Computer Science and Information Systems 14(2):329–346 DOI: 10.2298/CSIS160229011S

Integer Arithmetic Approximation of the HoG Algorithm used for Pedestrian Detection

Srđan Sladojević[1], Andraš Anderla[1], Dubravko Ćulibrk[1], Darko Stefanović[1*], and Bojan Lalić[1]

[1] Faculty of Technical Sciences, University of Novi Sad, Trg D. Obradovića 6, 21000 Novi Sad, Serbia
{sladojevic, andras, dculibrk, darkoste, blalic}@uns.ac.rs

Abstract. This paper presents the results of a study of the effects of integer (fixed-point) arithmetic implementation on classification accuracy of a popular open-source people detection system based on Histogram of Oriented Gradients. It is investigated how the system performance deviates from the reference algorithm performance as integer arithmetic is introduced with different bit-width in several critical parts of the system. In performed experiments, the effects of different bit-width integer arithmetic implementation for four key operations were separately considered: HoG descriptor magnitude calculation, HoG descriptor angle calculation, normalization and SVM classification. It is found that a 13-bit representation of variables is more than sufficient to accurately implement this system in integer arithmetic. The experiments in the paper are conducted for pedestrian detection and the methodology and the lessons learned from this study allow generalization of conclusions to a broader class of applications.

Keywords: computer vision, fixed-point, histogram of oriented gradients, pedestrian detection.

1. Introduction

The world witnessing a great growth in the number of embedded and mobile vision applications that promise to make roads safer, homes and cities more secure, work and chores easier, and spare time more fun. At the same time, the requirements for processing power, memory bandwidth, and battery power are increasing at a very fast rate and one could often wish the processor features are beating, rather than lagging behind the curve predicted by Moore's Law.

Processor and battery manufacturers are coming up with novel technologies to help get ideas to the market faster, such as highly parallel architectures, bigger and faster memory interfaces, lower power processes, and better batteries. However, if there is an idea for a great vision system or product, one should not just sit and wait for a good enough processor to come to the market. There are still things that the embedded and mobile vision community, can do to help and intercept the opportunities sooner. In addition to choosing algorithms that are more amenable to embedded implementation, carefully separating computations from data transfers, parallelizing algorithms whenever

possible, and using hardware specific instructions, one still have to push himself to use integer arithmetic (also known as fixed-point arithmetic) as often as he can. While many of the latest processors offer SIMD floating-point operations, even then there is a power dissipation advantage if integer math is used, not to mention reduction in data bandwidth and other possible advantages. Of course, the implementation of numerically sensitive algorithms in floating-point is still preferred. For example, any algorithm that involves matrix inversions or singular value decomposition should be implemented in floating-point, unless the dynamic range of input data is well understood and one can afford to spare the time and effort to implement the algorithm in integer arithmetic. Other vision algorithms, for the most part, can and should be implemented in integer arithmetic.

Pedestrian detection in images is a challenging task owing to the variability in appearance and the wide range of poses that people can adopt. Applications include surveillance, advanced robotics, automated personal assistance and automotive safety. With regard to automotive industry, today modern cars have systems that are able to detect pedestrians [1, 2], but because of the high price of these systems, they are reserved only for the high-end car models. Multiprocessor systems, such as GPUs with CUDA, support floating-point variables in double precision (64-bit), single precision (32-bit) and half precision (16-bit), but they are expensive, especially the embedded ones. Inexpensive processors with the same capabilities, but with integer arithmetic, could be used for pedestrian detection in order to increase the number of vehicles with the ability of pedestrian detection. Taking into account that most of today processors are at least 16-bit fixed point operations capable, this research is aiming to experiment with required number of bits ranging from 8 to 16, to show that 16-bits is more than enough, and that inexpensive processors with fix data size could be used for pedestrian detection. Another benefit is that the multiprocessor systems will exchange smaller amounts of data during processing. This is due to the fact that, in some implementations, floating-point variables require 4 bytes, whereas integer approximated values could require 2 bytes. In this way, the space complexity and the communication complexity of algorithms are reduced. In the approximated version of the algorithm, the influence of the reduction in the terms of spatial complexity could lead to two times reduction of memory usage for some operations, while the communication complexity could be decreased up to two times enabling transfer of just half of the data than in floating point data representation over the communication channels.

Many well-known algorithms were developed without much consideration for embedded or mobile vision challenges. That is why open-source implementations of vision algorithms are often not written in an embedded-friendly manner. In particular, open source code often uses floating-point when not really necessary. Because of this, a floating-point implementation of an algorithm is sometimes considered as a reference for accuracy and performance. In this paper a commonly referenced open-source people detection system is considered [3] and investigated how its performance deviates from the reference performance as integer arithmetic with different bit-width in several critical parts of the system is introduced. It is considered that the performance of the floating-point code is the ground truth, even though sometimes the integer arithmetic detects a person (i.e., gets a truly correct answer), while the floating-point implementation does not. In other words, the purpose of this study is not to determine which implementation does a better people detection, but to see how the performance of

the system deviates from the floating-point reference code as the bit-width of variables in different sections of the code are decreased.

The contribution of this paper is another proof that 13-bit representation of variables is sufficient to accurately implement the system in integer arithmetic. There is no loss with 13-bit implementation, but it is also shown that in some cases it is even possible to use less than 13 bits to store the values with fixed-point approximations. In addition, another important contribution of this paper is the analysis how and which crucial segment of the HoG algorithm reacts on fixed-point approximation, and how entire algorithm behaves while mathematical operations are approximated. The effects of the integer implementation were examined both separately and jointly for all four key segments of the HoG algorithm. Its segments with good reaction on performed approximation are selected, thus showing which segments of pedestrian detection using HoG could be changed or optimized to better suit the fixed-point arithmetic. Therefore, this analysis is paving the way to more researches regarding pedestrian detection in a real time manner running on inexpensive processors with lower capabilities.

The rest of this paper is organized as follows: Section 2 deals with the related published work. Section 3 describes the approach used to implement integer arithmetic with different bit-widths. Section 4 provides an overview of the experiments conducted and results achieved. Section 5 presents our conclusions.

2. Related Work

Over the past two decades, a number of papers within the field of computer vision has grown significantly. They provide an overview of the methods used, algorithms, and especially, different purposes of computer vision applications [4, 5, 6, 7].

