Multimed Tools Appl (2017) 76:23589–23605
DOI 10.1007/s11042-016-4129-0

Body orientation estimation with the ensemble of logistic regression classifiers

Ali Sebti[1] · **Hamid Hassanpour**[1]

Received: 3 May 2016 / Revised: 5 September 2016 / Accepted: 3 November 2016 /
Published online: 21 November 2016
© Springer Science+Business Media New York 2016

Abstract Orientation of human body is an important feature that can be used for behavioral analysis in surveillance systems. This cue contains useful information such as the direction of movement or attention. Difficulties such as low quality images, cluttered background and partial occlusion harden orientation estimation. In this paper, we propose a novel approach for determining body orientation using the ensemble of logistic regression classifiers.The logistic regression is a discriminative model that is very efficient in time and space complexities. In addition to these desirable properties, we show that this classifier provides a good classification performance in our problem. These classifiers are trained using Histogram of Oriented Gradient (HOG) descriptors which are extracted from four regions in the bounding box of the subjects. Two types of regions are considered in our method: static and dynamic regions. Static regions include: the whole body, upper half and the lower half of the body. Dynamic region includes the region of head and shoulder, which is located dynamically in various images and should be localized for each instance. To enhance the output of each classifier, we propose a weighting scheme based on the inherent characteristics of the orientation estimation problem and finally combine these outputs in an ensemble method to improve the accuracy. Experimental results show the superiority of the proposed method in accuracy and time complexity as compared to the state-of-the-art methods.

Keywords Body orientation · Logistic regression · Ensemble methods

✉ Ali Sebti
ali.sebti@shahroodut.ac.ir

Hamid Hassanpour
h.hassanpour@shahroodut.ac.ir

[1] Laboratory of Image Processing and Data Mining,
Shahrood University of Technology, Shahrood, Iran

1 Introduction

Analysing human behaviors and activities, is one of the main objectives in machine vision systems. In fact, many applications are considered as a preprocessing to these purposes. People tracking, re-identification and activity recognition are examples of such applications. One of the useful characteristics in these applications is body orientation in the scene. For example, in a video surveillance system, the orientation of a body or head determines the direction of movement or attention of the person [20]. Knowing the direction of attention can be used in behavioral analysis. As another application, body orientation can be used in the people reidentification task [4]. Pose estimation and, in particular, orientation estimation are important issues in human computer interaction. There are several problems in machine vision applications, including surveillance applications which have difficulties in the operations of detection and recognition such as: low quality surveillance cameras, various types of noise, cluttered backgrounds. As a result, it is important to design the classifier and extract features resistant to these problems.

Designing a good classification system depends on the extraction of discriminative features and employing an appropriate classifier. Sometimes a more discriminative feature or classifier is computationally complex. Combining multiple classifiers is a common way to overcome this issue [18]. There is often a tradeoff between the classification accuracy and computational complexity. In this paper we propose a method that meets both of these performance criteria. We use histogram of oriented gradient (HOG) as a main feature. Four classifiers corresponding to four regions are trained using the HOG feature. An ensemble of the classifiers determines the final result. These regions include: the whole body, the upper half, the lower half and head and shoulder regions. In the proposed method, the calculation of HOG is performed only once for each body region in the scene. We use logistic regression as the classifier. In this classifier, decision boundary is linear in the feature space and logistic kernel provides the class membership probabilities. The linearity property, sweets the generality and simplicity of the classifier. Mathematical description of this classifier, its effectiveness in classification performance, and time complexity are discussed in this paper. In the orientation estimation, some of the existing methods such as [8, 23], try to determine the body orientation in the four main directions (front, back, right and left). In our proposed method we estimate body orientation more precisely in eight directions.

One of the prerequisite steps in orientation estimation is human detection in the scene. Accuracy of the orientation estimation methods owes the performance of this step. There are many techniques in literature for human detection and pose estimation. In these techniques, many features and classifiers are used to describe image data and train their models. The features, such as HOG, local binary patterns, covariance descriptor, Haar-like feature [10, 15, 25] are used to describe the bounding box of people in an image. Many types of classifier and learning schemes, such as support vector machines (SVM) and boosting [7, 26] can be used to detect the location of the bounding box. For example, in [7, 10] human detection is performed with the Haar-like features and SVM classifier. A comprehensive survey on the recent development of human detection was provided in [17].

