DEVELOPMENT OF IMAGE PROCESSING SYSTEM FOR PERSON DETECTION

DOMINIK STURSA, DANIEL HONC, PETR DOLEZEL

University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control, Pardubice, Czech Republic

DOI : 10.17973/MMSJ.2020_10_2020032

dominik.stursa@upce.cz

With the rise of modern technology in computer science and engineering, as well as with the growing population in big cities around the world, many new approaches for person detection have become a very interesting and demanding topic. Person detection is a necessary building block for people monitoring systems and, therefore, various detection methods must be inspected comprehensively in order to select the one with the most suitable performance and accuracy. In this paper, a set of different image processing techniques applied to images captured from a high angle were used for people detection. To be more specific, selected feature extraction techniques, like edge detectors, local binary patterns, pixel intensities or histograms of oriented gradients, were used in combination with several classification algorithms. The combinations of each feature extractor and its best classifier were selected for performance comparison. As a result of the comparison, the most suitable image processing method for person detection in high angle image is presented at the end of the paper.

KEYWORDS

Person detection, Image processing, Pattern recognition

1 INTRODUCTION AND RELATED WORK

With the advance of modern methods in computer science and device engineering, as well as with growing population in big cities among the world, many new approaches for person detection have become a very interesting and relevant topic.

Person detection is an initial step in every system for person tracking or counting. As such, person detection has an indispensable importance for safety in public transport, safety in crowded areas, safety in production areas or for purposes of surveillance systems. Person detection is the necessary building block for all people monitoring systems and as such, different detection methods must be examined in detail in order to select the one with the greatest performance and accuracy.

The problem of person detection has been studied by many researchers. It is definitely a very extensive topic and can be analyzed from several points of view.

Various technologies and physical principles like radar [Choi 2018], chemical sensors, pressure sensitive plates, infrared detectors [Ahmed 2005], 3D scanners [Akamatsu 2015], or image acquisition and its processing [He 2019] can be used for people detection.

If the physical principle is known, other criteria must be specified closely. One important criterion is the relationship between an object and the measurement system in terms of their relative location. This location directly affects features of the object, which can be detected by a measurement system (sensor).

Technology and scene knowledge then narrow down computer science methods, which can be applied in the specific scenario.

1.1 Detection based on image processing

Camera systems are being installed increasingly in private and public areas for security and surveillance reasons. More image processing methods are tested as image acquisition is more available. Therefore, people tracking, counting or detection systems are often implemented using computer vision techniques and video processing algorithms. Different ways of image and video processing can be considered in general. Methods based on shape feature, model learning, or area estimation are widely used [Wu 2014].

Shape feature evaluation, human pose, orientation, movement and appearance are typically used as inputs for further processing [Pore 2016].

If the detection system is going to be installed in public areas, it is appropriate to avoid identification of persons (especially faces). Thus, the acquisition from a high angle tends to be a natural solution of the mentioned difficulty. An example of an image from a high angle (above people heads) is shown in Fig. 1.

Figure 1. Image with people heads captured from above

Only a few approaches capturing a scene like the mentioned one have been proposed. The method, based on a combination of depth and RGB images, has been used by [Fu 2012]. Authors [Gao 2016] provide a technique, which combines convolutional neural networks and cascade Adaboost methods. Both articles do not consider strict downward image acquisition.

Detection based on exactly the same scenario was used in the previous authors' publication [Dolezel 2019], where the histograms of oriented gradients were used as the feature extraction technique, and support vector machine based pattern recognition systems were used as a classifier.

1.2 Example of industry application scenario

With the rise of individual production lines, the need for precise object detection is increasing. Not only products, but also people during the production must be detected to ensure safety.

A camera system installed on the hall ceiling is capable of covering a big workspace, where special zones can be defined. The hall can be divided into dangerous zones, robotic stations, conveyor belt areas, safe zones, etc. The top view has a great advantage for possible "collision" prediction, as the intersection of person and any other zone can be easily detected.

A small irregular space can be selected from the whole image, as the critical parts are mainly lines close to dangerous zones.

The selected space can be then examined to detect any object of interest. This binary detection can be realized with a combination of a feature extraction method and a classification technique.