There is an ever-growing pressure to accelerate computer vision applications on embedded processors for wide-ranging equipment including mobile phones, network cameras, and automotive safety systems.

Dedeoglu et al. proposed a software library approach [8] that eases common computational bottlenecks by optimizing over 60 low- and mid-level vision kernels. Optimized for TI's C64x+ core, which has been deployed in many embedded vision systems, the library was designed for high-performance and low-power requirements. The algorithms are implemented in integer arithmetic and support block-wise partitioning of video frames so that a direct memory access engine can efficiently move data between on-chip and external memory. The authors highlight the benefits of this library for a baseline video security application, which segments moving foreground objects from a static background. Benchmarks show a ten-fold acceleration over a bit-exact unoptimized C implementation, creating more computational headroom to embed other vision algorithms.

Kisacanin et al. [9] provide a general discussion and practical examples for the following categories of algorithmic techniques: fast algorithms, reduced dimensionality and mathematical shortcuts. Additionally, they discuss the importance of software techniques such as the use of fixed-point arithmetic, reduced data transfers, and cache-friendly programming. In their experience, each of these techniques is a key enabler for real-time embedded vision systems. Even if the processor has native support for floating-

point, it is beneficial to use fixed-point because on SIMD processors shorter representation of numbers leads to proportionally faster implementation. The results presented in this paper served as one of the motivation factors for our research.

According to Coors et al. [10], algorithm implementation in fixed-point arithmetic can improve performance on the fixed-point DSPs, but in some cases it can also compromise the accuracy. Determining the optimum fixed-point representation can be time consuming if assignments are performed by trial and error. Often, more than 50% of the implementation time is spent on the algorithmic transformation to the fixed-point level for complex designs once the floating-point model has been specified.

The approach based on Histograms of Oriented Gradients (HoG), proposed by Dalal and Triggs [3], is one of the best and most widely used for people detection. An implementation of the algorithm is available as part of the OpenCV library [11]. In presented paper, the changes in original OpenCV HoG implementation are introduced and made comparisons against its original implementation. The authors compared the performances by using the INRIA dataset and PETS videos from OpenCV.

As Dalal and Triggs conclude, when it comes to human detection, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization turn out to be the best strategy, presumably because they permit limbs and body segments to change appearance and move from side to side, provided that they maintain upright orientation. Dalal and Triggs tested their approach using the MIT [12] and the INRIA datasets [13]. As the algorithm achieves near perfect results on the MIT set, we evaluated both the original and the integer implementation of the algorithm on the INRIA data set, which contains over 1800 pedestrian images with a large variety of poses and backgrounds. In addition, initial evaluation was done using one of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS) videos available within OpenCV [14].

The approach proposed by Dalal and Triggs [3] is one of the most successful and popular algorithm in its class and it has been extensively used in [15, 16, 17, 18]. The authors use HoG descriptors with linear SVM. In the presented paper the authors used the same approach with the linear SVM classifier. An alternative would be to replace the linear SVM with a Gaussian kernel, but as Dalal and Triggs state in their paper, it would increase the performance by about 4% at the cost of much higher run times. Although HoG features provide excellent performance to other existing edge and gradient-based features, it comes with high computational costs that are often too high to allow real-time processing [15, 19, 20, 21].

During the past few years, a number of different methods were developed which provide significant speed-ups. Shet et al. [22] used bilattice based logical reasoning approach that exploits contextual information and knowledge about interactions between humans, and augments it with the output of different low level detectors for human detection. The authors employ a boosted cascade of gradient histograms based detector to detect individual body parts. Tuzel et al. [23] developed a method which belongs to the group of methods based on sequentially applying a classifier at all the possible subwindows in a given image. They use covariance features as human descriptors and show the performance of the proposed approach on the INRIA database. Zhang et al. [24] described a method for detecting and tracking humans with extensive pose articulations in challenging situations on small training sets. Pose clusters are learned

from an embedded silhouette manifold. A set of object detectors are trained based on Object-Weighted Appearance Model.

Recently Xiaoyin et al. published a paper where the effects of reduced bit-width on the accuracy and performance of the HoG object detection algorithm implemented on an FPGA is explored [25]. The authors show that reducing the bit-width to 13 bits preserves the same detection accuracy as the original floating-point. All current fixed point HoG implementations use large bit-width to maintain detection accuracy, or perform poorly at reduced data precision. They introduce the full-image evaluation methodology to explore the FPGA implementation of the HoG using reduced bit-width. This approach decreases the required area resources on the FPGA and increases the clock frequency and hence the throughput per device through increased parallelism. Authors then evaluate the detection accuracy of the fixed-point HoG by applying some of computer vision pedestrian detection evaluation metrics and show it performs as well as the original floating-point code from OpenCV. Afterwards, a speed performance comparison is made between single FPGA implementation and a high-end CPU, a high-end GPU, and against the same implementation using floating-point on the same FPGA. The consumption comparison for different platforms showed that fixed-point FPGA implementation uses less power than any other implementation. The main difference between [25] and our work is that instead of using FPGA, or any other real environment, we only test the performances of reference implementations from the OpenCV library. We do not make comparisons regarding power consumptions nor the speed performances, we analyze how and which crucial segment of the HoG algorithm reacts on fixed-point approximation, and how entire algorithm behaves while mathematical operations are approximated. This could help introducing the optimization of specific parts of HoG algorithm in order to provide better performance when used with fixed-point approximations. It is considered that it is more important to optimize algorithm for fixed-point platform usage than to compare execution speeds on different platforms.

Also, Xiaoyin et al. used following benchmarks: Daimler Mono Pedestrian Detection [26], Caltech Pedestrian Detection [27], TUD-Brussels [28] and ETH Benchmarks [29] for measuring HoG detection accuracy for fixed-point, while we used PETS and INRIA dataset. In addition, they used images with resolution 640x480, while we used both bigger, more demanding images, and small images. Results methodology also differs between their and proposed work. They compare detection to ground truth data from datasets, and we compared number of pedestrian detection between floating-point and fixed-point approximation of the HoG algorithm. Even though we used different metrics and methodology, it is shown in our work, same as in [25], that the fixed-point implementation is suitable for embedded platforms if the precision of at least 13 bits can be assured, but sometimes even with less precision.

