Descriptors for Object Detection in Image Recognition

VŠB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Computer Science

Summary of Philosophiæ Doctor Thesis

Author: Ing. Radovan Fusek
Supervisor: doc. Dr. Ing. Eduard Sojka

January, 2016
Acknowledgments

I would like to thank my parents for their love. I also want to thank my supervisor doc. Dr. Ing. Eduard Sojka for his help, encouragement and guidance.
Contents

1 Introduction 1

2 State of the Art 3
   2.1 HOG-based Descriptors ......................... 4
   2.2 Haar Wavelet-based Descriptors ................. 7
   2.3 LBP-based Descriptors ......................... 9
   2.4 Interest Point Descriptors ..................... 11

3 Proposed Descriptors 13
   3.1 ETF-based Descriptors ......................... 13
       3.1.1 The Mean Energy in the Regular Cells ........ 15
       3.1.2 The Mean Energy Around in the Sources ....... 17
       3.1.3 The Mean Energy at Different Time .......... 18
       3.1.4 Hierarchical ETF .......................... 19
       3.1.5 ETF with DCT .............................. 21
   3.2 Distance-based Descriptors ..................... 24

4 Summary of the Results 28
   4.1 Face Detection ................................. 28
   4.2 Pedestrian Detection ........................... 34
   4.3 Conclusion ...................................... 39

5 Bibliography 42

6 Author’s Bibliography 50
   6.1 Publications Related to the Thesis .............. 50
1 Introduction

The area of computer vision includes many tasks that are well researched in recent years. In this thesis, we focus on the problem of object description and detection. It is clear that the images contain many objects of interest. The goal of the object detection systems is to find the location of these objects in the images (e.g. cars, faces, pedestrians). For example, the vehicle detection systems are crucial for traffic analysis or intelligent scheduling, the people detection systems can be useful for automotive safety, and the face detection systems are a key part of face recognition systems. Typically, in the area of feature-based detectors, the detection algorithms consist of two main parts. The extraction of image features is the first part. The second part is created by the trainable classifiers that handle a final classification (object/non-object).

In this thesis, the contribution is focused on the first part; on the extraction of image features. The extraction of relevant features has a significant influence on the successfulness of detectors. The large number of features slows down the training and detection phases; on the other hand a very small number of features may not be able to describe the properties of object of interest. The quality of training set is also equally important.

The proposed features are slightly inspired by the image features that are based on the histogram of oriented gradients (HOG) that was presented by Dalal and Triggs [18]. In their approach, the sliding window is divided into the small regions (cells). The histogram of gradient directions is computed within the regions. The regions are normalized across the larger regions (blocks) to provide better illumination invariance. The HOG descriptors are computed in every position of sliding window. In their paper, the authors used the classifier based on the support vector machine (SVM). Many works showed that the HOG descriptors are very useful in the various detection tasks. Nevertheless, the classical HOG descriptors suffer from the large number of features, which causes that the training and detection phases can be time consuming. The sufficient amount of training data is also needed to find a separating hyperplane by the SVM classifier. Sometimes, it is desirable to use the methods for the dimensionality reduction of feature vector.

These shortcomings became the motivation for creating novel methods for the extraction of image features which is the main contribution of this thesis. The first proposed method presented in this thesis is called energy-transfer features (ETF). The main idea behind ETF is based on the fact that the appearance of objects can be described by the energy distribution in the image. In this method, energy sources are defined inside the image. From the sources, the energy is transferred in the image.
the transfer process, the energy distribution is investigated. The appropri-
ately chosen values of energy function are then used for composing the feature vector. The feature vector is then used as an input for the SVM classifier. The investigation is one of the key parts of the method. Therefore, the different ways of coding the energy distribution are presented in this thesis. Namely, the investigation based on the mean value is proposed in Sections 3.1.1, 3.1.2, and 3.1.3. The hierarchical way is presented in Section 3.1.4. The method using the discrete cosine transform is described in Section 3.1.5.

Additional to this, we propose one another method how to describe the properties of objects. In that method, the geodesic distance values are computed in the image. The appropriately chosen distance values are then used to create the feature vector for the SVM classifier. This method is described in Section 3.2. In the next section, we provide the overview of the features that can be extracted from the images.
2 State of the Art

In the process of detection, we use the sliding window technique. In general, the sliding window technique represents the popular and successful approach for object detection. The main idea of this approach is that the input image is scanned by a rectangular window at multiple scales (Fig. 2.1(a)). The result of the scanning process is a large number of various sub-windows (Fig. 2.1(b)). A vector of features is extracted from each sub-window. The vector is then used as an input for the classifier (in our case, the SVM classifier). During the classification process, some sub-windows are marked as the objects. Using the sliding window approach, the multiple positive detections may appear, especially around the objects of interest (Fig. 2.1(c)). These detections are merged to the final bounding box that represents the resulting detection (Fig. 2.1(d)). The classifier that determines each sub-window is trained over the training set that consists of positive and negative images.

The key point is to find what values (features) should be used to effectively encode the image inside the sliding window. In the feature-
2.1. HOG-based Descriptors

In recent years, the object detectors that are based on edge analysis that provides valuable information about the objects of interest were used in many detection tasks. In this area, the histograms of oriented gradients (HOG) \[18\] are considered as the state-of-the-art method. The HOG descriptors are inspired by the scale-invariant feature transform (SIFT) \[41\]. The idea of the SIFT descriptor is that the interesting points (keypoints) of the objects can be extracted to provide the key information about the objects. The gradient magnitude and orientation are computed around the keypoint location; the histograms are then summarized over subregions. The keypoints are extracted from the reference image (that contains the object of interest) and also from the target image (that possibly contains the object of interest). The extracted keypoints are matched to find similarity between the images. In \[42\], the authors showed that the SIFT features achieved a very good performance compared with other local descriptors.

In essence, the HOG descriptors can be regarded as a dense version of the SIFT features. In HOG, a sliding window is used for detection. The window is divided into small connected cells in the process of obtaining HOG descriptors. The histograms of gradient orientations are calculated in each cell. It is desirable to normalize the histograms across a large block of image. As a result, a vector of values is computed for each position of window. This vector is then used for recognition, e.g. by the Support Vector Machine (SVM) classifier \[9\]. Dalal and Triggs experimented with the size of detection window and they suggested the rectangular window with the size $64 \times 128$ pixels. They also tried to reduce the size of the window to $48 \times 112$ pixels. Nevertheless, they obtained the best detection result with the size $64 \times 128$ pixels.

Let us express the method more formally in the following steps.

- For gradient computation, the image without Gaussian smoothing is filtered with the $[1, 0, -1]$ kernel to compute the horizontal ($G_x$) and vertical ($G_y$) derivatives.

- Then the derivatives are used to compute the magnitude of the gradient $|M| = \sqrt{G_x^2 + G_y^2}$ and orientation $\theta = \arctan \frac{G_y}{G_x}$.
2.1. HOG-based Descriptors

- In the next step, the image is divided into the cells and the cell histograms are constructed. The histogram bins are spread over 0 to 180 degrees or 0 to 360 degrees. The corresponding histogram bin is found for each pixel inside the cell. Each pixel contributes a weighted vote for its corresponding bin. The pixel contribution can be the gradient magnitude.

- Next step represents contrast normalization. For this purpose, the cells are grouped into the large blocks (i.e. 2×2 cells are considered as blocks). The histograms are normalized within the blocks (e.g. using L2-norm). In the paper, the two main block geometries are presented; rectangular and circular.

- The final HOG descriptor is represented by histogram vectors of all blocks within the detection window.

The classical HOG descriptors suffer from the large number of features, which causes that the training and detection phases can be time consuming. The sufficient amount of training data is also needed to find a separating hyperplane by the SVM classifier. Sometimes, it is desirable to use the methods for the dimensionality reduction of feature vector. In addition the that, the classical HOG descriptors are not rotation invariant. These shortcomings became the motivation for creating many variations of HOG-based detectors. Many methods and applications based on HOG were presented in recent years.