2 PROBLEM FORMULATION AND METODOLOGY

The main aim of our project is to develop an efficient, robust and reliable person detector in real life gray scale images for an indoor application. The obvious initial attempt is to select proven and well-known image processing techniques, and adapt them to solve this task. Hence, this procedure, including a comprehensive validation of each technique, is depicted in the following sections of the article.

The individual images for processing are supposed to be derived from a video (frame sequence) acquired from above people's heads. Each image is then processed according to the procedure, which consists of four steps, as follows.

In the first step, the object image is obtained from a full-scale frame of the video stream. Image preprocessing is performed in the second step. In the third step, the feature extraction is provided, and as a result, the feature vector is gained. And the final step represents the classification of the object image using the feature vector.

In this approach, input to the detector is the size normalized gray scale image cropped from a frame in a real life RGB video. The output from the detector is the class of the object. As the detector should recognize only two classes (Head vs. Not head), the classification can be reduced to the binary problem. The illustration of the functionality for both classes is shown in Fig. 2.

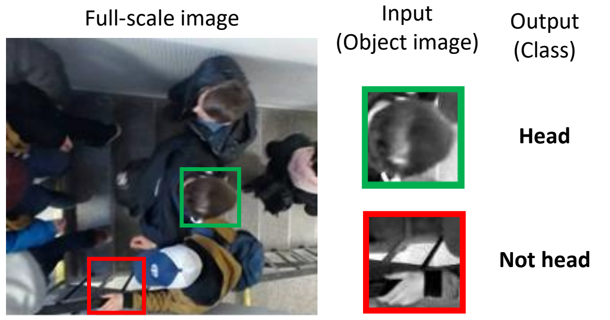

Figure 2. Person detector functionality

The structure of the detector is based on authors' previous experience published in [Skrabanek 2015, 2016 and 2017]. Globally, the structure fits on any pattern recognition system, but each part is redesigned in order to fit the person detection. All steps are described in detail in the next sections.

2.1 Image preprocessing

The object image is cropped from the RGB real life image. Preprocessing consists of two steps.

The conversion to gray scale must be done first, as the provided detector input is supposed to be a gray scale object image. As such, a gray scale image format according to the ITU-R recommendation BT.601 [ITU-R 2011] is created from the three-color channels.

As a second step, contrast normalization is done, which is a preliminary phase for a lot of image recognition algorithms.

The used object image preprocessing example is shown in Fig 3.

Selected feature extraction techniques and classification techniques are described in detail in following sections.

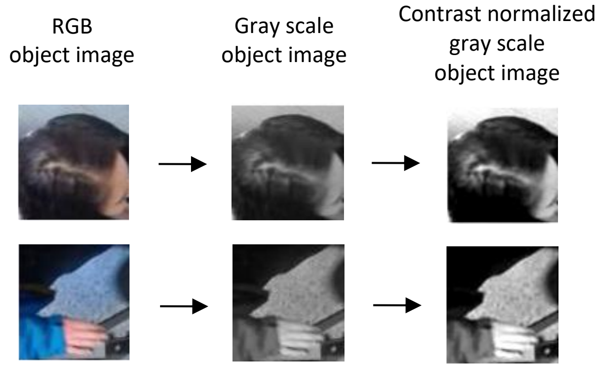

Figure 3. Preprocessing of the object image example

3 FEATURE EXTRACTION TECHNIQUES

Generally, many approaches can be used for image processing. In this work, proven and very well-known feature extraction techniques were selected for testing. Selected feature extraction techniques were based on edge and curve detection algorithms, blob detection algorithms, binary local patterns, histograms of oriented gradients, and pixel intensities. In particular, ten feature extractors were tested.

All the extractors are defined by some tunable parameters. For the purposes of this article, the parameters are set using a combination of heuristic recommendations found in the cited literature sources and initial tests with a specific dataset. The final values are mentioned below.

Edge, curve and blob detections are returning only points, where the defined features were found. As the number of found points is variable, the procedure, where feature points were imprinted into a white background image, is implemented in order to get the normalized number of features.

