C)}{\sqrt{\sum_i \sum_j U(A_i, A_j)}} \tag{2.14}$$

where C is the class attribute and the indexes i and j range over all attributes in the set.

Attribute selection is normally done by searching the space of attribute subsets and evaluating each set. Search can be performed exhaustively, using a simple genetic algorithm, randomly, or by greedy hill-climbing with or without backtracking. In our experiments we have used the CFS implementation provided by [166].

Chapter 3

Machine Learning Algorithms for Pedestrian Detection

This section performs an overview of the machine learning algorithms that are frequently used for the particular task of pedestrian detection, namely Bayesian Networks, AdaBoost, Multi-layer perceptron and Support vector machines.

These methods operate on a set of feature vectors and walk through a large area of machine learning techniques. The choice of a particular learning strategy has a great influence on the overall speed and accuracy of the pedestrian detection method. For example non-linear classifiers such as Radian Basis Function for Support Vector Machines have a good accuracy but are slow. On the other hand linear classifiers such as linear SVMs, Random/Hough Forests and different variants of Boosting are used. Other approaches perform probabilistic reasoning or try to linearly approximate nonlinear kernels.

3.1 Bayesian Networks

In the methods we propose we have used a Bayesian network learning scheme. Bayesian networks or belief networks have been used successfully for pedestrian classification in [167], [168].

Belief networks [169] are used to model the statistical dependencies among the component features. They take the topological form of a directed acyclic graph where each link is directional and there are no loops. They allow efficient and effective representation of the joint probability distribution over a set of random variables. In our solutions each node or unit represents a visual feature. For a given instance, the probability of each class value can be predicted using conditional probability tables that are given by the relative frequencies of the associated combinations of attribute values in the training data.

In order to build a learning algorithm for Bayesian networks two components must be defined: a function for evaluating a given network based on the data and a method for searching through the space of possible networks [165]. The quality of a given network is measured by the probability of the data given the network. The probability that the network accords to each instance is computed by adding the logarithms of the probabilities over all instances.

In some configurations the nodes in the network are predetermined, one for each attribute including the class. So, our Bayesian network U is a pair $B = \langle G, \Theta \rangle$, where G is a directed acyclic graph whose vertexes correspond to attributes in the training set, and whose edges represent direct dependencies between attributes. We model the set of attributes by the random variables $A_1, \ldots, A_n$. The graph G encodes independence assumptions: each variable A_i is independent of its non-descendants given its parents in G. Θ represents the set of parameters that quantifies the network. It contains a parameter $\theta_{a_i|\Pi_{a_i}} = P_B(a_i|\Pi_{a_i})$ for each possible value a_i of A_i, and Π_{a_i}, of Π_{A_i}, where Π_{A_i} denotes the set of parents of A_i in G. The Bayesian network B defines a unique joint probability distribution over U given by:

$$P_B(A_1, \ldots, A_n) = \prod_{i=1}^{n} P_B(A_i|\Pi_{A_i}) = \prod_{i=1}^{n} \theta_{A_i|\Pi_{A_i}} \tag{3.1}$$

So, being given a set of attributes $a_1, \ldots, a_n$ and an attribute describing the class, c, the classifier based on B returns the label c that maximizes the posterior probability $P_B(c|a_1, \ldots, a_n)$.

3.2 Boosting

Boosting [170] is based on the observation that finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule. In order to apply the boosting approach, one should start with a method or algorithm for finding the rough rules of thumb. The boosting algorithm calls this "weak" or "base" learning algorithm repeatedly, each time feeding it a different subset of the training examples or, to be more precise, a different distribution or weighting over the training examples. Each time it is called, the base learning algorithm generates a new weak prediction rule, and after many rounds, the boosting algorithm must combine these weak rules into a single prediction rule that, hopefully, will be much more accurate than any one of the weak rules.

Adaptive Boosting was introduced as a practical algorithm of the boosting theory. I will exemplify it on a binary classification problem. The task of the binary classification is to find a rule, which, given a set of patterns, assigns an object to one of the two classes.

Let X be the input space which contains the objects and denote the set of possible classes by Y. In the binary classification case, $Y = \{-1, +1\}$). The task of learning can be summarized as follows: estimate a function $f : X \to Y$, using input, output training data pairs generated independently at random from an unknown probability distribution $P(x, y)$, $(x_1, y_1), \ldots (x_n, y_n) \in R^d \times \{-1, +1\}$ such that f will correctly predict unseen examples (x,y). The label assigned to an input x is $y = f(x)$.

The performance of the classifier is assessed by:

$$L(f) = \int \lambda(f(x), y) dP(x, y) \tag{3.2}$$

where λ is a chosen loss function. The risk $L(f)$ is often called the generalization error.

For binary classification, the loss function used is:

$$\lambda(f(x), y) = I(yf(x) \leq 0) \tag{3.3}$$

where $I(E) = 1$ if the event E occurs and 0 otherwise.

In other words:

$$\lambda(f(x_i), y_i) = \begin{cases} 1, & \text{if } x_i \text{ is misclassified} \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

Since the probability distribution $P(x, y)$ is unknown, this risk $L(f)$ cannot be directly minimized. So we have to estimate a function as close as possible from $f_{optimal}$ based on the available information, i.e. the training examples and the properties of the function class F from which f is chosen. One classical solution is to approximate the generalization error by the empirical risk defined as follows:

$$\hat{L}(f) = \frac{1}{N} \sum_{n=1}^{N} \lambda(f(x_n), y_n) \tag{3.5}$$

This is the case if the examples are uniformly distributed. If the training set is large enough, we expect that:

$$\lim_{N \to \infty} \hat{L}(f) = L(f) \tag{3.6}$$

However, one stronger condition is required to validate formula 3.6: the risk error $\hat{L}(f)$ has to converge uniformly over the class of functions F to $L(f)$

While this condition is possible for large size training sets, for small samples size large deviations are possible and over-fitting might occur. If it is the case, the generalization cannot be obtained by minimizing the training error $\hat{L}(f)$.

The AdaBoost algorithm Let $h_1, h_2, \ldots h_T$ be a set of simple hypothesis and consider the composite ensemble of hypothesis:

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{3.7}$$

There are many approaches for selecting the coefficients α_t and the base hypothesis h_t in equation 3.7. We will present next the Adaptive Boosting algorithm.

It is called Adaptive in the sense that examples that are misclassified get higher weights in the next iteration, for instance the examples near the decision boundary are harder to classify and therefor get high weights in the input set after the first iterations.

1. **Input**: S=$(x_1, y_1), \ldots, (x_n, y_n)$, Number of iterations T
2. **Initialize:** $d_n^{(1)} = \frac{1}{N}$ for all $n = 1, \ldots, N$
3. for $t = 1, \ldots, T$ **do**
   (a) Train classifier with respect to the weighted sample set $S, d^{(t)}$ and obtain hypothesis $h_t : x \to \{-1, +1\}$, i.e. $h_t = L(S, d^{(t)})$
   (b) Calculate the weighted training error e_t of h_t:

$$e_t = \sum_{n=1}^{N} d_n^{(t)} I(y_m \neq h_t(x_n)) \tag{3.8}$$

(c) Set:

$$\alpha_t = \frac{1}{2} log \frac{1 - e_t}{e_t} \tag{3.9}$$

(d) Update the weights:

$$d_n^{(t+1)} = \frac{d_n^{(t)} e^{-\alpha_t y_n h_t(x_n)}}{Z_t} \tag{3.10}$$

where Z_t is a normalization constant such that $\sum_{n=1}^{N} d_n^{(t+1)} = 1$

4. **Break if:** $e_t = 0$ or $e_t \leq \frac{1}{2}$ and set $T = t - 1$

5. **Output:** $f_T(x) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T} \alpha_t} h_t(x)$

where x is the pattern to be classified, y is its target label and $f(x)$ is the decision function.

A weight $d^{(t)} = (d_1^{(t)}, \ldots, d_N^{(t)})$ is assigned to the data at step t and a weak learner h_t is constructed based on $d^{(t)}$. This weight is updated at each iteration. The weight is increased for the examples which have been misclassified in the last iteration.

The weights are initialized uniformly: $d_n^{(1)} = 1/N$.

To estimate if an example is correctly or badly classified, the weak learner produces a weighted empirical error defined by:

$$\epsilon_t(h_t, d^{(t)}) = \sum_{n=1}^{N} d_n^{(t)} I(y_n \neq h_t(x)_n) \tag{3.11}$$

Once the algorithm has selected the best hypothesis h_t , its weight $\alpha_t = \frac{1}{2} log \frac{1-\epsilon_t}{\epsilon_t}$ is computed such that it minimizes a loss function. One of the possible loss function considered in AdaBoost is:

$$G^{AB}(\alpha) = \sum_{n=1}^{N} e^{-y_n(\alpha h_t(x_n) + f_{t-1}(x_n))} \tag{3.12}$$

where f_{t-1} is the combined hypothesis of the previous iteration given by:

$$f_{t-1}(x_n) = \sum_{r=1}^{t-1} \alpha_r h_r(x_n) \tag{3.13}$$

The iteration loop is stopped if the empirical error ϵ_t equals 0 or $\epsilon_t \geq \frac{1}{2}$. If $\epsilon_t = 0$, the classification is optimal at this stage and so it is not necessary to add other classifiers. If $\epsilon_t \geq \frac{1}{2}$, the classifiers do not respect the weak condition anymore. They are not better than random selection so AdaBoost cannot be efficient.

Finally, all the weak hypotheses selected at each stage h_t are linearly combined as follows:

$$f_T(x) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T} \alpha_r} h_t(x) \tag{3.14}$$

The final classification is a simple threshold which determines if an example x_i is classified as positive or negative.

Other similar algorithms such as LogitBoost or Arcing algorithms use different loss functions.

Weak learners

The adaptive boosting algorithm calls the "weak" or "base" learning algorithm repeatedly, each time feeding it a different subset of training examples (actually, a different distribution or weighting over the training samples). Each time it is called, the base learning algorithm generates a new weak prediction rule, and after many rounds the boosting algorithm must combine these weak rules into a single prediction rule, that hopefully, will be more accurate than any one of the weak rules.

For weak learners one can use:

- Bayes decision rule [171]
- Decision trees having a root node and two children and using as splitting criteria:
  - Miss-classification error;
  - Entropy;
  - Gini index;
- Other decision rules.

We have used the training data for building a decision tree with one level a root and two children as depicted in Figure 3.1.

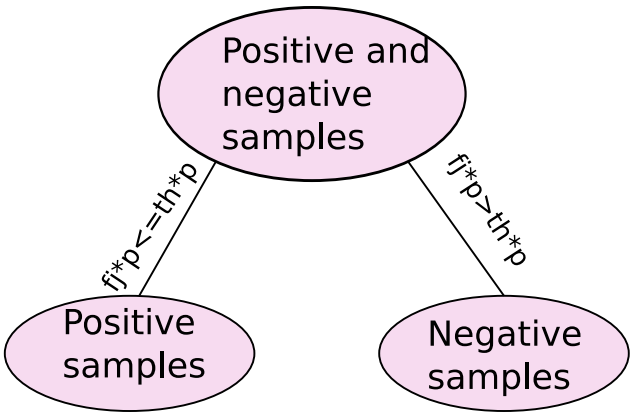


Figure 3.1: Decision tree with root and two children

Much of the work in designing trees focuses on deciding which property test or query should be performed at each node. With non-numeric data, there is no geometrical interpretation of how the test at a node splits the data. However, for numerical data, there is a simple way to visualize the decision boundaries that are produced by decision trees. For example, suppose that the test at each node has the form "is $x_i \leq x_{is}$?" This leads to hyperplane decision boundaries that are perpendicular to the coordinate axes.

The fundamental principle underlying tree creation is that of simplicity: we prefer decisions that lead to a simple, compact tree with few nodes. This is a version of Occam's razor, that the simplest model that explains data is the one to be preferred [169]. To this

end, we seek a property test T at each node N that makes the purity data reaching the immediate descendant nodes as "pure" as possible. In formalizing this notion, it turns out to be more convenient to define the impurity, rather than the purity of a node. Several different mathematical measures of impurity have been proposed, all of which have basically the same behavior [169]. Let $i(N)$ denote the impurity of a node N. In all cases, we want $i(N)$ to be 0 if all of the patterns that reach the node bear the same category label, and to be large if the categories are equally represented.

The most popular measure is the *entropy impurity* (or occasionally *information impurity*):

$$i(N) = -\sum_j P(\omega_j) \log_2 P(\omega_j) \tag{3.15}$$

where $P(\omega_j)$ is the fraction of patterns at node N that are in category ω_j .By the well-known properties of entropy, if all the patterns are of the same category, the impurity is 0; otherwise it is positive, with the greatest value occurring when the different classes are equally likely.

Another definition of impurity is particularly useful in the two-category case. Given the desire to have zero impurity when the node represents only patterns of a single category, the simplest polynomial form is:

$$i(N) = P(\omega_1)P(\omega_2) \tag{3.16}$$

This can be interpreted as a *variance impurity* since under reasonable assumptions it is related to the variance of a distribution associated with the two categories. A generalization of the variance impurity, applicable to two or more categories, is the *Gini impurity*:

$$i(N) = \sum_{i \neq j} P(\omega_i)P(\omega_j) = 1 - \sum_j P^2(\omega_j) \tag{3.17}$$

This is just the expected error rate at node N if the category label is selected randomly from the class distribution present at N. This criterion is more strongly peaked at equal probabilities than is the entropy impurity (see Figure 3.2).

The *miss-classification impurity* can be written as:

$$i(N) = 1 - \max_j P(\omega_j) \tag{3.18}$$

It measures the minimum probability that a training pattern would be misclassified at N. Of the impurity measures typically considered, this measure is the most strongly peaked at equal probabilities. It has a discontinuous derivative, though, and this can present problems when searching for an optimal decision over a continuous parameter space. Figure 3.2 shows these impurity functions for a two-category case, as a function of the probability of one of the categories.

Given a partial tree down to node N, what value s should be chosen for the property test T? An obvious heuristic is to choose the test that decreases the impurity as much as possible. The drop in impurity is defined by:

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R) \tag{3.19}$$

where N_L and N_R are the left and right descendant nodes, $i(N_L)$ and $i(N_R)$ their impurities, and P_L is the fraction of patterns at node N that will go to N_L when property test T is

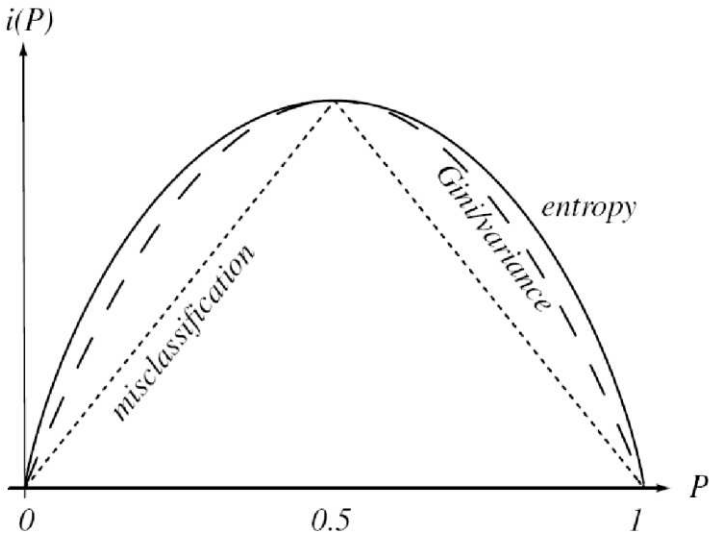


Figure 3.2: For the two-category case, the impurity functions peak at equal class frequencies and the variance and the Gini impurity functions are identical. To facilitate comparisons, the entropy, variance, Gini and miss-classification impurities have been adjusted in scale and offset to facilitate comparison; such scale and offset does not directly affect learning or classification. [169]

used. Then the "best" test value s is the choice for T that maximizes $\Delta i(T)$. If the entropy impurity is used, then the impurity reduction corresponds to an information gain provided by the query. Since each query in a binary tree is a single "yes/no" one, the reduction in entropy impurity due to a split at a node cannot be greater than one bit

For each weak classifier, the values are sorted. The optimal threshold with respect to the weighted error can be computed in a single pass over this sorted list. For each possible threshold, four sums are evaluated:

- Total sum of positive image weights T^+;
- Total sum of negative image weights T^-;
- Sum of positive weights below the current threshold S^+;
- Sum of negative weights below the current threshold S^-;

The weighted error for a threshold is:

$$e = \min\left(S^+ + (T^- - S^-), S^- + (T^+ - S^+)\right) \tag{3.20}$$

or the minimum of the error of labeling all images with a feature value below the current threshold negative and labeling the images with a feature value above positive (parity, p_j equals to -1 in the expression of the weak classifier) versus the error of the converse (parity p_j equals +1 in the expression of the weak classifier). The error of the weak learner is computed by summing the weights of the images (positive and negative) which are incorrectly classified by it.

3.3 Multiple Layer Perceptron

Multiple layer perceptron (MLP) also known as feed forward neural networks represents a series of logistic regression models stacked on top of each other, with the final layer being either another logistic regression or a linear regression model, depending on whether we are solving a classification or regression problem [172].

Multilayer perceptrons consist of several layers of units (perceptrons) that are connected and each connection has a weight. The first layer is connected to the input representing the attributes in the data. Next it has the hidden layer that has no direct connection to the environment and performs the actual processing. A multilayer perceptron has the same expressive power as, say, a decision tree. Two aspects are considered when dealing with multiple layer perceptrons: learning the structure of the network and learning the connection weights. Back-propagation can be used in order to determine the weights given a fixed network structure. Often a single hidden layer is all that is necessary, and an appropriate number of units for that layer is determined by maximizing the estimated accuracy.

Mathematically speaking, if the MLP has two layers and a regression problem is considered the model has the form [172]:

$$p(y|x,) = \mathcal{N}(y|w^T z(x), \sigma^2) \tag{3.21}$$

$$z(x) = g(Vx) = [g(v_1^T x), \ldots, g(v_H^T x)] \tag{3.22}$$

"where g is a non-linear activation or transfer function (commonly the logistic function), $z(x) = \Phi(x, V)$ is called the hidden layer (a deterministic function of the input), H is the number of hidden units, V is the weight matrix from the inputs to the hidden nodes, and w is the weight vector from the hidden nodes to the output. It is important that g be nonlinear, otherwise the whole model collapses into a large linear regression model of the form $y = w^T(Vx)$. One can show that an MLP is a universal approximator, meaning it can model any suitably smooth function, given enough hidden units, to any desired level of accuracy

To handle binary classification, the output is passed through a sigmoid, as in a GLM":

$$p(y|x, \theta) = Ber(y|sigm(w^T z(x))) \tag{3.23}$$

3.4 Support Vector Machine

Support vector machines use linear models to derive nonlinear class boundaries. They work by transforming the input space into a new space. With a nonlinear mapping, a straight line in the new space does not look straight in the original instance space. A linear model constructed in the new space can represent a nonlinear decision boundary in the original space. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In our experiments we use the implementation of SVM provided by [166] that implements the sequential minimal optimization (SMO) algorithm for training a support vector classifier, using polynomial or Gaussian kernels.

Sequential minimal optimization is an algorithm for solving the quadratic programming problem that arises during the training of support vector machines.

3.5 Aggregated Channel Features

Recently [86] proposed a model for multiple resolution image feature approximation instead of actual feature computation. The model can be applied to a generic object detector and it was applied to the pedestrian detection task in the Aggregated Channel Features (ACF) framework [86], [87].

The idea of ACF is that the computational bottleneck of many pedestrian detectors raised by feature computation at every scale over a finely sampled image pyramid can be removed by feature approximation via extrapolation from nearby scales. For a broad family of features this process does not reduce the performance of the detection. [86] find that features computed at octave-spaced scale intervals are sufficient to approximate features on a finely-sampled pyramid. Extrapolation is inexpensive as compared to direct feature computation. As a result, the proposed approximation yields considerable speedups with negligible loss in detection accuracy.

A feature pyramid represents a multiple scale representation of an image I. For each scale s channels $C_s = \Omega(I_s)$ are computed. These channels can be gradients, histograms and other textural features. The basic steps of the ACF method are:

- Given an input image I, compute several channels $C = \Omega(I)$, sum every block of 4×4 pixels in C, and smooth the resulting lower resolution channels.
- Instead of computing the features for each scale the ACF method computes I_s and $C_s = \Omega(I_s)$ for only a sparse set of s (once per octave).
- At intermediate scales C_s is computed by approximation.
- Scan the pyramid with a sliding window of dimension 32×64.

The important definitions described in detail by [86] for feature approximation are:

- Ω any low-level shift invariant function that takes an image I and creates a new channel image $C = \Omega(I)$
- C is a per-pixel feature map such that output pixels in C are computed from corresponding patches of input pixels in I (thus preserving overall image layout).
- C may be down-sampled relative to I and may contain multiple layers k.
- Define a feature $f_\Omega(I) = \sum_{ijk} w_{ijk} C(i, j, k)$
  - This forms includes gradient histograms, linear filters, color statistics etc.
- Given two scales s_1, s_2 Dollar. et. al. show that:
  - $f_\Omega(I_{s1})/f_\Omega(I_{s2}) = (s_1/s_2)^{-\lambda_\Omega} + \epsilon$
  - λ_Ω is determined empirically
  - ϵ – deviation from the power law for a given image

The method uses 27 scales. The four mentioned features are computed for the predefined scales equal to 1, 0.5, 0.25, 0.125, while the values of the features for 7 intermediate scales are approximated from neighboring scales. Exact feature scaling computation and approximation methodology is detailed in [86].

3.6 Random Forests

A random forest is composed of binary decision trees [143]. The training of a random forest consists in assigning each leaf node in each tree a binary test that is applicable to any data sample. Depending on the result of the test a sample can follow the path of one of the two children of a given non-leaf node. The training involves tree construction and also the assignement to each leaf node the class distribution. In order to classify a test sample its features are passed down all the trees of the forest and the classification score is computed by averaging the distributions recorded at the reached leaf nodes.

Contextual information is integrated in random forests and applied for the task of semantic labeling and pedestrian detection by [144]. They augment the random forest structure with label information and provide a novel split function evaluation criterion that makes use of the joint distribution observed in the structured label space.

3.7 Hough Forests

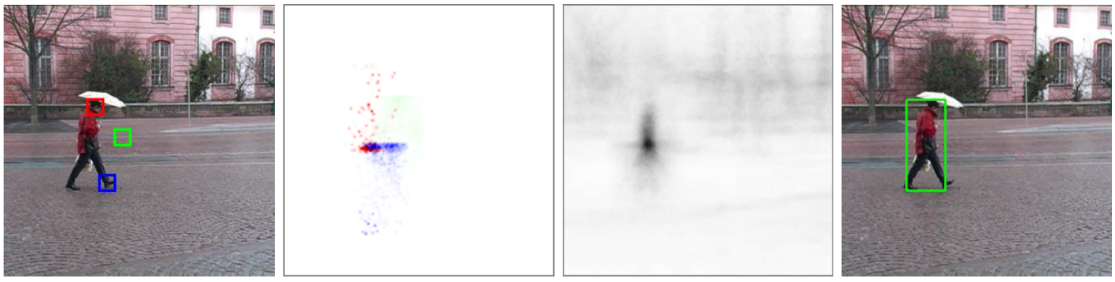
The Hough forests are in many aspects similar to random forests but as described by [19] they have some specific properties. As stated by [19] the set of leaf nodes of each tree in a Hough forest is a discriminative codebook. Each leaf node makes a probabilistic decision on a patch and tells if it corresponds to a part of the object or to the background. Also, a leaf node provides a vote about the centroid position of the whole object with respect to the patch center. The trees in a Hough forest are built such that the leaves produce probabilistic votes with small uncertainty. Each tree is constructed considering a collection of patches from the training samples. The method is supervised in the sense that at the construction of the forest it has to be known whether a patch comes from a background or an object, in the latter case, which part of the object does it come from

In the generalized Hough transform the detections of individual object parts cast probabilistic votes for possible locations of the centroid of the whole object. The final detection hypotheses correspond to the maxima of the Hough image that accumulates the votes from all parts.

[19] demonstrate that Hough forests improve the results of the Hough-transform object detection. A sample detection is shown in Figure 3.3. Using random forests [19] learn a direct mapping between the appearance of an image patch and its Hough vote. For detecting pedestrians the learned model is applied to the patches in the test image and the resulting votes are accumulated in the Hough image where the maxima are found.

The Hough-transform is also used by [46] that learn the class specific implicit shape model (ISM) that represents a codebook of interest point descriptors.

Two improvements on the Hough Forest object detection framework are proposed by [42]: they infer precise probabilistic segmentations for the object hypotheses and use those



(a) – Original image with three sample patches emphasized | *(b)* – Votes assigned to these patches by the Hough forest | *(c)* – Hough image aggregating votes from all patches | *(d)* – The detection hypothesis corresponding to the peak in *(c)*

Figure 3.3: The Hough transform used for pedestrian detection by the work of [19]

segmentations for improving the final hypothesis selection. They also develop an efficient cascaded voting scheme that significantly reduces the effort of the Hough voting stage without loss in accuracy.

A probabilistic framework related to Hough transform is proposed by[20] and extended by [66]. They formulate the object detection task as a problem of finding the finite subset of the Hough space that corresponds to objects that are present in the image. For the pedestrian detection task the votes are obtained in a probabilistic manner using Hough forests. Figure

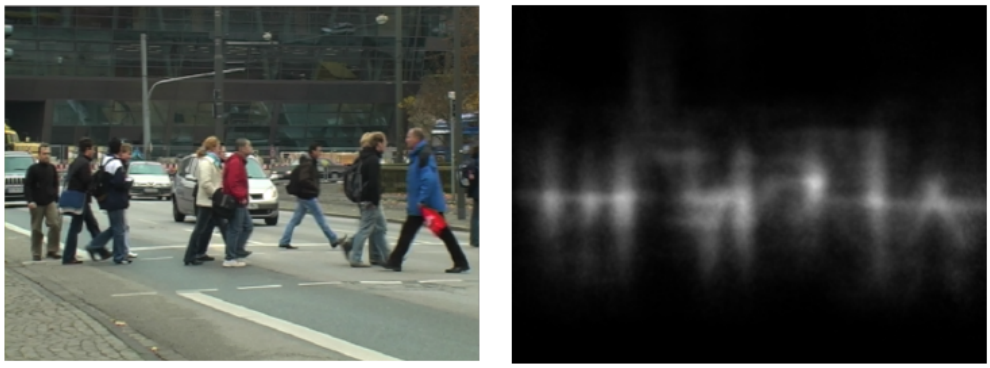


Figure 3.4: Pedestrian detection with Hough forests used by [20]

3.4 shows their approach.

3.8 Neural Networks

A multiple layer neural network trained using adaptive local receptive field features (of 5×5 features) is used by [99], [51]. A deep learning model computed on the deformable part pedestrian detector is employed by [106]. In their model each hidden variable of the network represents the visibility of a part. The object detection problem in general is regarded as a multi-layered model by [65]: with segmentation as first layer and a segment classification as second layer. They provide bounding boxes as object candidates using a single Deep Neural Network. Each bounding box has an associated confidence score that represents how likely

is the box to contain an interest object. This is the localization network. Then a classifier network (also a Deep Neural Network) returns the final classification score.

A Convolutional neural network model is employed by [146]. Their model uses multi-stage features, connections that skip layers in order to integrate global shape information with local distinctive motif information and an unsupervised method based on convolutional sparse coding to pre-train the filters at each stage.

A Switch Deep Network (SDN) is proposed by [76]. The SDN automatically learns hierarchical feature representations that correspond to body parts and the whole body. The SDN is composed of a convolutional layer that is initialized (or pre-trained) using a set of Gabor filters. It learns to extract low- and mid-level features. On the next level the SDN comprises four switchable layers (modeled by a Switchable Restricted Boltzmann Machine – SRBM) and they model high-level mixture representations and salience maps of the whole body and of three body parts: head-shoulder, upper-body, and lower-body. The last layer performs logistic regression and predicts labels.

Chapter 4

Collections of Annotated Image and Evaluation Metrics

4.1 Color Image Collections

A summary of available pedestrian datasets dealing with monocular intensity sequences for pedestrian detection is described in [33] and [39]. In order to asses the accuracy of the proposed algorithms we have used several benchmark pedestrian detection datasets.