In [19], Dalal and Triggs combine their descriptors with the optical flow-based motion descriptors. The motivation was to create the detector for standing and moving people in videos with moving cameras and backgrounds. The pedestrian detection method using the infrared images and histograms of oriented gradients combined with the SVM classifier was presented in [54]. The near real-time human detection system using the cascade of rejectors with the histograms of oriented gradients was proposed in [66]. The authors exploit the fast way of calculating the HOG features with the use of the integral image and they integrated the HOG features into the cascade framework that was presented by Viola and Jones in [57].

The PHOG (pyramid histograms of orientation gradients) descriptors was proposed in [8]. This method uses the combination of the image pyramid representation of Lazebnik et al. [30] and the histograms of orientation gradients [18]. These descriptors were successfully extended and used, e.g. for smile recognition in [4]. The authors reported that the features with a lower number of dimensions were extracted with the use of the PHOG descriptors in comparison with the Gabor features [40] for this detection task. Finally, the authors used the combination of PHOG and Gabor features for better results.
2.1. HOG-based Descriptors

In [29], the authors applied the principal component analysis (PCA) to the HOG feature vector to obtain the PCA-HOG vector. This vector contains the subset of HOG features and the vector is used as an input for the SVM classifier. Their method was used for pedestrian detection with the satisfactory results.

In [15], the authors proposed augmented histograms of oriented gradients (AHOG) for human detection from a non-static camera. Their approach extended the classical HOG features by adding the human shape properties. The authors reported that the method achieved a good performance at many views of targets.

The classical HOG-based detector was used for an upper-body detector for an automated upper-body pose estimation in [22]. In this work, the authors used the upper-body HOG-based detector to obtain a weak model of person and reduce the search space for body parts. Similarly, the HOG-based head-shoulder detector was used in the method that was focused on estimating the number of people in surveillance scenes in [33]. This method consists of two parts. In the first part, the foreground segmentation is used to obtain the active areas. In the second part, the head-shoulder detector based on HOG is used. The authors showed (in experiments) that the HOG-based head-shoulder detector outperformed the Haar features [57] and SIFT descriptors [41]. This method was extended in [34]. In this work, the authors accelerated the detection part using the combination of Viola-Jones type classifier and the HOG feature-based AdaBoost classifier.

In [21], Felzenszwalb et al. proposed the part-based detector that is based on HOG. In this method, the objects are represented using the mixtures of deformable HOG part models and these models are trained using a discriminative method.

A method for vehicle detection in low-altitude airborne videos using boosting HOG features was presented in [11]. The authors used the bins of the histogram as a week classifier. The AdaBoost algorithm is then used to train the strong classifier. In the final stage, all boosting HOG features are combined to the final feature vector to train the linear SVM classifier.

In [47], the authors proposed the speed-up of the HOG+SVM algorithm without sacrificing the classification accuracy using the sub-cell based interpolation algorithm to accelerate the calculation of the HOG features in one block combined with the strategy that can reduce the redundancy of HOG features inside the detection window. The authors reported that the technique is five times faster than the traditional method. Additionally, the authors established a top-view human database.
2.2 Haar Wavelet-based Descriptors

The main idea behind the Haar-like features is that the features can encode the differences of mean intensities between the rectangular areas. For instance, in the problem of face detection, the regions around the eyes are lighter than the areas of the eyes; the regions below or on top of eyes have different intensities that the eyes themselves. These specific characteristics can be simply encoded by one two-rectangular feature, and the value of this feature can be calculated as the difference between the sum of the intensities inside the rectangles.

The Haar-like features, which are similar to Haar basis functions, were proposed by Papageorgiou and Poggio [48]. In their paper, the Haar-like features were combined with the SVM classifier. The authors used the three types of Haar features (vertical, horizontal, diagonal) that were able to encode the changes in the intensities at various locations, scales, and orientations. The authors also reported promising performance in the tasks of face, car, and people detection.

The paper of Viola and Jones [57] contributed to the popularity of Haar-like features. The authors proposed the object detection framework based on the image representation called the integral image combined with the rectangular features, and the AdaBoost algorithm [23]. With the use of integral image, the rectangular features are computed very quickly. The AdaBoost algorithm helps to select the most important features. The features are used to train classifiers and the cascade of classifiers is used for reducing the computational time. An extension of Haar feature set was presented by Lienhart et al. [39]. The authors presented the $45^\circ$ rotated features that are able to reduce the false alarm and achieve more accurate face detection. The improvement of the weak classifiers combined with the Real AdaBoost for the fast multi-view face detection system was presented by Wu at al. [59].

Since Viola and Jones popularized the Haar-like features for face detection, the Haar-like features and their modifications were used in many detection tasks (e.g. pedestrian, eye, vehicle). In the area of pedestrian detection, in [43], the authors presented the component-based person detector that is able to detect the occluded people in clustered scenes in static images. The detector uses the Haar-like features to describe the components of people (heads, legs, arms) combined with the SVM classifier. The Viola and Jones detection framework was successfully extended for moving-human detection in [58]. In [53], the authors proposed the method for estimating the walking direction of pedestrian. The Haar-like features are used to generate the feature vector and the 16 classifiers are trained (each classifier represents a specific direction) using SVM in this approach. The 3D Haar-like features for pedestrian detection were presented in [17]. The authors extend the classical Haar-like features
using the volume filters in 3D space (instead of using rectangle filters in 2D space) to capture motion information. The 3D features are then combined with the SVM classifier. To compute the 3D Haar-like features using the integral image like the classical 2D features, the authors introduced Integral Volume that extends 2D integral image to the three dimensions.

In [36], the authors proposed the local Haar-like features in the edge maps (LHEP) to describe the shape of pedestrian. The selection of the LHEP features was done using the Real AdaBoost algorithm combined to the cascade framework. The method that combines the Haar-like features and HOG features for pedestrian detection was proposed in [61]. In this method, the detection process is divided into the phases. The first phase consists of a rectangular feature classifier that removes a lot of detection windows without the objects of interest. The second phase consists of the HOG classifier that manages further detection of windows from the first phase. The authors reported that this way can improve the detection speed with preserving high accuracy. In [26], the authors proposed a modified version of Haar-like features that more properly reflect the shape of the pedestrians than the classical Haar-like features. The authors reported that the detector based on these features achieved better detection results than the classical Haar-like features.

In the area of vehicle detection, the comparison of Haar-like features, the HOG features, and the combination of these features were presented in [45]. The vehicle detection system based on the Haar and triangular features was proposed in [25]. The authors presented 2D triangle features with triangular shape in contrast to the Haar-like features (rectangular shape). In [60], the authors proposed the front-view car and bus detection method based on the AdaBoost and Haar-like features. The forward collision warning system that consists of vehicle detection using the Haar-like features combined with the AdaBoost algorithms was presented in [16]. The method for detecting cars was presented by Zheng and Liang [65]. Their strip features are calculated using the integral image. Combined with the RealBoost framework, a good performance was achieved. Nevertheless, the authors mentioned that the strip features discard some statistical information compared with the more complex descriptors such as HOG [18].

The eye localization method using a combination of Haar-like features with the HOG descriptors was proposed in [44]. In this paper, the authors used the upper part face detector based on the Haar-like features to preselect the eye candidates at different scales in a short period of time. This step was also used to reject many negative sub-windows very quickly. In order to reduce the false positive eye candidates, the best candidates are selected using the SVM classifier combined with the HOG descriptors in the next step. The authors reported that the HOG
descriptors are computationally more expensive than the Haar-like features. Nevertheless, the HOG descriptors are less sensitive to the illumination changes. Due to the fact that the HOG descriptors were used only for reducing the number of eye candidates, the overall computational time of HOG descriptors was relatively small.

Recently, the improvement of Haar-like features for efficient object detection under a wide range of illumination conditions was proposed in [49]. In this method, the contrast (intensity differences between the regions divided by their average intensity) of region is computed instead of intensity difference between the rectangular regions. The authors created the car, pedestrian, and face detectors based on the proposed Haar contrast features. The authors reported that their detectors obtained very satisfactory results under a wide range of illumination conditions compared with LBP and the classical Haar features.

The comparison of the face and facial feature detectors based on the Viola-Jones general object detection framework was presented in [12]. The authors tested the public domain face and facial feature (e.g., left eye, right eye, nose, mouth) classifiers.

