Distance-based Descriptors for Pedestrian Detection

Radovan Fusek and Eduard Sojka

Technical University of Ostrava, FEECS, Department of Computer Science,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
{radovan.fusek, eduard.sojka}@vsb.cz

Abstract. In this paper, we propose an improvement of the detection approach that is based on the distance function. In the method, the distance values are computed inside the image to describe the properties of objects. The appropriately chosen distance values are used in the feature vector that is utilized as an input for the SVM classifier. The key challenge is to find the right way in which the distance values should be used to describe the appearance of objects effectively. The basic version of this method was proposed to solve the face detection problem. As we observed from the experiments, the method in the basic form is not suitable for pedestrian detection. Therefore, the goal of this paper is to improve this method, and create the pedestrian detector that outperforms the state-of-the-art detectors. The experiments show that the proposed improvement overcomes the accuracy of the basic version by approximately 10%.

Keywords: object detection, sliding window, distance function, SVM

1 Introduction

In the process of detection, we use the sliding window technique that 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. Many windows represent the result of the scanning process. A vector of features is obtained for each window. The vector is then used as an input for the classifier (in our case, the SVM classifier). During the classification process, some windows are marked as containing the object. Using the sliding window approach, multiple positive detections may appear, especially around the objects. The detections are merged into the final bounding box that represents the resulting detection. The classifier that determines whether or not the window contains the object is trained over the training set that consists of positive and negative images. The key point is to find which quantities should be used to effectively encode the image inside the sliding window.

In the method we propose, the geodesic distances are computed inside the sliding window to describe the objects. The preliminary version of this approach was presented in [9], where the properties of the method were shown in the area

of face detection. The image inside the sliding window is divided into the cells in which the distances are computed. The distance values are used to create the feature vector. This vector is then used as an input for the SVM classifier. Unfortunately, only one way of how to encode the distance values inside the image is shown in [9]. We experimented with this detector, and we observed that the method (in the version from [9]) is not suitable for describing the pedestrians. Therefore, we propose an extension of the method, thanks to which the method can be effectively used in the area of pedestrian detection. The improvement is based on extending the way of how the distance values are encoded. Using this improvement, we are able to outperform the basic version of the distance detector as well as the state-of-the-art detectors in the area.

The rest of this paper is organized as follows. In the next section, we provide the overview of the features that can be extracted from images. We focus on the methods that use the sliding window technique. Thereafter, we describe the main idea of the proposed improvement. Finally, we show the experiment results. The last section is a conclusion.

2 Related Work

The Haar-like features represent a popular and famous approach that is very often used for object description. 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 brighter than the areas of the eyes; the regions bellow or on top of the eyes have different intensities that the eyes itself. These specific characteristics can be encoded by one two-rectangular feature, and the value of this feature can be calculated as the difference between the sum of the pixel values inside the rectangles. The Haar-like features were proposed by Papageorgiou and Poggio [16]. In their paper, the Haar-like features are combined with the SVM classifier to create the face, car, and pedestrian detectors. Viola and Jones [20] proposed the very efficient way of how these features can be used in the combination of the integral image and AdaBoost algorithm. An extension of the Haar feature set was presented by Lienhart et al.[14]. The authors proposed the 45° rotated features. The rotated features are able to reduce false detections and achieve more accurate results. A comparison of face and facial feature detectors based on the Viola-Jones general object detection framework was presented in [2]. Recently, the improvement of Haar-like features for efficient object detection under a wide range of illumination conditions was proposed in [18].

The local binary patterns (LBP) represent an another way of how to encode the shape of objects. The LBP operator was proposed by Ojala et al.[15] for texture analysis. Since then, due to their positive properties (e.g. invariance to lighting changes), LBP were used in many detection tasks, especially for facial analysis. For example, LBP were used to create the face detector in low-resolution images in [10]. In [22], multi-block local binary patterns (MB-LBP) for face detection were proposed. In this method, the authors encode the rectangular

regions by the local binary pattern operator and the gentle AdaBoost is used for feature selection. The authors showed that MB-LBP are more distinctive than the Haar-like features and the original LBP features. The comprehensive study of facial expression recognition methods using LBP was proposed in [19]. The survey of facial analysis methods using LBP was presented in [11].