The rest of this paper is organized as follows. In Section 2, we review the literature for the recently developed approaches in the area of body orientation estimation. The proposed method is presented in Section 3 in details. The performance of the proposed method in two aspects: accuracy and computational complexity is evaluated in Section 4. Finally, the paper is concluded and the future research issues are discussed in Section 5.

2 Related works

Many researches have been conducted on pose and orientation estimation in recent years. Some researches have merely focused on the head pose and orientation estimation [9, 13]. These works require good quality images as an input. In [9], the detailed information is extracted from the head image using the depth camera. For training model, the random forest regression is used and finally, a precise localization and a 3D pose estimation are obtained.

In [13], the Supervised Local Subspace Learning method was used to learn a mixture of local subspaces. It was shown that this method outperforms other earlier existing methods in head pos estimation. This generative model is robust to noisy and occluded data, non-uniformly sampled data, lack of training samples and overfitting issue. A comprehensive review of the head pose estimation methods are presented in [16].

Many researches were conducted for the whole body pose estimation. Most of them require a high quality image as an input. For example in [1], body parts corresponding to skeleton structure are extracted and precise details of body pose are achieved. In this paper, a generative model is constructed using the appearance-based features associated with different part of the body to propose a kinematic model of human skeleton. This model is a probabilistic graphical model that utilizes the belief propagation algorithm to infer and estimate the articulated human pose in the image. The authors in [5] defined a new concept called poselet for detecting and localizing different body parts. A poselet is a particular region of a human image that describes a similar 3D configuration of body parts in many examples. In this paper a two-layer model is presented. The first layer containes the classifiers that detect the poselets. The second layer combines the outputs of previous layer with a 3D relationship between the parts in a max-margin framework, hence localizes the body part precisely. In [24] a full relational model is used for the upper body parsing. In this research, the results of body segment detectors are scored considering their appearance and spatial relations between all pairs of segments and finally are parsed by an approximation maximization procedure. The authors in [2] proposed a three-stage process to recover 3D pose for a single person. In this method 2D articulations and viewpoints obtained from each frame of a sequence are employed to estimate a reliable 3D pose of a person.

In body orientation estimation, many types of features and classifiers can be used. For example the authors in [11] used HOG as the feature descriptor and SVM as the classifier. In [19], the wavelet coefficients are used as a feature vector and SVM is used as a classifier. In many methods, pedestrian detection in a frame and orientation estimation are performed separately. But in some works, these two phases are combined in a unified fashion [8, 22]. In [8] a probabilistic framework is used to directly approximate the probability density of body orientation. This model consists of two parts: sample dependent cluster priors and discriminative expert classifiers that are integrated in a Bayesian fashion. In [22] the pedestrian localization and orientation estimation are performed in a modified random decision forest.

Covariance descriptor is a region descriptor that can be used in object detection and texture classification. In this descriptor, covariance between several feature vectors of the same region is used as characteristic feature. In [23], covariance descriptor is used for estimating the head and body orientation. For this purpose, the whole image of the body is divided into smaller regions and the covariance descriptor is calculated based on the bank of features. Covariance descriptors like any matrices, locates on the Reimannian manifold and in this space, the distance between two matrices is not an Euclidian distance. Many distance measures have been introduced in literature [6]. In [23] a distance measure is described. For each subregion, an SVM is trained. For extracting the feature vector for every subregion, firstly, a number of random images in the training set are selected as reference points. Therefore, the feature vector of a subregion is a concatenation of the distance values between their covariance descriptor to the corresponding sub-region of the reference images based on the proposed measure. Finally, combination of the SVMs for all the sub-regions determines the orientation.