3.1 Edge Detection

The Canny edge detector, Sobel operator, Prewitt operator, Roberts cross operator and Zero crossing with LoG filter edge detection methods were implemented. The algorithms can be found e.g. in [Lim 1990] and [Parker 1997]. The Canny edge detector finds edges by looking for local maxima of the gradient of the image [Canny 1986]. The edge function calculates the gradient using the derivative of a Gaussian filter. This method uses two thresholds to detect strong and weak edges, including weak edges in the output, if they are connected to strong edges. By using two thresholds, the Canny method is less likely than the other methods to be fooled by noise, and more likely to detect true weak edges. The Sobel operator finds edges at those points where the gradient of the image is maximum, using the Sobel approximation to the derivative. The Prewitt operator finds edges at those points where the gradient of the image is maximum, using the Prewitt approximation to the derivative. The Roberts cross operator finds edges at those points where the gradient of the image is maximum, using the Roberts approximation to the derivative. The Zero crossing with LoG filter finds edges by looking for zero-crossings after filtering the image with a Laplacian of Gaussian (LoG) filter. The last applied edge detection method uses gray scale pictures as vectors and evaluates pixel intensities.

Features from all edge detectors were selected with the block size value set to 11 and the feature size set to 64. Both vertical and horizontal directions were used for edge detections. The sigma parameter (standard deviation) for the Canny detector was set to the square root of 2. Edges selected by the Canny edge detector were in the range of threshold values between 0 and 1. The Laplacian of Gaussian filter was used with standard deviation set to the value of 2.

3.2 Blob Detection

Two blob detectors were tested. Binary robust invariant scalable key points (KAZE) detect keypoints in a 2-D grayscale image [Alcantarilla 2012].

For purposes of experiments with extractors, the scale, in which the interest points are detected, was set to the value of 1.6. The strength of response and orientation were set to the zero value.

The Maximally Stable Extremal Regions (MSER) detector incrementally steps through the intensity range of the input image to detect stable regions [Nister 2008, Matas 2004, Obdrzalek 2009 and Mikolajczyk 2005]. The threshold parameter determines the number of increments the detector tests for stability.

Through all experiments, several parameters were set. Step size between intensity threshold levels was set to 2. The possible size of the region was set to range between 30 and 1400 pixels. A rectangular interest area was set to the full size of an object image. As the final parameter, the maximum area variation between external regions was set to 0.25.

3.3 Feature Extraction

Histograms of oriented gradients (HOG) return extracted features from a true color or grayscale input image [Dalal 2005]. The returned features encode local shape information from regions within an image.

Based on the previous experiments with histograms of oriented gradients, the method parameters were set according to Tab. 1.

Table 1. HOG features parameters

Parameter	Value	Description
Cell size	[8 8]	Size of HOG cell
Block size	[2 2]	Number of cells in a block
Block overlap	[4 4]	Number of overlapping cells
Num. of bins	9	Num. of orientation histogram bins
Orientation	1	Selection of orientation values

Local binary patterns (LBP) return an extracted uniform local binary pattern from a grayscale image [Ojala 2002]. The LBP features encode local texture information.

The number of neighbors used to compute the LBP for each pixel in the input image was set to 8. Then, the radius of circular pattern used to select neighbors for each pixel was set to 1. The linear interpolation was chosen as an interpolation method. To include the individual rotation of object image, the rotation invariance was set.

3.4 Pixel intensities

As a last feature extraction approach, grayscale intensities of pixel values in the range from 0 to 255 were implemented. Therefore, the object image was transformed into a grayscale image and serialized to a row vector.

The grayscale value of each pixel was calculated by forming a weighted sum of R, G, B color components by following equation (1) as described in [ITU-R 2011].

$$E = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \qquad (1)$$

4 CLASIFICATION TECHNIQUES

The aim of a classification process is to decide the category of an object captured in an object image. In this contribution, only two categories are considered. These classes are the 'Head' and 'Not head', where 'head' stands for positive, and 'Not head' stands for a negative class.

In accordance with the aim of this study, the well-established and many-times-proven classification techniques were selected for testing. Specifically, decision trees, support vector machines (SVM) and nearest neighbor algorithms were included in this selection. From across mentioned groups, ten classifiers were tested. Again, the tunable parameters used in our experiments are mentioned directly in the text below.