The histograms of oriented gradients (HOG) proposed in [5] are considered as the state-of-the-art method in the area of pedestrian detection. In HOG, the gradient magnitudes and orientations are computed and composed into the feature vector that is used as an input for the SVM classifier. Many methods and applications based on HOG were presented in recent years. The method that combines principal component analysis (PCA) with the HOG features was proposed in [12]. Co-occurrence histograms of oriented gradients (CoHOG) that encode the spatial relationship between the pairs of pixels were proposed in [21]. The face recognition method using the HOG features was proposed in [6].

Due to the fact that the geodesic distance is used in the paper, we mention the works in which this distance was used in the area of image processing. Image segmentation and object detection methods based on the geodesic distance were presented in [17, 7, 1, 4].

3 Proposed Method

The main idea of the presented detector is based on the fact that the shape (appearance) of objects can be described using the distance values. If we speak about the distance values in this paper, we have in mind the geodesic distance. However, any appropriate distance function can be used (e.g. resistance distance, diffusion distance).

Consider the image that consists of one object of constant brightness. Suppose the point located in the mentioned object. From this point, the distances are computed inside the whole image. After that, the distance values can be investigated. Since the object has constant brightness, the distances inside the object are shorter than the distances investigated outside the object, which is the fact that the edges of the object create the barrier, and this barrier is reflected in the distance values. Therefore, we can conclude that the values of distances are different inside and outside of object and this assumption can be used to encode the properties of objects.

In the real images (e.g pedestrian images), the situation is more complicated. One point placed inside the objects will not probably be enough to cover all areas and important places of objects. Therefore, the image inside the sliding window is divided into the small cells. Inside each cell, the center point of the cell is determined. From this point, the distances are calculated to all remaining points within the cell. In [9], the authors proposed only one pattern how to encode the distance values. This pattern is shown in Fig. 1. The red point represents the center point of cell. The green points represent the places around the center point that are taken into account; i.e. the distances from the center point to these places are than used in the feature vector. To achieve better resistance to

illumination changes, the cells are grouped into the larger blocks (similarly to HOG) and the distance values are normalized within the blocks.

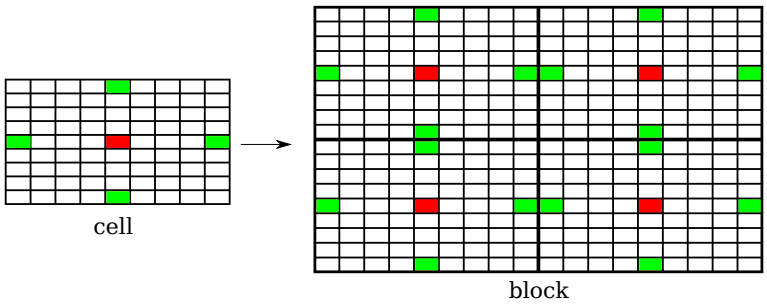

Fig. 1. An example of 9×9 cells that are grouped into one block. In this particular case, four values are used in the feature vector (depicted by green color) from each cell. The centers of cells are represented by red color. From the centers, the distances are computed.

We experimented with this pattern and we observed (as can be seen in Section 4) that this pattern is not suitable for pedestrian detection. The four values that are used in each cell are not enough to describe the shape of pedestrians. Therefore, we propose the extended versions of this pattern. The proposed patterns can be seen in Fig. 2.

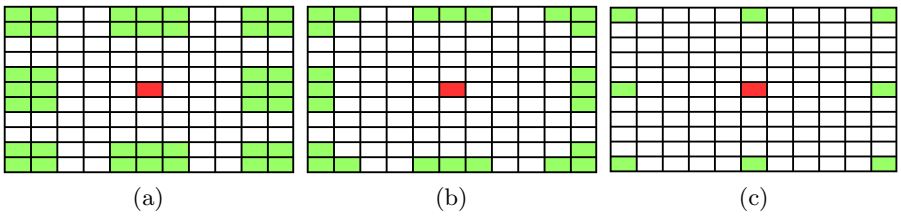

Fig. 2. An example of three 11×11 cell patterns that are used inside the blocks. In this particular case, eight values (areas) are used in the feature vector (depicted by green color). The centers of cells are represented by red color. From the centers, the distances are computed.