Daimler

The Daimler Stereo Pedestrian Detection Benchmark Dataset [1] introduced in [173] is an extension of the monocular dataset described in [33]. For our experiments we have used the monocular images. The annotations contain fully-visible pedestrians, pedestrian groups, partially occluded pedestrians, bicyclists and motorcyclists. The images are grayscale and have a size of 640×480 pixels. They provide a nice summary of the available pedestrian datasets recorded from a moving platform in an urban environment.

ETHZ

The ETHZ dataset [2] is described in [174]. The annotations contain few very small pedestrians having the size smaller than 60 pixels. The dataset contains color images of dimension 640×480 pixels. We use the setup 1 (chariot Mk I) in our evaluations.

Caltech

The Caltech Pedestrian Dataset [3] was introduced by [175]. It is a monocular color dataset and it is two orders of magnitude larger than existing datasets. The images have been recorded

[1] http://www.gavrila.net/Datasets/Daimler_Pedestrian_Benchmark_D/Daimler_Stereo_Ped__Detection_/daimler_stereo_ped__detection_.html

[2] http://www.vision.ee.ethz.ch/~aess/dataset/

[3] http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

from a moving vehicle and contain also low resolution images and frequently occluded people. The annotations refer to single pedestrians, occluded pedestrians and groups of pedestrians.

TUD-Brussels

The dataset was introduced by [114]. The data is divided in three sequences: (1) 'tud-pedestrians' that contains 250 images with 311 side-view fully visible pedestrians with significant variation in clothing and articulation; (2) 'TUD Crossing Sequence' that contains 201 images with 1008 annotated pedestrians. Most of the pedestrians are in side-view and many are partially occluded for the whole sequence. (3) 'TUD Campus Sequence' – contains 71 image with 303 annotated pedestrians. All pedestrians are in side-view and many are partially occluding each other.

NICTA

The dataset [4] has been introduced by [176] that analyze different characteristics of a a dataset such as image size, aspect ration, geometric variance and the relative scale of positive class instances within the training window. The goal is to determine what characteristics are desirable for a pedestrian dataset. The dataset contains positive and negative images at different resolutions (in our experiments we have used 16×40 and 64×80 image dimensions). The final dataset contains 25551 unique pedestrians, allowing for a dataset of over 50 000 images obtained using mirroring.

INRIA

Person Dataset [5] collected as part of research work on detection of upright people in images and video [116]. The authors of the database describe its content as follows: the database contains two sets of images:

1. positives: normalized positive training or test images centered on the person with their left-right reflections

2. negatives: containing original negative training or test images.

These two sets are further divided into training set containing images of dimension 96×160 pixels (a margin of 16 pixels around each side), and test set containing images of dimension 70×134 pixels (a margin of 3 pixels around each side). "This has been done to avoid boundary conditions (thus to avoid any particular bias in the classifier)". It is suggested to use the centered 64×128 pixels window for the pedestrian detection task and this is the dimension we have used.

[4] `http://www.nicta.com.au/research/projects/AutoMap/computer_vision_datasets`
[5] http://pascal.inrialpes.fr/data/human/

MIT

The training database [177] of people was generated from color images and video sequences taken in Boston and Cambridge in a variety of seasons using several different digital cameras and video recorders. The data was initially used in [178]. The pose of the people in this dataset is limited to frontal and rear views.

Each image was extracted from raw data and was scaled to the size 64x128 and aligned so that the person's body was in the center of the image; the height of these people is such that the distance from the shoulders to the feet is approximately 80 pixels.

4.2 Infrared Image Collections

Dataset Description

Before describing the infrared image data a short overview of the infrared domain will be provided. All matter (gases, planets, etc) emits some amount of electromagnetic radiation across a range of energies (or wavelengths). Infrared refers to the part of the electromagnetic spectrum where biological life-forms emit the most light, at wavelengths slightly longer than what we perceive as the color red. The human beings are not able to see infrared, but they can sense it through what is commonly called heat. Physical touch is the most direct way of observing it.

Objects generally emit infrared radiation across a spectrum of wavelengths, but sometimes only a limited region of the spectrum is of interest because sensors usually collect radiation only within a specific bandwidth. Therefore, the infrared band is often subdivided into smaller sections. The commonly used sub-division scheme comprises near-infrared ($0.75 - 1.4\mu m$), short-wavelength infrared ($1.4 - 3\mu m$), mid-wavelength infrared ($3 - 8\mu m$), long-wavelength infrared ($815\mu m$), far infrared ($15 - 1,000\mu m$). The long wavelength infrared corresponds to the "thermal imaging" region, in which sensors can obtain a completely passive picture of the outside world based on thermal emissions only and requiring no external light or thermal source such as the sun, moon or infrared illuminator. Forward-looking infrared (FLIR) systems use this area of the spectrum. This region is also called the "thermal infrared".

LWIR Sensor Information

The images have been collected with a PathFindIR camera that is 'designed primarily for driving vision enhancement (DVE) applications. PathFindIR is a hermetically sealed system, rated to IP-67, with an integrated, automatic window heater. using a 12VDC input power source, standard NTSC or PAL video is output for compatibility with most monitors or displays.' [6]. The characteristics of the camera are:

- $320x240$ uncooled VOx microbolometer, 38 micron pitch
- 8-14 micron LWIR

[6] http://www.flir.com/cvs/cores/view/?id=51221&collectionid=551&col=51218

- Hermetically sealed, IP67 enclosure
- Fixed 19mm (36 degree HFOV) high impact optic
- Built in automatic lens heater
- Nominal 12VDC power input
- Standard NTSC, or PAL video output
- Compact size: $71.4mm \times 57.4mm \times 56.1mm(2.8'' \times 2.3'' \times 2.2'')$ – Standard System cable adds 3" to bottom.
- Weight 360 grams.
- Persistent FLIR logo on image

Infrared Dataset

We have used our own annotated dataset that was formed from the several traffic sequences taken in autumn and in winter. The dataset contains 1924 pedestrian images having the heights varying from 12 pixels to 150 pixels and aspect ratios (width/height) in the range $(0.3, 0.4, 0.5, 0.6)$.

Besides fully visible pedestrians we have annotated occluded pedestrians, groups of pedestrians and occluded groups. Those annotations were used for assessing the performance of the region of interest generator. For classification we have only used the annotations of fully visible pedestrians without any occlusion handling. For testing we have employed another set of sequences that contains frames captured both at night and at daytime.

4.3 Evaluation Metrics

A proper evaluation methodology is crucial in the assessment of the good behavior of a pedestrian detector. We follow a standard evaluation protocol for the classifiers we propose. As stated by [36] this protocol quantifies and ranks detectors performance in a realistic, unbiased and informative manner.

Per window evaluation using confusion matrices

A protocol for evaluating detectors based on binary classifiers is to measure their per-window (PW) performance on cropped positive and negative image windows [36]. Per window evaluation is commonly used to compare classifiers – as we did in chapter 5.

The per window evaluation is captured in a tabular form in the confusion matrix that allows the visualization of the performance. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. Table 4.1 shows the confusion matrix for a two class classification problem dealing with pedestrian and non-pedestrian categories.

	Ground truth pedestrian	Ground truth non-pedestrian
Predicted pedestrian	True positive	False positive
Predicted non-pedestrian	False negative	True negative

Table 4.1: Confusion matrix

The measures that are derived from the confusion matrix are:

$$\text{True positive rate} = \frac{\sum \text{True positive}}{\sum \text{Ground truth positive}} \tag{4.1}$$

The true positive rate represents the amount of correctly detected pedestrians from the total number of ground truth pedestrians present in the evaluation.

$$\text{False negative rate} = \frac{\sum \text{False negative}}{\sum \text{Ground truth positive}} \tag{4.2}$$

The false negative rate represents the amount of ground truth pedestrians that are classified as negatives from the total number of ground truth pedestrians present in the evaluation.

Log average miss rate

The log average miss rate was proposed by [179] and used extensively by [36]. It is a metric used to report the detection performance on a full image as miss rate versus false positives per image (FPPI).

As stated by [36] "a detection system needs to take an image and return a BB and a score or confidence for each detection. The system should perform multiscale detection and any necessary non-maximal suppression (NMS) for merging nearby detections. Evaluation is performed on the final output: the list of detected BBs. A detected BB (BB_{dt}) and a ground truth BB (BB_{gt}) form a potential match if they overlap sufficiently. Specifically, the PASCAL measure is employed, which states that their area of overlap must exceed 50%:

$$a_0 = \frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} > 0.5 \tag{4.3}$$

For larger of the threshold the performance degrades."

In our experiments we use the evaluation protocol provided by [36], [87]. In describing their evaluation protocol they state that "each $BB_{dt}$and BB_{gt} may be matched at most once. They resolve any assignment ambiguity by performing the matching greedily. Detections with highest confidence are matched first; if a detected BB matches multiple ground truth BBs, the match with highest overlap is used (ties are broken arbitrarily). Unmatched BB_{dt} count as false positives and unmatched BB_{gt} as false negatives."

The log average miss rate (lamr) is defined by [179] as "the average miss-rate sampled from the lowest false positive rate to a false positive rate of 1 FPPI.

The log-average miss rate is used in [36], [87] to "summarize detector performance, computed by averaging miss rate at nine FPPI rates evenly spaced in log-space in the range 10^{-2} to 10^{0} (for curves that end before reaching a given FPPI rate, the minimum miss rate achieved is used).

The log-average miss rate is similar to the performance at 10^{-1} FPPI (meaning a negative at every 10 frames) but in general gives a more stable and informative assessment of performance. ”

Chapter 5

Monocular Color Pedestrian Detection

5.1 Survey of Current Approaches

Pedestrian detection is presented in the context of collision avoidance systems by the survey in [32]. They make a comparison of different sensor modalities (like visible cameras, near and thermal infrared cameras, RADAR, LASER scanner) for pedestrian detection. They analyze the approaches for pedestrian detection and tracking in the visible and infrared domain and the methods that combine several sensor cues. They also revise methods for behavior analysis and collision prediction.

An overview of pedestrian detection approaches related to monocular systems is provided by [33]. Their survey is two-fold: first they revise state of the art methods in pedestrian detection and secondly they describe an experimental study of the most popular approaches for pedestrian detection: cascade of AdaBoost classifier trained on wavelet features, linear SVM with HOG features, Neural Networks with Local Receptive Fields and combined shape-texture detection. The survey identifies and describes three main components of the pedestrian detection pipeline: Region of Interest selection, Classification comprising generative and discriminative models. Generative models regard the appearance of a pedestrian as a class conditional density function and the posterior probability for the pedestrian class can be inferred using a Bayesian approach. These models are based mainly on shape cues and some combine shape and texture. On the other hand discriminative models learn the parameters of a discriminant function between the pedestrian and non-pedestrian classes based on a set of training samples. The study identifies main features, classification architectures and multiple part representations.

An extensive literature review is done by [1]. They divide the problem of pedestrian detection into different processing steps, also identified in [34]: preprocessing, foreground segmentation, object classification, verification/refinement, tracking and application. These steps are analyzed in the framework of monocular and stereo visible sensors, near and thermal infrared sensors and in the perspective of multiple fused sensors. The preprocessing methods comprise exposure time, gain adjustments, camera calibration. The foreground segmentation extracts the regions of interest from the image. These regions are sent to the classification module. [1] shortly review methods that are applied to monocular images (comprising scan window based approaches, biologically inspired visual attention algorithms [2], symmetry based methods [35]), infrared images (including symmetry based approaches [35],

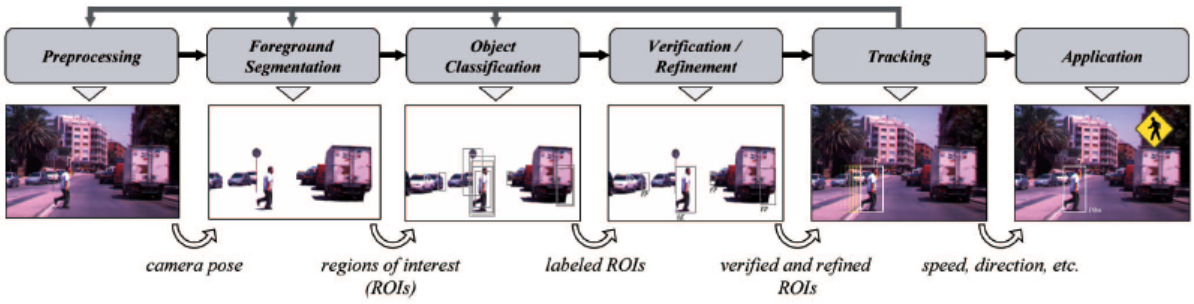


Figure 5.1: Processing steps in pedestrian detection methods surveyed by [1]

intensity thresholding, histogram projection combined with thresholding, hypermutation networks used for pixel classification), stereo based algorithms and motion based segmentation algorithms. In the step of object classification they identify two classes of methods: silhouette matching and appearance based algorithms that define a set of image features (descriptors) and a classifier is trained on those features. For the verification step [1] identify methods to filter out false positive detections and in the refinement step a fine segmentation of the pedestrian is performed. Tracking methods are also surveyed and their usage is nicely described: false detection removal over time, prediction of future pedestrian positions, inference about pedestrian trajectory and behavior. Discussion and review of benchmarking procedures are also presented.

Reference state of the art algorithms are described and evaluated by [36]. They also describe the Caltech dataset and study the statistics of size, position and occlusion patterns for pedestrians. A comparison between pedestrian detection datasets is made in terms of samples used for training and testing, pedestrian heights, and different properties like color images, evaluation methodology, temporal analysis. They also propose an elaborate evaluation methodology. For the reviewed algorithms they focus on sliding windows detection processes. They revise Haar based detectors using SVM [37] or AdaBoost [38], HOG based methods, shape based template matching methods, motion based methods and approaches that combine multiple feature representations: HOG, Haar, shapelets, edgelets, Local Binary Patterns, Integral Channel Features. They also revise the methods that improve the learning framework and methods that explore very large feature spaces. Finally they refer to part based detectors and pose specific detectors.

A recent survey [39] analyze the remarkable progress of the last decade by discussing the main ideas explored in 40+ detectors available in Caltech pedestrian detection benchmark. Based on the classification technology, [39] identify three classes of approaches: (1) deformable part models (DPM) variants , (2) Deep networks and (3) decision forests. In the pattern recognition chapter we will revise each.

We conclude that the above mentioned studies identify a modular approach in the development of a pedestrian detection scheme containing the following components:

(a) Generation of possible pedestrian location hypotheses: regions of interest – ROI or foreground segmentation;

(b) Definition of the pedestrian data model: full body, part based representations, components;

(c) Choice of the data representation model: contours, gradient histograms, wavelets;

(d) Pattern recognition or classification: matching, SVM, boosted classifiers;

(e) Refinement in which multiple scale overlapping detections are analyzed and the best detection is kept.

(f) Tracking of detected pedestrians.

(g) Benchmarking procedure and evaluation methodology.

The pedestrian hypothesis generation depends on the type of system used (stereo, monocular, infrared). Pedestrian hypothesis refers to the generation of possible locations for a pedestrian. Therefore we will name this process either pedestrian localization, pedestrian hypothesis generation or region of interest selection. We will present existing approaches for three types of sensorial systems: monocular, stereo and infrared.

5.2 Pedestrian Representation Model

The classification is done using a set of features extracted on different parts of the pedestrian body, on the whole body or it combines multiple parts or components and multiple views or poses of the pedestrian.

There are an extremely large number of methods that use one of these approaches and it is probably impossible to mention them all. Yet our study tries to capture the evolution of the pedestrian data representation model from simple whole body representation, to rigid multiple part description up to hierarchical deformable models and multiple view multiple part combined models.

When dealing with a pedestrian data representation model an important aspect is given by the model dimension. There are approaches that consider a single model dimension when training a classifier and there are methods that create multiple size models and for each size they train a classifier.

At a first representation level we have divided the approaches into monolithic ones that consider the pedestrian data as a whole and into part based model that regard the pedestrian as a combination of parts and the whole body. For each of the two division single scale or multiple-scale methods have been developed. These representations try to capture the high variance of the pedestrian appearance and some underline the multiple views or poses that pedestrians may have.

5.2.1 Monolithic Models

The pedestrian model is regarded as an indivisible and uniform structure. The variations of this model are given by the detector size and the pedestrian views enclosed. The multi-resolution monolithic models are formalized by [60]. They identify fixed resolution models

that we divide into no-view and multiple view single scale models, multiple fixed-resolution models that we categorize in no-view and multiple view fixed resolution models that consider separate model for each interest detection size.

No View Single Scale Model

The simplest and most popular representation is given by a single size model that is not view centric, that is no special views of pedestrians are considered. Basically all views or attitudes of a pedestrian are comprised without any separation between them. This approach has been used by [98] that propose a method for discriminating between pedestrians, bicyclists and motor-cyclists is described in [98]. For pedestrian detection they employ HOG features and Co-occurrence semantic HOG features that are input to a Fisher's linear discriminant function. MRF tracking is also used.

Multiple View Single Scale

The multiple view model considers specific views of pedestrians and trains focused classifiers for these views. This approach is encountered in the work of [99], [42].

In the work of [42] the multi-view pedestrian model comprises side-viewed front, rear, or diagonal pedestrians. They model the views by the rotation angle with respect to a side viewed pedestrian that has a rotation of 0 degrees, 45, 90/270 and 135 degrees.

Four view-related classifiers or mixture of experts are proposed by [23]. The considered views or poses correspond to front, left, back and right views of pedestrians.

No View Multiple Scale Model

A popular approach that speeds up the pedestrian detection process is employed by [30] that consider as canonical classifier the Integral Channel Feature based classifier of [15] and train it for a predefine set of scales (a scale for each octave). They obtain a remarkable speed-up improvement with negligible loss in accuracy because they perform object detection without image resizing.

Detectors encompass different scales of the single view model: [4].

A scan window based approach for pedestrian detection is presented in [5]. Their classification comprises a Discrete AdaBoost classifier with Weighted Fisher Linear Discriminant as weak learner trained on several macrofeature layouts. Macrofeatures encode a set of low-level features in a neighborhood. [5] employ three shape layouts (line, triangle and pyramid) as shown in Figure 5.2. These layouts are composed of HOG feature blocks closely located to each other in a multi-scale feature pyramid. They train a classifier on INRIA dataset and evaluated it on challenging data sets like: INRIA, ETH, TUD-Brussels, Caltech, Daimler. The proposed method is evaluated in terms of detection performance (relationship between detection rate and the number of false positives per image) and it terms of localization performance (mean of PASCAL VOC overlap ratios given false positives per image). They measure how is the alignment of the detected bounding boxes with respect to the ground truth annotation.

The algorithm is implemented using CUDA and it runs at 9.43 fps.

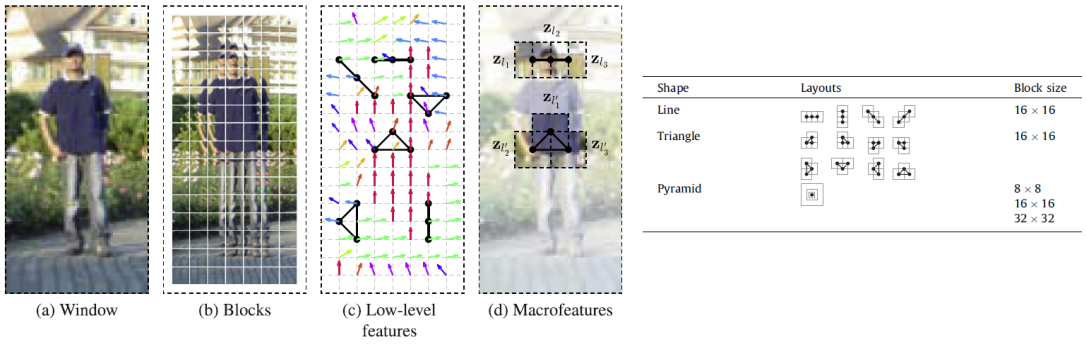


Figure 5.2: Macrofeatures selection and layout types employed by [5]

Multiple View Multiple Scale Model

Detectors are trained for several scales and for multiple views of the same object.

In the work of [68] the pedestrian views on which classifiers are trained are divided into four categories: back, front, left, right. The scale is represented by dividing the target distance up to 60m into three ranges: near, middle and far.

5.2.2 Part Based Models

These representations regard the pedestrian as a collection of parts that can have a certain degree of deformation.

When dealing with component based approaches more expert classifiers are trained and a voting or a combination scheme is used for the final classification.

Usually three main stages [100] are identified when dealing with part based models: part localization, part refinement and part combination. Part localization provides a score to the feature that describes the location of the part in an absolute framework (commonly referred to as 'star model' [101]) or with respect to other parts [102].

Part refinement may take several forms: part decomposition into subparts, re-training of the part mask for increased discriminative power, or binding the SIFT descriptor of the part with additional, hopefully orthogonal, descriptors (e.g. of texture or color).

Part combination may take the form of 'and' or 'or" operators applied to component parts, with and without spatial constraints. Applying 'and' operators corresponds to simple monomials introducing non-linearity when no spatial constraints are imposed, and to 'doublets' [102] if such constraints exist. Applying 'or' operators can create 'semantic parts' which may have multiple, different appearances yet a

Rigid Part Based Model

This model does not make a separation between the multiple poses of the parts used. It just decomposes the pedestrian into a collection of rigid parts.

A combination of HOG-SVM upper, lower and whole body classifiers is employed by [72], [62].

The individual human is modeled as an assembly of natural body parts [12]. Part detectors are based on edgelet features and are learned by boosting. "Responses of part detectors are combined to form a joint likelihood model that includes cases of multiple, possibly inter-occluded humans. The human detection problem is formulated as maximum a posteriori (MAP) estimation."

A flexible model is also employed by [12] and extended by [103] that use four components corresponding to full body, head-shoulder, torso and legs and also three poses of pedestrians: front/rear, left profile and right profile for which they train an AdaBoost classifier.

A part hierarchy is defined by [6] that model each part as a sub-region of its parent except for a whole-object node. The hierarchy model is depicted in Figure 5.3. For each

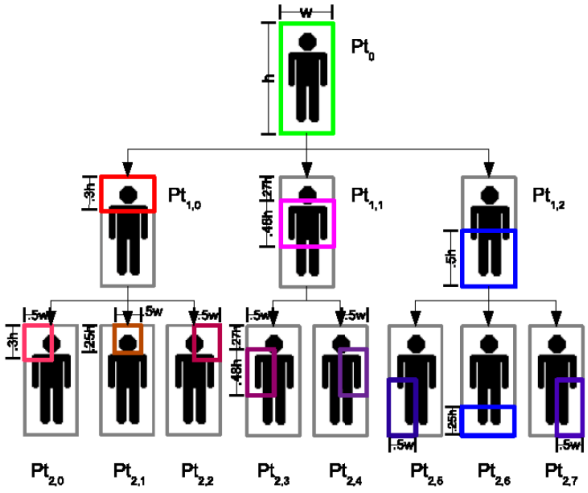


Figure 5.3: The part based hierarchy of [6]

part a Cluster Boosted Tree is trained using edgelet features. "A child node in the hierarchy inherits image features from its parent node and if a target performance can not be achieved only from the inherited features, more features are selected and added to the child node. For the whole-object node, in addition to the detector, a pixel-level segmentor is learned. They formulate segmentation as a binary classification problem and train the segmentor by a supervised learning algorithm. In the training procedure, for each feature in a large feature pool, a pair of weak classifiers for detection and segmentation is built".

Flexible Models

The flexible or deformable models represent a pedestrian in a collection of parts arranged in a deformable configuration. They capture the variations of parts in appearance and define parametric relations between parts.

Humans are modeled as flexible assemblies of parts in the work of [104]. The parts are modeled by co-occurrences of local features which captures the "spatial layout of the parts appearance. Feature selection and the part detectors are learnt from training images using AdaBoost".

One remarkable work in articulated part based models is given by the pictorial structures introduced by [7]. They consider that "an object is modeled by a collection of parts arranged

in a deformable configuration. Each part encodes local visual properties of the object, and the deformable configuration is characterized by spring-like connections between certain pairs of parts. The best match of such a model to an image is found by minimizing an energy function that measures both a match cost for each part and a deformation cost for each pair of connected parts". A tree like pictorial structure is depicted in Figure 5.4. In their work

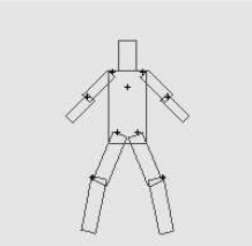


Figure 5.4: Tree like pictorial structure used in [7]

the authors provide statistical models for learning pictorial structures from examples and methods for finding multiple good matches of a model to an image. Matching is formulated as an energy minimization or Maximum A-posteriori Estimate.

A very popular multiscale deformable part model (DPM) is proposed by [8] and extended in [71] and [105]. In their work the models are not bound to a unique position relative to the detection window. The initial model of [8] consists of a global "root" filter and several part models composed of a spatial representation and a part filter. The spatial representation defines a set of allowed placements for a part relative to a detection window and a deformation cost for each placement. The classification score of a scan window is a combination between the root filter and the parts filters all trained on modified HOG features. The detection using

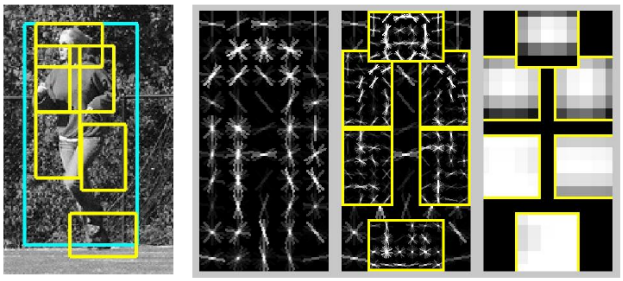


Figure 5.5: Deformable part based model proposed by [8]

the DPM model is depicted in Figure 5.5. For the classification component they introduce a new formalism called latent SVM.

A general method for building star-cascade classifiers from DPM models is presented by [71].

The deformable part model (DPM) has been adopted and exploited in the work of [77], [64], [24], [106], [69], [107], [86], [108], [49], [4].

Pictorial structures are used for pedestrian detection in the work of [109]. Densely sampled shape context descriptors and discriminatively trained AdaBoost classifiers model the

appearance of body parts. The normalized margin of each classifier is interpreted as a likelihood in a generative model and non-gaussian relationships between parts are represented as Gaussians in the coordinate system of the joint between parts.

A new insight on the deformable part model is given by [28] that perform object detection with grammar models. The formalism of grammar models has been introduced by [110]. In grammar models objects are represented by means of other objects using compositional rules. The relative motion of parts with respect to one another is described by deformation rules. This constructs hierarchical deformable part models. The grammar model for pedestrians allows plenty of flexibility in describing the amount of the person that is visible. "The parts in the model, such as the head part, are shared across different interpretations of the degree of visibility of the person. The grammar model also includes subtype choice at the part level to accommodate greater appearance variability across object instances. [28] use parts with subparts to benefit from high-resolution image data, while also allowing for deformations." They explicitly model the source of occlusion for partially visible objects.

A multi-task form of DPM is proposed by [64]. They partition the resolution space into two classes: low resolution containing scan windows in range 30–80 pixels and high resolution that comprises scan windows of dimension higher than 80 pixels. Their method considers the commonness and the differences of samples from different resolutions which are captured by a multi-task strategy. They build a resolution-invariant subspace by using a mapping of features from different resolutions to a common subspace. On this subspace a shared detector is trained in order to capture the structural commonness. The execution time is less than 1s on a standard PC.