2.3 LBP-based Descriptors

The local binary patterns (LBP) were introduced by Ojala et al. [46] for the texture analysis. The main idea behind LBP is that the local image structures (micro patterns such as lines, edges, spots, and flat areas) can be efficiently encoded by comparing every pixel with its neighboring pixels. In the basic form, every pixel is compared with its neighbors in the $3 \times 3$ region. The result of comparison is the 8-bit binary number for each pixel; in the 8-bit binary number, the value 1 means that the value of center pixel is greater than the neighbor and vice versa. The histogram of these binary numbers (that are usually converted to decimal) is then used to encode the appearance of region. The important properties of LBP are the resistance to the lighting changes and a low computational complexity. Due to their properties, LBP were used in many detection tasks, especially in facial image analysis.

In [24], LBP were used for solving the face detection problem in low-resolution images. In this approach, the $19 \times 19$ face images are divided into the 9 overlapping regions in which the LBP descriptors are computed. Additionally, the LBP descriptors are extracted from the whole $19 \times 19$ image. The descriptors are then used to create the feature vector, and the SVM classifier with a polynomial kernel is used for the final classification. In [2, 1], the similar method was used in which the authors presented the use of LBP for the face recognition task.

In [28], the authors noted that classical LBP can miss some local structures under certain circumstances. Therefore, they proposed improved
local binary patterns (LBP) combined with the Bayesian classifier for face detection. Multi-block local binary patterns (MB-LBP) for face detection were proposed in [62, 37]. In this method, the authors encode the rectangular regions by the local binary pattern operator and the Gentle AdaBoost is used for feature selection. Their results showed that MB-LBP are more distinctive than the Haar-like features and the original LBP features. In [14], another method based on LBP was proposed for face recognition. The authors combined the multi-scale local binary pattern representation with the linear discriminant analysis. The paper of Tan and Triggs [55] proposed the face recognition method with robust preprocessing based on the difference of Gaussian image filter combined with LBP in which the binary LBP code is replaced by the ternary code to create local ternary patterns (LTP).

Many other variations of LBP were proposed for face recognition. For example, multi-resolution histograms of local variation patterns (MH-LVP) were proposed in [63]. In this paper, a multi-resolution image description is computed using the Gabor wavelets to obtain multiple Gabor feature maps (GFMs). Each GFM is then divided into the non-overlapping rectangle regions from which the LBP histograms are extracted. The histograms are combined to the final feature vector that represents the face model. The framework for face recognition that combines the 2D intensity images and 3D shape models was proposed in [35]. In this framework, LBP are used to represent the 3D and 2D face models with AdaBoost that is applied for the feature selection and the classifier learning. In experiments, the authors presented that the results were better than the PCA-based method. The method based on the kernel Fisher discriminant analysis (KFDA) combined with LBP for the face verification was proposed in [64]. The authors reported that their proposed LBP based KFDA method outperforms the original LBP and KFDA algorithms. Another face recognition approach based on LBP was proposed in [38]. In this paper, elongated LBP (ELBP) and average maximum distance gradient magnitude (AMDGM) features are proposed. In classical LBP, the neighborhood pixels are defined on a circle. In ELBP, the neighboring pixels are defined on an ellipse. The authors noted that ELBP is able to capture anisotropic information of faces in contrast to the classical LBP. The authors also mentioned that LBP and ELBP do not take gradient information in account. Therefore, they proposed AMDGM to capture such information. Finally, the combination of both features achieved a very satisfactory recognition rate compared with the PCA, LDA, and LBP recognition methods.

LBP were also successfully used for the facial expression analysis. The comprehensive study of facial expression recognition using LBP was proposed in [52], the survey of facial image analysis using LBP was presented in [27].
2.4 Interest Point Descriptors

The most of the previously mentioned methods for object description were based on the fact that the descriptors were extracted over the whole image (sliding window) that was usually divided into the overlap or non-overlap regions. Inside these regions, the descriptors were calculated and combined to the final feature vector that was used as an input for the classifier.

In this section, we present the state-of-the-art descriptors that are based on the fact that the regions (within which the descriptors are extracted) are selected using the keypoint detectors. Since the proposed descriptors (in the next chapter) are also computed over the whole image divided into the regular regions (e.g. square blocks) like classical HOG, in contrast with the SURF and SIFT-like features that are based on arbitrarily located feature points, we present the key point detectors only in brief.

As was mentioned in Section 2.1, one of the most popular descriptors based on the interest points was proposed by David Lowe [41]. The method is called scale invariant feature transform (SIFT). This method uses the difference of Gaussian (DoG) keypoint detector. The histogram gradient orientations inside the regions around the keypoints are calculated.

The speeded up robust feature (SURF) descriptor by Bay et al. [6, 5] is also one of the widely used keypoint descriptors. In this method, the Hessian matrix-based measure is used to find the points of interest. The sum of the Haar-wavelet responses within the neighborhood of interest point is calculated. The authors also use the fast calculation via the integral image thanks to which SURF is faster than SIFT.

A very fast method called binary robust independent elementary features (BRIEF) was proposed by Calonder et al. [10]. The authors reported that the method outperforms SURF in the term of speed, and the recognition rate in many cases. In BRIEF, a binary string that contains the results of intensity differences of pixels are used and the descriptor similarity is evaluated using the Hamming distance. In [51], the authors proposed another binary descriptor with rotation and noise invariant properties called oriented fast and rotated BRIEF (ORB).

Leutenegger et al. [32] proposed binary robust invariant scalable keypoints (BRISK). The method provides both scale and rotation invariance. BRISK is a binary descriptor like BRIEF and ORB, it means that the binary string that represents a region around the keypoint is composed. In BRISK, a concentric circle pattern of points near to the keypoint is used. In this pattern, the blue circles represent the sampling locations and Gaussian blurring is computed to be less sensitive to noise; the radius of red circles denotes a standard deviation of blurring kernel. The standard de-
vation of the Gaussian kernel is increased with the increasing distance from the feature center to avoid aliasing effects. The final descriptor is determined by the comparison of sample points.

In [3], the authors proposed the fast retina keypoint (FREAK) descriptor that also uses the binary strings. The method is biologically inspired by a human visual system; more exactly by the retina. In this paper, the authors proposed a retinal sampling pattern. The pattern is divided into the areas (foveal, fovea, parafoveal, and perifoveal) similar to the human retina. In this pattern, the pixels are overlapped and concentrated near to the center. The binary strings is computed by comparing the point pairs of image intensities within the pattern.
3 Proposed Descriptors

In this chapter, we present the main idea of the proposed image features. In the first section, we describe the energy-transfer features (ETF) and we show the properties of these features in the area of face, pedestrian, and vehicle detection. In the last section, we describe the features that are based on the geodetic distance, and we show their application in the area of face and pedestrian detection. We also compare the proposed features with the state-of-the-art methods. The parts of these sections were presented in [67, 68, 69, 70, 71, 72, 73, 74, 75, 76].

3.1 ETF-based Descriptors

The main idea of the proposed features is that the appearance of objects can be described by the distribution of energy in image. If we speak about energy transfer, we have in mind transfer of heat. The image can be considered as a rectangular plate with certain thermal conductivity properties that are determined by the gradient of brightness (big gradients indicate the low conductivity and vice versa). In the area of image, the distribution of energy can be solved by making use of physical laws. We solve the distribution for the point sources of constant temperature that are appropriately located in the image. At \( t = 0 \), the temperature is zero in the whole area of image, except the temperature sources. We suppose that the heat transfer starts at \( t = 0 \) and, theoretically, it can be infinitely long. Nevertheless, we stop the transfer at a suitable time \( t > 0 \). During the whole time of transfer, the temperature at source points is held on a chosen initial value. After the transfer, the distribution of temperature is investigated. Since the contours of object correspond to the places with high gradients and since the values of gradients correspond to the value of thermal conductivity, we can conclude that the shapes of objects are encoded in the distribution of temperature that can be obtained by the process described above.

The usefulness and motivation to use the temperature distribution function can be described as follows. Suppose that the object of interest with very thin edges is analyzed by the functions of gradient sizes and directions. The meaningful sample values of this function can be difficult to obtain; it is difficult to obtain (by making use of the samples) the information about thin edges (it may happen that the samples do not hit the this edges). In contrast, the function of temperature distribution does not make problems during sampling. In this function, the areas with approximately constant temperature values are important and it is an easy matter to hit them by samples.