In [12], multiple classifiers are trained for some common human postures. In the testing phase, when an image is given, its overall pose is determined using the above mentioned classifiers. Then, the best contour which is matched with the edge of the image is retrieved. This matching is done by traversing a hierarchical clustering structure. These trees are constructed for each common posture separately. This contour provides a more accurate posture of the person.

Some approaches apply a postprocessing on output of the classification. In these methods the output of the classifiers are a real value that determines a confidence level of classification [3, 8]. These works estimate a global continuous probability density function using a mixture of Gaussian distributions. In [3], a set of extremely randomized trees classifiers are trained on HOG feature in quantized directions and finally, the exact direction is calculated by the above mentioned method.

3 Proposed method

In the proposed method, body orientation estimation is formulated as a classification problem. For this purpose, we quantize the orientation in eight directions. The classification system consists of two main parts: feature selection and the use of a discriminative model. HOG descriptor is used as the basis feature. This descriptor is calculated over four regions in bounding box of the subject. Two types of regions are considered in the proposed method: static and dynamic regions. The dynamic region should be first localized. This localization is performed efficiently in our proposed method. Logistic regression as a discriminative model has been used for both the classification task and localization of the dynamic region. We show that this classifier is efficient in classification performance and time complexity. Four classifiers corresponding to the four regions are trained using the HOG feature. Finally we combine outputs of the four classifiers in an ensemble method. The HOG descriptor and multinominal logistic regression are introduced in the following subsections.

3.1 HOG feature descriptor

HOG is one of the most popular feature descriptor in image processing and computer vision. This descriptor was first introduced for human detection [7]. In fact, this feature, represents the structural characteristics of a localized portion of an image. The main idea behind this descriptor is that, shape and structure of an image are described in the edges and gradient information. The main steps of the HOG method is the breaking of the whole image into small regions which are called cells. The cell size is defined by the user. In each cell, histogram of gradients is computed in different orientations bins. For reducing the effect of illumination changes and shadowing, a normalization stage is done across a higher division level. This division is a group of cells which are called blocks. These blocks can be overlapped along x or y axes. The final descriptor is a concatenation of the histograms in blocks for the whole image. The advantages of the HOG descriptor are robustness to illumination change, low geometric changes, low displacement changes and reasonable computational complexity. Figure 1 shows the summary steps of HOG descriptor computation for a sample image.

3.2 Multinomial logistic regression

Multinomial logistic regression is a classification method and known as a discriminative model. This method directly computes the conditional probability of dependent variable y, given to independent variable x. Suppose that the dependent variable is categorical or nominal and we assign it to a discrete variable (response variable). Thus, the classification task is formulated as regression of dependent variable to independent variable (feature vector). Logistic regression measures this regression by using a logistic function. A logistic function has a sigmoid shape that its output is in the range from zero to one. Binary logistic regression models the posterior class probability $Pr(Y = 1|X = x)$ according to (1).

$$Pr(Y_i = 1|X_i = x_i) = \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} = \frac{1}{1 + e^{-\beta^T x_i}} \tag{1}$$

In this equation, β is a vector of regression coefficient, and x is the feature vector. Due to simplicity of the equation, constant 1 is appended to the x feature vector. In the training phase, maximum likelihood (ML) estimates the parameters of the model. During the test, the class membership is measured by thresholding of the posterior probability value. Multinomial logistic regression is the generalized form of binary case. In this case, parameter β is jointly estimated using the maximum a posteriori (MAP) for all classes. Posterior class probabilities $Pr(Y = j|X = x)$ for class j is in the form of (2):

$$\begin{cases} Pr(Y_i = j|X_i = x_i) = \frac{e^{F_j(x_i)}}{\sum_{k=1}^{J} e^{F_k(x_i)}} \\ F_j(x_i) = \beta_j^T .x_i \end{cases} \tag{2}$$

In order to achieve a more generality, we use boosting version of the logistic regression in the proposed method that is described in the following subsection.