3. Integer Approximation

The HoG method is based on evaluating well-normalized local histograms of image gradient orientations. In practice, this is implemented by dividing image into small spatial regions ("cells") and accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels in each cell. To achieve greater invariance to

illumination, shadowing, etc., it is useful to contrast-normalize the local responses. This can be done by accumulating a measure of local histogram energy over somewhat larger spatial regions ("blocks") and using the result to normalize all the cells in a block. Normalized descriptor blocks are called HoG descriptors. Based on these descriptors, object detection is done by tilling a detection window with a dense grid of HoG descriptors and using these as a combined feature vector to be fed into a Support Vector Machine (SVM) classifier [30]. A sliding detection window is used to test each part of the image and detect objects.

The HoG technique counts occurrences of gradient orientation in localized portions of an image. It is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

When calculating the HoG descriptors, four key operations can be identified: the calculation of the magnitude and orientation of the image gradient, as well as the normalization of weighted votes. Finally, in addition to the calculation of the descriptor, the SVM classification is a key operation in people detection. Integer approximation will be performed for all four key operations.

The gradient of an image has been obtained with two one-dimensional filters applied on the image:

- horizontal gradient of the image with filter: [-1 0 1]
- vertical gradient of the image with filter: $[-1\ 0\ 1]^T$

Gradients can be "unsigned" or "signed". In this paper, unsigned gradients were considered with values from 0 to π.

Orientation of the gradient $O_g(x,y)$ at position (x,y) is calculated with the following equation:

$$O_g(x,y) = atan\left(\frac{G_H(x,y)}{G_V(x,y)}\right) \quad (1)$$

where $G_H(x,y)$ is the horizontal gradient at position (x,y) and $G_V(x,y)$ is the vertical gradient at position (x,y).

For better invariance to illumination and contrast changes the block histograms are normalized. Different normalization schemes are possible and in this paper, L1-norm was used to calculate the normalization factor f:

$$f = \frac{v}{\|v\|_1 + e} \quad (2)$$

where v is the non-normalized vector containing all histograms in a given block and e is a small regularization constant. The value of e is needed as sometimes empty gradients are evaluated and it does not have influence on the results.

The Support Vector Machines (SVM) classifier is a binary classifier algorithm that looks for an optimal hyperplane as a decision function in a high-dimensional space. A linear SVM classifier is used and the effect of the specified bit-width integer arithmetic on each of the key descriptor calculation operations and the SVM classifier was considered separately.

The SVM classifier is not re-trained for the purpose of the experiment. Original, OpenCV's pre-trained classifier is used, both for floating-point and fixed-point experiment. The idea for the experiment was to check how sensitive is the original implementation to fixed-point approximation and most important fact is that both implementations must use the same classifier. There is intention to introduce approximations in the classifier training for the future work.

Following OpenCV functions from HoG module have been changed: computeGradient(), normalizeBlockHistogram(), getDefaultPeopleDetector() and getDaimlerPeopleDetector() in order to approximate the effect of integer implementation on the precision of values resulting from calculations conducted within the key operations. The following equation has been used to replace used floating-point variables, commonly represented here by *x*:

$$x = (\text{floor}(\text{FACTOR} * x + 0.5)) / \text{FACTOR} \quad (3)$$

where FACTOR is an input parameter used to simulate different register bit-widths available to store the values on a fixed-point processor. By changing the values of this parameter between 2^8 and 2^{16}, register bitwidths (precisions) between 8 and 16 bits were simulated. As an example, the equation above (3) applied for magnitude calculation:

$$\text{magnitude} = \text{sqrt}(\text{gradX}^2 + \text{gradY}^2) \quad (4)$$

is given below:

$$\text{magnitude} = (\text{floor}(\text{FACTOR} * \text{sqrt}(\text{gradX}^2 + \text{gradY}^2) + 0.5)) / \text{FACTOR} \quad (5)$$

For calculation of the (1) atan function is used. Fixed-point arithmetic is not used for this calculation and any other in the algorithm. It is used only for data representation. Functions from all other libraries are used in their original implementation.

Other HoG parameters of interest were set as follows: scale = 1.05, Window Stride Width = 8 and Window Stride Height = 8.

When calculating the HoGs OpenCV can use the hardware acceleration, but this do not have the influence on the results. It is possible to switch between the CPU and the GPU version in the OpenCV HoG algorithm implementation, but both use the same model (default people detector). It just effects the calculation speed, but not the classification accuracy. In presented research, CPU mode of calculating descriptors is used. Additional extensive libraries of multicore-ready, or highly optimized software functions for digital media and data-processing applications like Intel® Integrated Performance Primitives, have not been used.

Images from the INRIA dataset, and frames from the PETS video used in the experiment have 3 channels with 8-bit pixel data representation. Input image data is used as a set of unsigned chars, so it did not affect the algorithm output. Presented approximation was the only method used for calculating influence on different bit-widths.

4. Experiments and results

The experiments were conducted using OpenCV 2.3. In particular, the OpenCV's pedestrian detection demo was modified to simulate fixed-point execution and it was used as bases of new developed application. The detection results of the original (floating-point precision) and modified (fixed-point precision) implementation were compared in the application output. Performance of the floating-point code represented by the number of pedestrians detected in each frame of the video or in every image is considered to be the ground truth. Even though sometimes the integer arithmetic detects a person (i.e., gets a truly correct answer), while the floating-point implementation does not, it is used as a reference for errors calculation.

The effect of a fixed-point implementation was evaluated both separately and jointly, for:

- HoG descriptor magnitude calculation,
- HoG descriptor angle calculation,
- normalization, and
- SVM classification.

When evaluated for a single operation, all other operations were done using floating-point arithmetic. In the joint evaluation, all four key operations were approximated using the same precision.

Initial evaluation was conducted using a publicly available PETS video, which is typically used within the OpenCV pedestrian detection demo. More extensive experiments were carried out using the INRIA person dataset [13].

In each case, the number of detections per image/frame was tracked and finally using those data, the results are calculated.

Application output contains a matrix with the first column containing the number of pedestrian detections in each frame/image using floating-point (original implementation) and second column containing the number of pedestrians detected using simulated fixed-point arithmetic:

$$data = \begin{bmatrix} 2 & 2 \\ 3 & 4 \\ 3 & 2 \\ \dots & \\ 4 & 4 \end{bmatrix} \quad (6)$$

If M is considered for the total number of detected pedestrians by the original implementation:

$$M = \text{sum}(data(:,1)) \quad (7)$$

Following errors are calculated:

$$\text{SumAbsError} = \text{sum}(\text{abs}(data(:,1) - data(:,2))) \quad (8)$$

$$\text{MeanAbsError} = \text{SumAbsError} / M \quad (9)$$

$$\text{RootMeanSquareError} = \text{sqrt}(\text{sum}((data(:,1) - data(:,2))^2) / M) \quad (10)$$

where SumAbsError is the sum of all differences in detection, MeanAbsError is Mean Absolute Error (MAE) and RootMeanSquareError is Root Mean Square Error (RMSE).