For better understanding, let us firstly consider the following simple
3.1. ETF-based Descriptors

![Diagram](image.png)

Figure 3.1: The image with one object and one source of temperature. The value of temperature is depicted by the intensity of red color.

Theoretical image containing one rectangular object of constant brightness on the background (the gradient of brightness of this theoretical image is shown in the second row in Fig. 3.1). The problem of segmentation can be transformed into the problem of solving heat transfer as follows. At places where the size of gradient is zero, the thermal conductivity equals to infinity; where the size of gradient is greater than zero, the conductivity is zero. In this introductory example, we have only one source of temperature that is placed into a point lying inside the object (say into the center of gravity). For all \( t \geq 0 \), the temperature at the source point is equal to 1. For \( t = 0 \), the temperature at all other places in the image is equal to 0 (Fig. 3.1(a)). After some time \( t > 0 \), the distribution changes into the form as is depicted in Fig. 3.1(b). Clearly, the distribution of temperature reflects the shape of the object. It follows that the distribution of temperature can be used for recognizing. We note that in this particular example, the time of transfer does not play a substantial role; the same distribution is achieved for every \( t > 0 \) due to the assumption about the infinite and zero conductivities.

The presented example also shows that one source point will not be sufficient for the real-life images (Fig. 3.2(a)). The reason can be easily understood. If we have more objects, if the objects are more complicated, and if we drop the assumption that the conductivity can only be either zero or infinity, more sources are apparently needed. Generally, the sources can be placed into a regular grid (Fig. 3.2(b)). The transfer of temperature starts from all sources at the same time. After the transfer that was carried out during a suitably chosen time, we obtain a temperature distribution. The distribution reflects the presence of objects and their parts, which is the main idea of the method we propose. The
visualization of temperature distribution is shown in Fig. 3.2(c).

Figure 3.2: The real-life image (a). The regular grid of sources (b). The visualization of distribution of temperature from the sources (c). The value of temperature is depicted by the level of brightness.

Generally, the distribution of temperature is a function that has values of uncountably many points. For practical use, the function must be compressed into an acceptable amount of values. In the next sections, we propose the ways how to encode the values of this function into the feature vector that can be used as an input for trainable classifiers.

3.1.1 The Mean Energy in the Regular Cells

For the purpose of recognition, as was said before, the function of temperature should be sampled. In [75], we introduce the first way how to encode the values of temperature function. We can either simply take the values at a point grid or to carry out the sampling by integration. We regard the second approach as more robust. For this purpose, we divide the input image into cells and we investigate the mean temperature in each cell. Generally, the position of the sources of temperature can be chosen arbitrarily and independently on the cells. In the following text, however, we put the sources into each cell. As the position of sources, we use the gravity centers of cells.

Let us express the things more formally. Let $I(x, y, t)$ stand for the value of temperature at a position $(x, y)$ and at a time $t$; the mean temperature in the $i$-th cell is denoted by $I\mu_i$. We can compute these mean temperatures for all cells in the whole input image. In the process of detection, we use a sliding window. The vector of features for each position of sliding window is assembled from the mean temperatures in the cells that fall into the window in its actual position (Fig. 3.3). The $i$-th item in the feature vector is the mean temperature $I\mu_i$ in the $i$-th cell in window. The vector of features is then used in the SVM classifier.

For practical realization of the method, it is important to mention that the thermal field over the input image can be solved by making use of
3.1. ETF-based Descriptors

Figure 3.3: The vector of features for a momentary position of sliding window.

the following equation [50]

\[
\frac{\partial I(x,y,t)}{\partial t} = \text{div}(c \nabla I),
\]

(3.1)

where \( I(x,y,t) \) represents the temperature at a position \((x,y)\) and at a time \(t\), \(\text{div}\) is a divergence operator, \(\nabla I\) is the temperature gradient and \(c\) stands for the thermal conductivity. For the source points and arbitrary time \(t \in [0,\infty)\), we set \( I(x_s,y_s,t) = 1 \), where \((x_s,y_s)\) are the coordinates of source points (i.e. we hold the temperature constant during the whole process of transfer, which is in contrast with the usual diffusion approaches). In all remaining points, we take into account the initial condition \( I(x,y,0) = 0 \). We solve the equation iteratively. The conductivity in Eq. 3.1 is determined by

\[
c = g(\|E\|),
\]

(3.2)

where \(E\) is an edge estimate. We define the edge estimate \(E\) as the gradient of original image \(E = \nabla B\), where \(B\) is the brightness function. The function \(g(\cdot)\) has the form of [50]

\[
g(\|\nabla B\|) = \frac{1}{1 + \left( \frac{\|\nabla B\|}{K} \right)^2},
\]

(3.3)

where \(K\) is a constant representing the sensitivity to the edges [50]. Once the temperature field over the input image is obtained (at a chosen time \(t\)), the mean cell temperature \(I_{\mu i}\) can be obtained by making use of the formula

\[
I_{\mu i} = \frac{\int \int I(x,y,t)dxdy}{|M|},
\]

(3.4)

where \(M\) stands for the cell area, and \(|M|\) is its size. Based on the principle from the previous paragraphs, we have developed the face detector
3.1. ETF-based Descriptors

using the proposed descriptors; more details (about results of this detector) can be found in [75].

3.1.2 The Mean Energy Around in the Sources

Since we hold the temperature \( I(x_s, y_s, t) = 1 \) constant during the whole process of transfer at the source points, the information at the points of the sources is not meaningful. In [74], we focus on elimination of this problem using the 8-neighborhood or 4-neighborhood of the sources that can be investigated. In the process of extracting the proposed features in this way, the image inside the sliding window is divided into the regular blocks (Fig. 3.4). We use the gravity centers of these blocks as the places in which we put the temperature sources. For the purpose of obtaining the distribution of temperature, the blocks are divided into the small cells (Fig. 3.4).

Let \( I(x, y, t) \) be a value of temperature at a time \( t \) and at a position \( (x, y) \). Inside each cell, the mean temperature \( I_{\mu i} \) of the \( i \)-th cell at a time \( t \) can be calculated. The final feature vector is composed of these mean values. We note that the temperature transfer is computed in the whole image inside the sliding window and the temperature transferred from one source can influence every cell inside the sliding window; the blocks and cells are formed only for distribution measurement.

Figure 3.4: The blocks and cells inside the sliding window.
Figure 3.6: The visualization of proposed descriptors of pedestrian images. The value of temperature is depicted by the level of brightness. The features are designed with the size of blocks $15 \times 15$; with the size of temperature sources $5 \times 5$, and with the 150 iterations for the temperature transfer.

Figure 3.5: The design of every block in the $Energy_{480}$ configuration (a). The design of every block in the $Energy_{240}$ configuration (b).

To test this approach, we created the pedestrian detector in [74]. We experimented with the parameters of proposed features and we suggested two optimal block configurations (Fig. 3.5). The visualization of the proposed features of positive samples is shown in the Fig. 3.6. The parameters and results are discussed in [74].

3.1.3 The Mean Energy at Different Time

In [73], we experiment with different times (different numbers of iterations) in which the function of temperature can be investigated and we use the method for solving the problem of car detection. It is clear that, with the use of different numbers of iterations, we get various information about objects the (Fig. 3.7). For the description of small areas by
3.1. ETF-based Descriptors

temperature distribution, a lower number of iterations during which the temperature transfer is carried out is required; small areas are filled with a certain amount of energy (that is sufficient for the description of these areas) in a relatively short period of time. On the other hand, the temperature sources that are located in the big areas require more iterations to affect the whole areas of the objects. For instance, the shape of the side windows is visible with the use of 50 iterations ($it = 50$), however, the shape of the hood is recognizable with the use of 150 iterations ($it = 150$), this can be visible in Fig. 3.7. For this reason, we compute the temperature distribution for various times, i.e. various numbers of iterations. For example, we can compute two different distributions; one with $it_1$ iterations, the second with $it_2$. In this particular case, the final feature vector includes the information from both these distributions. We can take the values of both distributions and compose them sequentially one after another into the final vector. The next possibility is to combine the values from both distributions together (e.g. by averaging them). In the first case, the size of the final vector is two times larger than is the size in the second case. We regard the second approach as often more suitable for recognition due to the fact that the dimensionality of final vector is not increased. For the purpose of investigating the distribution with a chosen number of iterations, each block is divided into the cells that are placed into the neighborhood of source like in Fig. 3.4. The parameters and results of car detection are discussed in [73].