A latent deformable template model with a locally affine deformation field and an inference procedure adapted for the method is proposed by [9]. Their template is allowed to deform according to a locally affine deformation field. The idea is based on refining the template "beyond translation and scaling with an additional transformation selected from a finite set of possible perturbations covering aspect ratio change and small in plane rotations." The

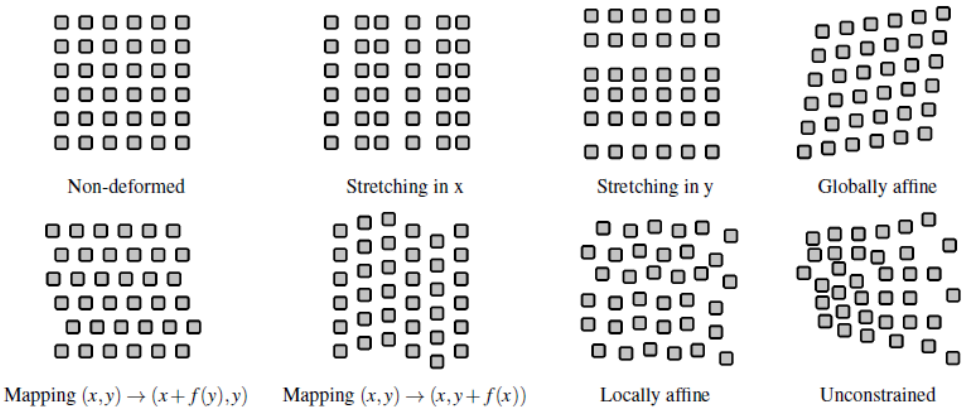
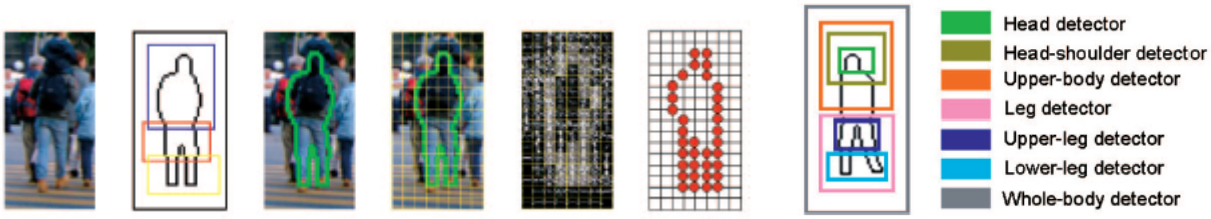


Figure 5.6: The locally affine deformation field proposed by [9]

locally affine deformation field is shown in Figure 5.6. They also prove that the deformation field can be used to measure the similarity of the training samples and hence use it for clustering samples of similar viewpoints and poses.

A hierarchical part based template matching is combined with HOG based discriminative learning in the work of [10]. They first learn a tree of part based contour templates. A histogram of gradient weighted edge orientation histograms is build. For each possible scan window they estimate the optimal pedestrian pose using template matching. Then block features closest to each pose contour point are concatenated in a feature vector as shown in Figure 5.7(a). Given the feature vector and the part part templates for a scan window,



(a) Descriptors computed on contour points from the part templates (b) Part detectors for occluded pedestrians

Figure 5.7: Approach used in [10].

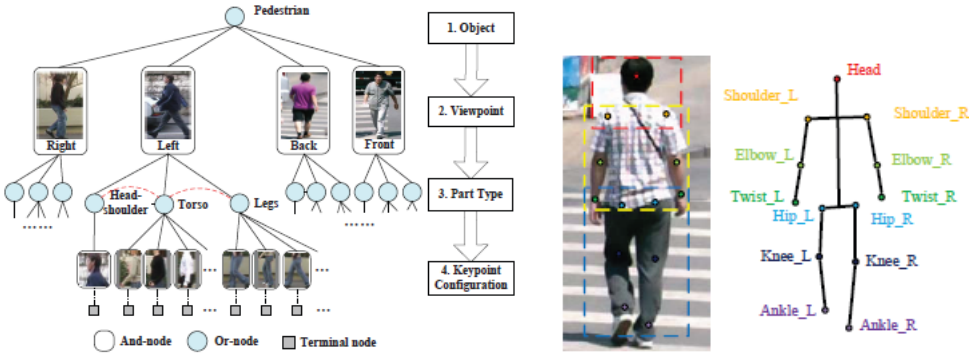
an optimal tree path is estimated by computing the part template score as an average of gradient magnitudes of corresponding orientation bins in each block of the feature vector. The score is defined by orientation consistency instead of distances between edges as in traditional Chamfer matching. Special occlusion handling is modeled by the score given by part combinations (that is weighted sum of individual part responses). The parts are depicted in figure 5.7(b). In order to speed up the scan window detection process and knowing the camera parameters they estimate the expected location of a head point given an arbitrary foot point in the image. They use a homography matrix that is estimated by least-squares method using four or more pairs of annotated foot and head points. Background subtraction is also employed. For evaluation they use the INRIA and MIT-CBCL data sets. The execution time for the occlusion based method is about 5fps on a standard CPU.

Pedestrians are represented in the framework of a grammar dictionary composed of clustered poselets models [11]. The clustered poselets select representative keypoint configurations of human parts and use HOG-SVM to learn the appearance models. HOG features are extracted at multiple resolutions for a test image. Convolution is employed for providing activations for all clustered poselet filters. Maximum a posterior solution is provided by bottom-up inference and connect the activations into the pedestrian full-body. The inference obtains locations the whole body and also for the parts. The execution time with no optimizations on a standard CPU is of 3 seconds per frame.

Statistical component based pedestrian shape model is referred by [16] that divide the pedestrian body into three parts: the head, the upper body and the lower body for which they compute Haar like templates.

Pedestrians are modeled as a hierarchy by four switchable layers of a Deep Neural Network in the work of [76]. They propose a root layer for the whole body and three sub-layers for head-shoulder, upper-body, and lower-body, respectively.

An iterative part based feature synthesis has been proposed by [100]. At each iteration of their method they perform feature generation that provides candidate features and feature



(a) The and-or grammar (b) Pedestrian keypoints

Figure 5.8: Pedestrian representation grammar a keypoints for poselets used by [11]

selection that has as outcome a subset of selected features and a learned linear classifier. Their process is based on a feature hierarchy model that comprises part-based features and operators that are used for part localization, part refining and part combination. Predictive feature selection is used for feature pruning. A large pool of rectangular image fragments is sampled from aligned positive training images. The features employed are HoG features, global maximum features, sigmoid features, localized features, subpart features, LDA features, OR features, cue integration and 'and' features. Their work is extended by [111] that propose an Accelerated Feature Synthesis that reduces the number of locations searched for each part by KDFerns method that compares each image location to only a subset of the model parts. Spatial inhibition object-level coarse-to-fine strategy is used to reduce candidate part locations. Their method achieves about 10fps on a regular CPU and on images of 640×480 pixels.

Multiresolution contextual models are proposed by [60]. They act as a deformable part-based model when scoring large pedestrian instances and as a rigid template when dealing with small size pedestrian instances.

5.3 Attitude Based Pedestrian Detectors

Within the context of a traffic scenario, pedestrians may have several attitudes or perform different actions: wait at the traffic light, cross the street, run for a bus or a taxi, walk or run on the pavement. When performing all these actions, pedestrians have different attitudes: stand, walk, run. We have studied those attitudes and the contexts in which they appear. A fine partition of the pedestrian appearances comes to improve the overall detectin rate. We prove that specific classifiers trained on particular attitudes provide a better overall performance than a generic classifier trained on a non-partitioned pedestrian space.

Attitude based pedestrian detectors have been largely studied, designed and implemented by the author.

Given the large appearance space of pedestrians in monocular intensity images capturing traffic scenes we propose (a) a coarse partitioning scheme into basic attitudes like run, stand walk and (b) a fine partitioning of the space into complex attitudes that combine the basic

attitudes with motion direction information (left, right, front, rear). The two partitioning schemes are included in a dataset we have designed and developed for modeling the variance in pedestrian attitudes.

We study several visual descriptors like HOG, Directional Derivatives, Anisotropic Gaussians, Gabor features and we design and implement an original feature mixture model mapped on an input space partitioned into distinct attitudes.

We create a pool of pattern classifiers that comprises AdaBoost, Bayesian Network, Neural Network and Support Vector Machines. We analyze the performance of those pattern classifiers in the context of a multiple attitude partitioned space and we introduce an original meta-classification scheme that combines several classifiers trained on different attitudes. We prove that the meta-classification scheme has better results than a generic pattern classifier trained on the un-partitioned input space.

We design and implement two novel meta-classification schemes:

1. Basic attitude meta-classifier that is trained on a coarse partition of the input space. The division comprises three main attitudes: stand, run, walk. As features we use histogram of gradient orientations, directional derivatives and anisotropic Gaussians. As pattern classifiers we employ Adaptive Boosting and Bayesian Networks.

2. Complex attitude meta-classifier that is trained on a fine partition of the input space. We propose a segmentation based on semantic concepts that comprise a combination between the actions that pedestrians perform: stand, run, walk and the direction of movement front, back, lateral left, lateral right.

Both meta-classification schemes are evaluated and compared to standard pedestrian detection methods. They prove to have an increased accuracy: an average true positive rate of 90% and a false negative rate smaller than 5%. For a monocular system, the execution time of the basic meta-classification schemes is about 17fps while the complex attitude meta-classifier achieves a speed of 14fps depending on the classifiers and on the features embodied in the model.

We enrich the whole-body meta-classification model with a part based model and introduce the so called "part based meta-classifier". The method considers four pedestrian models: front, rear, lateral left, lateral right (named attitudes or poses). We train a root classifier on all pedestrian attitudes. This root classifier has a high true positive rate but the false positive rate is not very low. Yet it has the role of identifying pedestrian hypotheses fast. These hypotheses are further refined by the specific classifiers trained for different attitudes. The attitude classifiers combine different five body part components. The log average miss rate of the star classifier on Daimler dataset is about 45% outperforming the classical HOG classifier. For evaluation we have used the per image evaluation measure [87]. This measure is hit for pedestrians having height greater than 50 pixels and partially occluded (that is at least 50% of the body is visible). For pedestrians having a height greater than 100 pixels and not occluded the log average miss rate is of about 20%. That is the star classifier detects correctly about 80% of the pedestrians that are not occluded and are closer to the camera.

5.3.1 Basic Attitude Meta-Classifier

For the basic attitude meta-classifier we have considered a coarse division of the pedestrian input space into three attitudes: sanding, walking and running. For those three attitudes we study the relevance of three types of features: histogram of gradient orientations, directional derivatives and anisotropic Gaussians. Based on this analysis we develop a novel attitude based model based on histogram of gradient orientations. We also compare the accuracy of two popular pattern classification methods, namely Bayesian Networks and AdaBoost classifiers. Based on this comparison we design and implement a meta-classification scheme that encompasses AdaBoost classifiers with histogram of gradient orientations extracted on the coarse partition of the attitude space.

Bayesian Networks as Basic Attitude Meta-Classifiers

We have considered three pedestrian attitudes: standing, walking and running. For each pedestrian category we have trained two Bayesian Belief Networks with identical setups but one uses HOG features and another exploits the absolute value of first order partial derivatives computed in four directions. We have compared the results obtained using these two feature sets, HOG providing the best results. We have grouped the classifiers trained for each category in a meta-classifier or a hierarchy of classifiers. The work was presented in [183]. The steps we performed for constructing the meta-classification scheme are:

- build a hierarchy of pedestrian attitudes;
- for each image from the hierarchy of attitudes, relevant features are extracted and selected;
- for each of the three pedestrian attitudes a classifier is trained.

The flow of our pedestrian recognition algorithm is presented in Figure 5.9. With this approach we want to answer the question if a meta-classifier composed of binary classifiers trained on each class of attitudes has better performance than a single binary classifier trained on the whole set?

An important step is the feature selection that identifies and removes irrelevant and redundant information. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. In our experiments we have used CFS feature selection for the directional derivatives.

Given the four directional derivative images as shown in Figure 5.10 after applying CFS the points remaining are grouped on the pedestrian contour and on the pedestrian relevant body parts.

The images in Figure 5.10 depict the selected features for each direction and all the overlapping of all the features selected. One may notice that a certain form of pedestrian can be observed.

Evaluation

Dataset description For our experiments we have have considered three pedestrian attitudes. We have annotated images from INRIA [184] and MIT [177] datasets. The dataset

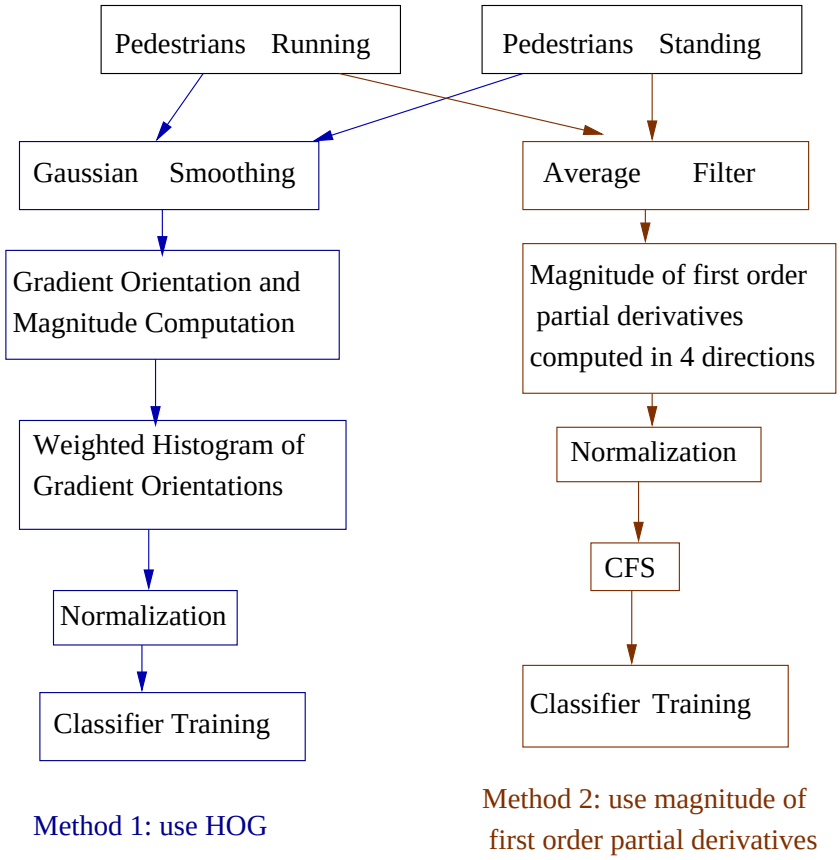


Figure 5.9: Flow of the pedestrian detection algorithm for two categories: pedestrians running and pedestrians standing.

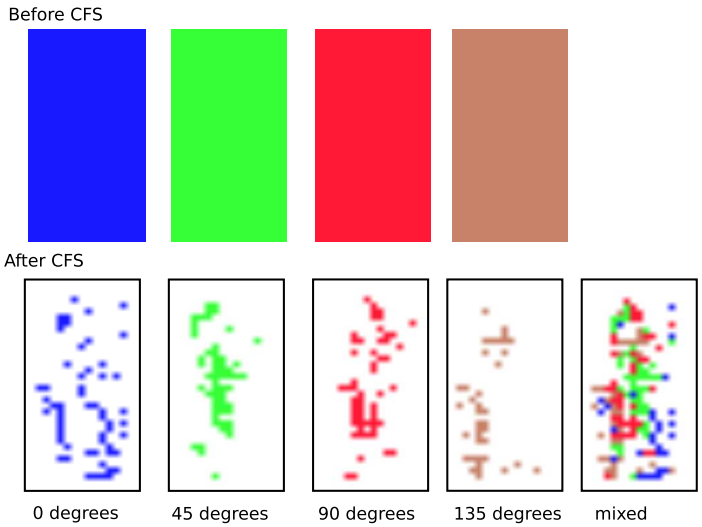


Figure 5.10: Feature selection using CFS

Pedestrians standing (front+rear view)	Pedestrians walking (lateral view)	Pedestrians running (lateral view)
2100 positive train samples 400 positive test samples	1000 positive train samples 400 positive test samples	1100 positive train samples 400 positive test samples
Same negative images: 10000 for training , 2000 for testing.		

Table 5.1: Dataset description for stand, walk and run attitudes

structure is presented in Table 5.1. For each category we have considered images of dimension 18x36 pixels.

Experimental set-up For all three categories, pedestrians running, walking and pedestrians standing, we have computed the two feature sets: histograms of gradient orientations and magnitude of first order partial derivatives computed in four directions. We experimented numerous combinations of parameters for each attribute set:

1. For magnitude of first order partial derivatives computed in four directions the possible parameters are:
   - block size: 3x6, 3x3, 6x6, 6x12 pixels
   - search strategy for correlation-based feature selection: best first, genetic search, random search
2. For HOG the set of parameters is given by:
   - cell size: 3x3, 3x6, 6x6, 6x12 pixels.
   - number of bins in the histogram: 4, 8, 16
   - block size in number of cells: 3x3, 2x2

We retained the parameters that provide optimal results for the detection window of 18x36 pixels:

1. For magnitude of first order partial derivatives computed in four directions: a block size of 6x12 pixels and best first search resulted in a feature set of 316 attributes.
2. For HOG: a cell size of 3x6 pixels with a histogram having 8 bins, and a block size of 3x3 cells resulted in a feature set of 288 attributes.

Concerning the parameters of the Bayesian Network we have used TAN as learning algorithm. It determines the maximum weight spanning tree and returns a Naïve Bayes network augmented with a tree.

As shown in Figure 5.11 we have trained a classifier for each attitude and also, we have trained a classifier that contained in the training set the three training sets of pedestrians having different attitudes. We have used this classifier for comparing the results to the results obtained with our meta-classifier.

Classifiers trained on HOG features

Table 5.2 shows the results of classifiers trained separately on the three attitudes and a

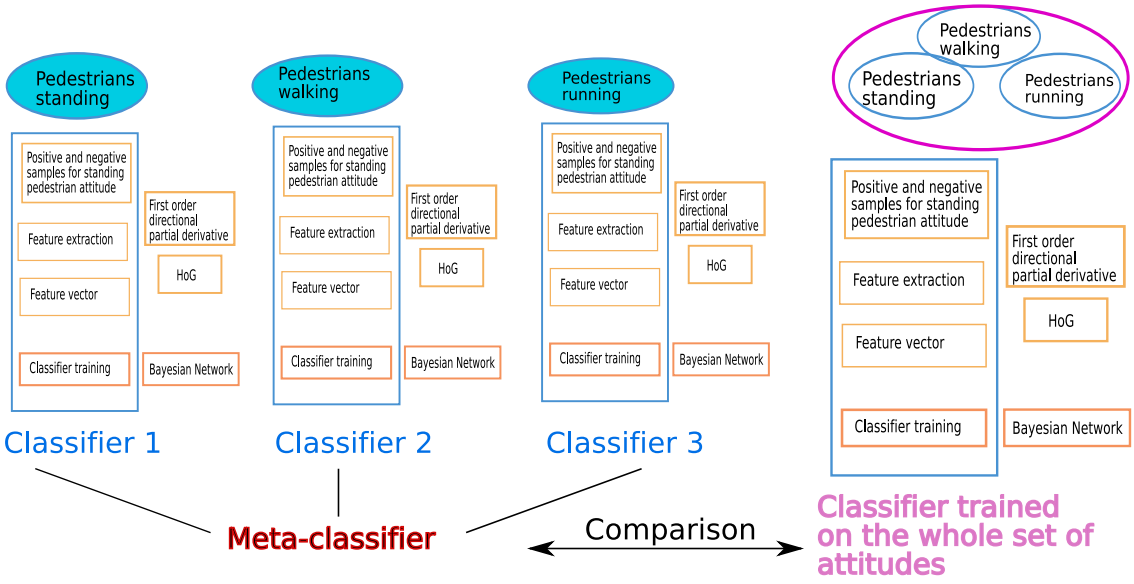


Figure 5.11: Methodology for evaluating run, stand, walk pedestrian classifiers

Classifier trained on / tested on	Pedestrians standing	Pedestrians walking	Pedestrians running	Pedestrians all attitudes
standing	TP = 0.898	TP = 0.72	TP = 0.68	TP = 0.86
walking	TP = 0.79	TP = 0.847	TP = 0.77	TP = 0.81
running	TP = 0.55	TP = 0.76	TP = 0.90	TP = 0.76

Table 5.2: Comparing results of meta-classifiers with a classifier trained on all attitudes for HOG features

Classifier trained on / tested on	Pedestrians standing	Pedestrians walking	Pedestrians running	Pedestrians all attitudes
standing	TP = 0.866	TP = 0.67	TP = 0.69	TP = 0.72
walking	TP = 0.72	TP = 0.875	TP = 0.78	TP = 0.76
running	TP = 0.64	TP = 0.76	TP = 0.805	TP = 0.72

Table 5.3: Comparing results of meta-classifiers with a classifier trained on all attitudes for directional derivative features

classifier trained on all attitudes, all using HOG features. In red we have underlined the best true positive results. We notice that specific classifiers have a true positive rate greater with at least 3% than an classifier trained on all attitudes.

Classifiers trained on directional derivative features

Table 5.3 shows the results of classifiers trained separately on the three attitudes and a classifier trained on all attitudes, all using directional derivative features. In red we have underlined the best true positive results. The results for directional derivatives are not so promising as in the case of HOG, but these features can still be used in the future for other types of classifiers.

We conclude that the recognition rates for the HOG are better than the ones for the magnitude of first order partial derivatives computed in different directions. Nevertheless, the performance of the detector trained on pedestrians standing using first order partial derivatives is quite accurate and its results can be improved. We have considered this set of features because our module is applied in the context of a real-time detection system in which fast computation of features is a must. Also, the CFS feature selection scheme can be replaced by more accurate feature selection methods.

For both sets of features, the resulting meta-classifier formed of three attitude specialized classifiers outperforms the detection rate of a classical learner trained on the whole pedestrian feature space.

Bayesian Networks and AdaBoost Meta-Classifiers

The previous section presented the results for a meta-classifier trained on three pedestrian attitudes. Due to the large overlapping of pedestrians walking with the situations of pedestrians running and pedestrians standing we propose a scheme that is focused only on pedestrians running and standing.

The contributions reside in the development of a mixed classification scheme for pedestrian recognition based on a partitioned pedestrian space. We have trained different classifiers, namely AdaBoost and Bayesian Networks for two categories of pedestrian attitudes: standing and running. We show that the obtained meta-classifier outperforms previous approaches that use the whole un-partitioned pedestrian space. The work was published in [185].

The idea of our research was to divide the complex pedestrian space into several different attitudes. For each attitude we have trained a classifier. We have obtained by this, a tree of classifiers that were all grouped into a meta-classifier. We show that the performance of the meta-classifier is better than the detection rate of a single classifier trained on the whole

un-partitioned object space. Figure 5.12 shows the methodology we have employed for this

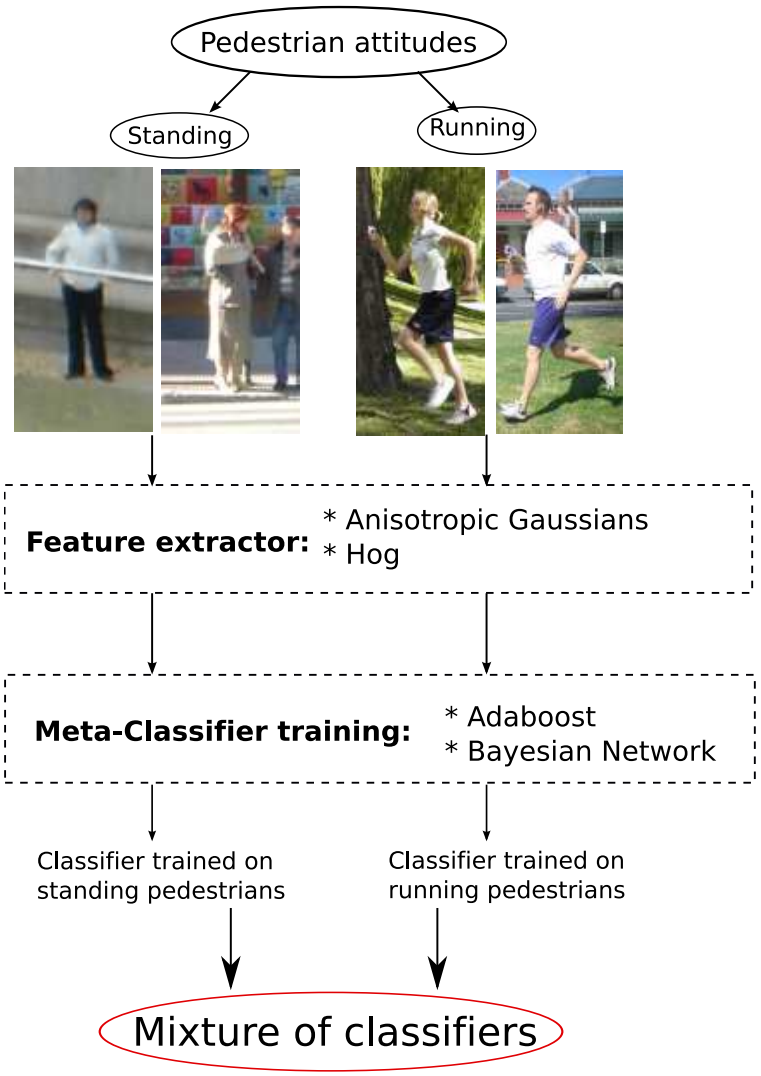


Figure 5.12: Methodology for running and standing pedestrian attitudes

meta-classification class.

Dataset description We have considered two classes of attitudes for pedestrians: standing and running. For each class we have used a training set and a testing set. We have collected our samples in the category standing from the MIT pedestrian database [177] and from the INRIA pedestrian database [184] that contain images of pedestrians in city scenes. We have collected pictures for the category 'running pedestrians' from several images taken from the Internet or from our personal photos and some from the INRIA [184] database. We build a set of 300 images of different running pedestrians. In all the pictures the pedestrians occupy the central position. We have applied a 4 small in-plane rotations with 5°, 10°, -5°and -10°to each image from both classes of attitudes, hence we have obtained a dataset of 1500 pictures of running pedestrians and 2500 pictures of standing pedestrians.

Methodology For each category we have considered images of dimension 18x36 pixels. We have divided the datasets for the standing pedestrian attitude into 2100 training samples

and 400 testing samples and the dataset for running pedestrians was split into 1100 train pictures and the 400 test pictures. The initial negative training set had 12000 images (also of dimension 18x36 pixels) sampled randomly from person-free training photos. An initial training is made and the obtained detector is tested on a larger set of negative images that were not in the initial negative training set. All the false positives are added to the training set and the process is repeated until we reach a good accuracy of the detection. Hence, we obtain a classifier for each category of the considered pedestrian attitudes i.e. running and standing. For each category we have trained two separated classifiers, one that uses HOG features and another that exploits Anisotropic Gaussians. We have compared the results obtained using these two feature sets, HOG providing the best results for pedestrians running and Anisotropic Gaussians gave best results for pedestrians standing. Figure 5.13

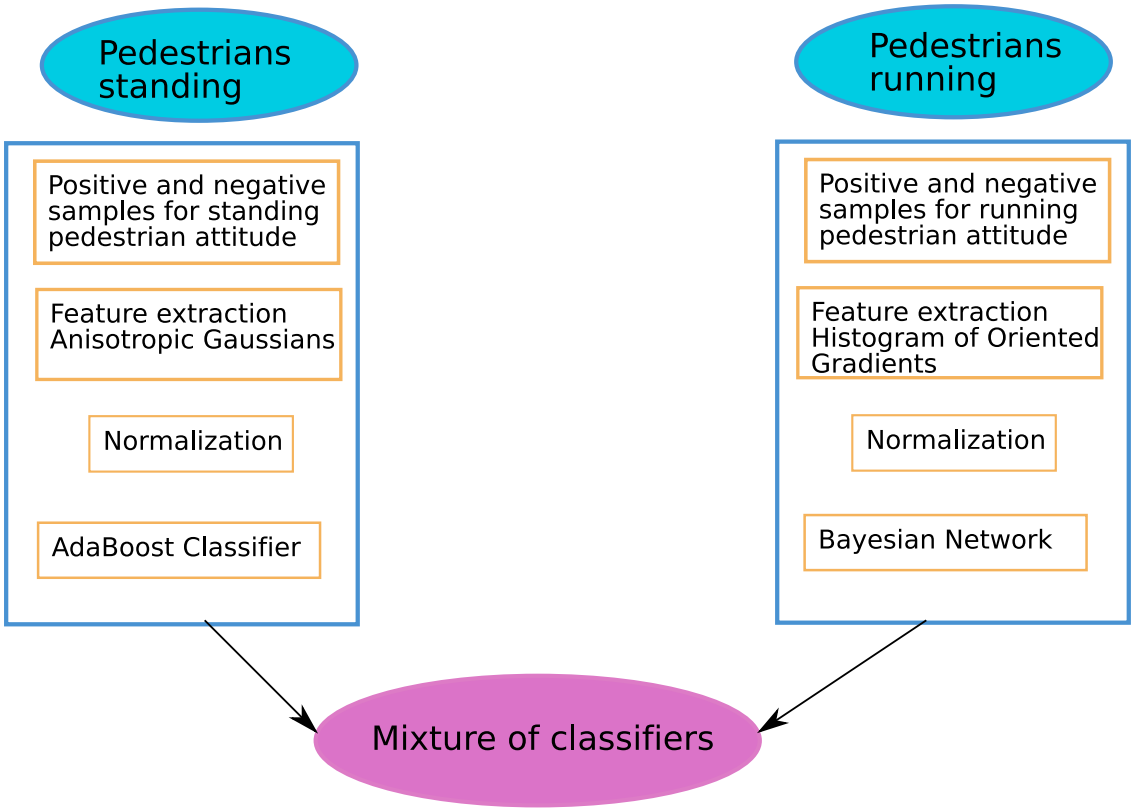


Figure 5.13: Bayesian meta-classifier for running and standing pedestrians

shows the main modules of the proposed system.