4.1 Decision Trees

Decision trees, or classification trees and regression trees, predict responses to data [Breiman 1984]. Decision trees are easy to interpret, fast for fitting and prediction, and low on memory usage, but they can have low predictive accuracy. Coarse, Medium and Fine Tree methods differ in model flexibility because of the number of splits and hence different distinction ability between the classes. The following tree algorithms were tested: Fine tree, Medium tree, Boosted trees, Bagged trees and RUS boosted trees.

In the case of a fine tree, the maximum number of splits was set to 100. For all other trees, this value was set to 20. Fine and medium trees used Gini's diversity index as a split criterion. Both boosted and bagged trees used decision trees as a learner type with 30 learners, and a learning rate set to 0.1. A boosted tree was assembled with AdaBoost. A bagged tree was assembled by the bag method.

4.2 Support Vector Machine (SVM) Classification

A SVM classifies data by finding the best hyperplane that separates data points of one class from those of the other class [Lihong 2013]. Their prediction speed is medium for linear or slow for others. Similarly, their memory usage is medium for linear and multiclass, and large for binary problems. They are easy to interpret for linear and hard for all other kernel types. Linear and Coarse Gaussian SVM have low model flexibility with simple separation between classes. Model flexibility of Quadratic Cubic and a Medium Gaussian SVM is medium. The Fine Gaussian SVM exhibits high model flexibility and makes finely detailed distinctions between the classes. Linear, Quadratic, Medium Gaussian and Coarse Gaussian SVM were tested.

The performance of the SVM classifier is influenced by a regularization constant C. Performance of the Gaussian SVM is further influenced by a kernel width value σ. According to our previous studies, C was set to 1 and σ to 31 for purposes of the Coarse Gaussian SVM and to a value of 7.7 in the case of the Medium Gaussian SVM.

4.3 Nearest Neighbors

Nearest neighbor classifiers typically have good predictive accuracy in low dimensions but might not in high dimensions [Weinberger 2009]. They have medium prediction speed, medium memory usage and are not easy to interpret. Fine, Medium and Coarse methods differ in the number of neighbors. Cosine, Cubic and Weighted methods use different distance metrics. The Cosine Nearest Neighbor method was tested.

For the Cosine Nearest Neighbor classifier, the number of neighbors was defined, which was set to a value of 10.

5 EXPERIMENT PROCEDURE

The process of person detection was based on the structure of a typical pattern recognition system and is shown in Fig. 4.

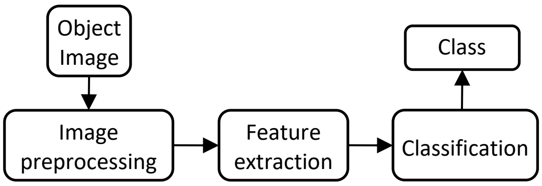

Figure 4. Person detection block scheme

For purposes of possible experiment procedure, the dataset composed from object images was created.

5.1 *Dataset creation*

The important step for a person detection system is a preparation of appropriate training and evaluation sets. The source data should be acquired within the conditions as close to the real application as possible.

Therefore, the video acquisition was carried out on an indoor public place with various light conditions. The video sequences with people walking on the staircase were captured with a monocular camera. Then, the frames with significant shift between person head positions in two consecutive frames were selected.

From the selected frames, object images were cropped sequentially for both classes. Eventually, 736 original object images were acquired with the size normalized to 81px × 81px. In order to support the generalization of the detector, the sets were artificially enhanced - each object image was transformed to provide three more descendants using 90, 180 and 270 degrees rotation. The data were divided into four subsets according to Tab. 2.

Table 2. Dataset groups

Dataset	Training set	Testing set
Positive	1065	355
Negative	1144	380

5.2 *Experiment procedure*

All object images in the dataset were preprocessed. Then each feature extraction technique was applied on the dataset, from which ten groups of data were created.

This procedure creates a matrix for each feature extractor, where a column stands for the feature vector and row number represents position number of an object image. The matrix was then extended for a column, where the object class was added.

After that, each classification technique is applied to data provided by each feature extraction method. The holdout validation was performed by the testing set. All used extractors and classifiers are listed in Tab. 3.