![Figure 3.7](image)

Figure 3.7: The visualization of temperature distribution for various numbers of iterations (for various times). The value of temperature is depicted by the level of brightness.

3.1.4 Hierarchical ETF

In [72], we experiment with various sizes of the cells. The image is divided into the cells of variable sizes and the values of the temperature function is investigated inside each cell.

As was mentioned before, once the temperature transfer inside the image is obtained, the function of temperature distribution inside the image is investigated. In this method, the image is iteratively divided into the finer spatial cells; i.e. we recursively divide the image into the
cells of varying size (Fig. 3.8). In general, the image at hierarchical level \( l \) has \( 4^l \) cells. Inside each cell, the distribution is investigated.

Figure 3.8: The different hierarchical levels of the cells.

Let \( I(x, y, t) \) be the function of temperature (at location \( (x, y) \) and the time \( t \)) that is determined. We can compute the mean temperature in every cell; \( I_{\mu_{it}} \) stands for the mean temperature of the \( i \)-th cell at the time \( t \). We use the mean cell temperatures as the values in the feature vector. For the additional information and for the precise description of the temperature distribution, we use the histogram of the temperature distribution that is also determined inside the cells. Each cell is represented by the \( \sqrt{|M|} + I_{\mu_{it}} \) dimension vector, where \( |M| \) is the size of the cell, and \( \sqrt{|M|} \) represents the number of histogram bins; the final vector of features is composed of the mean temperature and from the histograms of each cell at each hierarchical level. For example, for one \( 20 \times 20 \) cell (400 pixels), we compute the 20 histogram bins and the mean temperature of the cell \( I_{\mu_{it}} \), and the feature vector of this cell is the vector with the dimension \( d = 21 \). The process of composing the final feature vector for the levels \( l = 0 \) and \( l = 1 \) is shown in Fig. 3.9.

To test the hierarchical approach, we create the face detector. For the training phase, the positive set consists of 2300 faces and 4300 non-faces. We used the face images from the BIOID database combined with the Extended Yale Face Database B [31]. We manually cropped these images on the area of faces only. The negative set consists of 3000 images that were obtained from the MIT-CBCL database combined with the 1300 hard negative examples. The training images (for the proposed method) were resized to the size of 80 \( \times \) 80 pixels.

As we said before, we use the sliding window technique in the detection phase. The size of detection window is set to 80 \( \times \) 80 pixels (the size of training samples). We use the fixed size of window that scans the image in 12 different resolutions of input image. The thermal field is computed for each resolution.

We experimented with the proposed method and we suggest the following configuration. The size of temperature sources is 1 pixel and the distance between the sources is 5 pixels. The number of iterations (time
3.1. ETF-based Descriptors

![Figure 3.9: The process of composing the final feature vector.](image)

Figure 3.9: The process of composing the final feature vector. In the case that the image has the size of $80 \times 80$ pixels, the vector dimension is $d = 81$ at the level $l = 0$ (80 bin histogram + mean temperature $I_{\mu_{l_0}}$). The level $l = 1$ contains 4 cells with size $40 \times 40$ pixels; the 40 bin histogram with the mean temperature $I_{\mu_{l_1}}$ is calculated in each cell (41 values for each cell) and the vector dimension is $d = 164$. In this case, the final feature vector is composed as the concatenation of vectors at the level $l = 0$ and $l = 1$ (not merging them).

The parameters and results are discussed in [72].

3.1.5 ETF with DCT

In the basic version of energy-transfer features (ETF), the idea behind ETF is that the appearance of objects can be successfully described using the function of temperature distribution in image. Inside the image, the temperature sources are placed and the heat is transferred from the sources during a certain chosen time. The values of temperature function
ETF-based Descriptors

have to be reduced into a reasonable number of values. The process of reducing can be simply solved by sampling; the input image is divided into the regular cells. The mean of the values is calculated inside each cell. The values are then considered as the vector that is used as an input for the SVM classifier.

In [67, 71], we propose an improvement of this process. Instead of the mean temperature, we use the discrete cosine transform (DCT) coefficients to encode the function of temperature distribution inside each cell. After DCT, the DC coefficients represent the average temperatures in the cells, and the AC coefficients represent the temperature changes across the cells. To reduce the quantity of coefficients, we use the patterns in which the coefficients are grouped into regions. In each region, the mean value of coefficients is computed. The mean values from the particular regions are then used for recognition since they effectively encode the function of temperature distribution. Since we hold the temperature constant during the whole process of transfer, the image information at the places of source points is lost. We propose a post-processing step that eliminates this problem. Finally, PCA (principal component analysis) is also used to create the feature vector with a relatively small dimension.

Let us express the method in detail. Once the temperature field over the input image is obtained (at a chosen time $t$) using 3.1, the temperature values should be sampled. For this purpose, we divide the input image inside the sliding window into the blocks. The blocks are divided into the cells (Fig. 3.10).

![sliding window](image)

Figure 3.10: The blocks and cells inside the input image (sliding window).

We experimented with different sizes of blocks and cells. We observed that the best results were obtained using $16 \times 16$ blocks and $8 \times 8$ cells; inside the cells the DCT coefficients are computed and composed
3.1. ETF-based Descriptors

to the final feature vector. The coefficients that are located in the upper left corner contain the most important information (low frequencies). On the other hand, the bottom right coefficients correspond to the higher frequencies that can be discarded. Therefore, instead of encoding the whole matrix of coefficients, we encode the upper left coefficients only.

To encode the coefficients, we create three patterns of coefficients (Fig. 3.11) for our experiments (similarly as in [56]). In the patterns, three AC regions are created. The regions correspond to the different frequencies; different information can be encoded using different patterns. To reduce the quantity of coefficients, the mean of coefficient values is calculated for each region. It means that each $8 \times 8$ cell is represented by four values: 1 DC coefficient + 3 averages of AC coefficients. These values are included into the final feature vector.

The DC coefficient that represents the average energy of cell is the most important coefficient. Therefore, the pattern Fig. 3.11(a) is constructed to use the coefficients that are close to the DC coefficient. In the pattern from Fig. 3.11(b), the coefficients are grouped into horizontal, vertical, and diagonal areas; this pattern is constructed to capture information in different directions. The pattern Fig. 3.11(c) is constructed as an another alternative for the pattern from Fig. 3.11(b). In this pattern, the differences between the sizes of the areas are reduced; the regions have approximately the same size. We experimented with different variations of the patterns. However, these three patterns achieved the best results. Using this approach to encode the coefficients, the dimensionality of the feature vector of each block is 16. One $16 \times 16$ block that contains four $8 \times 8$ cells is encoded by 16 values: 4 DC coefficients + 12 averages of AC coefficients.

![Figure 3.11: The three different options of AC patterns. The areas are depicted by three different colors. In the areas, the averages of coefficients are calculated.](image)

In the face detector, this dimensionality is further reduced using PCA. Finally, the support vector machine classifier with the radial basis function kernel is trained over the proposed descriptors to create the final
3.2 Distance-based Descriptors

In [68, 69], we propose the descriptors based on the geodesic distance function; the main idea of the method is similar to the energy descriptors but instead of the energy values, the distance values are computed inside the image. The proposed distance method is based on the fact that the properties of the image (especially the properties of the objects) can effectively be described by the distance function. The goal is to obtain more meaningful values for recognition than the classical state-of-the-art methods use. The usefulness of the distance function can be described in the following way.

Consider the simple theoretical image that contains one object of constant brightness (Fig. 3.12(a)). The appearance (shape) of this object can be described using the gradient of this object (Fig. 3.12(b)). In the classical sliding window methods (HOG, Haar, LBP), the samples (e.g. blocks, rectangular features) must hit the places with the intensity differences (edges) to obtain the information about the object. In the situation that edges can be very thin (theoretically, infinitely thin), it is difficult to hit the places with the edge information, and many samples contain the redundant information without the gradient (edges) information.