All of the represented detection results (both original and approximated HoG) are calculated using some form of non-maximum suppression (KDE algorithm is not used), with purpose to filter out multiple detections (in scale and space) of the same person and to filter out singular cases of false positives. The same non-maximum suppression algorithm is used in OpenCV's PeopleDetector example.

Experiments were conducted on two different datasets, The first PETS video dataset [31] contains video with resolution of 768x576 pixels and the number of frames varies between 250 and 800, acquired at a frame rate of 7 to 14 frames per second. It has four video sequences, the first, S0, contains only background images with almost no people, which is therefore used to train the approach. The remaining sequences (S1, S2, S3) are more complex in terms of number of persons, containing a large number of groups. For the experiment S2.L1 walking scenario was used. Total number of frames used in presented research was 800 where every frame contained at least one pedestrian.

Another dataset is the most famous and challenging INRIA Person dataset [3, 13]. It was collected as part of research work on detection of upright people in images and

videos. The dataset is divided in two formats: original images with corresponding annotation files, and positive images in normalized 64x128 pixel format with original negative images. Only upright persons from images are used for the experiment. The dataset contains images from several different sources: images from GRAZ 01 dataset [32], images from personal digital image collections taken over a long-time period cropped to highlight persons and images taken from the web using google images. For presented research, 1130 small INRIA images containing from 0 to 3 pedestrians were used.

4.1. Detecting People in PETS Video Frames

The number of detections achieved by the original and the fixed-point implementation in video frames and calculated RMSE and MAE across the whole sequence were compared. Since the neighboring frames are similar and to make for more computationally efficient implementation, once a frame is processed, the next frame is skipped.

A sample frame from the sequence illustrating the different detections achieved by the two implementations is shown in Fig. 1.

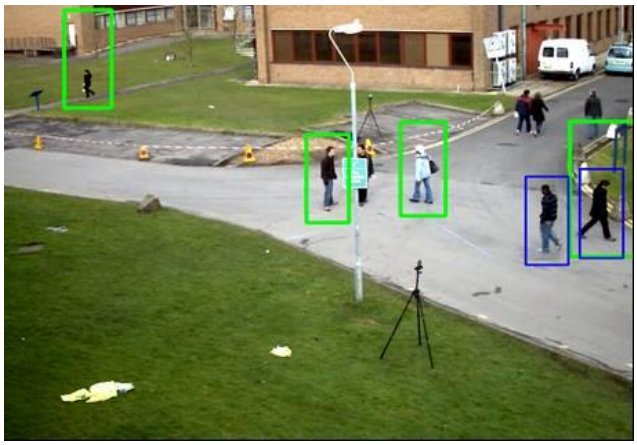

Fig. 1. Example of people detection in video with original (*green rectangles - 4 people detected*) and approximated HoG algorithm (*blue rectangle - 2 people detected*)

In this case there was only one detection in common for the two implementations, the right-most person.

The magnitude calculation proved to be quite robust in terms of fixed-point implementation and caused no detection errors when precision was 10 bits wide (FACTOR = 2^{10}). Fig. 2 shows the plot of MAE and RMSE values obtained for different precisions. Y axes in Fig. 2 and in following graphs are displayed as logarithmic ones in order to respond to skewness towards large values, cases in which one or a few points are much larger than the bulk of the data and to show percent change or multiplicative factors.

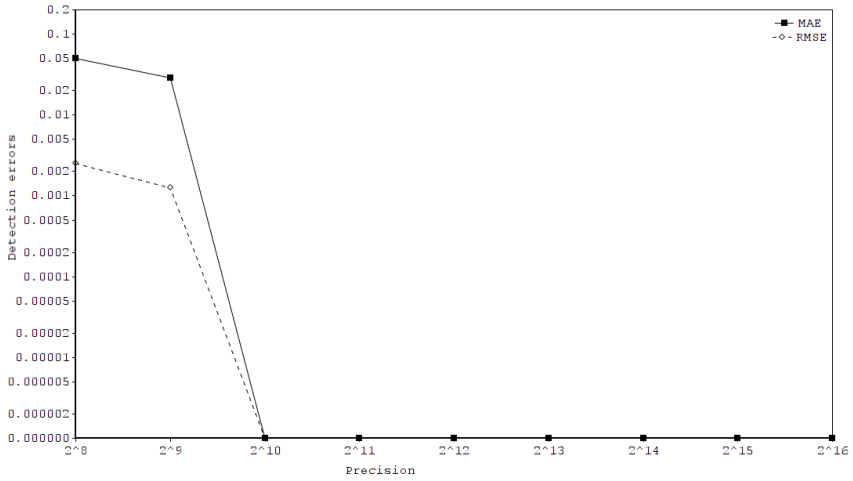

Fig. 2. Detection errors (MAE, RMSE) for different precision approximation of gradient magnitude on PETS video

The image gradient orientation (angle) calculation has to be done with more precision. At least 12 bit wide fixed-point values are needed for the gradient angle to eliminate detection errors, as shown in Fig. 3.