All proposed patterns use eight values from each cell in the feature vector. In the patterns (a) and (b), the averages of distance values are calculated inside the green areas. In the pattern (c), one value from each green point is used. In the next section, we propose the pedestrian detectors based on the original

pattern and the new patterns to compare these detectors with the state-of-the-art approaches.

4 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 [3] with the images from the Daimler benchmark [8] for the positive set. For the negative set, the images were randomly sampled from the INRIA Person Dataset [5].

Fig. 3. The visualization of the distance function values inside each cell of the $Dist_2$ configuration. The values of distance function are depicted by the level of brightness.

To find the optimal sizes of sliding windows, cells, and blocks, we experimented with the following three configurations: $Dist_1$, $Dist_2$, $Dist_3$. The configurations use the original pattern from Fig. 1. Some examples of the values of distance function are shown in Fig. 3.

The parameters of the $Dist_1$ configuration are as follows. The size of sliding window is 70×140, the size of blocks is 14×14, the size of cells is 7×7, the horizontal block step size is 7. The detector using this configuration has 2736 descriptors for one position of sliding window; each window consists of 171 overlapping blocks and each block consists of 4 cells, i.e. 684 cells are defined in each window. Finally, each cell is described using 4 distance values, i.e. 2736 descriptors are used (684×4).

The parameters of the $Dist_2$ configuration are as follows. The size of sliding window is 72×144, the size of blocks is 18×18, the size of cells is 9×9,

the horizontal block step size is 9. The detector using this configuration has 1680 descriptors for one position of sliding window; each window consists of 105 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. 1680 descriptors are used (420×4).

The parameters of the $Dist_3$ configuration are as follows. The size of sliding window is 88 × 176, the size of blocks is 22 × 22, the size of cells is 11 × 11, the horizontal block step size is 11. The detector using this configuration has 1680 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 4 distance values, i.e. 1680 descriptors are used (420×4).

Let us compare these configurations. For the test, we collected 85 images from [5]. In the detection stage, the sliding window scanned 10 different resolutions of input image. Before the process of performance calculation, the positive detections were merged to one if at least 3 positive detections hit approximately one place in image. In Table 1, the detection performances are shown. For evaluation, we use the following quantities; Precision = TP/(TP+FP), Sensitivity = TP/(TP+FN), F1 score (harmonic mean of precision and sensitivity) = 2 × Precision × Sensitivity/(Precision + Sensitivity); TP = number of true positives, FP = number of false positives, FN = number of false negatives.

Table 1. The pedestrian detection results of the $Dist_1$, $Dist_2$, and $Dist_3$ configurations.

	Precision	Sensitivity	F1 score
$Dist_1$	81.21%	77.91%	79.53%
$Dist_2$	88.98%	68.48%	77.40%
$Dist_3$	79.89%	84.24%	82.01%

From Table 1, it can be seen that the detection results are not convincing. As we mentioned in the previous section, this experiment shows that if only the four values from each cell are included into the feature vector, it is non sufficient to describe the shape of pedestrian properly. However, due to the fact that the best detection results achieved the detector that used the $Dist_3$ configuration (F1 82.01%), the further experiments were based on this detector, because we wanted to verify whether this detector could be improved using the proposed extended patterns.

Let us consider the detector $Dist_{3(a)}$ with the same settings like in $Dist_3$, however with the change that the pattern is from Fig. 2(a). Similarly, the pattern from Fig. 2(b) is used in $Dist_{3(b)}$, and the pattern from Fig. 2(c) is used in $Dist_{3(c)}$. The new detectors have 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 (420×8). The detection results of these detectors are shown in Table 2.