Say that the samples (e.g. blocks, rectangular features) are placed inside the image in the way as is depicted in Fig. 3.12(c). In such a case, the samples do not detect any important information; the values of gradient sizes and directions are null (HOG principle), as well as the intensity differences inside the samples (Haar principle). This situation was motivation to use another way how to encode the appearance of the objects inside images.

Figure 3.12: The image with one object with constant brightness (a). The gradient of the image (b). The samples (red color blocks) in which the information about the object is encoded (c).

3.2. Distance-based Descriptors

Suppose that an arbitrary point that is placed inside the previously mentioned object. Say that it is located in the gravity center of the object (Fig. 3.13(a)); we will call it the centroid $c_i$. Let us compute the geodesic distance function $d$ from the centroid $c_i$ to all other points inside the image. The visualization of the distance function values is shown in Fig. 3.13(b). In this particular case, we use the geodesic distance, nevertheless, it is important to note that any appropriate distance function can be used in the proposed detection framework (e.g. resistance distance, diffusion distance). In general, the geodesic distance $d(c_1, c_2)$ between two points $c_1, c_2$ computes the shortest curve that connects both points along the image manifold.

![Figure 3.13](image)

Figure 3.13: The image with one object with constant brightness that contains the centroid point (red color) (a). The visualization of the distance function (b). The values of distance function are depicted by the level of brightness. The samples (red color blocks) in which the information about the object is encoded (c).

Consider the same distribution of samples as in the previous case (Fig. 3.13(c)). The main contribution of using the distance function is that the values of this function are different inside and outside the object of interest. In essence, the values of distance function reliably reflect the image information and the appearance of objects, and the meaningful values can be reliably obtained by sampling. Even the simple samples in Fig. 3.13(c) can be used to describe the properties of the image; the sample values can be used to encode the properties (shape) of the objects.

It is clear that the situation is more complicated in the real images and one centroid will not be enough to cover more complicated image structures. Therefore, we divide the whole image into the cells. Instead of the centroid, the gravity centers of each cell are used; the distance is computed from these points to all other points inside each cell.

The visualizations of geodesic distances inside the cells of different sizes are shown in Fig. 3.14. Based on the cell sizes, information with various levels of details is obtained. To compress the information con-
3.2. Distance-based Descriptors

Figure 3.14: The visualization of the distance function values inside each cell. The example of face image (a). The sizes of cell $15 \times 15$ (b), $25 \times 25$ (c), $35 \times 35$ (d).

Figure 3.15: An example of $9 \times 9$ cells that are grouped into one block. In this particular case, from each cell, four values (depicted by green color) are used in the feature vector. The centers of cells are represent by red color. From the centers, the distance is computed within the cells.

In our experiments, we use the overlapping blocks; the second half of one block corresponds to the first half of the next block. The final feature vector is then used as an input for the SVM classifier.

In [69], we tested the properties of the proposed method for the case of solving the problem of face detection using the classical sliding window technique. The examples of visualization of distance function values of face training images are shown in Fig. 3.16.

In [68], we tested the detector in the area of pedestrian detection. The examples of visualization of distance function values of pedestrian
3.2. Distance-based Descriptors

Training images are shown in Fig. 3.17.

Figure 3.16: The visualization of the distance function values inside cells (9 × 9 pixels); faces.

Figure 3.17: The visualization of the distance function values inside cells (9 × 9 pixels); pedestrians.
4 Summary of the Results

In the previous sections, we have presented the papers in which we propose the ways how the values of energy function (after the energy transfer process) and geodesic distance values can be encoded into the feature vector that is used as an input for the trainable classifier to create the object detector (face, pedestrian, car). Due to the fact that some of the presented methods were tested on different datasets or different objects (e.g., faces only), in this section, we compare the methods on the reputable face and pedestrian datasets for the final comparison of the proposed methods.

4.1 Face Detection

In this chapter, we compare the proposed descriptors in the area of face detection. For the training phase of face detection, we used 2300 faces and 4300 non-faces similarly as in the previous chapters. The faces were obtained from the BIOID database combined with the Extended Yale Face Database B [31]. The negative set consists of 3000 images that were obtained from the MIT-CBCL database combined with the 1300 hard negative examples. In the detection process, the sliding window scanned 10 different resolutions of input image. For the test, we used the following methods based on the previously presented ideas; the approaches with the mean temperature inside the cells, with DCT, with the hierarchy improvements, and with the geodesic distance.

The method (with the mean temperature of cells) presented in 3.1.1 is slightly modified in this test. The idea of mean temperature inside each cells is preserved but the image inside the sliding window is divided into the blocks and cells as is depicted in the Fig 4.1. This modification is done due to the promising results that were obtained using this settings.

Figure 4.1: The blocks and cells inside the input image (sliding window).
4.1. Face Detection

in the version with DCT (Section 3.1.5) and geodesic distance (Section 3.2). The next reason is to create conditions how to fairly compare the approaches. The method is designed with the size of sliding window 80 × 80 pixels, with the size of temperature sources 1 pixel, with the 4-pixel distance between the sources, with 100 iterations for the transfer of temperature, with the size of block 16×16 pixels, with the size of cell 8×8 pixels (four cells inside each block), and with the horizontal step size between blocks 8 pixels (blocks are overlapped). The final feature vector consists of 324 descriptors for one position of 80 × 80 sliding window. This configuration represents the method in which only the mean temperature is investigated. The detector based on this configuration is denoted as ETF\textsubscript{mean} in the test.

The next method used in this test is the hierarchical version that is proposed in Section 3.1.4. We use the following settings. The size of sliding window is 80 × 80 pixels, the size of temperature sources is 1 pixel, the distance between the sources is 4 pixels, the number of iterations for the transfer of temperature is 100. The mean temperature is computed inside each cell at the levels 0, 1, 2, 3, 4, and the histogram of the temperature distribution is computed at the levels 0, 1, 2, 3. The final feature vector consists of 1541 descriptors for one position of 80 × 80 sliding window. The detector based on this configuration is denoted as ETF\textsubscript{hierarch} in the test.

The DCT version of the method proposed in Section 3.1.5 is tested with the following settings. The size of sliding window is 80 × 80 pixels, the size of temperature sources is 1 pixel, the distance between the sources is 4 pixels, the number of iterations for the transfer of temperature is 100, the size of block is 16×16 pixels, the size of cell is 8×8 pixels (four cells inside each block), and the horizontal step size between blocks is 8 pixels (blocks are overlapped). Based on the experiments in Section 3.1.5, we use the pattern from Fig. 3.11(b) in which the coefficients are grouped into horizontal, vertical, and diagonal form; this pattern is constructed to capture the information in different directions. The configuration gives 1296 descriptors for one position of sliding window. The detector based on this configuration is denoted as ETF\textsubscript{DCT} in the test.

The method that uses the geodesic distance to encode the properties of the objects is based on the following configuration. The size of sliding window is 72 × 72 pixels, the size of blocks is 18 × 18 pixels, the size of cells is 9 × 9 pixels, and the horizontal block step size is 9 pixels. This configuration gives 784 descriptors for one position of sliding window. Each window consists of 49 overlapping blocks and each block consists of 4 cells, i.e. 196 cells are defined in each window. Finally, each cell is described using 4 distance values, i.e. 784 descriptors are used (196×4). The name of the configuration in the test is Geo\textsubscript{dist}. The post-processing step that includes the removal of temperature sources (presented in Sec-
4.1. Face Detection

For comparison, we used the detectors that are based on the HOG features, LBP (Local Binary Patterns) features [37] and Haar features (Viola-Jones detection framework). For the HOG features, we used the size of the sliding window $80 \times 80$ pixels like for the proposed features. We used the typical parameters of HOG descriptors. The size of block is $16 \times 16$ pixels, the size of cell is $8 \times 8$ pixels, the horizontal step size is 8 pixels, and 9 bins are used. This configuration gives 2916 HOG descriptors for one position of sliding window. The configuration is denoted as HOG. The Support Vector Machine classifier with Radial basis function kernel is trained over the HOG descriptors to create the final classifier (similarly also in the proposed descriptors).

We resized the training images to $19 \times 19$ pixels for the detector based on the Viola-Jones detection framework and for the LBP-based detector. The detector based on the Viola-Jones detection framework is denoted as Haar, the detector based on the LBP features is denoted as LBP. It is important to mention that for these features, we created the cascade classifiers. For HOG, Haar, and LBP, we used the identical training sets like for the proposed features (2300 positive and 4300 negative samples).