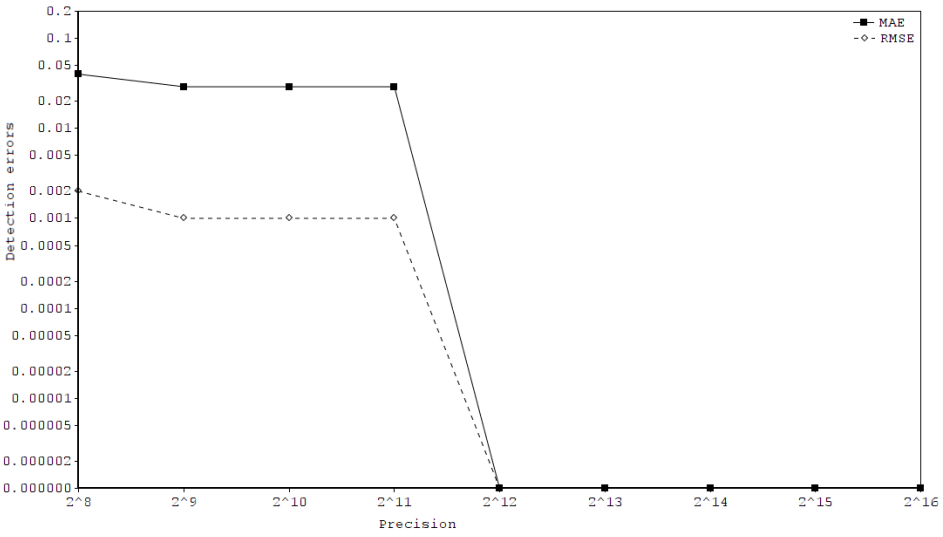

Fig. 3. Detection errors (MAE, RMSE) for different precision approximation of gradient angle on PETS video

Normalization and SVM classification exhibit somewhat more interesting behavior as shown in Fig. 4 and Fig. 5, respectively. As Fig. 4 shows, with a 11-bit approximation of the normalization procedure, the approximated fixed-point and reference floating-point implementation of the HoG algorithm do not differ in their performance.

The SVM classifier proved to be the operation most sensitive to fixed-point implementation. However, the fixed-point approximation using 13 bits to store the values is able to match the performance of the floating-point implementation.

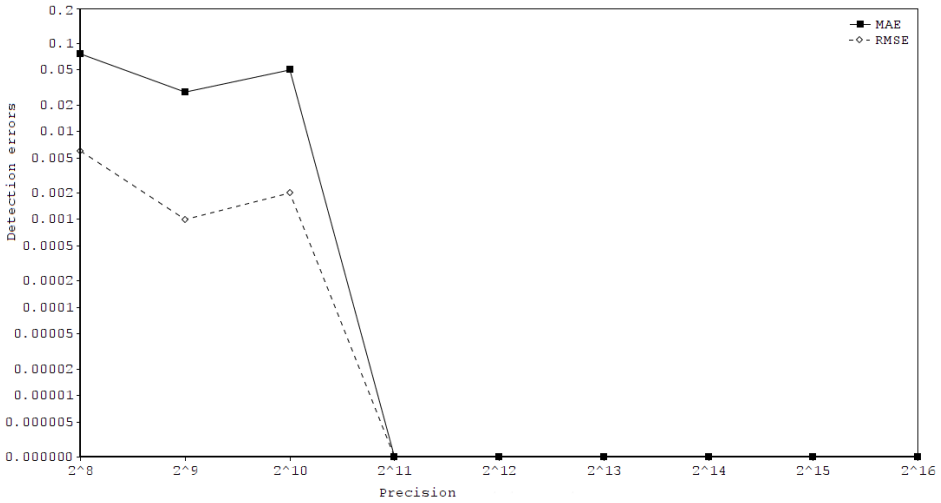

Fig. 4. Detection errors (MAE, RMSE) for different precision approximation of normalization on PETS video

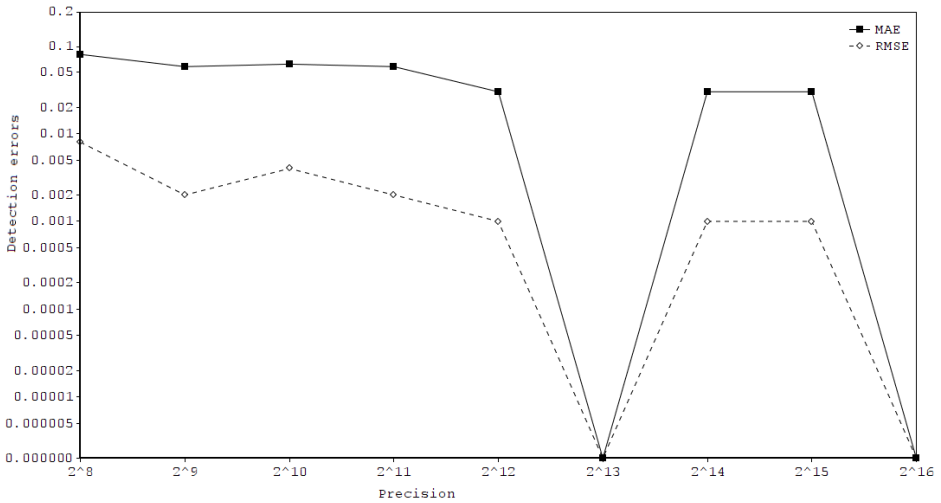

Fig. 5. Detection errors (MAE, RMSE) for different precision approximation of SVM classification on PETS video

Finally, the case when all of the key operations are done in fixed-point, using the same precision was considered. Fig. 6 shows the plot obtained for this experiment. Not surprisingly, the behavior exhibited in this case correlates with the behavior of the most sensitive operation, the SVM classifier.

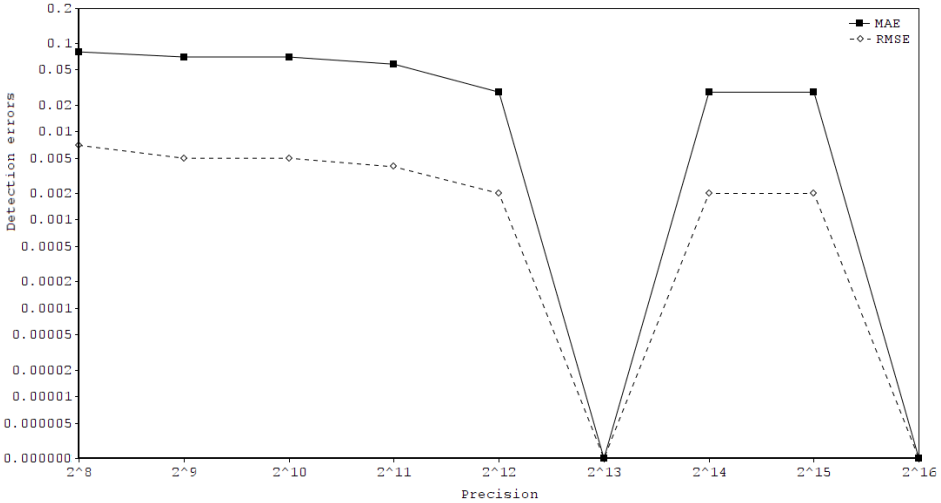

Fig. 6. Detection errors (MAE, RMSE) for different precision approximation of all four key operations on PETS video

It could be seen in Fig. 6 that there is a non-zero error for the case when more than 13 bits are used for approximation (almost same error for 15-bit and 8-bit precision). This is explained due to the nature of errors used. Taking into account that MAE is a quantity used to measure how close forecasts or predictions are to the eventual outcomes and RMSE is a measure of the differences between values predicted by a model or an estimator and the values actually observed, here the authors compare number of pedestrians detected by original and modified algorithm. For the approximations below 13 bits there is MAE and RMSE because fixed-point approximated algorithm provides less detections, while with 14 and 15 bits, surprisingly, approximated version provides better results, so better approximation to the original floating-point algorithm yields better detections and increases MAE and RMSE values.