Table 3. List of used extractors and classifiers

Feature extractor	Classifier
Canny edge detector	Fine Tree
Sobel edge detector	Medium Tree
Prewitt edge detector	Linear SVM
Roberts edge detector	Quadratic SVM
LoG edge detector	Medium Gaussian SVM
KAZE detector	Coarse Gaussian SVM
MSER detector	Cosine KNN
HOG features	Boosted Trees
LBP features	Bagged Trees
Pixel intensities	RUSBoosted Trees

In the experiments, the numbers of true positives, true negatives, false positives and false negatives for every tested combination of feature extractor and classifier were evaluated.

6 RESULTS AND DISCUSSION

The aim of this section is to evaluate all proposed combinations of feature extractors and classifiers. Accuracy of the classifiers over the testing set was used, as a standard and good practice evaluation method for image classification. For more comprehensive classifier evaluation, the recall and precision metrics were also used. All mentioned metrics are described by following equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

In the equations above, TP (true positive) is the number of correctly classified positive images, FP (false positive) is the number of misclassified negative image, TN (true negative) is the number of correctly classified negative images and FN (false negative) is the number of misclassified positive images.

In addition, the average computational time is evaluated for each feature extraction data group creation.

As the dataset was split into two groups, the training set was used during the training of classifiers and the testing set was used only for purposes of evaluation.

Testing results of accuracy, recall and precision for all tested combinations of feature extractors and classifiers are summarized in Tab. 4.

The relative computational time is evaluated in Fig. 5.

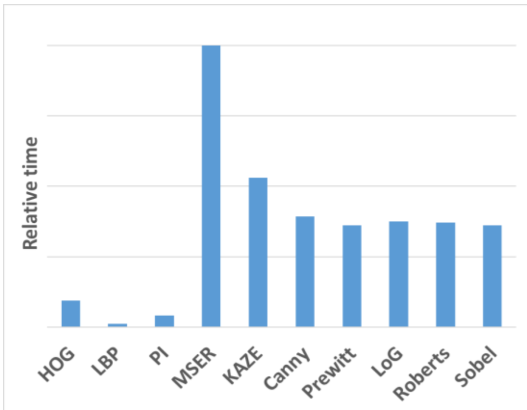

Figure 5. Computational time comparison

The resulting values, in all three tables, indicate several interesting outcomes. First of all, the best precision, recall and accuracy is provided by the HOG feature extractor in combination with the medium Gaussian SVM classifier. Only two feature extractors provide accuracies greater than 90%. Most of the remaining feature extractors provide accuracy greater than 80%. The medium tree and RUS Boosted tree fail in accuracy rates for all feature extractors.

The relative computational time comparison for feature extractors indicates that the performance of edge detectors is similar. The MSER detection is computationally the most demanding. The best performance is provided by the LBP feature extractor. The HOG features extractor is 7 times more computationally demanding than LBP.