To test the detectors, we used 450 images from the Faces in the Wild dataset [7]. Before the process of performance calculation, the positive detections were merged to one if at least 3 positive detections hit approximately one place in the image.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Sensitivity</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETF$_{mean}$</td>
<td>90.23%</td>
<td>78.97%</td>
<td>84.23%</td>
</tr>
<tr>
<td>ETF$_{hierarch}$</td>
<td>90.57%</td>
<td>82.05%</td>
<td>86.70%</td>
</tr>
<tr>
<td>ETF$_{DCT}$</td>
<td>97.21%</td>
<td>83.42%</td>
<td>89.79%</td>
</tr>
<tr>
<td>Geo$_{dist}$</td>
<td>97.24%</td>
<td>90.26%</td>
<td>93.62%</td>
</tr>
<tr>
<td>HOG</td>
<td>75.54%</td>
<td>95.04%</td>
<td>84.18%</td>
</tr>
<tr>
<td>Haar</td>
<td>80.85%</td>
<td>87.11%</td>
<td>83.86%</td>
</tr>
<tr>
<td>LBP</td>
<td>72.58%</td>
<td>73.54%</td>
<td>72.11%</td>
</tr>
</tbody>
</table>

The detection results are shown in Table 4.1. From the results, it can be seen that the method based on the mean temperature only (ETF$_{mean}$) achieved the worst F1 score (84.23%) among the proposed methods. It is due to the fact that we use the more complete testing environment than in [75] and Section 3.1.1. However, the F1 score of ETF$_{mean}$ is relatively similar to Haar and HOG (84.18% and 83.86%), and better than LBP (72.11%). We note that that this method uses the lowest number of descriptors for one position of sliding window (324). The ETF$_{hierarch}$
4.1. Face Detection

Figure 4.2: The face detection result examples of $ETF_{DCT}$ (first column), $Geo_{dist}$ (second column), and $HOG$ (third column) methods. The results are without the postprocessing (the detection results are not merged).

method achieved the F1 score of 86.10% which is better than the score of $ETF_{mean}$. The $ETF_{DCT}$ method achieved the F1 score of 89.79%. It is the best detection result among the energy based methods. The best detection results (among the proposed methods) were achieved by the method based on the geodesic distance (F1 93.62%). This method outperformed the state-of-the-art methods as well as the methods based on the energy transfer. Moreover, the geodesic distance based detector uses 784 descriptors only (for one position of sliding window) in comparison with the HOG based detector that uses 2916 descriptors (for one position of sliding window).
4.1. Face Detection

Figure 4.3: The face detection results of Geo_{dist} method. The results are with the postprocessing (the detection results are merged).
4.1. Face Detection

Figure 4.4: The face detection results of Geo_dist method in which the method fails. The results are with the postprocessing (the detection results are merged).

Examples of the differences between the detection results of the methods that achieved the best detection results (ETF_DCT, Geo_dist, HOG) are shown in Fig. 4.2. From the examples, it can be seen that the Geo_dist method has a good balance between the number of false positives and false negatives. It is also visible in Table 4.1. The distance-based method achieved the precision of 97.24% and sensitivity of 90.26% which are the best results in the whole tests. Comparing to the other methods (especially to the HOG detector), we can conclude that the properties of the faces can be effectively encoded using the proposed distance-based detector with the reasonable dimensionality of feature vector without need for the methods for reducing the feature space (e.g. PCA). The examples of detection results of the proposed distance based method are shown in Fig. 4.3. The examples in which this method failed are shown in Fig. 4.4.
4.2 Pedestrian Detection

In next experiments, we will focus on the pedestrian detection.

In this section, we compare the proposed descriptors in the area of pedestrian detection. For the training phase, we collected 2500 positive images and 10000 negative images. We combined the pedestrian images from the CBCL Pedestrian Database \[13\] with the images from the Daimler benchmark \[20\] for the positive set. For the negative set, the images were randomly sampled from the INRIA Person Dataset \[18\].

Similarly to the previous face detection tests, we use the following methods based on the previously presented ideas. The approach with the mean temperature inside the cells, with DCT, with the hierarchical modification, and with the geodesic distance.

The method based on the mean temperature only (Section 3.1.1) is designed as follows. The size of sliding window is $64 \times 128$ pixels, the size of temperature sources is 1 pixel, the distance between the sources is 2 pixels, the number of iterations for the transfer of temperature is 100, the size of block is $16 \times 16$ pixels, the size of cell is $8 \times 8$ pixels (four cells inside each block), and the horizontal step size between blocks is 8 pixels (blocks are overlapped). The name of the configuration in the test is $ETF_{mean}$.

The DCT version of the proposed method (Section 3.1.5) is designed as follows. The size of sliding window is $64 \times 128$ pixels, the size of temperature sources is 1 pixel, the distance between the sources is 2 pixels, the number of iterations for the transfer of temperature is 100, the size of block is $16 \times 16$ pixels, the size of cell is $8 \times 8$ pixels (four cells inside each block), and the horizontal step size between blocks is 8 pixels (blocks are overlapped). Based on the face detection results, we selected the Fig. 3.11(b) pattern. The final feature vector consists of 1680 descriptors for one position of $64 \times 124$ sliding window; each window consists of 105 overlapping blocks and each block consists of 4 cells, i.e. 420 cells are defined in each window. Finally, each cell is described using 4 values ($1 \times DC + 3 \times AC$), i.e. 1680 descriptors are used ($420 \times 4$). The name of the configuration in the test is $ETF_{DCT}$.

The hierarchical version of the proposed method (Section 3.1.4) is designed with the following settlings. The size of sliding window is $64 \times 128$, the size of temperature sources is 1 pixel, the distance between the sources is 2 pixels, the number of iterations for the transfer of temperature is 100. The mean temperature is computed inside each cell of 64, 32, 16, 8, 4 pixels. The histogram of the temperature distribution is computed inside each cell of 64, 32, 16, 8 pixels. The final feature vector consists of 2602 descriptors for one position of $64 \times 124$ sliding window. The name of the configuration in the test is $ETF_{hierarch}$. 
4.2. Pedestrian Detection

The method that uses the geodesic distance values to encode the properties of the objects is tested with the configuration that achieved the best detection results in Section 3.2. The detector is designed with the following settings. The size of sliding window is $88 \times 176$ pixels, the size of blocks is $22 \times 22$ pixels, the size of cells is $11 \times 11$ pixels, and the horizontal block step size is $11$ pixels. This detector has 3360 descriptors for one position of sliding window; each window consists of 105 overlapping blocks and each block consists of 4 cells, i.e. 420 cells are defined in each window. Finally, each cell is described using 8 distance values, i.e. 3360 descriptors are used ($420 \times 8$). The name of the configuration in the test is $Geo_{dist}$.

For comparison, we used the detectors that are based on the HOG features, LBP (Local Binary Patterns) features [37] and Haar features (Viola-Jones detection framework). For the HOG-based detectors, we created two detectors with the different settings. The detectors are denoted by $HOG_1$ and $HOG_2$. We used the classical sizes of blocks and cells. Nevertheless, we used two different numbers of bins to create different numbers of descriptors for one position of sliding window. Therefore, we can compare the proposed detectors and the HOG-based detectors in the sense of descriptor numbers. The parameters of $HOG_1$ are as follows. The size of block is $16 \times 16$ pixels, the size of cell is $8 \times 8$ pixels, the horizontal step size is 8 pixels, and the number of bins is 4. The detector based on these parameters produces 1680 descriptors for one position of sliding window. The parameters of $HOG_2$ are as follows. The size of block is $16 \times 16$, the size of cell is $8 \times 8$ pixels, the horizontal step size is 8, and the number of bins is 8. The detector based on these parameters produces 3360 descriptors for one position of sliding window. The training images (for the HOG detectors) were resized to the $64 \times 128$ pixels (the size of sliding window was also set to this size).

For the detectors based on the Viola-Jones detection framework with the Haar features and with the features that are based on LBP, we created the cascade classifiers. For these classifiers, we resized the training images to the size of $24 \times 48$ pixels. The detectors are denoted as $LBP$ and $Haar$.