4.2. Detecting People in the INRIA Dataset

In order to evaluate the fixed-point implementation further, the INRIA person data set was used. The images originally provided for testing were used and cropped to avoid image-border related effects.

A sample image with pedestrian detection results for both implementations is shown in Fig. 7.

When evaluated on the INRIA dataset, the fixed-point approximation of magnitude and angle calculation caused no detection errors when compared to the floating-point implementation. By using 8-bit approximation, HoG algorithm behaves the same as with floating-point implementation. Evaluation with less than 8 bits approximation is not considered for any of the key operations because acceptable results have not been expected. Unlike PETS video, INRIA images have shown better pedestrian detection capabilities for HoG algorithm, and less sensitivity for gradient magnitued and angle

approximations. Those results could be explained because only small images with up to 3 pedestrians, often even without pedestrians from INRIA dataset are used.

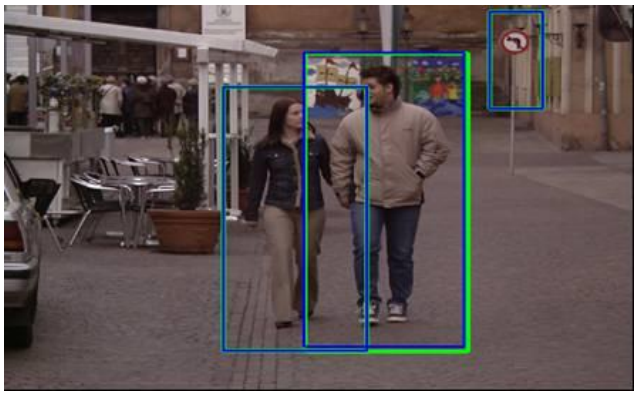

Fig. 7. Sample detection result for and image from the INRIA dataset. Original (*green rectangles*) and approximated (*blue rectangles*) HoG algorithm

Figures 8 and 9 show the plots of detection errors obtained for normalization and SVM classification key operations of HoG algorithm.

Once again, the normalization required at least 11-bit wide fixed-point approximation to match the performance of the floating-point.

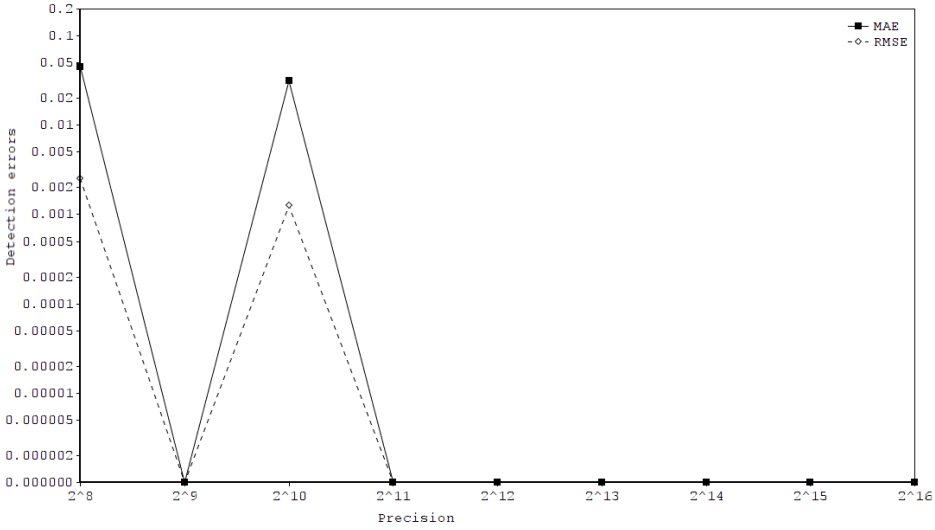

Fig. 8. Detection errors (MAE, RMSE) for different precision approximation of normalization on INRIA dataset

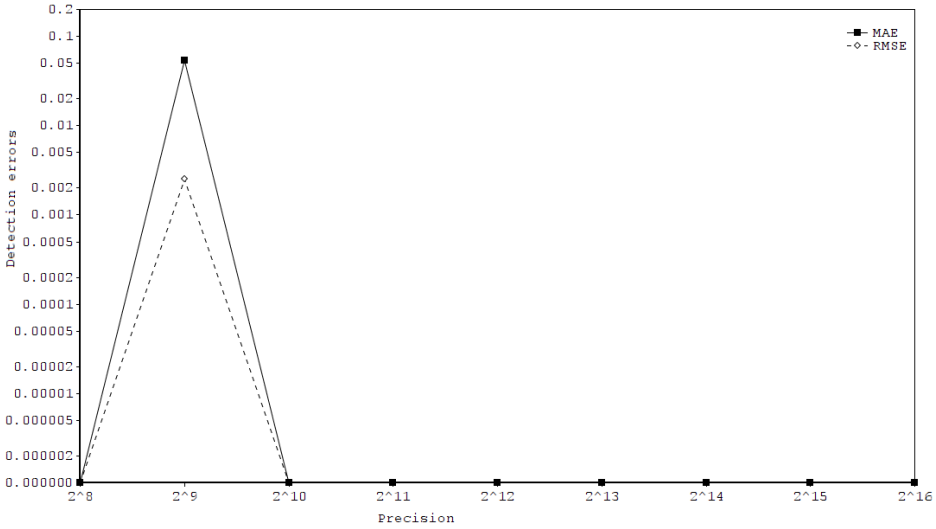

Fig. 9. Detection errors (MAE, RMSE) for different precision approximation of SVM classification on INRIA dataset

Surprisingly, the SVM classifier was more robust to the fixed-point implementation, when evaluated on the INRIA data set. When the values are stored with 10 bits or more, the SVM classification works as well as in the floating-point implementation.