Table 4. Accuracy, Recall and Precision results

Accuracy										
Classifier / Extractor	**Canny**	**Sobel**	**Prewitt**	**Roberts**	**LoG**	**MSER**	**KAZE**	**LBP**	**PI**	**HOG**
Fine Tree	56.8%	70.7%	73.6%	74.0%	59.4%	65.9%	58.2%	83.7%	74.7%	76.9%
Medium Tree	59.0%	55.2%	58.3%	58.2%	52.4%	68.5%	60.1%	78.7%	75.1%	77.3%
Linear SVM	67.1%	78.9%	80.8%	82.6%	81.5%	78.3%	70.7%	85.6%	55.4%	93.9%
Quadratic SVM	67.7%	78.9%	81.9%	81.1%	79.1%	77.9%	70.4%	89.9%	85.1%	94.0%
Medium Gaussian SVM	70.7%	80.4%	80.4%	82.9%	81.8%	79.8%	71.7%	90.9%	85.7%	96.2%
Coarse Gaussian SVM	56.7%	78.0%	78.7%	79.5%	78.4%	77.4%	70.8%	85.9%	64.0%	94.8%
Cosine KNN	82.1%	82.6%	82.3%	82.2%	82.1%	73.9%	68.9%	87.1%	80.7%	95.5%
Boosted Trees	63.3%	62.5%	65.4%	64.9%	63.7%	76.6%	67.8%	87.9%	81.7%	92.8%
Bagged Trees	59.0%	79.5%	81.8%	82.3%	70.8%	78.1%	65.9%	89.0%	87.9%	93.3%
RUSBoosted Trees	60.3%	64.3%	67.1%	67.5%	57.3%	75.4%	62.8%	80.3%	79.9%	83.0%
Recall										
Classifier / Extractor	**Canny**	**Sobel**	**Prewitt**	**Roberts**	**LoG**	**MSER**	**KAZE**	**LBP**	**PI**	**HOG**
Fine Tree	46.2%	83.1%	84.5%	85.4%	75.2%	71.0%	51.8%	82.8%	76.6%	76.1%
Medium Tree	41.7%	92.1%	95.5%	96.9%	86.2%	68.7%	55.2%	69.3%	84.5%	76.6%
Linear SVM	58.6%	84.2%	87.6%	89.3%	83.1%	76.6%	70.1%	87.6%	47.9%	94.6%
Quadratic SVM	61.4%	83.1%	86.5%	85.4%	80.8%	76.1%	68.7%	94.1%	90.4%	94.4%
Medium Gaussian SVM	69.3%	80.8%	83.1%	85.6%	79.4%	77.5%	79.2%	95.8%	92.4%	96.9%
Coarse Gaussian SVM	13.5%	90.4%	91.8%	90.1%	82.8%	67.6%	75.2%	89.9%	82.3%	96.9%
Cosine KNN	71.8%	83.4%	84.8%	84.8%	78.6%	62.5%	46.2%	95.2%	77.5%	93.5%
Boosted Trees	34.6%	90.4%	91.3%	95.2%	80.8%	82.3%	62.0%	87.0%	93.2%	92.4%
Bagged Trees	33.8%	72.7%	77.2%	79.7%	59.2%	76.9%	59.2%	89.0%	87.0%	92.4%
RUSBoosted Trees	49.9%	86.2%	91.0%	92.4%	80.3%	82.8%	57.7%	69.6%	89.0%	83.1%
Precision										
Classifier / Extractor	**Canny**	**Sobel**	**Prewitt**	**Roberts**	**LoG**	**MSER**	**KAZE**	**LBP**	**PI**	**HOG**
Fine Tree	56.4%	65.4%	68.3%	68.6%	55.9%	63.0%	57.3%	83.3%	72.5%	76.1%
Medium Tree	60.9%	52.0%	53.8%	53.7%	50.4%	66.8%	59.2%	83.7%	70.1%	76.4%
Linear SVM	68.6%	75.1%	76.2%	77.9%	79.5%	77.9%	69.4%	83.4%	54.3%	92.8%
Quadratic SVM	68.3%	75.6%	78.3%	77.7%	76.9%	77.6%	69.5%	86.3%	80.9%	93.3%
Medium Gaussian SVM	69.7%	79.1%	77.8%	80.2%	82.2%	79.9%	67.7%	86.7%	80.8%	95.3%
Coarse Gaussian SVM	80.0%	71.5%	71.8%	73.4%	75.0%	82.5%	67.8%	82.4%	59.1%	92.7%
Cosine KNN	88.9%	81.1%	79.8%	79.6%	83.3%	79.0%	81.2%	81.3%	81.6%	97.1%
Boosted Trees	76.4%	57.0%	59.1%	58.4%	59.1%	72.8%	68.3%	87.8%	74.9%	92.7%
Bagged Trees	64.2%	82.7%	83.8%	83.0%	75.0%	77.6%	66.5%	88.3%	87.8%	93.7%
RUSBoosted Trees	60.8%	58.8%	60.6%	60.7%	53.9%	71.0%	62.3%	87.0%	74.4%	81.9%

7 CONCLUSIONS

In this article, the set of feature extraction techniques in combination with the set of classifiers for person detection is introduced, designed and tested. According to the results, the object image feature extraction, using histograms of oriented gradients in combination with medium gaussian SVM or cosine KNN as a classifier, looks like an effective solution for such an issue.