We have grouped the classifiers trained for each category in a mixture of classifiers. The next step was the comparison of our method based on the partitioning of the pedestrian space into several classes, with a method that trains a classifier using the whole pedestrian set. In order to perform this comparison we build a third classifier that used as positive training set 1100 images from the category running and the 2100 train images from the category standing and the same set of negative samples as the classifiers build separately for each category.

Figure 5.14 shows samples from the collected dataset.

Considering the two test sets (for pedestrians running and for pedestrians standing) we evaluated the performance of a Bayesian detector trained using the whole object space,

(a) (b)

Figure 5.14: Samples for running and standing pedestrians

an adaptive boosting classifier trained on the whole object space and the performance of the mixture of classifiers (that uses belief networks for pedestrians running and AdaBoost for pedestrians standing). The best results of pedestrian recognition are obtained when applying the mixed scheme.

The detection window is of 18x36 pixels. We have not described the way in which the mixture of classifiers is applied for larger images because the recognition module is designed to work within an existing probabilistic pedestrian detection framework [186], [187] which provides the pedestrian hypotheses.

Evaluation

In this part we will present some results of the system. For both categories, pedestrians running and pedestrians standing, we have computed the two feature sets: histograms of gradient orientations and anisotropic Gaussians. For our detection window of 18x36 pixels we have retained for HoG features: a cell size of 3x6 pixels with a histogram having 8 bins, and a block size of 3x3 cells resulted in a feature set of 128 attributes.

We have evaluated our method with 400 positive samples of running pedestrians, 400 positive images of standing pedestrians and 10000 non-pedestrians. The positive images were collected from INRIA [184] and MIT [177] datasets.

As shown in figure 5.15 we have used three datasets: a dataset that comprises all pedestrians without any attitude separation (we will refer it as ALL_DB), a dataset for pedestrians standing (referred as STAND_DB) and a dataset for pedestrians running (referred as RUN_DB). We have trained separately a classifier that uses anisotropic Gaussians on the ALL_DB, a classifier that uses HOG and Bayesian Network on ALL_DB.

Then we have separated the attitudes and trained on separate attitudes datasets.

Table 5.4 makes a comparison of the recognition rates for the Bayesian Network trained using HOG features and the meta-classifier proposed in our research. We depict the values of the true positive rate (TP) and of the true negative rate (TN). The results show that our approach which trained a classifier for each class of objects and formed a meta-classification

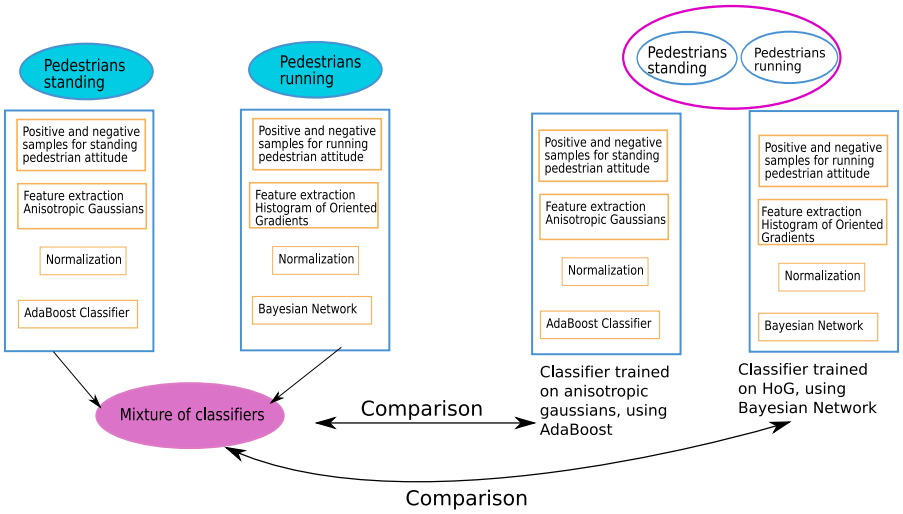


Figure 5.15: Bayesian meta-classifier for running and standing pedestrians

Classifier trained on	ped. running		ped. standing		running & standing	
	TP	TN	TP	TN	TP	TN
Run test	0.90	0.95	0.55	0.91	0.76	0.91
Stand test	0.68	0.90	0.89	0.91	0.86	0.91

Table 5.4: Detection rates obtained using a Bayesian Network trained on the unpartitioned object space using HOG features and the proposed metaclassification scheme

Classifier trained on	ped. running		ped. standing		running & standing	
	TP	TN	TP	TN	TP	TN
Run test	0.75	0.70	0.52	0.78	0.70	0.82
Stand test	0.57	0.63	0.87	0.85	0.79	0.82

Table 5.5: Detection rates obtained using boosted classifiers trained on the ALL_DB object space using anisotropic Gaussians and the proposed meta-classification scheme.

scheme gives better results than the Bayesian Network trained on the mixed set of pedestrians (running and standing). Table 5.5 makes a comparison of the recognition rates for the Adaptive Boosting trained using anisotropic gaussian features and the meta-classifier proposed. We present the values of the true positive rate (TP) and of the true negative rate (TN). The results show that our approach which trained a classifier for each class of objects and formed a meta-classification scheme gives better results than the boosted classifiers trained on the mixed set of pedestrians (running and standing). For both sets of features, the resulting meta-classifier outperforms the detection rate of a classical learner (either Bayesian Network or Adaptive Boosting classifier) trained on the whole pedestrian feature space.

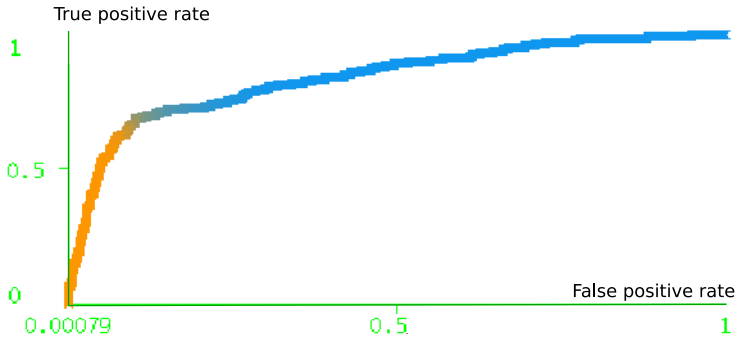


Figure 5.16: Results for classifier trained on pedestrians running

Figure 5.16 shows the detection rate of the classifier for the category pedestrians running(PedRun). The negative training set is shortly referenced with NonPed. We have depicted the true positive rate (TP), precision (Prec.), recall (Rec) and area under ROC (ROC-A). Figure 5.16 also includes the ROC for the classifier trained on pedestrian running set.

	TP	Prec	Rec	ROC-A
PedStand	0.87	0.57	0.87	0.674

Recognition rate of the classifier trained on the running set using HOG

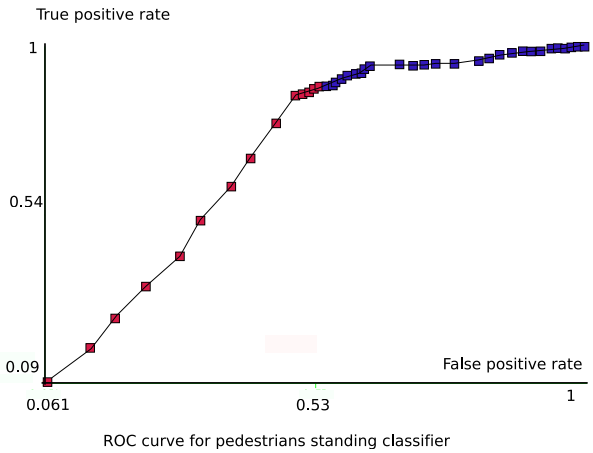


Figure 5.17: Results for classifiers trained on pedestrians standing

Figure 5.17 shows the ROC curve for the classifier trained on pedestrians standing.

Basic attitude Meta-Classifier Application

We have applied the meta-classification scheme as a stand-alone classification module for monocular intensity images.

In a monocular setup the meta-classifier is applied following the standard pipeline of a pedestrian detector as shown in Figure 5.18 Given an input image the monocular region

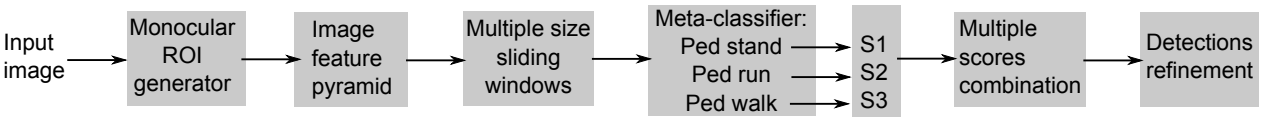


Figure 5.18: Basic meta-classifier in the context of monocular images

of interest generator is applied. Next HOG features are extracted and a multiple scale image feature pyramid is created. For this meta-classification scheme we used no feature approximation techniques. Given the image pyramid, each layer of the pyramid is scanned with three detection models of different sizes corresponding to pedestrians stand, walk and run classifiers. Each location is provided with a score from the three classifiers. Then the scores are combined using "at least one" class label combination scheme. This means that if at least one of the classifiers provides a positive score than the final detection will be "pedestrian". If none of the classifiers gives a positive than the location does not contain a pedestrian.

The execution time of the method is about 17 fps and its speed-up boost is given by the large space pruning factor of the region of interest.

Evaluated on Daimler dataset with occluded pedestrians and pedestrians of all heights the method has a log average miss rate of about 49% outperforming the classical HOG classifier that has a log average miss rate of 52%. When evaluated on pedestrians with at least 80% of the body visible and with a height greater than 100 pixels the log average miss rate is of 30%.

5.3.2 Complex Attitude Meta-Classifier

Using the results of the previous chapters and keeping in mind that we could infer motion information from any monocular or stereo based system we create a fine partition pedestrian attitude space. The fine partition combines in the form of semantic concepts three main attitudes: pedestrians running, standing, walking with direction information like front, back, lateral left and lateral right. The work was published in [188] and further used in [189].

AdaBoost, Neural Networks and SVM meta-classifiers for different pedestrian attitudes

Figure 5.19 shows the semantic concepts that form the fine partition scheme of the pedestrian input space:

We propose a model that has the role of analyzing the correlation between semantic concepts, visual features and pattern classifiers as shown in the scheme in Figure 5.20. Based

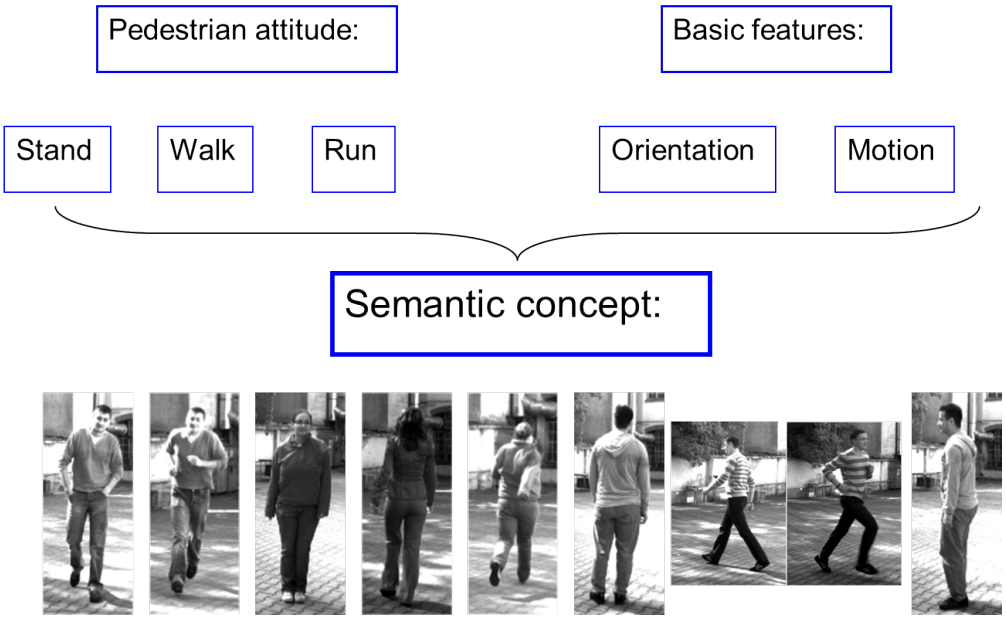


Figure 5.19: Semantic concepts identified for traffic scenes

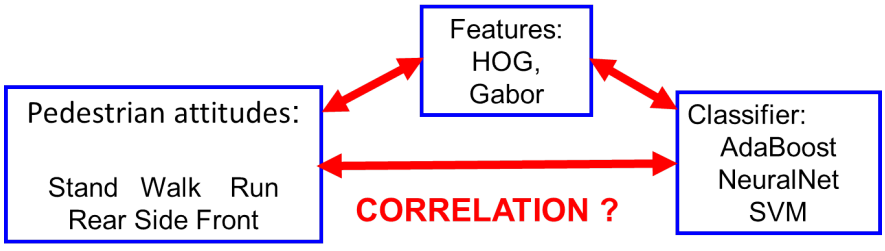


Figure 5.20: The objective for a semantic concept correlation analysis

on the correlation model we design and implement a complex attitude meta-classifier.

As features we have used: Histograms of Oriented Gradients and Gabor wavelets. For classification we have experimented Adaptive Boosting, Neural Networks and Support Vector Machines.

The method can be integrated in a stereo-vision system that can provide the speed and orientation of the objects that appear in a traffic scene. Actually a stereo based pre-classification step is used for determining the speed, orientation and the dimension of the pedestrian pattern.

If that obstacle is probably to be a pedestrian, we can apply a specialized classifier trained on a particular attitude This results in a higher detection accuracy.

Definition of Semantic Concepts

We have collected video sequences of traffic scenes and from them we have extracted three types of attitudes: stand, walk, run. These attitudes are characterized by two basic features: orientation and speed. Using these two basic features we have constructed semantic concepts as presented in Table 5.6. The constants α_1 and α_2 are determined experimentally, α_1 is

Attitude	Orientation	Speed	Semantic concept
Run	80°- 100 °	$s \geq \alpha_1$	Side run
Run	0°- 10 °	$s \geq \alpha_1$	Front run
Run	0°- (-10) °	$s \geq \alpha_1$	Rear run
Walk	80°- 100 °	$\alpha_2 \leq s \leq \alpha_1$	Side walk
Walk	0°- 10 °	$\alpha_2 \leq s \leq \alpha_1$	Front walk
Walk	0°- (-10) °	$\alpha_2 \leq s \leq \alpha_1$	Rear walk
Stand	80°- 100 °	$s = 0$	Side stand
Stand	0°- 10 °	$s = 0$	Front stand
Stand	0°- (-10) °	$s = 0$	Rear stand

Table 5.6: Relationship between semantic concepts and pedestrian attitudes

about 1m/s and α_2 is 0.5m/s.

Method description The architecture of our system comprises two modules: training and classification as shown in Figure 5.21

The training module takes the established semantic concepts and extracts features. Then, on each type of feature it trains a semantic classifier.

In the classification module the semantic classifiers are applied on the 2D pattern that results after analyzing the 3D box of an obstacle detected by the stereo vision system. A pre-classification module analyzes the speed, orientation and box dimension of each detected obstacle. Based on the values of these three characteristics, a precise semantic classifier is applied. A schematic view of the pre-classification module is given in Figure 5.22.

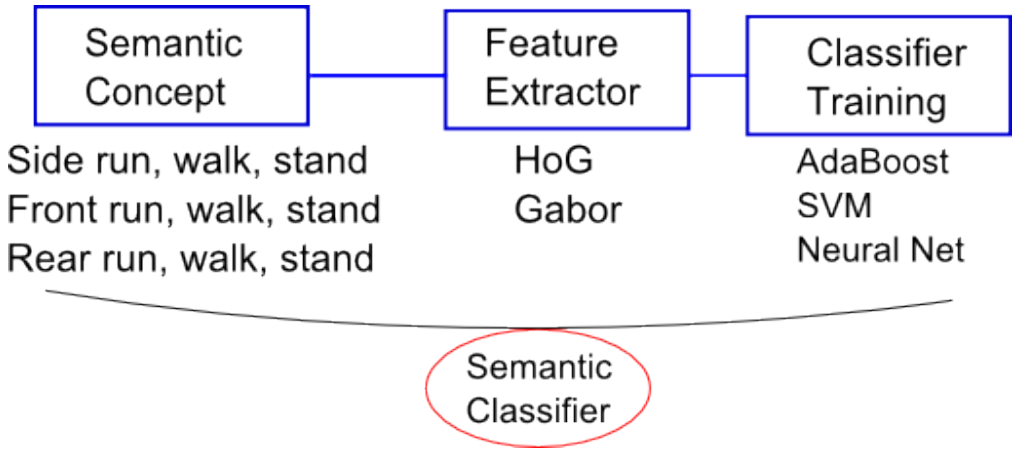


Figure 5.21: Semantic training module

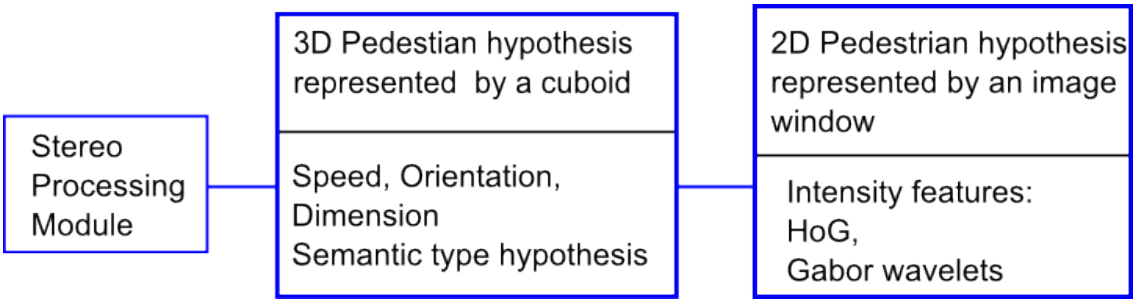


Figure 5.22: Stereo-based preclassification module

Evaluation

Dataset with pedestrian attitudes

We also created a dataset of semantic concepts for pedestrians in traffic scenes. The annotations and the number of instances for each semantic concept are defined in Table 5.7.

The dimension of the images that belong to front run, front stand, front walk, rear run, rear stand, rear walk, side stand is of 75×184 pixels. The pictures for side walk and side run have a dimension of 104×144 pixels. The dimension of the 2D model differs because side running and side walking attitudes require phisically a larger space.

	Front Walk	Front Run	Front Stand	Rear Walk	Rear Run	Rear Stand	Side Walk	Side Run	Side Stand
Nb Pos	600	600	600	600	600	600	600	600	600
Nb Neg	5000	500	5000	5000	5000	5000	5000	5000	5000
Size	74x184	74x184	74x184	74x184	74x184	74x184	104x144	104x144	74x184

Table 5.7: Semantic concepts – dataset description

In each negative set used for the subsequent methods we have included images of semantic concepts , different from the one on which the classification was done. For example, in the negative set of the concept 'walk front' we have included images of the concepts 'walk rear', 'run front', 'run rear' , 'stand rear', 'stand front', 'stand side'.

Experimental setup

The experimental setup for each feature was:

- HoG – cell width = 10, cell height = 10, number of bins = 8, normalization block dimension is 3×3.
- Gabor wavelets: $\sigma \in \{0.5, 1\}$, orientation $\in \{0°, 45°, 90°\}$, scale = 2, frequency $\in \{\sqrt{2}, 2\sqrt{2}\}$

Classifier training methodology We have trained classifiers on each semantic concept and for each semantic concept we have used different features and different classifiers as shown in Figure 5.23.

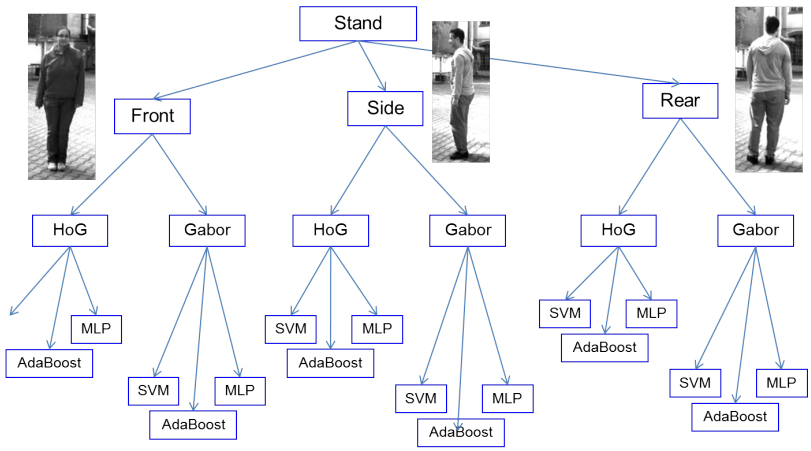
In our experiments we have used 600 positive images for each semantic concept and 5000 negative images. We have used cross-validation with 10 folds for evaluating the results of the classifiers in the dataset.

Confusion matrices

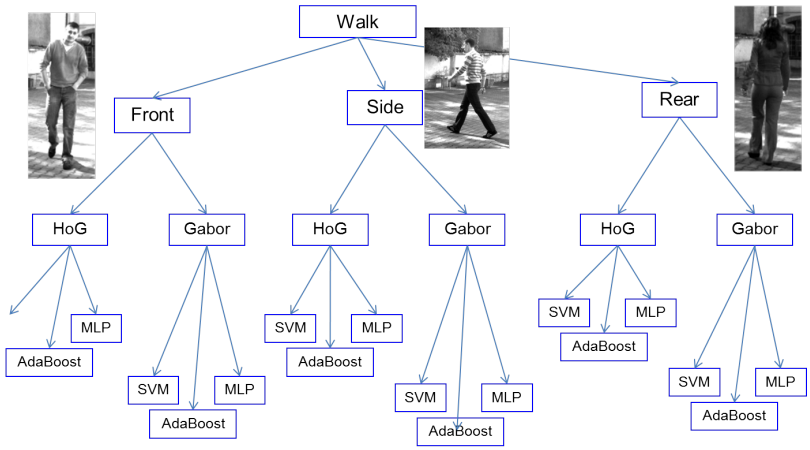
For training the classifiers we have used the functions offered by the library WEKA [165]. The results for the the classifiers trained using HoG features are presented in Table 5.8.

In most of the cases, support vector machines proved to be the best classifier. Yet, for some concepts like run rear, walk side or walk front the best results were obtained using AdaBoost.

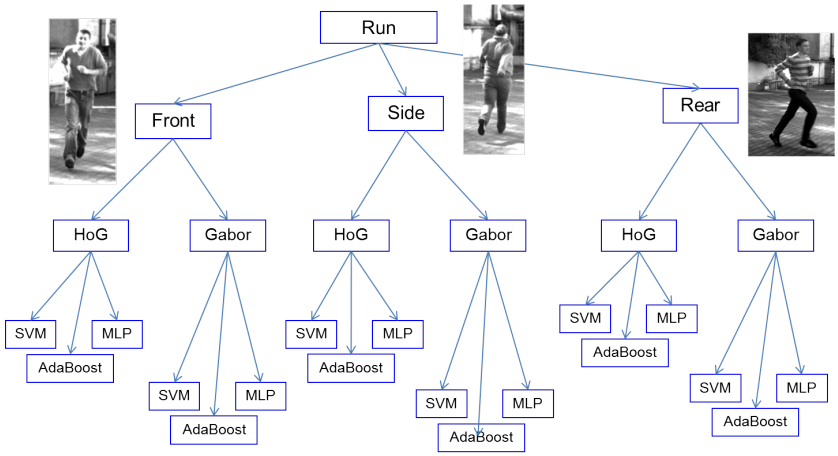
The concept 'stand rear' has not so good classification results using Histogram of Oriented Gradient features. As one can notice in Table 5.9 Gabor wavelets have better results on this



(a)



(b)



(c)

Figure 5.23: Semantic concept – classifiers trained

Concept	AdaBoost			SVM			MLP		
	TP	TN	FP	TP	TN	FP	TP	TN	FP
Stand front	82%	75%	25%	84%	80%	20%	80%	77%	23%
Stand rear	51%	59%	41%	70%	68%	32%	65%	57%	43%
Stand side	55%	65%	35%	79%	80%	20%	71%	82%	18%
Walk front	88%	70%	30%	86%	79%	21%	75%	76%	24%
Walk rear	67%	69%	31%	82%	70%	30%	72%	68%	32%
Walk side	85%	67%	33%	80%	80%	20%	71%	75%	25%
Run front	85%	78%	22%	88%	80%	20%	82%	79%	21%
Run rear	86%	80%	20%	84%	76%	24%	74%	75%	25%
Run side	80%	53%	47%	86%	81%	19%	72%	66%	34%

Table 5.8: Classification results for the recognition using HoG features

semantic concept. TP stands for True Positive rate, TN stands for True Negative rate, FP is used for denoting the false positive rate and FN represents the false negative rate.

Analyzing the results obtained on our dataset, as presented in Table 5.9 and Table 5.8 we can say that there is a slight correlation between concepts and classifiers. Most of the semantic concepts have been best detected by support vector classifiers, but there are semantic concepts for which boosted classifiers or multi layer perceptron gave the best results.

Regarding the features used, in most of the cases, Histogram of Oriented Gradients proved to be the most suitable for the pedestrian detection problem. There are semantic concepts like stand rear, stand side, walk rear that were recognized better using Gabor features rather than using HoG.

Concept	AdaBoost			SVM			MLP		
	TP	TN	FP	TP	TN	FP	TP	TN	FP
Stand front	62%	69%	31%	85%	79%	21%	80%	70%	30%
Stand rear	62%	69%	31%	85%	78%	22%	70%	72%	28%
Stand side	80%	65%	35%	85%	73%	27%	82%	70%	30%
Walk front	80%	83%	17%	87%	89%	11%	72%	84%	16%
Walk rear	85%	79%	21%	87%	78%	22%	80%	77%	23%
Walk side	84%	86%	14%	82%	87%	13%	80%	83%	17%
Run front	80%	78%	22%	82%	82%	18%	85%	80%	20%
Run rear	81%	79%	21%	84%	80%	20%	82%	69%	31%
Run side	73%	79%	21%	83%	82%	18%	82%	76%	24%

Table 5.9: Classification results for the recognition using Gabor wavelets

The disadvantage of Gabor features is their high execution time and this does not make them suitable for a real time execution.

Classifier	Feature	Frames per second	Log Average Miss rate
AdaBoost	HOG	14 fps	46%
AdaBoost	Gabor	2fps	42%
SVM	HOG	10fps	44%
SVM	Gabor	0.8 fps	40%

Table 5.10: Complex meta-classifier performance for pedestrians with heights greater than 50 pixels

Complex Attitude Meta-Classifier Application

In a monocular setup the complex meta-classifier is applied following the standard pipeline of a pedestrian detector as shown in Figure 5.24

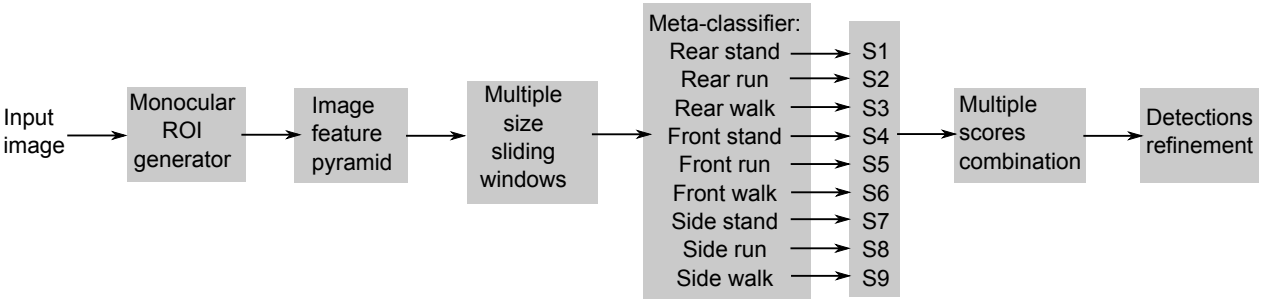


Figure 5.24: Complex meta-classifier in the context of monocular images

Given an input image the monocular region of interest generator is applied. Based on the feature relevance analysis and on the classifier relevance analysis we have used two feature setups, namely HOG and Gabor wavelets and AdaBoost and SVM classifiers and we have evaluated the meta-classification scheme on Daimler dataset. The results are presented in Table 5.10. There is a trade-off between speed and performance. One may notice that the AdaBoost trained on HOG is the fastest but has the smallest performance, while SVM trained on Gabor features has the highest accuracy but it is very slow.