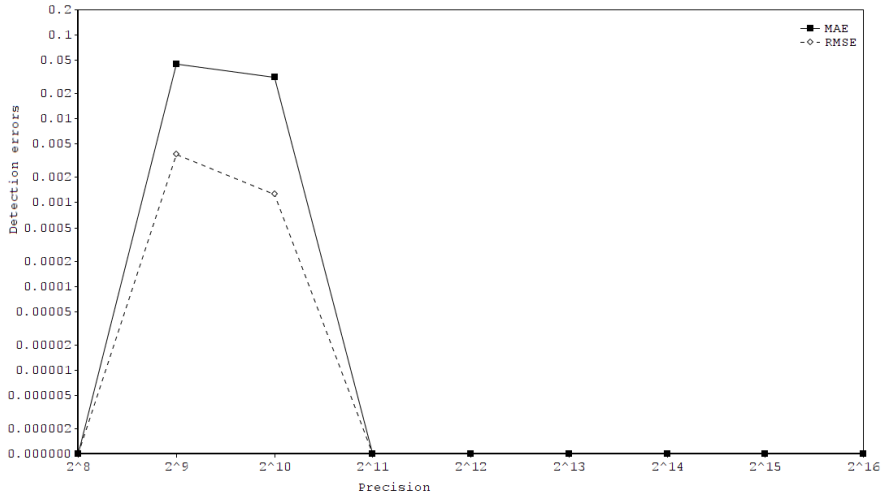

Fig. 10. Detection errors (MAE, RMSE) for different precision approximation of all key operations on INRIA dataset

Once again, when all the key operations are done using the fixed-point arithmetic, the behavior of the whole system is determined by the most sensitive operation, as shown in Figure 10. Based on the INRIA set, the most sensitive operation is the normalization and

the 11 bits wide fixed-point approximation of all key operations is able to achieve the performance of the reference algorithm.

This INRIA dataset experiment showed that in some cases, like in Fig. 10 less bits based approximation gives MAE and RMSE equal to zero. There is no detection error when using 8-bit precision for overall approximation. However, 8 bits are not enough for a successful approximation because 9 and 10-bits approximation has errors in detection, and those errors are approximations based. Original algorithm detects more pedestrians than the approximated one. First value suitable for approximation without errors is 11-bit precision. There is a similar situation in Fig. 9 and Fig. 10. All these inconsistencies originate from the nature of the dataset (complexness, image size, number of pedestrians per image), and type of errors evaluation selected.

Finally, Table 1 shows the summary of pedestrian detection experiment - how many bits is enough for each key operation fixed-point approximation to avoid errors in pedestrian detection.

Table 1. Pedestrian detection experiment summary - the required number of bits to avoid errors in detection when approximate fixed-point arithmetic is used

	Gradient magnitude	Gradient angle	Normalization	SVM classification	All 4 key operations
PETS	10	12	11	13	13
INRIA	8	8	11	10	11

5. Conclusion

Detecting pedestrians still remains an active area of research and has multiple uses. It is an essential and significant task in any intelligent video surveillance system and has an obvious extension to automotive applications due to the potential for improving safety systems, like advanced driver assistance. The results in proposed paper suggest that it is possible to use simpler embedded systems for accurate pedestrian detection. This means that presented implementation can be used in automotive industry for a wide range of cars, not only for high-end modern cars, because inexpensive microprocessors could satisfy the needs. In addition, it proves that such systems could be used in any other pedestrian detection low cost solution.

The suitability of a commonly referenced HoG based pedestrian detector for fixed-point implementation was evaluated. The aim of this research was to identify key operations and the effect of different bit-width fixed point implementation on each of them was examined, first separately and then jointly. When key operations are performed separately, it is shown that 10 bits are sufficient to accurately store data in integer arithmetic for gradient magnitude descriptor. It is found that the normalization and SVM classification are most prone to problems introduced through fixed-point implementation. However, if precision of at least 13 bits can be assured, the fixed-point implementation should behave identical to its floating-point counterpart. For some segments of interest, lower precision could be used for the same results in comparison to the floating-point implementation. This could, even improve performances, in multiprocessor systems, especially spatial and communication complexity. The

performed study was done on a set of examples from two widely used datasets. These preliminary findings may serve as motivation for other researchers in the field and these results could be extended. In addition, this paper is confirmation that presented results are not only dependent on the domain problem and dataset specifications, because other researchers in the field already achieved same precision of four other datasets by using different methodology.

Another observation is that the PETS dataset is more suitable for testing HoG algorithm and therefore gives more consistent results than the INRIA dataset.

The purpose of presented research was to theoretically explore which key operations have most influence on fixed-point implementation. It is shown that using smaller amount of data for each entity could improve performances in multiprocessor systems. Tests and experiments with simple embedded systems that work in real-time to confirm these findings are out of scope of this paper and they will be performed as future work.

In order to achieve more accurate results, the authors plan to re-train the SVM classifier with approximated values, and then to repeat the experiment with the new classification model. It is expected that the results could be similar to presented ones, but to converge more to original HoG implementation.

Additionally, as the original implementation is very demanding regarding speed of the execution, which leads to poor performances of embedded systems, the authors plan to investigate how to increase performances to enable real-time detection. As many uses of the algorithm are related to the detection of pedestrians from video sources, a work on parallelization of the classification tasks is proposed, as an optimal modification of the workaround, which is endorsed by many of the researchers in the field.