For the test, we collected 85 images from [18]. In the detection stage, the sliding window scanned 10 different resolutions of input image. Before the process of performance evaluation, the positive detections were merged to one if at least 3 positive detections hit approximately one place in image and we used the same training and testing images for all methods. In Table 4.2, the detection performances are shown.

Let us look at the detection performance of the proposed methods $ETF_{mean}$, $ETF_{hierarch}$, $ETF_{DCT}$, $Geo_{dist}$. Based on the face detection results presented in the previous section (face detection), we can conclude that the pedestrian results are not so different. The $ETF_{mean}$ detector has
4.2. Pedestrian Detection

Table 4.2: The pedestrian detection results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Sensitivity</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETF&lt;sub&gt;mean&lt;/sub&gt;</td>
<td>78.26%</td>
<td>65.45%</td>
<td>71.29%</td>
</tr>
<tr>
<td>ETF&lt;sub&gt;hierarch&lt;/sub&gt;</td>
<td>80.89%</td>
<td>76.97%</td>
<td>78.88%</td>
</tr>
<tr>
<td>ETF&lt;sub&gt;DCT&lt;/sub&gt;</td>
<td>82.64%</td>
<td>72.12%</td>
<td>77.02%</td>
</tr>
<tr>
<td>Geo&lt;sub&gt;dist&lt;/sub&gt;</td>
<td>94.27%</td>
<td>89.70%</td>
<td>91.93%</td>
</tr>
<tr>
<td>HOG&lt;sub&gt;1&lt;/sub&gt;</td>
<td>62.26%</td>
<td>80.00%</td>
<td>70.03%</td>
</tr>
<tr>
<td>HOG&lt;sub&gt;2&lt;/sub&gt;</td>
<td>82.82%</td>
<td>81.82%</td>
<td>82.32%</td>
</tr>
<tr>
<td>Haar</td>
<td>88.55%</td>
<td>89.09%</td>
<td>88.82%</td>
</tr>
<tr>
<td>LBP</td>
<td>86.93%</td>
<td>80.61%</td>
<td>83.65%</td>
</tr>
</tbody>
</table>

the higher number of the false positives (F<sub>1</sub> 71.29%), and the ETF<sub>hierarch</sub>, ETF<sub>DCT</sub> achieved relatively similar results (F<sub>1</sub> 78.88% and 77.02%).

Similarly as in the face detection, the detector based on the geodesic distance achieved the best detection result in the test (F<sub>1</sub> 91.93%). This detector even overcame all tested detectors that use HOG, Haar, and
4.2. Pedestrian Detection

Figure 4.5: The differences between the detection results of Haar detector and the proposed detector that uses the Geo_dist configuration. The results are without the postprocessing (the detection results are not merged).
Figure 4.6: The pedestrian detection results of $Geo_{dist}$. The results are with the postprocessing (the detection results are merged).
LBP features. From these detectors, the Haar-based detector achieved the best detection result (F1 88.82%), despite the fact that this detector is usually used in the face detection area. Moreover, it is important to note that the Haar-based detector combined with the AdaBoost classifier was trained for 45 hours on the 16-core CPU (2x Intel Xeon CPU E5-2640 v2). In contrast, HOG and the proposed methods (that use SVM classifier) were trained for approximately 5-10 minutes (depending on the dimensionality of the feature vector).

Finally, the difference between the detection results of the Haar-based detector and the proposed distance-based detector can be seen in Fig. 4.5. The examples of pedestrian detection results of the proposed distance based method are shown in Fig. 4.6. The examples in which this method failed are shown in Fig. 4.7.

4.3 Conclusion

At present, we are the witnesses of the situation that the detectors based on the sliding window are used more and more widely. The principle itself is straightforward. The image is scanned by an area (usually, a rectangular area is used). For each position and for a certain range of the sizes of the area, the goal is to detect whether or not the area contains the object that is to be recognised. The recognition is usually done on the basis of a certain set of numbers creating a feature vector, and a certain trainable classifier. Naturally, the recognition can only be successful to the extent that is determined by the information contained in the feature vector. Therefore, determining the feature vector is a key point.

The way how the feature vector is created should be universal, i.e. usable for various types of objects that should be recognised. The information contained in the vector should be sufficient to recognise the desired types of objects correctly. Finally, the information should be presented concisely and clearly (the number of vector entries should not be high) since, otherwise, the classifiers would require bigger training sets and longer times for training. It is not an easy matter to meet all these requirements simultaneously. Many approaches how the feature vector could be constructed appeared in the past (e.g. [1, 18, 57, 62]). In spite of this, we believed that the research in this area is not finished, which was the main motivation for this work. The challenge was to create the feature vector in such a way that it outperforms the respected methods at least in certain situations.

In this dissertation thesis, we present an approach whose main idea is to solve energy transfer (e.g. heat transfer) in the environment whose properties are defined by the values of contrast in the corresponding image that is to be analysed. Once we now how the energy is distributed after a chosen time of transfer (e.g. the function is known describing
the temperature in particular points), we derive the feature vectors from this information. We regard the new idea that has just been presented as a main contribution of this work. Naturally, the idea itself is not enough. It was also necessary to explain adequately why it should work and should be better, and to verify the expectations experimentally. This was presented in several papers. Since the amount of energy that is transferred between two points can be viewed as something which is similar to measuring a distance, we also generalised the approach for the distances. We give a more detailed overview of contributions in the subsequent paragraphs.

In Section 3.1.1, we proposed the first way in which the energy values inside the cells are calculated. In essence, the method is inspired by the HOG features. The sliding window is divided into the regular cells. However, instead of the gradient orientations, the mean of energy distribution is used. The method was presented in [75]. We tested the method in the area of face detection with the promising results which became the motivation for further research in this area.

In Section 3.1.2, we presented the modification of the proposed method. In this modification, the neighborhood of each energy source is investigated. The image (inside the sliding windows) is divided into the blocks. We use the gravity centers of the blocks as the places for the energy sources. Since the blocks are divided into small cells (Fig. 3.5), the energy in the neighborhood of the sources can be investigated. The method was tested in the area of pedestrian detection in [74].

In Section 3.1.3, we proposed the experiments with the different iterations (time for the energy transfer). The goal was to explore the influence of this parameter on the detection results. The experiments was presented in [73]; the method was tested in the area of car detection.

In Section 3.1.4, we presented the hierarchical way of the proposed (energy) features. In the method, the different cell sizes are used. In addition to this, we experimented with the histograms of energy distribution, and we tested this approach in the area of face detection. The results were presented in [72].

In Section 3.1.5, we presented the combination of energy values with DCT to obtain an efficient feature vector. In [71], the first version of this method was presented in the area of face detection. The extended experiments of the method were proposed in the areas of pedestrian detection and face detection in [102].

In addition to these energy-based methods, we proposed another way how to encode the appearance of object in Section 3.2. The method uses a similar principle as the energy-based method. Instead of the energy transfer, the method is based on the distance function. The method was proposed in [104] with very promising results. In [103], we proposed an extension of the method, thanks to which the method was effectively
used in the area of pedestrian detection.

We also compared the proposed methods with each other and with the state-of-the-art methods in the area of sliding window detectors. The comparison results are shown in Sections 4.1, and 4.2. In Section 4.1, the experiments are focused on the problem of face detection. In Section 4.2, the results of pedestrian detection are presented. From the results we obtained, we conclude that the distance-based method achieved the best detection results in all experiments (faces and pedestrians).

In future works, we would like to focus on further improvements (experiments) of the distance-based method. For example, on the possibility of using the histograms of distance values. The inspiration for extensions can also be found in LBP. It means that we will try to find a way how to encode distance values using a binary code. In next works, we will also try to combine the distance values with the rectangular (Haar-like) features. The main idea of Haar-like features is based on the fact that the appearance of objects can be encoded by the differences of mean intensities between the rectangular areas. We would like to use this idea. However, instead of the differences of intensities, the distance differences between the rectangular areas can be used to encode the shape of objects.

In our experiments, we used the SVM classifier. It is worth mentioning that various classifiers can be used as well (e.g. Random Forests). The experiments with different types of distances can also be carried out in future works.
5 Bibliography


42


45


6 Author’s Bibliography

6.1 Publications Related to the Thesis


6.1. Publications Related to the Thesis