Apparently, not only the accuracy, but also computational time is necessary to be tuned in order to provide a suitable tool for the monitoring of person flow in real life applications.

The results for local binary patterns feature extraction in combination with medium gaussian SVM, obtains an accuracy rate greater than 90% and is 7 times faster than the previous combination.

Tuning of this combination should provide similar accuracy while achieving a superior speed, which can be advantageous in time-critical applications.

The performance of feature extraction is directly affected by the size of the feature vector. Furthermore, this is also suitable for classifier training and prediction time. As such, the minimization of the feature vector size could lead to higher performance.

The edge and blob detectors are more suitable for finding the region of interest in a full-scale image, than for evaluating the object image, which is apparent from the results.

ACKNOWLEDGMENTS

Supported from ERDF/ESF. 'Cooperation in Applied Research between the University of Pardubice and companies, in the Field of Positioning, Detection and Simulation Technology for Transport Systems (PosiTrans)'

(No. CZ.02.1.01/0.0/0.0/17_049/0008394).

REFERENCES

[Ahmed 2005] Ahmed, A. and Siddiqui, N.A. Design and implementation of infrared based computer controlled monitoring system. In: 2005 Student Conference on Engineering Sciences and Technology. Karachi, Pakistan, 27 Aug. 2005, pp. 1-5. DOI: 10.1109/SCONEST.2005.4382890

[Akamatsu 2014] Akamatsu, S., Shimaji, N., and Tomizawa T. Development of a person counting system using a 3d laser scanner. In: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). Bali, Indonesia, 5-10 Dec. 2014, pp. 1983-1988. DOI: 10.1109/ROBIO.2014.7090627

[Alcantarilla 2012] Alcantarilla, P.F., Bartoli, A. and Davison A.J. KAZE Features. In: European Conference on Computer Vision, ECCV 2012, Part VI, LNCS 7577, p. 214-227. https://doi.org/10.1007/978-3-642-33783-3_16

[Breiman 1984] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone C. J. Classification and Regression Trees. Boca Raton, FL: Chapman & Hall, 1984.

[Canny 1986] Canny, J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, 1986, pp. 679-698. DOI: 10.1109/TPAMI.1986.4767851

[Choi 2018] Choi, J.W., Quan, X. and Cho, S.H. Bi-directional passing people counting system based on ir-uwb radar sensors. IEEE Internet of Things Journal, 2018, 5(2), pp. 512-522. DOI: 10.1109/JIOT.2017.2714181

[Dalal 2005] Dalal, N. and Triggs B. Histograms of Oriented Gradients for Human Detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20-25 June 2005, pp. 886–893. DOI: 10.1109/CVPR.2005.177

[Dolezel 2019] Dolezel, P., Stursa, D. and Skrabanek P. On Possibilities of Human Head Detection for Person Flow Monitoring System. In: Advances in Computational Intelligence, IWANN 2019, PT II. Lecture Notes in Computer Science, vol. 11507, pp. 402-413. https://doi.org/10.1007/978-3-030-20518-8_34

[Fu 2012] Fu, H., Ma, H., and Xiao H. Real-time accurate crowd counting based on rgb-d information. In: 19th IEEE International Conference on Image Processing, Orlando, FL, 30 Sept.-3 Oct. 2012, pp. 2685-2688. DOI: 10.1109/ICIP.2012.6467452

[Gao 2016] Gao, C., Li, P., Zhang, Y., Liu, J. and Wang L. People counting based on head detection combining adaboost and cnn in crowded surveillance environment. Neurocomputing, 2016, 208, 108-116. https://doi.org/10.1016/j.neucom.2016.01.097

[He 2019] He, M., Luo, H., Hui, B. and Chang Z. Pedestrian flow tracking and statistics of monocular camera based on convolutional neural network and Kalman filter. Applied Sciences, 9(8):1624, 2019.

[ITU-R 2011] ITU-R Recommendation BT.601, Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios, [online], 2011 [Apr. 2020]. Available from https://www.itu.int/rec/R-REC-BT.601/

[Lihong 2013] Lihong, S. and Huang, Y. Support Vector Machine (SVM) Classification: Comparison of Linkage Techniques Using a Clustering-Based Method for Training Data Selection. GIScience & Remote Sensing, 2013, pp. 411-423. https://doi.org/10.2747/1548-1603.46.4.411

[Lim 1990] Lim, J.S. Two-Dimensional Signal and Image Processing. Englewood Cliffs, NJ, Prentice Hall, 1990, pp. 478-488.