References

1. Broggi, A., Cerri, P., Ghidoni, S., Grisleri, P.: A new approach to urban pedestrian detection for automatic braking. IEEE Transactions on intelligent transportation systems, Vol. 10, No. 4, 594-605. (2009)
2. Prioletti, A., Mogelmose, A., Grisleri, P., Trivedi, M.M., Broggi, A., Moeslund, T.B.: Part-based pedestrian detection and feature-based tracking for driver assistance: real-time, robust algorithms, and evaluation. IEEE Transactions on intelligent transportation systems, Vol. 14, No. 3, 1346-1359. (2013)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition (CVPR), Vol. 1, 886-893. (2005)
4. Klancnik S., Ficko M., Balic J., Pahole I.: Computer Vision-Based Approach to End Mill Tool Monitoring. Int. Journal of Simulation Modelling, Vol. 14, No. 4, p. 571-583, doi:10.2507/IJSIMM14(4)1.301. (2015)
5. Anthimopoulos, Marios, et al.: Computer vision-based carbohydrate estimation for type 1 patients with diabetes using smartphones. Journal of diabetes science and technology 9.3: 507-515. (2015)
6. Cajal C., Santolaria J., Samper D., Garrido A.: Simulation of Laser Triangulation Sensors Scanning for Design and Evaluation Purposes. Int. Journal of Simulation Modelling, Vol. 14, No. 2, p. 250-264, doi:10.2507/IJSIMM14(2)6.296. (2015)
7. Töreyin, B. Uğur, et al.: Computer vision based method for real-time fire and flame detection. Pattern recognition letters 27.1: 49-58. (2006)

8. Dedeoglu, G., Kisacanin, B., Moore, D., Sharma, V.: An optimized vision library approach for embedded systems. Computer Vision and Pattern Recognition Workshops (CVPRW), Colorado Springs, United States, 8-13. (2011)
9. Kisačanin, B., Nikolić, Z.: Algorithmic and software techniques for embedded vision on programmable processors. Signal Processing: Image Communication, Vol. 25, No. 5, 352-362. (2010)
10. Coors, M., Keding, H., Lühtje, O., Meyr, H.: Design and DSP Implementation of Fixed-Point Systems. EURASIP Journal on Advances in Signal Processing, Vol. 2002, No. 1, 908-925. (2002)
11. Bradski, G., Kaehler, A.: Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Incorporated. (2008)
12. Ronfard, R., Schmid, C., Triggs, B.: Learning to parse pictures of people. The 7th ECCV, Copenhagen, Denmark, Volume 4, 700-714. (2002)
13. 'INRIA Person Dataset', [Online]. Available: http://pascal.inrialpes.fr/data/human/INRIAPerson.tar (current January 2016)
14. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), http://pets2013.net/ (retrieved on 17 February, 2013)
15. Zhu, Q., Yeh, M., Cheng, K., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 1491-1498. (2006)
16. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, Vol. 60, No. 2, 91-110. (2004)
17. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human Detection Based on a Probabilistic Assembly of Robust Part Detections. Pajdla T., Matas J. (eds.) ECCV LNCS Springer, Vol. 3021, 69-82. (2004)
18. Wu, B., Nevatia, R.: Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. ICCV, Vol. 1, 90-97. (2007)
19. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell., Vol. 34, No. 4, 743-761. (2012)
20. Wang, C.-C. R., Lien, J.-J. J.: Adaboost learning for human detection based on histograms of oriented gradients. Proceedings of the 8th Asian Conference on Computer Vision (ACCV), Springer, 885-895. (2007)
21. Enzeweiler, M., Gavrila, D.: Monocular pedestrian detection: survey and experiments. IEEE Trans. Pattern Anal. Mach. Intell., Vol. 31, No. 12, 2179-2195. (2009)
22. Shet, V. D., Neumann, J., Ramesh, V., Davis, L. S.: Bilattice-based reasoning for human detection. Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), 1-8. (2007)
23. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on Riemannian manifolds. Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), 1-8. (2007)
24. Zhang, L., Wu, B., Nevatia, R.: Detection and tracking of multiple humans with extensive pose articulation. Proceedings of the International Conference on Computer Vision (ICCV), 1-8. (2007)
25. Ma, X.L., Najjar, W., Roy-Chowdhury, A.K.: Evaluation and Acceleration of High-Throughput Fixed-Point Object Detection on FPGAs. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 25, No.6, 1051-1062. (2015)
26. Enzweiler M., Gavrila D.M.: Monocular pedestrian detection: Survey and experiments. IEEE transactions on pattern analysis and machine intelligence. Vol. 31, No.12, 2179-95. (2009)
27. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. Computer Vision and Pattern Recognition, IEEE Conference on CVPR, 304-311. (2009)
28. Wojek, C., Walk, S., Schiele, B.: Multi-cue onboard pedestrian detection. Computer Vision and Pattern Recognition, IEEE Conference on CVPR, 794-801. (2009)

29. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: A mobile vision system for robust multi-person tracking. Computer Vision and Pattern Recognition, IEEE Conference on CVPR, 1-8. (2008)
30. Steinwart, I., Christmann, A.: Support vector machines. Springer, (2008)
31. 'PETS 2009 Benchmark Data', [Online]. Available: http://www.cvg.reading.ac.uk/PETS2009/a.html (retrieved on 26 December, 2016)
32. 'Yet Another Computer Vision Index To Datasets (YACVID) - Details', [Online]. Available: http://riemenschneider.hayko.at/vision/dataset/task.php?did=22 (retrieved on 26 December, 2016)

Dr Srdjan Sladojevic, assistant professor at the University of Novi Sad. Expertise in the field of information and communication systems, image/video processing, computer vision, data mining and machine learning, electronic devices prototyping, embedded systems programming and manufacturing as well as project management. Computer engineer with 20+ years of experience in developing IT solutions and has been a lead engineer on projects for prestigious international development companies. Architect of many hardware-software commercial systems.

Dr Andras Anderla has a PhD in Industrial and Engineering Management from the University of Novi Sad. Research areas are in the field of information and communication systems, computer vision, image/video processing, medical imaging systems, computer aided design and manufacturing. Participated in many international projects.

Dr Dubravko Culibrk has a PhD in computer engineering from Florida Atlantic University, Boca Raton. Currently works as an Associate Professor at the University of Novi Sad. His research interests include video and image processing, computer vision, NNs and their applications, cryptography, hardware design and evolutionary computing.

Dr Darko Stefanovic has a PhD in Industrial and Engineering Management from the University of Novi Sad. He is head of Chair of Information and Communication Systems at Department of Industrial Engineering and Management at University of Novi Sad. His research interest includes implementation of Information Systems, ERP systems, e-learning systems, and e-government systems. Darko Stefanovic has published in several international information systems journals and took part in many projects in implementing various Information Systems.

Dr Bojan Lalic has a PhD in Industrial and Engineering Management and works as an Associate Professor at the University of Novi Sad. His research interest includes simulation modeling, e-business, and e-government. Bojan Lalic is director of Department of Industrial Engineering and Management at University of Novi Sad. He managed numerous projects including TEMPUS, CEEPUS, and implementation of Information Systems in Public Administration of Republic of Serbia.

Received: February 29, 2016; Accepted: March 15, 2017