Table 2. The pedestrian detection results of the $Dist_{3(a)}$, $Dist_{3(b)}$, and $Dist_{3(c)}$ configurations.

	Precision	Sensitivity	F1 score
$Dist_{3(a)}$	87.95%	88.48%	88.22%
$Dist_{3(b)}$	90.12%	88.48%	89.30%
$Dist_{3(c)}$	94.27%	89.70%	**91.93**%

From this table, it can be seen that with the use of the proposed patterns, the detectors are more efficient (by approximately 10%) than the original proposed pattern. The best detection results were achieved by the detector that used the $Dist_{3(c)}$ configuration with the pattern from Fig. 2(c). On the basis of the results, we can conclude that (in this particular application) using a simple distance value of each green area (in Fig. 2(c)) seems to be a more efficient approach than using the mean of the distance values.

Table 3. The summary of pedestrian detection results.

	Precision	Sensitivity	F1 score
$Dist_{3(c)}$	94.27%	89.70%	**91.93**%
HOG	82.82%	81.82%	82.32%
$Haar$	88.55%	89.09%	88.82%
LBP	86.93%	80.61%	83.65%

To compare the proposed $Dist_{3(c)}$ detector that achieved the best detection results with the state-of-the-art methods in the area of sliding window detectors, we used the detectors that are based on the HOG features, LBP (Local Binary Patterns) features [13] and Haar features (Viola-Jones detection framework). For HOG, we created the detector with the classical settings of HOG. The detector is denoted as HOG. The parameters of HOG are as follows. The size of block is 16×16, the size of cell is 8×8, the horizontal step size is 8, the number of bins is 8. The detector based on these parameters gives 3360 descriptors for one position of sliding window. The training images (for the HOG detector) were resized to 64×128 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 24×48 pixels. The cascade classifier of LBP features had 21 stages; the cascade classifier of Haar features had 24 stages. The detectors are denoted as LBP and $Haar$.