We refine our results to pedestrians having heights higher than 100 pixels and having at least 90% of the body visible. The results are shown in table 5.11 The performance rates are

Classifier	Feature	Frames per second	Log Average Miss rate
AdaBoost	HOG	14 fps	29%
AdaBoost	Gabor	2fps	25%
SVM	HOG	10fps	27%
SVM	Gabor	0.8 fps	24%

Table 5.11: Complex meta-classifier performance for pedestrians with heights greater than 100 pixels and 90% of the body being visible

better reaching up to 76% detection performance with only one false detection at every 10 frames.

5.3.3 Part Based Attitude Meta-Classifier

Considering the results of the previous sections we introduce part based representation to attitude based classifiers published in [190]. The method considers four pedestrian models: front, rear, lateral left, lateral right (named attitudes or poses). Our original contribution resides in the development of a classification scheme based on these models. We train a root classifier on all pedestrian attitudes. This root classifier has a high true positive rate but the false positive rate is not very low. Yet it has the role of identifying pedestrian hypotheses fast. These hypotheses are further refined by the specific classifiers trained for different attitudes. Hence the attitude classifiers have the role of refining the false positive detections of the root classifiers.

Another original contribution resides in the representation of the pedestrian model. We use a block based feature computation approach. Our classifiers are trained using Histogram of Oriented Gradient features (HOG) and Local Binary Patterns (LBP). The novelty resides in the combination of the multi-attitude representation of the pedestrian model with part based feature extraction. That is we define several pedestrian models corresponding to front, rear and lateral attitude and we do not compute the features on the whole pedestrian model, but we extract features parts of interest that are positioned along different pedestrian body parts. Each part is composed of overlapping blocks of dimension 16x16 pixels having an overlap of 8x8 pixels. This part based approach is useful for capturing the variations in shape and position of different pedestrian body parts. We try to model the variance in arms, legs and torso.

We have tested our method in the framework of a stereo-vision system [191]. This system performs several preprocessing operations, among which pedestrian hypotheses generation. The obtained intensity image corresponding to the pedestrian hypothesis is input to our star-based classification scheme that provides a confidence score.

We have also tested the method on monocular images from the Daimler pedestrian dataset. We have considered intensity images (without any stereo-vision information) and we applied a scanning window approach. These experiments show the method has a good accuracy even if for the particular test setup it was time consuming due to the lack of a fast pedestrian hypothesis generation (that we benefit from in the stereo-vision based framework). In order to obtain training data for the attitude classifiers we have analyzed the Daimler pedestrian benchmark data.

Method Overview

The method we propose makes use of the following:

- Stereo-based pedestrian hypothesis generation;
- Component based pedestrian representation by using parts of interest for multiple pedestrian attitudes;

- Visual descriptors based on Histogram of Oriented Gradients and Local Binary Patterns.
- Mixture of expert classifiers based on cascades of AdaBoost learners.
- Detection refinement based on non-maximum suppression.

Our method's originality resides in the development of a pose specific analysis (front view, lateral view and rear view) of pedestrians and in the usage of classification models for each pose based on a block feature computation scheme.

The architecture of the proposed approach is described in Figure 5.25. For each input

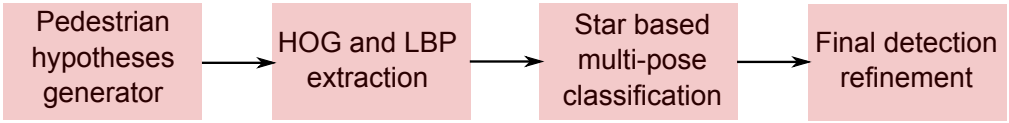


Figure 5.25: The framework for multi-pose pedestrian detection using HOG-LBP features

hypothesis our framework computes the HOG descriptors and the LBP descriptors defined on particular parts. Then the descriptors are concatenated into a single feature vector and it is passed to the classification component that returns a confidence score for the input hypothesis.

Part Based Representation

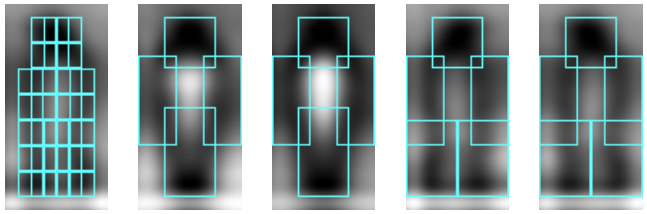
For each of the five pedestrian data representation we use different parts. In order to define these parts we made a block-based analysis of the degree of homogeneity in the gradient magnitude that captures the variations in intensity. For a given image we extract the gradient magnitude and then we divide the image into overlapping blocks of dimension 16×16 pixels. The overlapping factor is of 8×8 pixels. For each block we compute the normalized histogram of gradient magnitudes, $p(f)$, with $f \in [minG, \ldots, maxG]$, minG is the minimum value of the magnitudes in the image and maxG is the maximum value of the magnitudes in the whole image. Then for each block we measure the homogeneity, E:

$$E = \sum_{f=minG}^{maxG} (p(f))^2 \tag{5.1}$$

We analyze each attitude separately and for each we define parts that are comprised of bocks with minimum values of E. We have considered the parts such that we cover the variances of different body parts. Figure 5.26 shows the parts we have used for each attitude and for the root.

Star Classification Model

The pedestrian detection method we propose employs the combination of a root classifier with attitude-specific classifiers. We will refer this structure as being a star detection model. We build five classifiers: *(1) Root* – is a generic classifier trained on all pedestrian attitudes.



(a) Root (b) Front (c) Back (d) Left (e) Right

Figure 5.26: The parts used for extracting features

Its mission is to rapidly identify pedestrians on the trade-off of a larger false positive rate. Its detections will be further refined by the pose-specific classifiers that have a high true positive rate and a very low false positive rate (hence they will eliminate the possible false detections of the root classifier). *(2) Front* – is a specialized classifier considering as positive set only front pose pedestrians. *(3) Rear (or back)* – is a learner focused on rear pedestrian attitude. *(4) Lateral Left* – is a classifier concentrated on pedestrians facing left. *(5) Lateral Right* – is a particular classifier that considers only the pedestrians facing right. Figure 5.27 represents the architecture of the star classification scheme. Each of the five learners is a cascade of

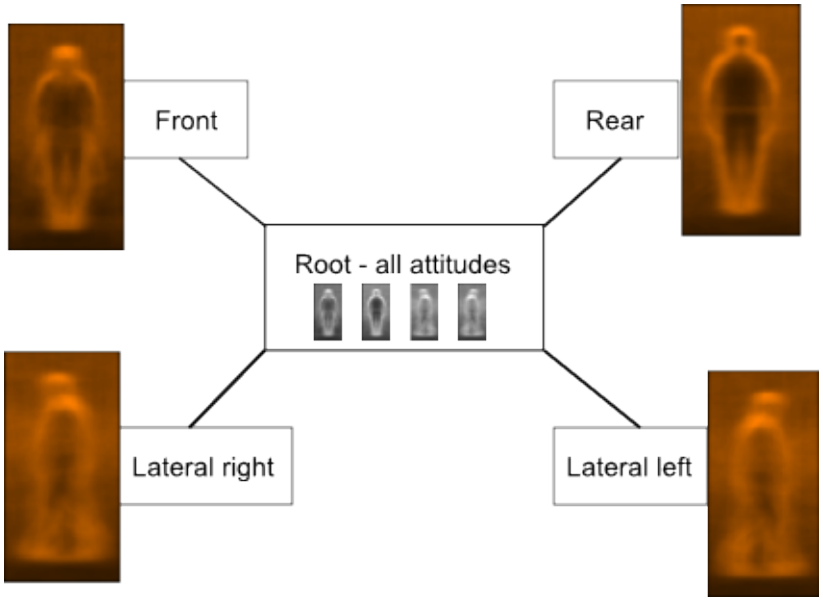


Figure 5.27: Star detection model: combination of Root and Attitude specific classifiers

AdaBoost classifiers containing 2 stages. The final classification score is a combined vote between root and attitude classifiers. The fusion of classification outputs is done using a majority vote filtered by the root classifier. We have five classifiers: c_1, c_2, c_3, c_4, c_5, with c_1 being the root classifier and the others are attitude classifiers. Each of them provides a response for a test image I. The set of responses is $r_1(I), r_2(I), r_3(I), r_4(I), r_5(I)$ and each response is a weighted combination of weak learners responses. If the root response $r_1(I)$ is greater than a given threshold t_1 then we have a high probability that I is a pedestrian and we analyze the votes of the attitude classifiers. The decision is based on the majority

voting scheme: $v = \max(r_2, r_3, r_4, r_5)$. If the maximum response v is greater than a threshold t_v then we consider the image I is a pedestrian, otherwise it is not. The thresholds are particular for each classifier and they have been chosen empirically. Figure 5.28 shows the voting scheme. The proposed method has a high applicability in systems for which motion

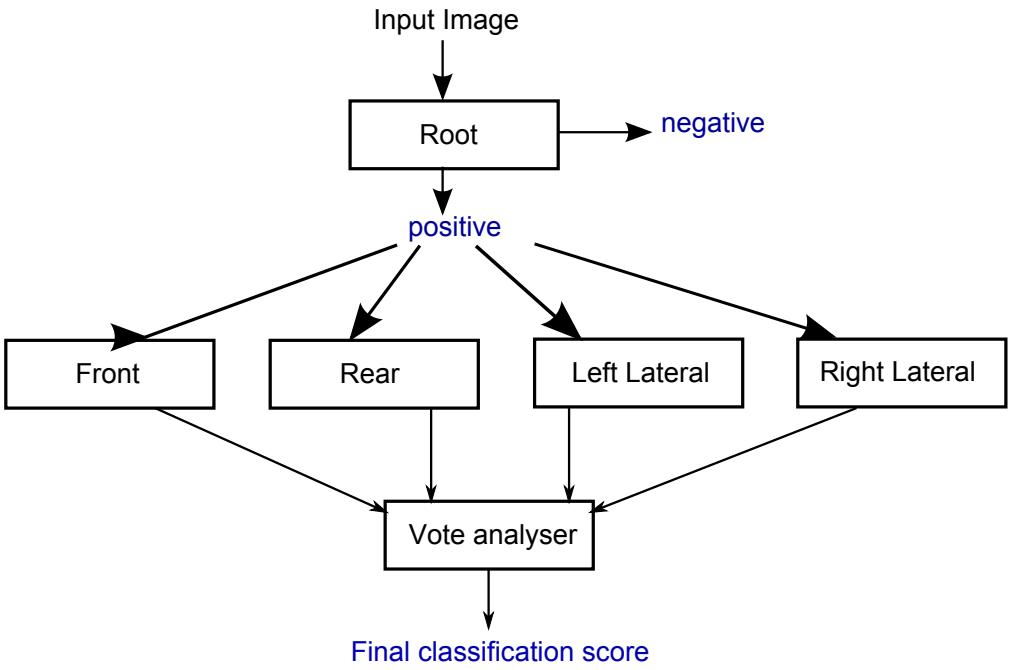


Figure 5.28: Voting scheme for multi-attitude star classification model

information can be captured. Based on the motion law (speed, direction), one can apply only the attitude-specific classifiers and this leads to a very small execution time for pedestrian classification. But for the purpose of this paper we validate our results without considering motion information.

Refining the Hypothesis Location

During the test and evaluation phase we perform a supplementary analysis of the detection window in order to identify optimal locations of the parts for each attitude. We find the areas having minimum homogeneity and based on those areas we define a penalty score. By this operation we try to capture small variations in body part positions with respect to the original parts that we have defined. For each attitude we consider separately the original parts of interest (shown in Figure 5.26) and we analyze a neighbouring region of each block. In that neighbouring region we find the optimal location of the block (for a test image) based on minimum homogeneity E. Figure 5.29 depicts the process. We perform this analysis for each part of a given attitude and we obtain best part locations. Based on these locations we introduce a penalty score that measures the displacement with respect to the original positions. The penalty score is high when parts are shifted in different directions with respect to the original position and it is small when parts are not shifted or they are all displaced with similar offsets with respect to the original. We will exemplify how we compute the penalty score for a given attitude a_j. Suppose that for a_j we have five parts

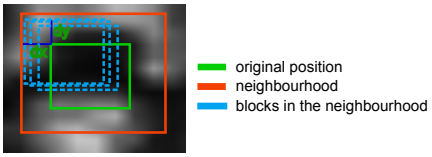


Figure 5.29: Analysis of block homogeneity for optimal part location generation

represented by top-left coordinates, width and height. Their original positions with which the training of the classifier was made is defined by: $o_1(x_{o1}, y_{o1}, w_{o1}, h_{o1}), \ldots o_5(x_{o5}, y_{o5}, w_{o5}, h_{o5})$. After the analysis of the neighbourhood region for each of the five blocks we obtain the best positions using minimum homogeneity analysis: $b_1(x_{b1}, y_{b1}, w_{b1}, h_{b1}), \ldots b_5(x_{b5}, y_{b5}, w_{b5}, h_{b5})$. We measure the displacements of the best block position with respect to the original positions: $d_{xi} = x_{oi} - x_{bi}; d_{yi} = y_{oi} - y_{bi}; i \in (1 \ldots 5)$. We compute the standard deviation of the displacements on the x axis: σ_x and on the y axis: σ_y. The penalty is directly proportional with the sum of standard deviation on both axes $\sigma_x + \sigma_y$. The features will be computed in the best matching positions and the score returned by the attitude classifier will be multiplied with the penalty score.

We also perform a non-maximal suppression step for eliminating multiple detections. For our experiments we have used a method that chooses the smallest subset of the bounding boxes such that each remaining bounding box is within overlap of one of the chosen bounding boxes. The score of each bounding box is set to the sum of the scores of the bounding boxes it covers.

Experimental Results

We have created a framework for the training and evaluation of the proposed method.

Training setup

For training the star-classifier we have used Daimler pedestrian dataset. We scaled all pedestrian images to have a dimension of 64×128. We made an analysis of this database and we manually labeled images corresponding to four different attitudes and this resulted in:

- Frontal pedestrian images: 1800
- Back (rear) pedestrian images: 4500
- Lateral left pedestrian images: 2600
- Lateral right pedestrian images: 2600

We left out the images that did not fall in any of these four categories. For each image in the training database we have extracted HOG and LBP features having the following parameters:

- HOG parameters: block size $= 16 \times 16$, cell size $= 8 \times 8$, number of bins $= 9$, normalization in blocks $= L2Hys$;

- LBP parameters: circle of radius 1, considering 8 neighbours. We have used uniform patterns and we had 59 labels; The LBP descriptors were grouped in overlapping blocks of dimension 16×16 with an overlapping of 8×8 pixels. A histogram of descriptors was computed for each block.

For training the star cascade classifier we have used the implementation of Real AdaBoost in [181]. As weak learners we employed decision stumps. Each classifier has 500 weak learners. The star-classification scheme is comprised of five learners: root and four attitude cascades, each having 2 stages.

Results and evaluation procedure

We have evaluated the proposed method in two different scenarios

(a) On intensity images from a standard dataset that has ground truth labels for test images.

(b) On pedestrian hypothesis obtained in the context of a stereo-vision framework.

The log average miss rate of the star classifier on Daimler dataset is about 45% outperforming the classical HOG classifier. For evaluation we have used the per image evaluation measure [87]. This measure is hit for pedestrians having height greater than 50 pixels and partially occluded (that is at least 50% of the body is visible). For pedestrians having a height greater than 100 pixels and not occluded the log average miss rate is of about 20%. That is the star classifier detects correctly about 80% of the pedestrians that are not occluded and are closer to the camera.

Another test case was made using the pedestrian hypothesis generated by the stereo-vision system. The results are shown in Figure 5.30 .

For the evaluation on the hypotheses generated by the stereo-vision system we took a sequence in an urban environment setting. We manually annotated the pedestrians and non-pedestrian objects (such as cars or poles). Then we run our star-based classification scheme on the annotated sequence.

We evaluated the performance when applying just the root classifier (blue curve in Figure 5.30) and when applying the star classification (red curve). It can be noticed that the star based classification has better results.

The execution time for one cascade is comparable to real-time execution but the application of five different cascades is time consuming and further improvements should be made on this aspect. We hit an execution time of 16 fps.

As future improvements we can either use motion and direction information and apply only the root and the specific pose classifier, or we can reduce the number of stages in each cascade. Some results on the images taken with our stereo-based framework are displayed in Figure 5.31.

5.3.4 Bag of Words For Pedestrian Detection

The bag of words model has been actively adopted by content based image retrieval and image annotation techniques. We employ this model for the particular task of pedestrian

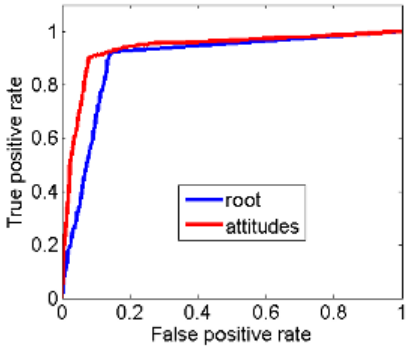


Figure 5.30: Evaluation of root vs. multi-attitude star classification model on stereo-based pedestrian hypotheses

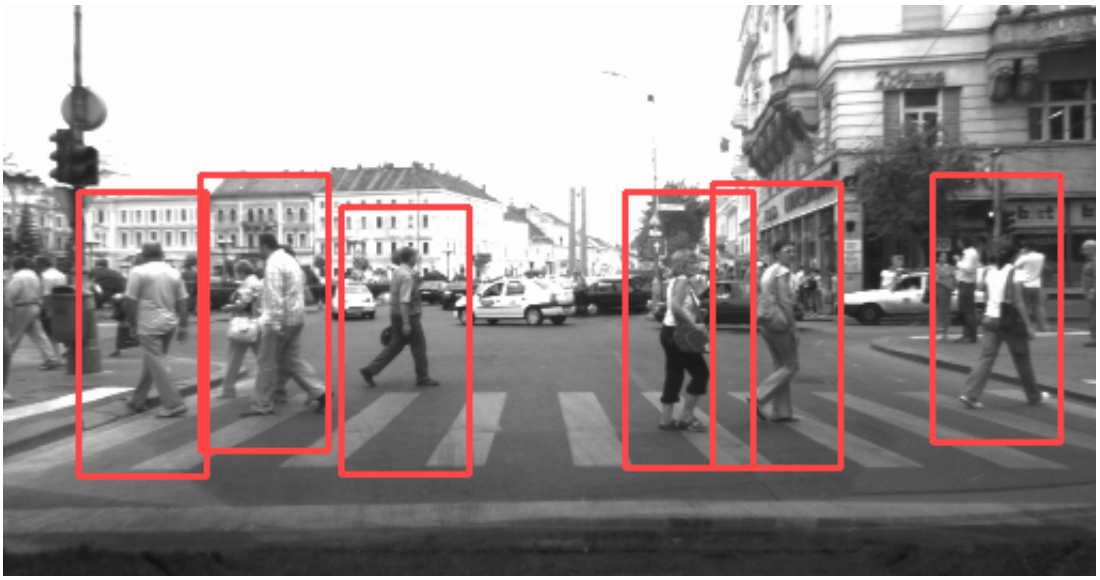


Figure 5.31: Pedestrian detection results

detection in two dimensional images, producing this way a novel approach to pedestrian detection. The work was published in [192] and used by [193].

We perform a study of two types of representations used for pedestrian detection:

- Representation based on primitive features: we have extracted relevant visual features in the field of pedestrian detection, namely Haar and Histogram of oriented Gradient (HoG).
- Representation based on codebook computed from primitive features.

We apply the same classification algorithm to both representation, for features computed on images in benchmark datasets. We notice that the codebook representation provides better detection results than the representation using primitive visual features.

Method Description

The methodology we employ is depicted in Figure 5.32. The main steps are the following:

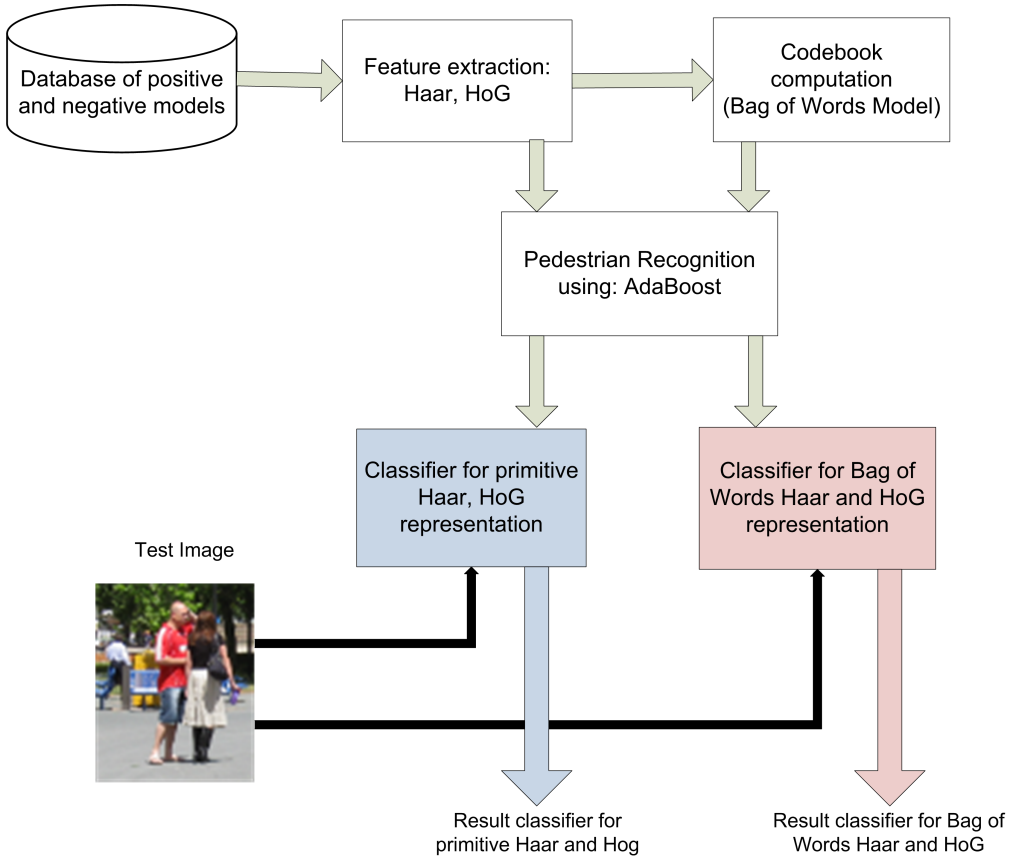


Figure 5.32: Methodology: pedestrian detection based on primitive features and based on the bag-of-words model of the primitive features

1. Extract primitive features from different datasets. By primitive features we mean: Haar wavelets, HoG features.

2. Randomly choose a number of positive images and generate for them the codewords. Then take all the images in the positive and negative training set and compute the extended codebook.

3. Feed the primitive features to a classification module (AdaBoost).

4. Send the code-books for each feature to the classification module.

5. Compare the detection results of the previous two steps.

Algorithm for Codebook Generation From Visual Features

The Bag of Words (BoW) model has been introduced by natural language processing techniques and during the last years it has been used extensively in computer vision for the object recognition task.

To represent an image using BoW model, an image can be treated as a document. For the image context we need to define the "word" concept. This concept has different meanings and representations depending on the task we need to solve, on the images and on the features extracted for them.

Three main computational steps are employed by the bag-of-words model [194]:

1. feature detection: extract several local patches (or regions), which are considered as candidates for basic elements, "words".

2. feature description: each image is abstracted by several local patches. Feature representation methods deal with how to represent the patches as numerical vectors. These methods are called feature descriptors.

3. The final step is to convert vector represented patches to "codewords" that are representative for several similar patches. One simple method is performing K-means clustering over all the vectors.

We have modified the classic approach of the bag-of-words model by transforming the local patches into features. Our idea is to find the most representative features by clustering and then, for each image compute a histogram representation that stores the information about how many features are in the clusters. The steps of our algorithm are:

1. Randomly choose p images from the training data set.

2. For each image compute the features. The number of features may differ from image to image. We denote f_i the number of features computed for the i^{th} image.

3. Construct a large feature space by putting all the features for all images in a single feature vector.

4. Perform a supervised clustering using all the features of the large feature vector.

5. The features representing the centers of the clusters will be the codewords.

These steps are done for each class of objects, in our case for the Pedestrian class and for the NonPedestrian class.

The second major step consists in representing an image by its codebook. To obtain the codebook representation of an image we do the following:

- Compute all the features of the image;
- Find the cluster to which each feature belongs;
- Count how many features are in each cluster;

The final codebook representation feature vector associated to an image has a number of elements equal to the number of clusters and the value of the element at position i is given by the number of features that belong to cluster i for the given image.

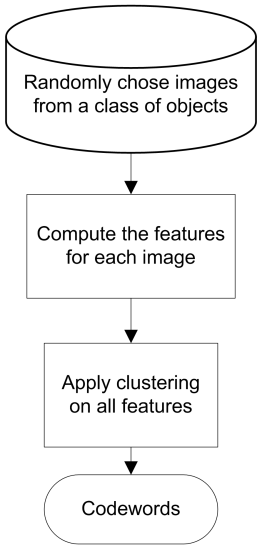


Figure 5.33: Codeword generation for images in a given class

Evaluation

We perform our evaluation on Daimler and NICTA pedestrian datasets.

Experimental set-up

For both features, Haar and HOG, and for both representations (codebook and primary) we have used the AdaBoost classification method. Our experiments have been done with the machine learning library, WEKA [1] [166]. The parameters of the ensemble learning algorithm were the following:

- Weak learner type: decision stump.
- Number of iterations: 10.

[1] http://www.cs.waikato.ac.nz/ml/weka/

As clustering method for generating the codewords we have used k-means. The general idea of k-means is that being given a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a d-dimensional real vector, the algorithm tries to partition the n observations into k sets $(k \leq n)$ $S = S_1, S_2, \ldots, S_k$ so as to minimize the within-cluster sum of squares (WCSS):

$$\text{argmin}_S \sum_{i=1}^{k} \sum_{x_i \in S_i} ||x_j - \mu_i||^2 \tag{5.2}$$

where μ_i is the mean of points in S_i.

Haar features influence on codebook based detection

The first experiment comprised Haar features computed on Daimler dataset.

Classification results based on primitive Haar features

For Daimler dataset we have randomly chosen 5000 negatives images and 4800 positives images and we have created a classification model. The number of Haar features was equal to 840 for each image.

For the test set we have considered all the other images, that is: 20000 negatives and 19195 positives. The results are the following:

- Correctly Classified Instances = 32981 that is 84.1459 %;
- Incorrectly Classified Instances = 6214 that is 15.8541 % of the total number of samples;
- Kappa statistic = 0.6831;
- Mean absolute error = 0.2644;
- Root mean squared error = 0.3515;
- Relative absolute error = 52.9121 %;
- Root relative squared error= 70.3168 %;

The detailed accuracy by class is:

The confusion matrix is given in table 5.13.