[Matas 2004] Matas, J., Chum, O., Urba, M., and Pajdla T. Robust wide baseline stereo from maximally stable extremal regions. Image and Vision Computing, 22(10), 2004, pp. 761-767. https://doi.org/10.1016/j.imavis.2004.02.006

[Mikolajczyk 2005] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Kadir, T. and Van Gool, L. A Comparison of Affine Region Detectors. International Journal of Computer Vision, 65, Numbers 1–2 / November, 2005, pp. 43–72. https://doi.org/10.1007/s11263-005-3848-x

[Nister 2008] Nister, D. and Stewenius, H. Linear Time Maximally Stable Extremal Regions. In: Lecture Notes in Computer Science, 10th European Conference on Computer Vision, Marseille, France, 2008, no. 5303, pp. 183–196. https://doi.org/10.1007/978-3-540-88688-4_14

[Obdrzalek 2009] Obdrzalek D., Basovnik, S., Mach, L., and Mikulik A. Detecting Scene Elements Using Maximally Stable Colour Regions. In: International Conference on Research and Education in Robotics, Communications in Computer and Information Science, La Ferte-Bernard, France; 2009, vol. 82, pp 107–115. https://doi.org/10.1007/978-3-642-16370-8_10

[Ojala 2002] Ojala, T., Pietikainen, M. and Maenpaa T. Multiresolution Gray Scale and Rotation Invariant Texture Classification With Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 24, Issue 7, July 2002, pp. 971-987. DOI: 10.1109/TPAMI.2002.1017623

[Parker 1997] Parker, J.R. Algorithms for Image Processing and Computer Vision. New York, John Wiley & Sons, Inc., 1997, pp. 23-29.

[Pore 2016] Pore, S.D. and Momin, B.F. Bidirectional people counting system in video surveillance. In: 2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT). Bangalore, India, 2016, pp. 724-727. DOI: 10.1109/RTEICT.2016.7807919

[Skrabanek 2015] Skrabanek, P. and Runarsson, T.P. Detection of grapes in natural environment using support vector machine classifier. In: 21st International Conference on Soft Computing, MENDEL 2015, pp. 143-150.

[Skrabanek 2016] Skrabanek, P. and Majerik, F. Evaluation of performance of grape berry detectors on real-life images. In : 22nd International Conference on Soft Computing, MENDEL 2016, pp. 217-224.

[Skrabanek 2017] Skrabanek, P. and Dolezel, P. Robust grape detector based on svms and hog features. Computational Intelligence and Neuroscience, 2017. https://doi.org/10.1155/2017/3478602

[Weinberger 2009] Weinberger, K.Q. and Saul, L.K. Distance Metric Learning for Large Margin Nearest Neighbor Classification. Journal of Machine Learning Research 10(9), 2009, pp. 207–244.

[Wu 2014] Wu, X. Design of person ow counting and monitoring system based on feature point extraction of optical flow. In: 2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications. pp. 376-380. DOI: 10.1109/ISDEA.2014.92

CONTACTS:

Ing. Dominik Stursa
University of Pardubice
Faculty of Electrical Engineering and Informatics
Department of Process Control
nám. Cs. legii 565, Pardubice, 530 02, Czech Republic
+420 466 037 124, dominik.stursa@upce.cz, fei.upce.cz

Ing. Daniel Honc, Ph.D.
University of Pardubice
Faculty of Electrical Engineering and Informatics
Department of Process Control
nám. Cs. legii 565, Pardubice, 530 02, Czech Republic
+420 466 037 107, daniel.honc@upce.cz, fei.upce.cz

doc. Ing. Petr Dolezel, Ph.D.
University of Pardubice
Faculty of Electrical Engineering and Informatics
Department of Process Control
nám. Cs. legii 565, Pardubice, 530 02, Czech Republic
+420 466 037 450, petr.dolezel@upce.cz, fei.upce.cz