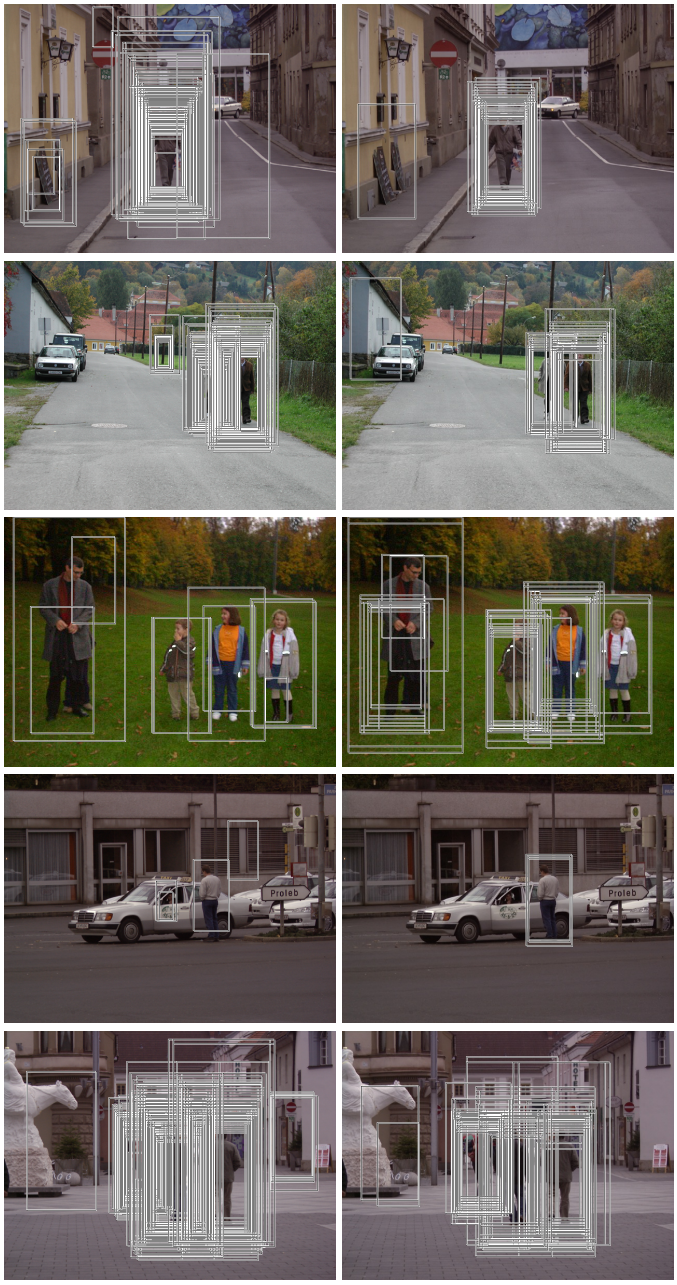

Fig. 4. The differences between the detection results of $Haar$ detector and the proposed detector that uses the $Dist_{3(c)}$ configuration. The results are without the postprocessing (the detection results are not merged).

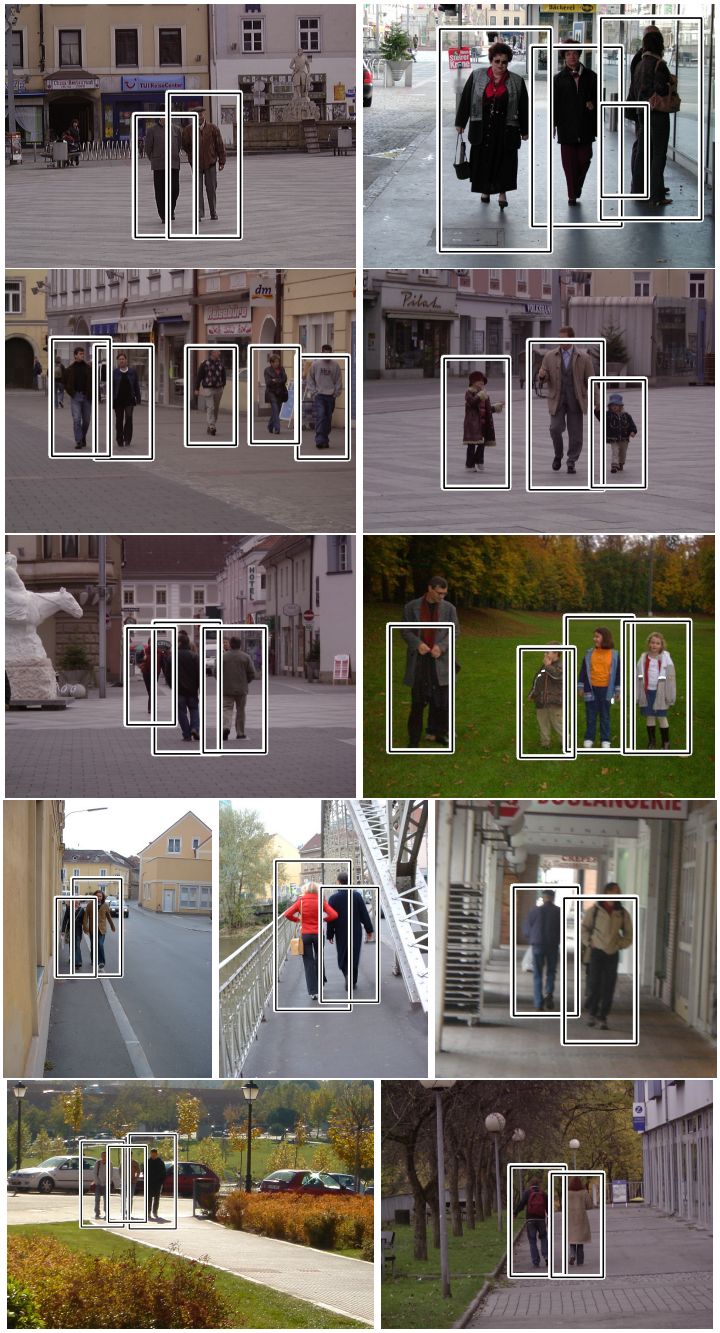

Fig. 5. The pedestrian detection results of the $Dist_{3(c)}$ configuration. The results are with the postprocessing (the detection results are merged).