Classification results based on Haar codebook representation

For the codebook representation of Haar features extracted on Daimler dataset we have randomly chosen 500 positives and 500 negatives. We have used them for generating a bag of words model for the positive data and a bag of words model for the negative data. Each of the models contained 70 codewords (we have generated 70 clusters). Next we have randomly chosen other 4800 positive images that we have used for generating the codebooks of positives and we have picked randomly 5000 negatives for generating the codebook of negatives. All the remaining images have been used for creating the test data that contained: 19208 pedestrians and 19984 non-pedestrians. The results are much better than in the case of primitive Haar features:

- Correctly Classified Instances = 39148 that is 99.8877 %;

Class	Measure	Value
Pedestrian	TP Rate	0.866
Pedestrian	FP Rate	0.182
Pedestrian	Precision	0.82
Pedestrian	Recall	0.866
Pedestrian	ROC-Area	0.9
NonPedestrian	TP Rate	0.818
NonPedestrian	FP Rate	0.134
NonPedestrian	Precision	0.864
NonPedestrian	Recall	0.818
NonPedestrian	ROC-Area	0.9

Table 5.12: Primitive Haar Detailed Accuracy on Daimler dataset

# Pedestrians	#NonPedestrians	Classified as
16617	2578	Pedestrians
3636	16364	NonPedestrians

Table 5.13: Confusion Matrix for Primitive Haar on Daimler dataset

- Incorrectly Classified Instances = 44 that is 0.1123 % of the total instances;
- Kappa statistic = 0.9978;
- Mean absolute error = 0.0014;
- Root mean squared error = 0.0259;
- Relative absolute error= 0.2776 %;
- Root relative squared error = 5.1845 %;

The detailed accuracy by class is given in table 5.14:

Class	Measure	Value
Pedestrian	TP Rate	0.999
Pedestrian	FP Rate	0.001
Pedestrian	Precision	0.999
Pedestrian	Recall	0.999
Pedestrian	ROC-Area	0.999
NonPedestrian	TP Rate	0.999
NonPedestrian	FP Rate	0.001
NonPedestrian	Precision	0.999
NonPedestrian	Recall	0.999
NonPedestrian	ROC-Area	0.999

Table 5.14: Codebook Haar Detailed Accuracy on Daimler dataset

# Pedestrians	#NonPedestrians	Classified as
19180	16	Pedestrians
28	19968	NonPedestrians

Table 5.15: Confusion Matrix for Codebook Haar on Daimler dataset

The confusion matrix is show in table 5.15.
The second experiment comprised Haar features computed on NICTA pedestrian dataset.
Classification results based on primitive Haar features

For NICTA training set we have considered 7000 positive images and 7000 negative images having a resolution of 16×40 pixels. The number of features extracted for each image equals 1152. For testing we have used 13785 pedestrian images and 23100 negative images. The classification results are as follows:

- Correctly Classified Instances = 32605 that is 88.3221 %;
- Incorrectly Classified Instances = 4311 that is 11.6779 % of the total instances;
- Kappa statistic = 0.7509;
- Mean absolute error = 0.1851;
- Root mean squared error = 0.291;

The detailed accuracy by class is provided in table 5.16:

Class	Measure	Value
Pedestrian	TP Rate	0.84
Pedestrian	FP Rate	0.091
Pedestrian	Precision	0.999
Pedestrian	Recall	0.848
Pedestrian	ROC-Area	0.949
NonPedestrian	TP Rate	0.909
NonPedestrian	FP Rate	0.16
NonPedestrian	Precision	0.999
NonPedestrian	Recall	0.904
NonPedestrian	ROC-Area	0.949

Table 5.16: Primitive Haar Detailed Accuracy on NICTA dataset

The confusion matrix is provided by table 5.17.
Classification results based on Haar codebook representation
For the codebook representation on the NICTA dataset we have randomly chosen 1000 positives and 1000 negatives with which we have generated the centers of the clusters. Then, the codebook representation for the training set was formed of 8000 positive images and 8000 negative images. The evaluation was done on a set that contained 13915 positives and 22999 negatives.

# Pedestrians	#NonPedestrians	Classified as
11695	2221	Pedestrians
2090	20910	NonPedestrians

Table 5.17: Confusion Matrix for Primitive Haar on NICTA dataset

The results are:

- Correctly Classified Instances = 36882 (99.9133 %),
- Incorrectly Classified Instances = 32 (0.0867 %),
- Kappa statistic = 0.9982;
- Mean absolute error = 0.0012;
- Root mean squared error = 0.0255;
- Relative absolute error = 0.2331 %;
- Root relative squared error = 5.1203 %

The detailed accuracy by class is depicted in table 5.18:

Class	Measure	Value
Pedestrian	TP Rate	0.999
Pedestrian	FP Rate	0.001
Pedestrian	Precision	0.998
Pedestrian	Recall	0.999
Pedestrian	ROC-Area	0.999
NonPedestrian	TP Rate	0.999
NonPedestrian	FP Rate	0.001
NonPedestrian	Precision	0.999
NonPedestrian	Recall	0.999
NonPedestrian	ROC-Area	0.999

Table 5.18: Codebook Haar Detailed Accuracy on NICTA dataset

The confusion matrix is displayed in table 5.19.

# Pedestrians	#NonPedestrians	Classified as
13908	7	Pedestrians
25	22974	NonPedestrians

Table 5.19: Confusion Matrix for Codebook Haar on NICTA dataset

HOG feature influence on codebook based detection

For Histogram of Gradient orientation features we have used the standard parameters of computation:

- cell size of dimension (8×8)
- block size of dimension (16×16)
- block stride of 8×8
- unsigned gradient representation
- L2Hys normalization

As the Daimler dataset contains small images $(18x36)$ the number of HoG descriptors having standard parameters is relatively small. That is why we have concentrated our experiments with HoG features on NICTA database working with images of dimension 64×80.

Classification results based on primitive HoG features

For the training set we have used 8000 positive images and 8000 negative images. For each image we have extracted 144 features. For testing we have used 16413 pedestrian images and 20503 non-pedestrian images.

The obtained results are as follows:

- Correctly Classified Instances = 30923 (83.7658 %),
- Incorrectly Classified Instances = 5993 (16.2342 %);
- Kappa statistic = 0.6662;
- Mean absolute error = 0.2136;
- Root mean squared error = 0.3352;

The detailed accuracy by class is shown in table 5.20:

Class	Measure	Value
Pedestrian	TP Rate	0.874
Pedestrian	FP Rate	0.185
Pedestrian	Precision	0.741
Pedestrian	Recall	0.874
Pedestrian	ROC-Area	0.922
NonPedestrian	TP Rate	0.815
NonPedestrian	FP Rate	0.126
NonPedestrian	Precision	0.915
NonPedestrian	Recall	0.815
NonPedestrian	ROC-Area	0.922

Table 5.20: Primitive HoG Detailed Accuracy on NICTA dataset

# Pedestrians	#NonPedestrians	Classified as
12168	1748	Pedestrians
4245	18755	NonPedestrians

Table 5.21: Confusion Matrix for Primitive HoG on NICTA dataset

The confusion matrix is displayed in table 5.21.

Classification results based on HoG codebook representation

For the codebook representation we have randomly chosen 1000 positives and 1000 negatives for which we have generated the clusters. The number of clusters is equal to 60. For computing the codebook representation of the training set we have chosen 8000 positives and 8000 negatives, while for testing we have analyzed 13825 positives and 23089 negatives.

The classification results are:

- Correctly Classified Instances = 35404 (95.9094 %);
- Incorrectly Classified Instances = 1510 (4.0906 %);
- Kappa statistic = 0.9128;
- Mean absolute error = 0.0601;
- Root mean squared error = 0.175 ;
- Relative absolute error = 12.0399 %;
- Root relative squared error = 35.0631 %.

The detailed accuracy by class is provided in table 5.22:

Class	Measure	Value
Pedestrian	TP Rate	0.943
Pedestrian	FP Rate	0.031
Pedestrian	Precision	0.949
Pedestrian	Recall	0.943
Pedestrian	ROC-Area	0.992
NonPedestrian	TP Rate	0.969
NonPedestrian	FP Rate	0.057
NonPedestrian	Precision	0.965
NonPedestrian	Recall	0.969
NonPedestrian	ROC-Area	0.992

Table 5.22: Codebook HoG Detailed Accuracy on NICTA dataset

The confusion matrix is shown in table 5.23.

# Pedestrians	#NonPedestrians	Classified as
13115	800	Pedestrians
710	22289	NonPedestrians

Table 5.23: Confusion Matrix for Codebook HoG on NICTA dataset

Conclusion

Several methods for pedestrian detection in monocular intensity images have been presented. The authors' original contributions reside in the creation of a two fold partitioning of the pedestrian attitude space: (1) coarse partition that comprises basic attitudes like run, sand, walk and (b) fine partitioning that combines the basis attitudes with motion direction information: side, rear or front motion.

In the context of the two partitioning schemes feature mixture model that maps relevant features to attitudes is proposed and developed. The considered features are Histogram of Gradient Orientations, Directional Derivatives, Anisotropic Gaussians and Gabor features.

A pool of pattern classifiers that comprises AdaBoost, Bayesian Network, Neural Network and Support Vector Machines was created. An analysis of the performance of those pattern classifiers in the context of the multiple attitude partitioned space was performed. An original meta-classification scheme that combines several classifiers trained on different attitudes was described in this chapter. Based on the experimental results the author proves that the meta-classification scheme has better results than a generic pattern classifier trained on the un-partitioned input space.

Two original meta-classification schemes were designed and implemented:

1. Basic attitude meta-classifier that is trained on a coarse partition of the input space. The division comprises three main attitudes: stand, run, walk. As features we use histogram of gradient orientations, directional derivatives and anisotropic Gaussians. As pattern classifiers we employ Adaptive Boosting and Bayesian Networks.

2. Complex attitude meta-classifier that is trained on a fine partition of the input space. We propose a segmentation based on semantic concepts that comprise a combination between the actions that pedestrians perform: stand, run, walk and the direction of movement front, back, lateral left, lateral right.

Both meta-classification schemes are evaluated in a stereo-vision framework and in a monocular system. For the stereo-vision system the basic attitude meta-classifier provides an overall improvement of about 5% and the complex attitude meta-classifier gives a pedestrian detection improvement of 8% leading to an overall detection accuracy of about 90%, while obeying the real time execution constraints.

For the monocular setup we use the Daimler pedestrian benchmark data. The basic meta-classifier evaluated with occluded pedestrians and pedestrians of all heights the method has a log average miss rate of about 49% outperforming the classical HOG classifier that has a log average miss rate of 52%. When evaluated on pedestrians with at least 80% of the body visible and with a height greater than 100 pixels the log average miss rate is of 30%. The basic meta-classifier achieves an execution time of 17 fps.

When evaluated on pedestrians having heights greater than 50 pixels and with at least 50% of the body visible the complex meta-classifier achieves a log average miss rate of 44% at 14 fps. If we refine the space by dealing with pedestrians that are closer (i.e. having a height greater than 100 pixels) and not heavily occluded (i.e. at least 90% of the body is visible) the performance rates are better reaching up to 76% detection performance with only one false detection at every 10 frames.

We enhance the multi-attitude scheme with the inclusion of part-based classifiers. We combine models for different pedestrian attitudes lateral, front, rear with a root classifier trained on all attitudes. The role of the root classifier is to identify pedestrians fast with the cost of admitting more false positives. The attitude classifiers come to refine the root decision and to eliminate the false detections and refine the positive detections. The root and attitude classifiers are trained on HOG and LBP features computed for different parts of interest defined based on edge homogeneity minimization function. The proposed method operates at 14 fps. The log average miss rate of the star classifier on Daimler dataset is about 45% outperforming the classical HOG classifier. For evaluation we have used the per image evaluation measure of the framework [87]. This measure is hit for pedestrians having height greater than 50 pixels and partially occluded (that is at least 50% of the body is visible). For pedestrians having a height greater than 100 pixels and not occluded the log average miss rate is of about 20%. That is the star classifier detects correctly about 80% of the pedestrians that are not occluded and are closer to the camera.

In our work we encompass a section in which we study the relevance of the codebook representation of different visual features like Haar and HOG. A general conclusion that can be drawn from the experiments we have performed is that, in terms of accuracy, the codebook representation overcomes the representation based on primitive features. Another advantage of the codebook is the dimension of data space that is much smaller and the classification algorithms work faster. Nevertheless, for a test image we still need to compute all the features in order to generate the codebook, hence the feature computation time is not reduced.

Chapter 6

Pedestrian Detection in Infrared Images

The infrared field is of high interest because it is sensitive to the heat emitted by objects. Hence the usage of an infrared camera can improve the accuracy of pedestrian detection for monocular visible cameras because infrared can be used in extreme conditions like rain, fog, snow and it overpasses the monocular cameras for night vision applications. The benefits of long wave infrared sensors (LWIR) are exploited and a solution for pedestrian localization and pedestrian detection in infrared images is proposed and developed. The 'infrared' term is used throughout the chapter to denote the long wave infrared band.

Most approaches for detecting pedestrians in automotive applications rely on stereo or monocular vision mainly for daytime conditions. A limitation of the visual spectrum is encountered at night or in difficult weather conditions when stereo vision is not feasible. For night vision pedestrian detection the infrared sensors have been used successfully because they capture the heat emitted by objects. The infrared sensors can also be used at daytime and enhance the accuracy of stereo vision or monocular vision based approaches by sensor information fusion.

The pedestrian appearance in infrared images is different from the look of a pedestrian in the visual field but the challenges that must be faced by a pedestrian detector hold due to the high variety of appearance given by clothing, accessories, body part positions, viewing angle (front, lateral, rear) and actions (walk, stand, run) performed by pedestrians. Another difficulty for a pedestrian detector on infrared images is given by the fact that an infrared image looks different at summer and at winter. In winter the pedestrian face appears as being lighter while the body is insulated by warm clothes and it is darker. In autumn and spring when temperatures are bellow 25 degrees the pedestrian head and body is more nicely visible because the clothes are less insulating than in winter. Usually the cars and heated buildings are also clearly visible in those conditions, while the road, the sky or other cold objects appear to be darker in the IR image. In summer when temperatures are very high (above 30 degrees) the road and the sky are hotter and appears lighter than the buildings. Depending on the environment temperature the pedestrians might be darker than the background in hot summer days. To overcome the difference between appearances in summer and winter, a polarity inversion algorithm can applied on IR images taken in hot summer days such that the pedestrians are lighter than the environment.

This chapter presents approaches for detecting pedestrians in long wave infrared images refereed in what follows 'infrared' and abbreviated as IR. The appearance of a pedestrian in IR images is usually lighter that the appearance of the environment, but it is influenced by several factors like: different levels of clothing insulation, environmental temperature (an IR image looks different at summer and at winter), accessories, body part positions, viewing angle (front, lateral, rear) and actions (walk, stand, run) that a pedestrian may perform. Hence the detection of pedestrians in infrared images is a challenging task.

A solution proposed by the author is also explained. The described solution considers the above mentioned constraints and difficulties and it is two-fold:

- First it generates a region of interest (ROI) that comprises parts of the image that have a high probability of containing a pedestrian.
- Second it applies a pedestrian detector in the identified ROI.

The region of interest generator was designed based on the following considerations:

1. Pedestrians are vertical structures in the image hence vertical edge information is extracted and uniform areas such the sky or parts of road and buildings are removed. Some of the vegetation is elliminated and areas having a high density of connected vertical edges are enhanced.
2. In an automotive setup closer pedestrians have a large height opposed to far pedestrians that are smaller.In each region of the image the dimension of the scanning windows that must be retained by the region of interest generator is determined.
3. Usually in all environment conditions the pedestrian head and legs appear as light spots in the image. In order to make the whole body uniform a combination of morphological operations and adaptive thresholding to keep as much as possible of the pedestrian body in the region of interest is applied.

The methods explored by the author also comprise the analysis of four channel features for pedestrian detection in infrared images. These features are: the intensity channel also referred as infrared channel, histogram of gradient orientations (HOG), normalized gradient magnitude (MN) and local binary patterns (LBP).

A dataset of pedestrian models in long wave infrared images is created. The dataset is created from video sequences taken in autumn and winter and it comprises over 3000 pedestrian instances annotated from traffic scene images. The proposed framework is suitable for night vision because in far-infrared images pedestrians generally appear warmer than the background.

6.1 Survey of Current Approaches

A survey of approaches is given by [32], [199] , [200] and [201]. They provide details on infrared based technologies for pedestrian collision avoidance systems.

Based on these surveys a generic scheme valid for pedestrian detection in images from all types of sensors is identified. This scheme comprises the generation of pedestrian hypothesis candidates, extraction of relevant features and the actual classification.

For the generation of the regions of interest (ROI) existing methods combine the presence of hot spots with edge and texture information [202], [203]. A horizontal segmentation of the infrared image is presented in [204]. Another approach is employed by [205] that besides thresholding and edge detection perform ROI generation based on symmetry by computing the local direction of gradients. Edge symmetry information is combined with intensity profile analysis and road detection using infrared stereo in the work of [199]. A candidate generation method driven by the search of pedestrian head is employed by [206], [207]. By simple thresholding they find the blobs corresponding to the head and grayscale correlation is used for validation of the bounding boxes. Next they compute the symmetry of vertical edges on a bounding box approximated from a set of geometric constraints. Then an adaptive linear gain curve is applied to areas of the image with a high vertical symmetry in order to extract the silhouette of the warm objects detected. In the work of [208] the ROI is extracted based on discrete keypoints computed from the phase coherence image using the maximum and minimum moment of covariance.

When using a classification method that learns a discriminative function for the detection of pedestrians several visual features have been explored. Such features may be edgelets [209], multi-block binary patterns(MLBP) [210], local binary patterns(LBP) and their variations [211], histogram of oriented gradients (HOG) [212], [213], combination of HOG with contour based features [214]. Other methods exploit the benefits of scale invariant descriptors like discrete keypoints [208], SURF [215], or codebook dictionaries build upon SURF descriptors [216] that use reciprocal nearest neighbor (RNN) search to group similar SURF features into a tree like code book. Their work is extended by [217] that add several fast computing global features that improve the discriminative power representation. Intensity Self Similarity (ISS), adapted to pedestrian detection in far IR images is proposed by [218]. The ISS encodes the distribution of color as repetition across the image. Contrast invariant features named HOPE are proposed by [219] that exploit the local information histogram of orientations of phase coherence. An evaluation of several combinations of features and classifiers is done in [220]. They include features like Principal Component Analysis (PCA), LBP, HOG and HOPE.

For the actual pedestrian detection [203] use a shape correlation algorithm based on shaded 3-D pedestrian models. Correlation with precomputed deformable probabilistic models is also employed by [206], [207]. [204] use a template matching method that compare the similarity between multi-dimensional shape-independent feature vectors for previously generated ROIs and one generic pedestrian template. The shape independent feature vector comprises brightness histograms, image inertial, and contrast. Pattern correlation and trained pattern classification has been explored by [221] that combine (1) hierarchical contour matching based on distance transform; (2) Haar based cascade classification and (3) several hyper permutation networks that transform the original input image into the per pixel likelihood image. Support vector machines are employed by [212] that frame the SVM detector in a complete system, which deals with stereo infrared images. SVM is also employed by [213], [216], [217], [214] and [208], [219]. The use of Dempster-Shafer theory (DST) to combine in a finer way the outputs of SVM classifiers is presented by [222]. An adaptation of a latent variable SVM for far infrared images is proposed by [220]. Other classification methods explored by state of the art algorithms are artificial neural networks [223] and boosting [224]. An an Implicit Shape Model is build by [215] in order to describe the codebook for the pedestrian class. Classification is done by SURF feature matching that cast votes for object center

locations in a 3D Hough voting space. Probabilistic models are employed by [225]. They use four different models in order to recognize the pose of the pedestrians: open, almost open, almost closed and fully closed legs are detected.

6.2 Detection by Means of Feature Scaling

For the method that uses feature scaling the main steps are as follows:

- Region of interest generation that has the role of rapid identification of the areas in the infrared image that may contain pedestrians.
- Feature extraction – that computes HOG, LBP and gradient magnitude channels from a fixed predefined set of infrared image scales.
- Multiple scale image feature approximation – that performs a fast computation of channel features for intermediate scales. The intermediate scales are in the range of the fixed scales at the previous step.
- Classification using AdaBoost classifiers – does the actual detection.
- Non maximal suppression – removes multiple overlapped close detections.

The region of interest generator is described in the previous section.

6.2.1 Feature Extraction

Several feature configurations in order to find a best setup for pedestrian detection in infrared images have been experimented.

Normalized gradient magnitude features computed as described by [31] and [86] are computed as:

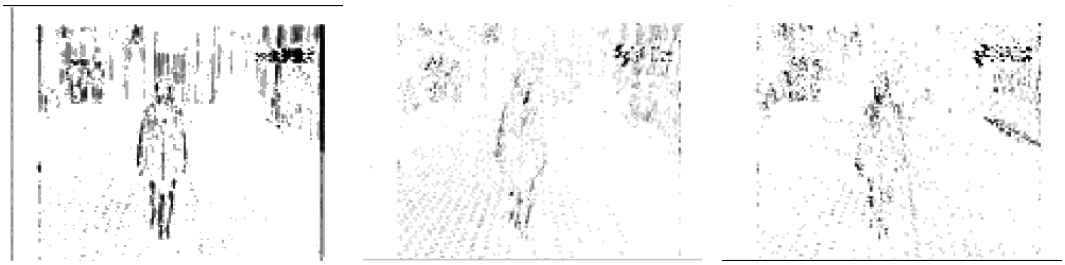
$$\widetilde{M}(i;j) = M(i;j)/(\overline{M}(i;j) + 0.005) \tag{6.1}$$

where $\overline{M}$ is the average gradient magnitude in each 11×11 image patch (computed by convolving M with an L_1 normalized 11×11 triangle filter).

Next local gradient histograms on image blocks of dimension 4×4 pixels are computed. For each pixel of an image I the gradient magnitude and the orientation are computed. Next the gradient histogram for an image is extracted. For the histogram each pixel provides a vote that is weighted by its gradient magnitude. The bin of each pixel corresponds to its gradient orientation. The approach in [86] is used and the image is divided into blocks of 4×4 pixels and then in each block the gradient histogram is computed. A number of bins equal to 6 is used. The votes for three of the bins are presented in sub figures 6.1(d), 6.1(e), 6.1(f). The other votes are not shown because they are not so visible in the given context of an infrared image.

Another employed feature is the local binary pattern [155]. For each point of an image I the LBP operator generates a binary code considering a threshold-ed difference of intensity values between the pixel and some points in its local neighborhood. The threshold value is

(a) Original Image (b) Normalized Gradient Magnitude (c) LBP applied on ROI



(d) HOG – 1 (e) HOG – 2 (f) HOG – 3

Figure 6.1: Features used

zero. The number of features extracted by the LBP operator can be reduced by using the so called uniform patterns [159]. These patterns are used to reduce the length of the feature vector and also implement a simple rotation-invariant descriptor. A local binary pattern is called uniform if the binary pattern contains at most two bit wise transitions from 0 to 1 or vice verse when the bit pattern is traversed circularly [160].

Experiments with several versions of LBP include:

- LBP59 – the LBP using uniform patterns with at most two transitions of 0 and 1s and this resulted in 59 different labels;
- LBP37 – the uniform 59 LBP was cyclically rotated such that the obtained binary number is minimum. There are 37 patterns obtained by this cyclic rotation.
- LBP256 – the classical LBP pattern.

The used features are depicted in Figure 6.1.

Multiple Scale Image Feature Approximation

This step is extremely important for ensuring a fast execution of the overall detection process. The classical methodology of a pedestrian detector is based on computing the features for one scale of the image, than slide the detection window and mark positive answers. Next, scale the image and recompute features and repeat the sliding process. Image scaling, feature computation and sliding window detection for a large number of scales is time consuming. In the proposed method the idea presented in [31] and [86] with the so called Aggregated Feature Channels [87] is used. The method uses 27 scales. The four mentioned features

are computed for the predefined scales equal to 1, 0.5, 0.25, 0.125, while the values of the features for 7 intermediate scales are approximated from neighboring scales. Exact feature scaling computation and approximation methodology is detailed in [86]. The LBP features were integrated in the framework of [87]

Shortly the main steps are:

- Given an input image I, compute several channels $C = \Omega(I)$, sum every block of 4×4 pixels in C, and smooth the resulting lower resolution channels.
- Instead of computing the features for each scale the ACF method computes I_s and $C_s = \Omega(I_s)$ for only a sparse set of s (once per octave).
- At intermediate scales C_s is computed by approximation.
- The pyramid obtained by computed and approximated features is scanned with a window of dimension 32×64

Nb. Ap.	Scale Factors computed ; approximated
0	1 0.91 0.84 0.77 0.70 0.65 0.59 0.54 0.50 0.45 0.41 0.38 0.35 0.32 0.30 0.27 0.25
2	1 0.91 0.84 0.77 0.70 0.65 0.59 0.54 0.50 0.45 0.41 0.38 0.35 0.32 0.30 0.27 0.25
4	1 0.91 0.84 0.77 0.70 0.65 0.59 0.54 0.50 0.45 0.41 0.38 0.35 0.32 0.30 0.27 0.25
7	1 0.91 0.84 0.77 0.70 0.65 0.59 0.54 0.5 0.45 0.41 0.38 0.35 0.32 0.30 0.27 0.25

Table 6.1: Examples of scale approximation factors

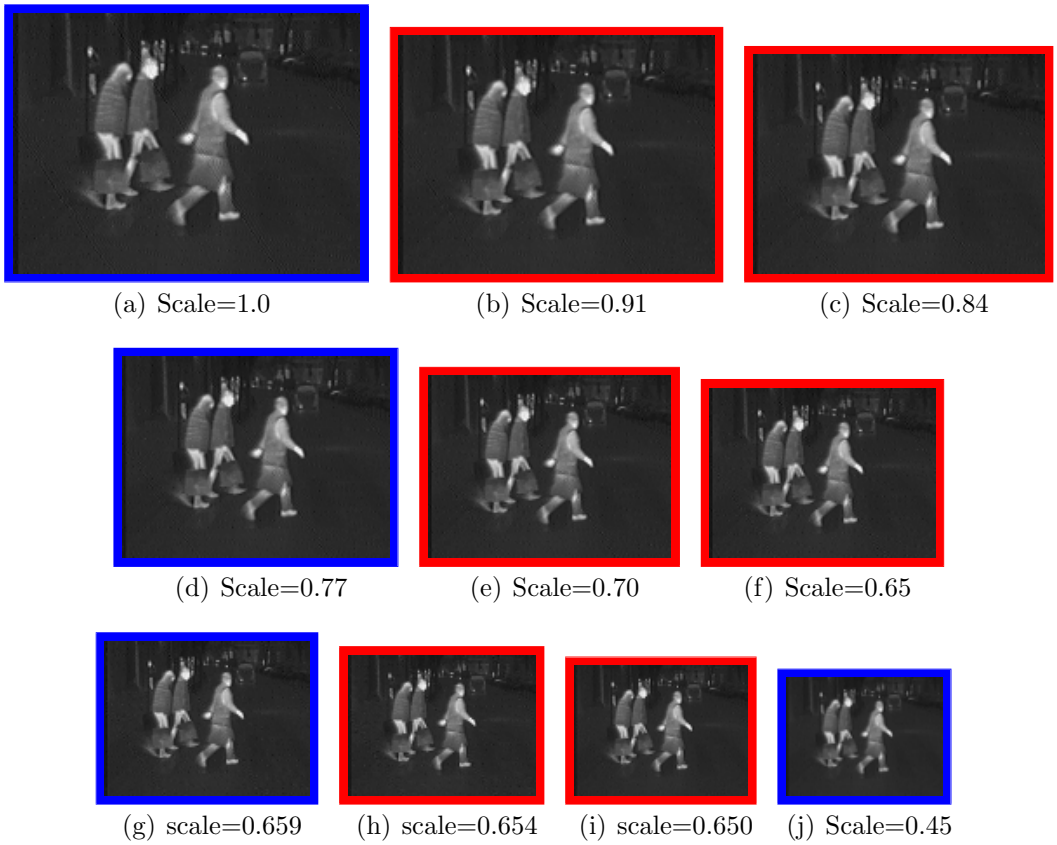
Table 6.1 shows with blue the scale factors that are computed and in red the scale factors that are approximated.

6.2.2 Classification Using AdaBoost

For classification a cascade of AdaBoost classifiers is used [87]. Their classification score is a linear combination of weighted weak learner responses.A cascade of such composite ensemble is used. The cascade has four stages and each stage has the same positive training set, while the negatives for each stage are the false positives of the previous stage. Each weak learner is a decision tree. The number of weak classifiers in the stages is $256, 512, 1024, 2048$.

6.3 Pedestrian Detection in IR With Multiple Scale Boosted Cascades

Another approach for the classification task is to train eight boosted cascades, each working with a specific size of training images. Hence the so called approach of one image scale and multiple detector sizes is adopted. For each size an aspect ratio of 0.5 is used. Each cascade is trained until it reaches specific false positive and true positive rates. Negatives' bootstrapping is applied for each stage of the eight cascades.



(a) Scale=1.0 (b) Scale=0.91 (c) Scale=0.84

(d) Scale=0.77 (e) Scale=0.70 (f) Scale=0.65

(g) scale=0.659 (h) scale=0.654 (i) scale=0.650 (j) Scale=0.45

Figure 6.2: Blue border – features are computed, Red border – features are approximated

By the sizes of the models the entire height dimension of infrared pedestrians captured by the IR sensor is covered and for this method we work with images having a resolution of 320×240 pixels in order to obtain a speed-up improvement.

6.4 Cascade of AdaBoost Classifiers

As classification method a cascade of boosted classifiers is employed. This concept was introduced by [154]. Each boosted classifier is a threshold-ed linear combination of a variable number of weak learners that are decision stumps. As input the number of stages (i.e. the number of boosted classifiers) and a target false positive rate and a target detection rate are provided. Each stage in the cascade reduces the false positive rate and decreases the detection rate. Each stage is trained by adding weak learners until the target detection and false positive rates are met.

The classical approach with scan window pedestrian detection is to scan the image with a window of constant size for which a classification model was build, and mark all positive detections. Then the image is scaled up and down with a certain number of factors and the scan window procedure is repeated. This process image pyramid scan window is extremely slow.

Instead of the image pyramid the model of [30] is adopted, that instead of rescaling the image learned models at few different scales.

Eight different scales are proposed for which models are trained. At each scale an aspect ratio of 0.5 is used. This ratio results from the analysis of more than 3000 annotated pedestrians in real traffic scenes.

The eight window sizes are: 24×48, 36×72, 48×96, 60×120, 72×144, 84×168, 86×192, 108×216. Each of the eight models is a cascade of AdaBoost classifiers that are trained until each stage in the cascade has a true positive rate of 0.99 and a false positive rate of 0.01.

A diagram of our cascade model is shown in the figure 6.3.

Evaluation

Evaluation of pedestrian detection using feature scaling

The Aggregated Channel Feature method uses 7 approximation scales per octave and in total there are 8 scales per octave.The number of approximated scales per octave was varied and the impact on accuracy and execution time was studied. First no approximations were adopted. This means the features are computed for every scale (27 scales in total). For comparison we have considered the following set of number of approximated scales per octave: $0, 2, 4, 7$. This was done for the ACF method that uses HOG, Normalized gradient and IR channels, and for the ACF method in combination with LBP256, LBP59 and LBP37. The performance of the detection was evaluated using full image evaluation as described by [36]. The miss rate against the percent of false positives per image using the evaluation toolbox provided by [87] is measured

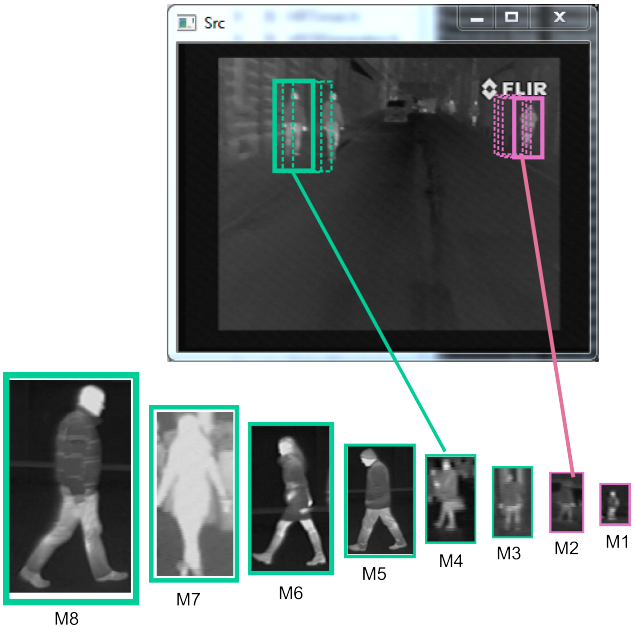


Figure 6.3: The scan windows for an input image are evaluated by the models having similar sizes in the multiple scale cascade

Table 6.2 shows the frames per second rate (fps) and the log average miss rate (lamr) for different channels and for several approximated scales per octave. Several configurations for LBP features in combination with the classical channels like: HOG, normalized gradient magnitude (MN)and LUV image were considered. Instead of the LUV color space the infrared image (IR) is used.

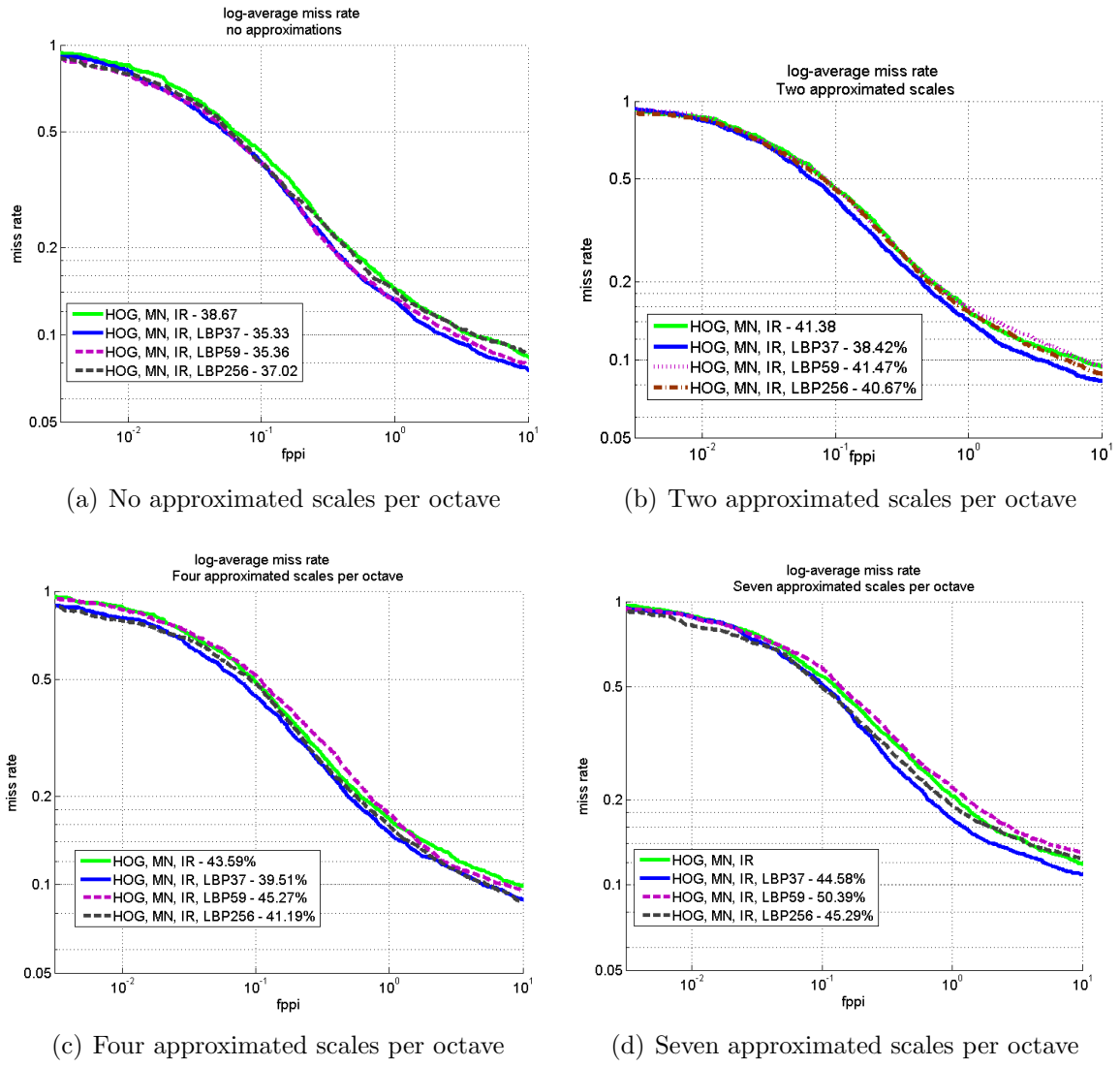
	Approximated scales per octave							
	0		2		4		7	
Channels	fps	lamr	fps	lamr	fps	lamr	fps	lamr
HOG MN IR LBP256	4.3	37.10%	10	40.67%	13	41.19%	16	45.29%
HOG MN IR LBP59	6	35.36%	11	41.47%	16	45.27%	18	50.39%
HOG MN IR LBP37	6	35.33%	13	38.42%	18	39.51%	22	44.58%
HOG MN IR	16	38.67%	24	41.38%	37	43.59%	38	48.95%

Table 6.2: Comparison of execution time and log average miss rate for different methods

One can notice that in all types of channel combinations the best performance is achieved when no feature channel approximation is done. Yet, for those cases the execution time is high. If the number of approximated scales is increased then the execution time decreases in disadvantage of a lower accuracy.

From the comparative analysis it can be noticed that the LBP features bring an improvement in terms of accuracy but lower the execution time. A reasonable setup, as shown in

table 6.2 would be the combination of HOG, MN, IR and LBP37 and their approximation for a number of 4 scales per octave. This leads to an accuracy of 39.5% that is close to the best accuracy obtained without feature approximation for HOG, MN and IR, that is 38.87%, having lower execution time of about 18fps. The log average miss rate curves are presented in Figure 6.4.



(a) No approximated scales per octave

(b) Two approximated scales per octave

(c) Four approximated scales per octave

(d) Seven approximated scales per octave

Figure 6.4: Log Average Miss Rate for different approximated scales per octave for infrared images

Sample detection results are presented in Figure 6.5.

Execution time analysis

The code was tested on an i7–3770K CPU machine. All the experiments are performed on images having the size 640 × 480 pixels. A hybrid implementation that combines C++ and Matlab is used For the region of interest generator the execution time for processing one

Figure 6.5: Pedestrian detection in IR images with feature scaling approach

Table 6.3: Per window evaluation

Model Size	TP rate	TN rate
24×48	60%	99%
36×72	96.5%	96.2%
48×96	99.2%	83%
60×120	99.5%	85.1%
72×144	97.2%	87%
84×168	98.76%	89.4%
86×192	93.2%	92.3%
108×216	92.8%	91.3%

frame is about 5 ms for a frame and the total execution time of the detector depends on the type of features used and on the number of approximated scales. As shown in Table 6.2 depending on the number of approximated scales per octave the total execution time ranges from 38fps – when seven approximation scales are used to 4.3 fps when no approximation is used and LBP256 is employed.

Evaluation of pedestrian detection using multiple scale boosted cascades

The performance of the detection was evaluated using two different measures proposed by [36].

- Per window evaluation that is we measure the performance on cropped positive and negative image windows.

- Full image evaluation in which we find the miss rate against false positives per image.

For each of the eight cascades a per window evaluation is applied, in order to find out the accuracy of each cascade in part on pedestrian and non-pedestrian images that have been resized in order to fit the detector size. The results of the per window evaluation are provided in table 6.3. TP rate refers to the true positive rate that is how many of the positives are correctly classified. TN rate refers to the true negative rate that is how many of the negatives are correctly classified as being negatives. Each classifier behaves relatively good when dealing to perfectly scaled small images.

In order to perform the per image evaluation test sequences are considered and for each the average number of false positive detections and the average true positive rate are measured. The true positive rate of a frame is given by the number of correctly detected pedestrian divided by the total number of pedestrians in that frame. The average true positive rate for a sequence is given by the sum of true positive rates per frames divided by the total number of frames that contain pedestrians. In order to prove the effectiveness of the multiple scale cascade structure its performance is compared with each of the eight cascades. The results are shown in table 6.4.

An evaluation of 1000 annotated frames of the test sequences is carried out. Only pedestrians having height greater than 30 pixels were considered.

Table 6.4: Per image evaluation

	$\epsilon = 0.3; th = 1$		$\epsilon = 0.2; th = 3$	
Model Size	Average TP rate	Average FP per window	Average TP rate	Average FP per window
24×48	58%	3.15%	58%	3.15%
36×72	75%	1.94%	59%	0.28%%
48×96	49%	18.73%	73%	4.4%
60×120	30%	5.38%	72%	46.3%
72×144	30%	2.39%	29%	42%
84×168	34.2%	1.75%	10.8%	39.2%
86×192	38.3%	1.706	12.8%	38.6%
108×216	22.5%	1.69%	12.4%	37.7%
Multi-scale:	**67%**	**0.49%**	34%	0.75%

In order to obtain these values a grouping of close enough bounding boxes based on their location and score was performed. All the input rectangles were clustered using the rectangle equivalence criteria based on similarity in size location [181].

The similarity is defined by the parameter ϵ that gives the relative difference between sides of the rectangles to be merged into a group. Then, the small clusters containing less than or equal to th rectangles are rejected. In each other cluster, the average rectangle is computed and put into the output rectangle list.

Table 6.4 shows the TP rate and the FP rate for two different parameter settings. When $\epsilon = 0.3$ and the grouping threshold is 1 the multi-scale cascade has a true positive rate of almost 70% and a false positive rate of 0.49, while the stand-alone cascades perform worse.

It can be noticed that the proposed detector works better than the classical HOG detector evaluated on the intensity images from INRIA dataset by [36].

Some examples of detection results are shown in Figure 6.6.

The sample detection results show that small pedestrians are not detected because a constraint of height being greater than 30 pixels was imposed, and in the detection results a false positive can be also noticed.

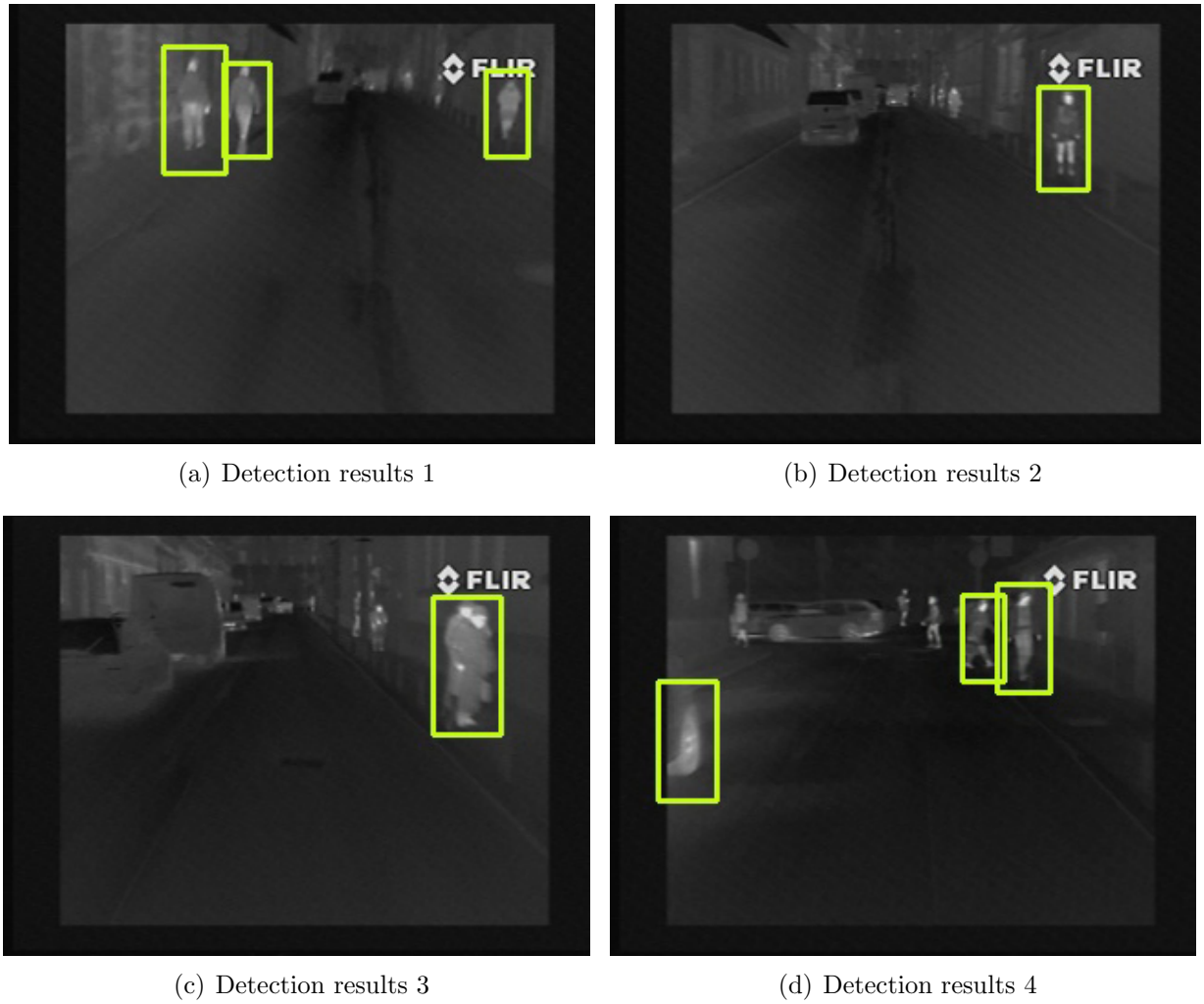
Execution time analysis

Our experiments have been run on an i7–3770K CPU. All the images used have a dimension of 320×240 pixels. For the region of interest generator the execution time for processing one frame is about 29 ms and this leads to an average execution time of 0.5fps.

Conclusion

This chapter presents two original solutions for detecting pedestrians in far infrared images: (1) detection by means of image feature approximation and (2) detection by multiple scale boosted cascades.

A popular and successful approach for monocular intensity pedestrian detection is based on the approximation (instead of computation) of image features for multiple scales based on



(a) Detection results 1

(b) Detection results 2

(c) Detection results 3

(d) Detection results 4

Figure 6.6: Pedestrian detection in IR images with multi-scale cascades

the features computed on set of predefined scales. This idea is ported to the infrared domain. The contributions of the author reside in the combination of four channel features, namely infrared, histogram of gradient orientations, normalized gradient magnitude and local binary patterns with the objective of detecting pedestrians for night vision applications dealing with far infrared sensors. Multiple scale feature computation is done by feature approximation. Another contribution is the study of different formulations for Local Binary Patterns like uniform patterns and rotation invariant patterns and their effect on detection performance. The detection speed is also boosted by the aid of a fast morphological based region of interest generator. A reasonable result hits a speed of 18fps with a log average miss rate of 39%.

Another proposed approach for the classification task is to train eight boosted cascades, each working with a specific size of training images. Hence the so called approach of one image scale and multiple detector sizes is followed. For each size an aspect ratio of 0.5 is used. Each cascade is trained until it reaches specific false positive and true positive rates. Negatives' bootstrapping is applied for each stage of the eight cascades. By the sizes of the models the entire height dimension of infrared pedestrians captured by the IR sensor is

covered. All the images have a resolution of 320×240 pixels. An execution time of about 20fps and a log average miss rate of 33% is obtained.

Chapter 7

Conclusions

This book addresses the problem of time-constrained pedestrian detection in monocular intensity and far infrared images. The described approaches involve a comprehensive study of the various modules of a pedestrian detector, namely the pedestrian representation models, the feature extraction process and the actual pattern classification.

Chapter 1 presents the motivation and the main challenges of generic pedestrian detection models for monocular images. A main motivation for developing pedestrian detection systems in the context of road environments is given by the need of active safety technologies that assist autonomous driving systems in the prevention of pedestrian collision. The objective of such active safety technologies is to minimize the occurrence and consequences of automobile accidents. The challenges that a pedestrian detection system must face are given by the large variety of appearances due to the actions they perform (walk, run, stand), due to the motion of different body parts, due to the clothing and accessories they wear. Hence pedestrians posses a large intra-class variability because they are highly deformable instances in a traffic scene and their appearance depends on numerous factors like: pose, orientation, shape, attitude, occlusions, imaging conditions, background.

Chapter 2 presents the mathematical models employed by various existing solutions. Details upon visual descriptors like first order partial derivatives, histogram of gradient orientations, Haar filters, local binary patterns, anisotropic Gaussians and Gabor wavelets are provided. The correlation based feature selection method that is employed for selecting relevant features is included.

Chapter 3 provides a description of the machine learning algorithms used in state of the art solutions for pedestrian detection. Insights on Bayesian networks, boosting, multiple layer perceptrons and support vector machines are provided.

Chapter 4 presents popular datasets used in assessing the performance of pedestrian detectors. The standard evaluation protocol used for assessing the performance of pedestrian detection methods is also detailed.

Chapter 5 discuses the difficulties of existing scan window based pedestrian detectors in monocular visible images and presents our several ways for tackling this problem.

A revision of techniques for monocular based pedestrian detection is deployed.

A summary of existing pedestrian representation models is given. The representation models refer to the pedestrian appearance captured by the classification model. The main identified categories are: (1) monolithic representations consider the pedestrian data as a

whole and (2) part based representations that regard the pedestrian as a combination of parts and the whole body. For each of the two representations single scale or multiple-scale methods have been developed. These representations try to capture the high variance of the pedestrian appearance and some of them underline the multiple views or poses that pedestrians may have.

Next the contributions of the author in this field are presented. An approach that considers several pedestrian attitudes specific for traffic environment is defined. Actions like pedestrian running, pedestrian standing and walking are identified as being representative in the process. A space of semantic concepts that are formed by adding direction information to the attitudes is described: for example walk left, right, pedestrian facing forward or backward. Several classification algorithms have been employed for learning different semantic concepts. Those classifiers are combined in a single reasoning module that called meta-classifier that comprises: (1) basic attitude meta-classifier that is trained on a coarse partition of the input space. The division comprises three main attitudes: stand, run, walk. (2)complex attitude meta-classifier that is trained on a fine partition of the input space. A segmentation based on semantic concepts that comprise a combination between the actions that pedestrians perform: stand, run, walk and the direction of movement front, back, lateral left, lateral right is proposed.

The complex meta-classifier is integrated in a stereo-vision system and also in a monocular system. Based on the motion information provided by the stereo-system the orientation of each pedestrian hypothesis is extracted. Using the orientation only three instead of nine classifiers are applied once. For example if the orientation information corresponds to front motion than only front walk, front run and front stand classifiers are applied. Given this context information the complex meta-classifier improved the overall pedestrian detection accuracy with about 8% leading to an overall true positive rate of about 90% with only a small decrease in execution time of the overall system.

Thirdly, a star based meta-classifier that combines a whole body (root) classifier with part based the multi-attitude models is described. Models for different pedestrian attitudes given by orientations like lateral left, lateral right, front, rear are proposed. The root classifier trained on all attitudes. The role of the root classifier is to identify pedestrians fast with the cost of admitting more false positives. The attitude classifiers come to refine the root decision and to eliminate the false detections and refine the positive detections. The root and attitude classifiers are trained on HOG and LBP features computed for different parts of interest defined based on an edge homogeneity minimization function. The log average miss rate of the star classifier on pedestrians from Daimler dataset having a height greater than 50 pixels and partially occluded is of about 45% For pedestrians having a height greater than 100 pixels and not occluded the log average miss rate is of about 20%. That is the star classifier detects correctly about 80% of the pedestrians that are not occluded and are closer to the camera. The star meta-classifier hits an execution time of 16 fps.

Chapter 6 outlines existing feature based classification approaches for detecting pedestrians in infrared images. The contributions of the author in this direction are underlined by two categories of methods: (1) detection by means of image feature approximation and (2) detection by multiple scale boosted cascades. Insights on the proposal of a single detection model applied at multiple scales are given. A pyramidal model with approximation of features values for different scales is used.

The formalisation and description of methods that comprise multiple size models applied at one image scale are also included. Several detection models of different dimensions are trained and applied in parallel without any image scaling. Hence the so called approach of one image scale and multiple detector sizes is achieved. For each size an aspect ratio of 0.5 is used. Each cascade is trained until it reaches specific false positive and true positive rates. Negatives' bootstrapping is applied for each stage of the eight cascades. By the sizes of the models the entire height dimension of infrared pedestrians captured by the IR sensor is being covered. For this method the images have a resolution of 320×240 pixels. An execution time of about 20fps and a log average miss rate of 33% is obtained.

References

[1] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.

[2] F. Miau, C. Papageorgiou, and L. Itti, "Neuromorphic algorithms for computer vision and attention," in *Proc. SPIE 46 Annual International Symposium on Optical Science and Technology*, B. Bosacchi, D. B. Fogel, and J. C. Bezdek, Eds., vol. 4479. Bellingham, WA: SPIE Press, Nov 2001, bu;mod;cv, pp. 12–23.

[3] D. Cheda, D. Ponsa, and A. Lopez, "Pedestrian candidates generation using monocular cues," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 7–12.

[4] M. Pedersoli, J. Gonzalez, X. Hu, and X. Roca, "Toward real-time pedestrian detection based on a deformable template model," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 1, pp. 355–364, Feb 2014.

[5] W. Nam, B. Han, and J. H. Han, "Macrofeature layout selection for pedestrian localization and its acceleration using gpu," *Computer Vision and Image Understanding*, vol. 120, pp. 46–58, 2014.

[6] B. Wu, R. Nevatia, and Y. Li, "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*, June 2008, pp. 1–8.

[7] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 55–79, Jan. 2005. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000042934.15159.49

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[9] L. Ladicky, P. H. S. Torr, and A. Zisserman, "Latent svms for human detection with a locally affine deformation field," in *British Machine Vision Conference*, 2012.

[10] Z. Lin and L. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 4, pp. 604–618, April 2010.

[11] B. Li, Y. Chen, and F. Wang, "Pedestrian detection based on clustered poselet models and hierarchical and-or grammar," *Vehicular Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[12] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, Oct 2005, pp. 90–97 Vol. 1.

[13] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.

[14] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 10, pp. 1713–1727, Oct 2008.

[15] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *BMVC*, 2009.

[16] S. Zhang, C. Bauckhageyz, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in *Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'14)*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014.

[17] Y. Ding and J. Xiao, "Contextual boost for pedestrian detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 2895–2902.

[18] D. Gavrila, "A bayesian, exemplar-based approach to hierarchical shape matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 8, pp. 1408–1421, Aug 2007.

[19] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 1022–1029.

[20] O. Barinova, V. Lempitsky, and P. Kohli, "On detection of multiple object instances using hough transforms," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 2233–2240.

[21] C.-K. Heng, S. Yokomitsu, Y. Matsumoto, and H. Tamura, "Shrink boost for selecting multi-lbp histogram features in object detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3250–3257.

[22] Q. Ye, J. Liang, and J. Jiao, "Pedestrian detection in video images via error correcting output code classification of manifold subclasses," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 193–202, March 2012.

[23] M. Enzweiler and D. M. Gavrila, "A multilevel mixture-of-experts framework for pedestrian classification," *Image Processing, IEEE Transactions on*, vol. 20, no. 10, pp. 2967–2979, 2011.

[24] W. Ouyang and X. Wang, "Single-pedestrian detection aided by multi-pedestrian detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3198–3205.

[25] G. Ma, S.-B. Park, A. Ioffe, S. Muller-Schneiders, and A. Kummert, "A real time object detection approach applied to reliable pedestrian detection," in *Intelligent Vehicles Symposium, 2007 IEEE*, June 2007, pp. 755–760.

[26] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila, "Multi-cue pedestrian classification with partial occlusion handling," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. IEEE, 2010, pp. 990–997.

[27] W. Ouyang, X. Zeng, and X. Wang, "Modeling mutual visibility relationship in pedestrian detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3222–3229.

[28] R. Girshick, P. Felzenszwalb, and D. McAllester, "Object detection with grammar models," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2011.

[29] S. Rujikietgumjorn and R. Collins, "Optimized pedestrian detection for multiple and occluded people," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3690–3697.

[30] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," in *CVPR*, 2012.

[31] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *BMVC*, 2010.

[32] T. Gandhi and M. Trivedi, "Pedestrian collision avoidance systems: a survey of computer vision based recent studies," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, Sept 2006, pp. 976–981.

[33] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.

[34] D. Gerónimo, A. López, and A. D. Sappa, "Computer vision approaches to pedestrian detection: Visible spectrum survey," in *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis, Part I*, ser. IbPRIA '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 547–554. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72847-4_70

[35] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi, "Shape-based pedestrian detection and localization," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 1, 2003, pp. 328–333 vol.1.

[36] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[37] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 4, pp. 349–361, 2001.

[38] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 734–741 vol.2.

[39] R. Benenson, M. Omran, J. Hosang, , and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *ECCV, CVRSUAD workshop*, 2014.

[40] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" in *BMVC*, 2014.

[41] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 10, pp. 1683–1698, Oct 2008.

[42] K. Rematas and B. Leibe, "Efficient object detection and segmentation with a cascaded hough forest ism," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 966–973.