We used the same training and testing images for all methods. In Table 3, the detection performances are shown.

Similarly as in the previous tests, the detector based on the geodesic distance achieved the best detection result in the test (F1 91.93%). This detector even overcame all tested detectors that use HOG, Haar, and LBP features. From these non-proposed 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. Nevertheless, it is important to note that the Haar-based detec-

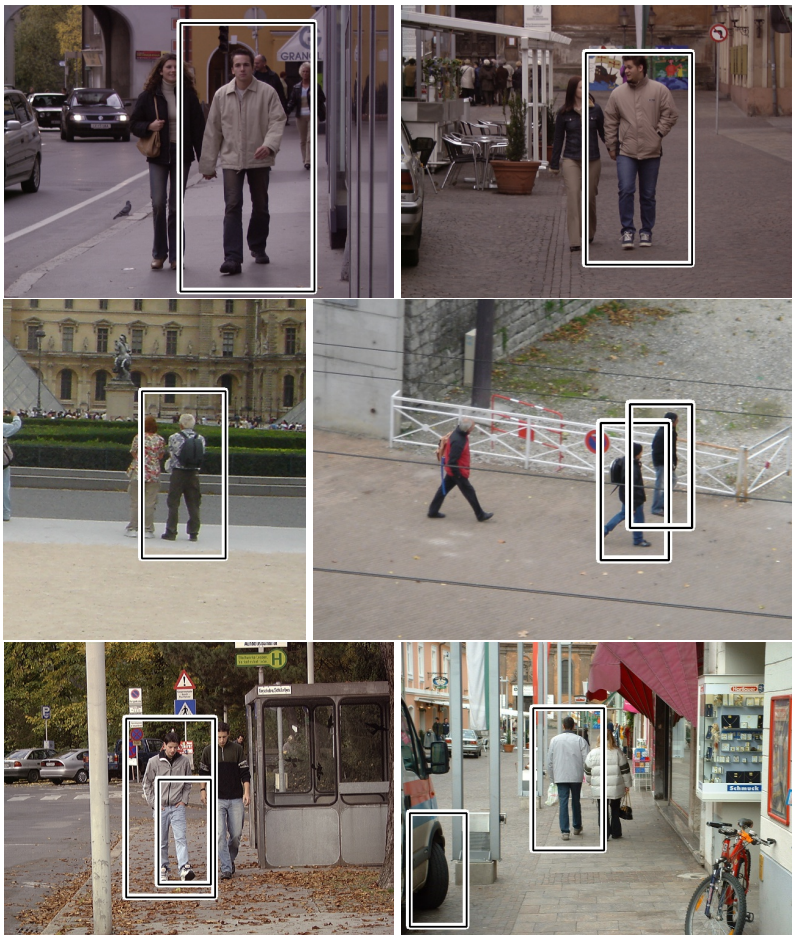

Fig. 6. The pedestrian detection results of the $Dist_{3(c)}$ configuration in that the method fails. The results are with the postprocessing (the detection results are merged).

tor with the AdaBoost classifier was trained for 45 hours on the 16-core CPU

(2× Intel Xeon CPU E5-2640 v2); all cores were used. HOG and the proposed method (SVM classifiers with RBF kernels) were trained for approximately 5-10 minutes (depending on the dimensionality of the feature vectors). In the proposed method, the calculation of geodesic distance and composing the feature vector took approximately 0.5 milliseconds for one position of sliding window on the 16-core CPU (2× Intel Xeon CPU E5-2640 v2); all cores were used. The recognition time depends on the chosen classifier.

The differences between the detection results of the Haar-based detector and the proposed distance-based detector can be seen in Fig. 4. The examples of pedestrian detection results of the proposed distance-based method are shown in Fig. 5. The examples in which this method failed are shown in Fig. 6.

5 Conclusion

In this paper, we presented an improvement of the distance-based object detector in which the distance values are used in the feature vector that is used as an input for the SVM classifier. The improvement is based on the extension of the basic pattern that was proposed in the first version of this detector. The newly proposed patterns outperformed the basic pattern by approximately 10%. The new version of detector also outperformed the state-of-the-art detectors in the area of pedestrian detection. In the current version of the presented approach, the geodetic distance is used. We leave the experiments with another distances for future work.

Acknowledgments

This work was supported by SGS in VSB - Technical University of Ostrava, Czech Republic, under the grant No. SP2015/141.

References

1. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. pp. 1–8 (Oct 2007)
2. Castrilln, M., Dniz, O., Hernndez, D., Lorenzo, J.: A comparison of face and facial feature detectors based on the viola-jones general object detection framework. Machine Vision and Applications 22(3), 481–494 (2011)
3. Center for Biological and Computational Learning: MIT CBCL Pedestrian Database #1 (2013), http://cbcl.mit.edu/software-datasets/PedestrianData.html
4. Criminisi, A., Sharp, T., Blake, A.: Geos: Geodesic image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) Computer Vision ECCV 2008, Lecture Notes in Comp. Sci., vol. 5302, pp. 99–112. Springer Berlin Heidelberg (2008)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886 –893 vol. 1 (june 2005)

6. Dniz, O., Bueno, G., Salido, J., la Torre, F.D.: Face recognition using histograms of oriented gradients. Pattern Recognition Letters 32(12), 1598 – 1603 (2011)
7. Economou, G., Pothos, V., Ifantis, A.: Geodesic distance and mst based image segmentation. In: European Signal Processing Conf., pp. 941–944 (2004)
8. Enzweiler, M., Gavrila, D.: Monocular pedestrian detection: Survey and experiments. Patt. Anal. and Mach. Intell., IEEE Trans. on 31(12), 2179–2195 (2009)
9. Fusek, R., Sojka, E.: Distance-based descriptors and their application in the task of object detection. In: Jiang, X., Hornegger, J., Koch, R. (eds.) Pattern Recognition, Lecture Notes in Computer Science, vol. 8753, pp. 488–498. Springer International Publishing (2014)
10. Hadid, A., Pietikainen, M., Ahonen, T.: A discriminative feature space for detecting and recognizing faces. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. vol. 2, pp. II–797–II–804 Vol.2 (2004)
11. Huang, D., Shan, C., Ardabilian, M., Wang, Y., Chen, L.: Local binary patterns and its application to facial image analysis: A survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Trans. on 41(6), 765–781 (Nov 2011)
12. Kobayashi, T., Hidaka, A., Kurita, T.: Neural information processing. chap. Selection of Histograms of Oriented Gradients Features for Pedestrian Detection, pp. 598–607. Springer-Verlag, Berlin, Heidelberg (2008)
13. Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: ICB. pp. 828–837 (2007)
14. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Image Processing. 2002. Proceedings. 2002 International Conference on. vol. 1, pp. I–900–I–903 vol.1 (2002)
15. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29(1), 51–59 (Jan 1996)
16. Papageorgiou, C., Poggio, T.: A trainable system for object detection. Int. J. Comput. Vision 38(1), 15–33 (Jun 2000)
17. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(3), 266–280 (2000)
18. Park, K.Y., Hwang, S.Y.: An improved haar-like feature for efficient object detection. Pattern Recognition Letters 42(0), 148 – 153 (2014)
19. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: A comprehensive study. Image Vision Comput. 27(6), 803–816 (May 2009)
20. Viola, P., Jones, M.: 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, pp. I–511 – I–518 vol.1 (2001)
21. Watanabe, T., Ito, S., Yokoi, K.: Co-occurrence histograms of oriented gradients for pedestrian detection. In: Wada, T., Huang, F., Lin, S. (eds.) Advances in Image and Video Technology, Lecture Notes in Computer Science, vol. 5414, pp. 37–47. Springer Berlin Heidelberg (2009)
22. Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z.: Face detection based on multi-block lbp representation. In: Proceedings of the 2007 international conference on Advances in Biometrics. pp. 11–18. ICB'07, Springer-Verlag, Berlin, Heidelberg (2007)