[43] S. Agarwal, A. Awan, D. Roth, and I. C. Society, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, p. 2004, 2004.

[44] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005, pp. 878–885 vol. 1.

[45] E. Seemann, M. Fritz, and B. Schiele, "Towards robust pedestrian detection in crowded image sequences," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.

[46] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, 2008.

[47] A. Costea and S. Nedevschi, "Multi-class segmentation for traffic scenarios at over 50 fps," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, June 2014, pp. 1390–1395.

[48] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998. [Online]. Available: http://dx.doi.org/10.1109/34.730558

[49] S. Chikkerur, T. Serre, C. Tan, and T. Poggio, "What and where: A bayesian inference theory of attention," *Vision Research*, vol. 50, no. 22, pp. 2233 – 2247, 2010, mathematical Models of Visual Coding. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698910002348

[50] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, "Salient object detection by composition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1028–1035.

[51] M. Enzweiler, P. Kanter, and D. Gavrila, "Monocular pedestrian recognition using motion parallax," in *Intelligent Vehicles Symposium, 2008 IEEE*, June 2008, pp. 792–797.

[52] S. Zhang, C. Bauckhage, D. Klein, and A. Cremers, "Moving pedestrian detection based on motion segmentation," in *Robot Vision (WORV), 2013 IEEE Workshop on*, Jan 2013, pp. 102–107.

[53] P.-H. Lee, Y.-L. Lin, S.-C. Chen, C.-H. Wu, C.-C. Tsai, and Y.-P. Hung, "Viewpoint-independent object detection based on two-dimensional contours and three-dimensional sizes," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1599–1608, Dec 2011.

[54] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.rti.2004.12.004

[55] H. Kim, Y. Shibayama, and S. Kamijo, "Acquisition of pedestrian trajectory using on-board monocular camera," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 544–549.

[56] S. Kamijo, K. Fujimura, and Y. Shibayama, "Pedestrian detection algorithm for on-board cameras of multi view angles," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 973–980.

[57] S. Álvarez, M. Á. Sotelo, I. Parra, D. F. Llorca, and M. Gavilán, "Vehicle and Pedestrian detection in eSafety Applications," in *WCECS ICIAR09*, oct. 2009.

[58] S. Alvarez, D. Llorca, M. A. Sotelo, and A. G. Lorente, "Monocular target detection on transport infrastructures with dynamic and variable environments," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 61–66.

[59] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 9, pp. 1820–1833, Sept 2011.

[60] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution models for object detection," in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 241–254. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888089.1888108

[61] P. Sudowe and B. Leibe, "Efficient use of geometric constraints for sliding-window object detection in video," in *Computer Vision Systems*, ser. Lecture Notes in Computer Science, J. Crowley, B. Draper, and M. Thonnat, Eds. Springer Berlin Heidelberg, 2011, vol. 6962, pp. 11–20.

[62] A. Prioletti, P. Grisleri, M. Trivedi, and A. Broggi, "Design and implementation of a high performance pedestrian detection," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 1398–1403.

[63] K. Yang, E. Du, P. Jiang, Y. Chen, R. Sherony, and H. Takahashi, "Automatic categorization-based multi-stage pedestrian detection," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 451–456.

[64] J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li, "Robust multiresolution pedestrian detection in traffic scenes," in *Proceedings of the 2013 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'13)*, ser. CVPR '13. Washington, DC, USA: IEEE Computer Society, 2013.

[65] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'14)*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014.

[66] O. Barinova, V. Lempitsky, and P. Kholi, "On detection of multiple object instances using hough transforms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1773–1784, Sept 2012.

[67] J. H. Joung, M. S. Ryoo, S. Choi, W. Yu, and H. Chae, "Background-aware pedestrian/vehicle detection system for driving environments," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 1331–1336.

[68] K. Goto, K. Kidono, Y. Kimura, and T. Naito, "Pedestrian detection and direction estimation by cascade detector with multi-classifiers utilizing feature interaction descriptor," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 224–229.

[69] H. Cho, P. Rybski, A. Bar-Hillel, and W. Zhang, "Real-time pedestrian detection with deformable part models," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 1035–1042.

[70] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014.

[71] P. Felzenszwalb, R. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 2241–2248.

[72] A. Mogelmose, A. Prioletti, M. Trivedi, A. Broggi, and T. Moeslund, "Two-stage part-based pedestrian detection," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 73–77.

[73] A. Prioletti, A. Mogelmose, P. Grisleri, M. Trivedi, A. Broggi, and T. Moeslund, "Part-based pedestrian detection and feature-based tracking for driver assistance: Real-time, robust algorithms, and evaluation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 3, pp. 1346–1359, Sept 2013.

[74] P. Geismann and G. Schneider, "A two-staged approach to vision-based pedestrian recognition using haar and hog features," in *Intelligent Vehicles Symposium, 2008 IEEE*, June 2008, pp. 554–559.

[75] L. Yu, W. Yao, H. Liu, and F. Liu, "A monocular vision based pedestrian detection system for intelligent vehicles," in *Intelligent Vehicles Symposium, 2008 IEEE*, June 2008, pp. 524–529.

[76] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'14)*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014.

[77] G. Chen, Y. Ding, J. Xiao, and T. Han, "Detection evolution with multi-order contextual co-occurrence," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1798–1805.

[78] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'14)*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014.

[79] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2189–2202, Nov 2012.

[80] G. Gualdi, A. Prati, and R. Cucchiara, "Multi-stage sampling with boosting cascades for pedestrian detection in images and videos," in *Proceedings of*

the 11th European Conference on Computer Vision: Part VI, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 196–209. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888212.1888229

[81] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *ECCV*, 2012.

[82] G. Gualdi, A. Prati, and R. Cucchiara, "Multistage particle windows for fast and accurate object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 8, pp. 1589–1604, Aug 2012.

[83] M. Pedersoli, J. Gonzlez, A. D. Bagdanov, and X. Roca, "Efficient discriminative multiresolution cascade for real-time human detection applications," *Pattern Recognition Letters*, vol. 32, no. 13, pp. 1581 – 1587, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865511001954

[84] M. Pedersoli, J. Gonzlez, A. Bagdanov, and J. Villanueva, "Recursive coarse-to-fine localization for fast object detection," in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin Heidelberg, 2010, vol. 6316, pp. 280–293. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15567-3_21

[85] M. Pedersoli, A. Vedaldi, and J. Gonzalez, "A coarse-to-fine approach for fast deformable object detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 1353–1360.

[86] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 8, pp. 1532–1545, Aug 2014.

[87] P. Dollár, "Piotr's Image and Video Matlab Toolbox (PMT)," http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

[88] C. H. Lampert, M. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[89] ——, "Efficient subwindow search: A branch and bound framework for object localization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 12, pp. 2129–2142, Dec 2009.

[90] A. Lehmann, B. Leibe, and L. Van Gool, "Feature-centric efficient subwindow search," in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 940–947.

[91] D. Llorca, M. Sotelo, A. Helln, A. Orellana, M. Gaviln, I. Daza, and A. Lorente, "Stereo regions-of-interest selection for pedestrian protection: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 25, no. 0, pp. 226 – 237, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X12000885

[92] D. Llorca, M. Sotelo, I. Parra, J. Naranjo, M. Gavilan, and S. Alvarez, "An experimental study on pitch compensation in pedestrian-protection systems for collision avoidance and mitigation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 469–474, 2009.

[93] S. Nedevschi, S. Bota, and C. Tomiuc, "Stereo-based pedestrian detection for collision-avoidance applications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 380–391, 2009.

[94] M. Enzweiler, M. Hummel, D. Pfeiffer, and U. Franke, "Efficient stixel-based object recognition," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 1066–1071.

[95] C. Keller, M. Enzweiler, M. Rohrbach, D. Fernandez Llorca, C. Schnorr, and D. Gavrila, "The benefits of dense stereo for pedestrian detection," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1096–1106, Dec 2011.

[96] C. Keller, D. Llorca, and D. Gavrila, "Dense stereo-based roi generation for pedestrian detection," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, J. Denzler, G. Notni, and H. Se, Eds. Springer Berlin Heidelberg, 2009, vol. 5748, pp. 81–90. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03798-6_9

[97] C. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. Gavrila, "Active pedestrian safety by automatic braking and evasive steering," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1292–1304, Dec 2011.

[98] T. Takahashi, H. Kim, and S. Kamijo, "Urban road user classification framework using local feature descriptors and hmm," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 67–72.

[99] M. Enzweiler and D. Gavrila, "Integrated pedestrian classification and orientation estimation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 982–989.

[100] A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg, "Part-based feature synthesis for human detection," vol. 6314, pp. 127–142, 2010. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15561-1_10

[101] A. Hillel, D. Weinshall, and T. Hertz, "Efficient learning of relational object class models," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, Oct 2005, pp. 1762–1769 Vol. 2.

[102] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, Oct 2005, pp. 370–377 Vol. 1.

[103] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *International*

Journal of Computer Vision, vol. 75, no. 2, pp. 247–266, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11263-006-0027-7

[104] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Springer Berlin Heidelberg, 2004, vol. 3021, pp. 69–82. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24670-1_6

[105] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[106] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3258–3265.

[107] J. Xu, D. Vazquez, A. Lopez, J. Marin, and D. Ponsa, "Learning a multiview part-based model in virtual world for pedestrian detection," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 467–472.

[108] R. Muhammad Anwer, D. Vzquez, and A. Lpez, "Color contribution to part-based person detection in different types of scenarios," in *Computer Analysis of Images and Patterns*, ser. Lecture Notes in Computer Science, P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano, and W. Kropatsch, Eds. Springer Berlin Heidelberg, 2011, vol. 6855, pp. 463–470. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23678-5_55

[109] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 1014–1021.

[110] P. Felzenszwalb, "Object detection grammars," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 691–691.

[111] D. Levi, S. Silberstein, and A. Bar-Hillel, "Fast multiple-part based object detection using kd-ferns," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 947–954.

[112] D. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision*, vol. 73, no. 1, pp. 41–59, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11263-006-9038-7

[113] S. Munder and D. Gavrila, "An experimental study on pedestrian classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 11, pp. 1863–1868, Nov 2006.

[114] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 794–801.

[115] F. Suard, A. Rakotomamonjy, and A. Bensrhair, "Model selection in pedestrian detection using multiple kernel learning," in *Intelligent Vehicles Symposium, 2007 IEEE*, June 2007, pp. 270–275.

[116] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *In CVPR*, 2005, pp. 886–893.

[117] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 1030–1037.

[118] A. Gepperth, M. Ortiz, and B. Heisele, "Real-time pedestrian detection and pose classification on a gpu," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 348–353.

[119] G. Overett, L. Petersson, L. Andersson, and N. Pettersson, "Boosting a heterogeneous pool of fast hog features for pedestrian and sign detection," in *Intelligent Vehicles Symposium, 2009 IEEE*, June 2009, pp. 584–590.

[120] Y.-F. Kao, Y.-M. Chan, L.-C. Fu, P.-Y. Hsiao, S.-S. Huang, C.-E. Wu, and M.-F. Luo, "Comparison of granules features for pedestrian detection," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 1777–1782.

[121] Y.-M. Chan, L.-C. Fu, P.-Y. Hsiao, and M.-F. Lo, "Pedestrian detection using histograms of oriented gradients of granule feature," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 1410–1415.

[122] M. Dikmen, D. Hoiem, and T. Huang, "A data driven method for feature transformation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3314–3321.

[123] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool, "Seeking the strongest rigid detector," in *CVPR*, 2013.

[124] R. Muhammad Anwer, D. Vzquez, and A. Lpez, "Opponent colors for human detection," in *Pattern Recognition and Image Analysis*, ser. Lecture Notes in Computer Science, J. Vitri, J. Sanches, and M. Hernndez, Eds. Springer Berlin Heidelberg, 2011, vol. 6669, pp. 363–370. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21257-4_45

[125] Y. Socarrs Salas, D. Vzquez Bermudez, A. Lpez Pea, D. Gernimo Gomez, and T. Gevers, "Improving hog with image segmentation: Application to human detection," in *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes

in Computer Science, J. Blanc-Talon, W. Philips, D. Popescu, P. Scheunders, and P. Zemk, Eds. Springer Berlin Heidelberg, 2012, vol. 7517, pp. 178–189. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33140-4_16

[126] S. Maji, A. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.

[127] B. Wu and R. Nevatia, "Simultaneous object detection and segmentation by boosting local shape feature based classifier," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.

[128] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on riemannian manifolds," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.

[129] C.-E. Wu, Y.-M. Chan, L.-C. Fu, P.-Y. Hsiao, S.-S. Huang, H.-H. Chen, P.-T. Huang, and S.-C. Hu, "Combining multiple complementary features for pedestrian and motorbike detection," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 1358–1363.

[130] M. Enzweiler and D. Gavrila, "A mixed generative-discriminative framework for pedestrian classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[131] G. D., S. A. D., L. A., and P. D., "Adaptive image sampling and windows classification for on-board pedestrian detection," in *The 5th International Conference on Computer Vision Systems*, 2007.

[132] D. Gernimo, A. D. Sappa, D. Ponsa, and A. M. Lpez, "2d3d-based on-board pedestrian detection system," *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 583 – 595, 2010, special issue on Intelligent Vision Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314210000330

[133] J. Marin, D. Vazquez, A. Lopez, J. Amores, and B. Leibe, "Random forests of local experts for pedestrian detection," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 2592–2599.

[134] A. González, S. Ramos, D. Vázquez, A. M. López, and J. Amores, "Spatiotemporal stacked sequential learning for pedestrian detection," *CoRR*, vol. abs/1407.3686, 2014. [Online]. Available: http://arxiv.org/abs/1407.3686

[135] M. Mathias, R. Benenson, R. Timofte, and L. Van Gool, "Handling occlusions with franken-classifiers," in *ICCV*, 2013.

[136] J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *CVPR*, 2013.

[137] A. D. Costea and S. Nedevschi, "Word channel based multiscale pedestrian detection without image resizing and using only one classifier," in *Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'14)*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014.

[138] W. Voravuthikunchai, B. Crémilleux, and F. Jurie, "Histograms of pattern sets for image classification and object recognition," in *Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'14)*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014.

[139] D. Gavrila, "Pedestrian detection from a moving vehicle," in *Computer Vision - ECCV 2000, 6th European Conference on Computer Vision, Dublin, Ireland, June 26 - July 1, 2000, Proceedings, Part II*, 2000, pp. 37–49. [Online]. Available: http://dx.doi.org/10.1007/3-540-45053-X_3

[140] J. Xu, S. Ramos, D. Vazquez, and A. Lopez, "Domain adaptation of deformable part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[141] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," pp. 3539–3546, June 2010.

[142] S. Maji, A. Berg, and J. Malik, "Efficient classification for additive kernel svms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 66–77, Jan 2013.

[143] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986. [Online]. Available: http://dx.doi.org/10.1007/BF00116251

[144] P. Kontschieder, S. Rota Bulo, M. Pelillo, and H. Bischof, "Structured labels in random forests for semantic labelling and object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[145] R. Appel, T. Fuchs, P. Dollár, and P. Perona, "Quickly boosting decision trees - pruning underachieving features early," in *ICML*, 2013.

[146] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3626–3633.

[147] M. J. Saberian and N. Vasconcelos, "Learning optimal embedded cascades," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 2005–2018, Oct 2012.

[148] L. Leyrit, T. Chateau, C. Tournayre, and J.-T. Lapreste, "Association of adaboost and kernel based machine learning methods for visual pedestrian recognition," in *Intelligent Vehicles Symposium, 2008 IEEE*, June 2008, pp. 67–72.

[149] P. Dollr, B. B. S. Belongie, and P. P. Z. Tu, "Multiple component learning for object detection," in *In Proc. of ECCV*, 2008.

[150] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, "Trainable classifier-fusion schemes: An application to pedestrian detection," in *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, Oct 2009, pp. 1–6.

[151] K. Ali and K. Saenko, "Confidence-rated multiple instance boosting for object detection," in *Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'14)*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014.

[152] J. Marín, D. Vázquez, A. M. López, J. Amores, and L. I. Kuncheva, "Occlusion handling via random subspace classifiers for human detection," *IEEE T. Cybernetics*, vol. 44, no. 3, pp. 342–354, 2014. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2013.2255271

[153] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *International Conference on Computer Vision*, 1998.

[154] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[155] M. Pietikäinen and T. Ojala, "Texture analysis in industrial applications," 1996.

[156] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037–2041, 2007.

[157] Y. Cao, S. Pranata, and H. Nishimura, "Local binary pattern features for pedestrian detection at night/dark environment," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. 2053–2056.

[158] Y. Zheng, C. Shen, R. Hartley, and X. Huang, "Pyramid center-symmetric local binary/trinary patterns for effective pedestrian detection," in *Computer Vision ACCV 2010*, ser. Lecture Notes in Computer Science, R. Kimmel, R. Klette, and A. Sugimoto, Eds. Springer Berlin Heidelberg, 2011, vol. 6495, pp. 281–292. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19282-1_23

[159] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[160] M. Pietikinen, "Local binary patterns," *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.

[161] L. Peotta, L. Granai, and P. Vandergheynst, "Very low bit rate image coding using redundant dictionaries," in *Proceedings of the SPIE, Wavelets: Applications in Signal and Image Processing X*, ser. Lecture Notes in Computer Science, vol. 5207. SPIE, 2003, pp. 228–239.

[162] D. Gabor, "Theory of communication," in *J. IEE (London)*, vol. 93, no. 26, 1946, pp. 429–457.

[163] J. Daugman, "Uncertainty relations for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," in *Journal of the Optical Society of America A*, vol. 2, 1895, pp. 1160–1169.

[164] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000, pp. 359–366.

[165] I. H. Witten and E. Frank, *Data Mining Practical Machine Learning Tools and Techniques*, 2005.

[166] "WEKA : Data mining software in java." [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/

[167] M. Dimitrijevic, V. Lepetit, and P. Fua, "Human body pose detection using bayesian spatio-temporal templates," in *Computer Vision and Image Understanding*, vol. 104, November 2006, pp. 127–139.

[168] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part," pp. 247–266, November 2007.

[169] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley, 2001.

[170] R. Schapire, "The boosting approach to machine learning: An overview," March 2001. [Online]. Available: http://www.research.att.com/~schapire/boost.html

[171] J. Meynet, "Fast face detection using adaboost," Master's thesis, Ecole Polytechnique Fédérale de Lausanne, 2003.

[172] K. P. Murphy, *Machine learning: a probabilistic perspective*, Cambridge, MA, 2012.

[173] C. Keller, M. Enzweiler, and D. Gavrila, "A new benchmark for stereo-based pedestrian detection," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 691–696.

[174] A. Ess, B. Leibe, K. Schindler, , and L. van Gool, "A mobile vision system for robust multi-person tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE Press, June 2008.

[175] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, June 2009.

[176] G. Overett, L. Petersson, N. Brewer, L. Andersson, and N. Pettersson, "A new pedestrian dataset for supervised learning," in *Intelligent Vehicles Symposium, 2008 IEEE*, June 2008, pp. 373–378.

[177] "Center for biological and computational learning at MIT: Pedestrian database." [Online]. Available: http://cbcl.mit.edu/software-datasets/PedestrianData.html

[178] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *CVPR*, 1997, pp. 193–99.

[179] C. Wojek, S. Walk, S. Roth, and B. Schiele, "Monocular 3d scene understanding with explicit occlusion reasoning," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 1993–2000.

[180] R. Brehar and S. Nedevschi, "Scan window based pedestrian recognition methods improvement by search space and scale reduction," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, no. 529–534. IEEE, Jun 2014.

[181] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[182] S. Álvarez, M. Sotelo, I. Parra, D. F. Llorca, and M. Gavilán, "Vehicle and pedestrian detection in esafety applications," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, October 2009.

[183] R. Borca-Muresan and S. Nedevschi, "Meta-classifier for pedestrian attitude recognition," in *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, Aug 2008, pp. 33–40.

[184] "INRIA pedestrian dataset." [Online]. Available: http://pascal.inrialpes.fr/data/human/

[185] R. Borca-Mureşan, S. Nedevschi, and F. Măguran, *Mixtures of Classifiers for Recognizing Standing and Running Pedestrians*, ser. Computer Vision and Graphics, L. Bolc, J. L. Kulikowski, and K. Wojciechowski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5337, no. 345–355.

[186] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, and T. Graf, "Stereovision approach for obstacle detection on non-planar roads," in *IEEE and IFAC International Conference on Informatics in Control, Automation and Robotics*, August 2004, pp. 11–18.

[187] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, and M. A. Obojski, "A sensor for urban driving assistance systems based on dense stereovision," in *Proceedings of Intelligent Vehicles*, June 2007, pp. 278–286.

[188] R. Borca-Muresan and S. Nedevschi, "Correlation between features and classifiers for semantic understanding of pedestrian attitudes in traffic scenes," in *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*, Aug 2009, pp. 149–152.

[189] R. Brehar, C. Fortuna, S. Bota, D. Mladenic, and S. Nedevschi, "Spatio-temporal reasoning for traffic scene understanding," in *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, no. 377–384. IEEE, Aug 2011.

[190] R. Brehar and S. Nedevschi, "Pedestrian detection in traffic scenes using multi-attitude classifiers," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, no. 1077–1082. IEEE, Oct 2013.

[191] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, and M. Obojski, "A sensor for urban driving assistance systems based on dense stereovision," in *Intelligent Vehicles Symposium, 2007 IEEE*, 2007, pp. 276–283.

[192] R. Brehar and S. Nedevschi, "A comparative study of pedestrian detection methods using classical haar and hog features versus bag of words model computed from haar and hog features," in *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, no. 299–306. IEEE, Aug 2011.

[193] N. Tomasev, R. Brehar, D. Mladenic, and S. Nedevschi, "The influence of hubness on nearest-neighbor methods in object recognition," in *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, no. 367–374. IEEE, Aug 2011.

[194] L. Fei-fei, "A bayesian hierarchical model for learning natural scene categories," in *In CVPR*, 2005, pp. 524–531.

[195] R. Brehar and N. S., "Localization and detection of pedestrians in infrared traffic scenes," *Automation, Computers, Applied Mathematics (ACAM)*, vol. 21, no. 2, pp. 161–168, 2012.

[196] R. Brehar and S. Nedevschi, "Pedestrian detection in infrared images using hog, lbp, gradient magnitude and intensity feature channels," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC 2014)*. IEEE, Oct 2014.

[197] R. Brehar, C. Vancea, and S. Nedevschi, "Pedestrian detection in infrared images using aggregated channel features," in *Proceedings - 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing, ICCP 2014*, no. 127–133, 2014.

[198] M. Muresan, R. Brehar, and S. Nedevschi, "Vision algorithms and embedded solution for pedestrian detection with far infrared camera," in *Proceedings - 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing, ICCP 2014*, no. 133–137, 2014.

[199] R. O'Malley, M. Glavin, and E. Jones, "A review of automotive infrared pedestrian detection techniques," in *Signals and Systems Conference, 208. (ISSC 2008). IET Irish*, June 2008, pp. 168–173.

[200] T. Gandhi and M. Trivedi, "Pedestrian protection systems: Issues, survey, and challenges," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, no. 3, pp. 413–430, Sept 2007.

[201] S. Krotosky and M. Trivedi, "On color-, infrared-, and multimodal-stereo approaches to pedestrian detection," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, no. 4, pp. 619–629, Dec 2007.

[202] M. Bertozzi, A. Broggi, P. Grisleri, T. Graf, and M. Meinecke, "Pedestrian detection in infrared images," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, June 2003, pp. 662–667.

[203] A. Broggi, A. Fascioli, M. Carletti, T. Graf, and M.-M. Meinecke, "A Multi-resolution Approach for Infrared Vision-based Pedestrian Detection," in *Procs. IEEE Intelligent Vehicles Symposium 2004*, Parma, Italy, Jun. 2004, pp. 7–12.

[204] Y. Fang, K. Yamada, Y. Ninomiya, B. Horn, and I. Masaki, "A shape-independent-method for pedestrian detection with far-infrared-images," *IEEE Transactions On Vehicular Technology*, vol. 53, no. 5, p. 1679, 2004.

[205] U. Meis, W. Ritter, and H. Neumann, "Detection and classification of obstacles in night vision traffic scenes based on infrared imagery," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 2, Oct 2003, pp. 1140–1144 vol.2.

[206] D. Olmeda, C. Hilario, A. de la Escalera, and J. Armingol, "Pedestrian detection and tracking based on far infrared visual information," in *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes in Computer Science, J. Blanc-Talon, S. Bourennane, W. Philips, D. Popescu, and P. Scheunders, Eds. Springer Berlin Heidelberg, 2008, vol. 5259, pp. 958–969.

[207] D. Olmeda, A. De la Escalera, and J. Armingol, "Detection and tracking of pedestrians in infrared images," in *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*, Nov 2009, pp. 1–6.

[208] D. Olmeda, J. Armingol, and A. de la Escalera, "Discrete features for rapid pedestrian detection in infrared images," in *Intelligent Robots and Systems (IROS), 2012 IEEE RSJ International Conference on*, Oct 2012, pp. 3067–3072.

[209] J. Li and Y. Wang, "Pedestrian tracking in infrared image sequences using wavelet entropy features," in *Computational Intelligence and Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on*, vol. 1, Nov 2009, pp. 288–291.

[210] D. Xia, H. Sun, and Z. Shen, "Real-time infrared pedestrian detection based on multi-block lbp," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, vol. 12, Oct 2010, pp. V12–139–V12–142.

[211] Y. Cao, S. Pranata, and H. Nishimura, "Local binary pattern features for pedestrian detection at night/dark environment." in *ICIP*, B. Macq and P. Schelkens, Eds. IEEE, 2011, pp. 2053–2056.

[212] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," in *Intelligent Vehicles Symposium, 2006 IEEE*, 2006, pp. 206–212.

[213] R. O'Malley, E. Jones, and M. Glavin, "Detection of pedestrians in far-infrared automotive night vision using region-growing and clothing distortion compensation," *Infrared Physics & Technology*, vol. 53, no. 6, pp. 439 – 449, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1350449510000770

[214] Y.-C. Lin, Y.-M. Chan, L.-C. Chuang, L.-C. Fu, S.-S. Huang, P.-Y. Hsiao, and M.-F. Luo, "Near-infrared based nighttime pedestrian detection by combining multiple features," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 1549–1554.

[215] K. Jungling and M. Arens, "Pedestrian tracking in infrared from moving vehicles," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 470–477.

[216] B. Besbes, A. Rogozan, and A. Bensrhair, "Pedestrian recognition based on hierarchical codebook of surf features in visible and infrared images," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 156–161.

[217] B. Besbes, A. Apatean, A. Rogozan, and A. Bensrhair, "Combining surf-based local and global features for road obstacle recognition in far infrared images," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, Sept 2010, pp. 1869–1874.

[218] A. Miron, B. Besbes, A. Rogozan, S. Ainouz, and A. Bensrhair, "Intensity self similarity features for pedestrian detection in far-infrared images," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 1120–1125.

[219] D. Olmeda, A. de la Escalera, and J. Armingol, "Contrast invariant features for human detection in far infrared images," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 117–122.

[220] D. Olmeda, C. Premebida, U. Nunes, J. M. Armingol, and A. de la Escalera, "Pedestrian detection in far infrared images," *Integrated Computer-Aided Engineering*, vol. 20, no. 4, pp. 347–360, 2013.

[221] M. Mählisch, M. Oberlander, O. Lohlein, D. Gavrila, and W. Ritter, "A multiple detector approach to low-resolution fir pedestrian recognition," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, June 2005, pp. 325–330.

[222] B. Besbes, S. Ammar, Y. Kessentini, A. Rogozan, and A. Bensrhair, "Evidential combination of svm road obstacle classifiers in visible and far infrared images," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 1074–1079.

[223] U. Meis, M. Oberlander, and W. Ritter, "Reinforcing the reliability of pedestrian detection in far-infrared sensing," in *Intelligent Vehicles Symposium, 2004 IEEE*, June 2004, pp. 779–783.

[224] D. Gavrila and J. Giebel, "Shape-based pedestrian detection and tracking," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 1, June 2002, pp. 8–14 vol.1.

[225] M. Bertozzi, A. Broggi, C. Gomez, R. I. Fedriga, G. Vezzoni, and M. Del Rose, "Pedestrian detection in far infrared images based on the use of probabilistic templates," in *Intelligent Vehicles Symposium, 2007 IEEE*, June 2007, pp. 327–332.