3.2.1 Logitboost

Logitboost is an additive boosting method that is categorized as a supervised learning method. In boosting methods, a set of weak learners are added to the model iteratively, and

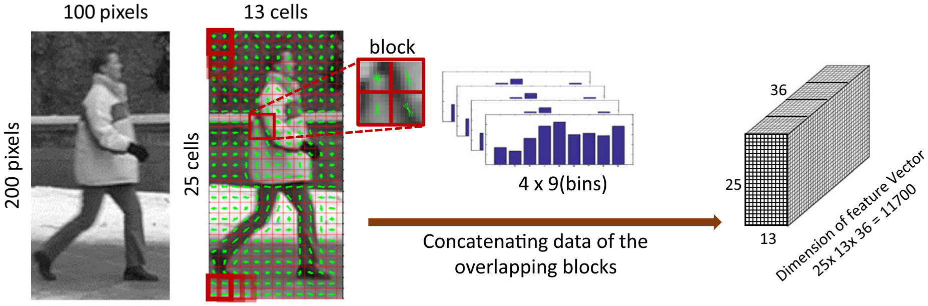

100 pixels
200 pixels
13 cells
25 cells
block
4 x 9(bins)
Concatenating data of the overlapping blocks
36
25
13
Dimension of feature Vector 25x 13x 36 = 11700

Fig. 1 Summary steps of HOG descriptor computation with commonly used parameter's values (Cell size = 8 × 8 pixels, block size = 2 × 2 cells, block overlapping = {1, 1}, 9 orientation bins)

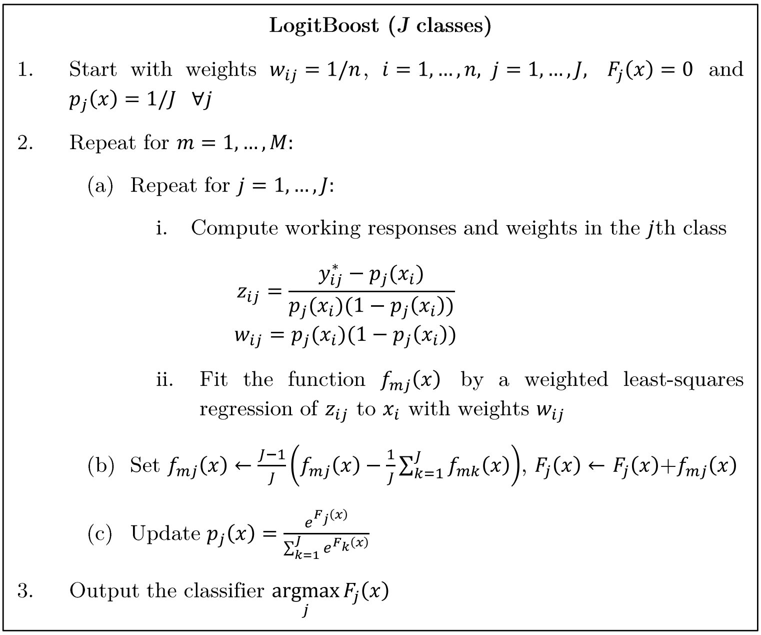
a strong classifier is produced. In each iteration, a new weak learner learns a weighted version of the training data. This weighting makes the focus of new learner to the data that is misclassified by the last learner. Finally, linearly combination of the weak learners form the strong classifier. The steps of logitboost algorithm are shown in Fig. 2. In this algorithm, parameter estimation of the logistic model is performed by minimizing the negative log likelihood loss function (which is equivalent to maximizing the log likelihood). Root of the derivative of the loss function is the solution. The Newton's method is used to estimate the root. In the Logitboost algorithm, the z_{ij} stands for the step term in Newton method. w_{ij} represents weight of training sample in each boosting iteration. In each iteration, the relationship (f_{mj}) between input data (x) and z vector is estimated by using weighted least square regression. Finally, according to Newton's method, F_j iteratively accumulates the f_{mj}, which contains the β coefficient of the logistic model. The first term in step (b) of the algorithm guaranties that $\sum_{k=1}^{J} F_k(x_i) = 0$ and an unique solution is obtainable. In this algorithm, y_{ij}^* determines the membership of x_i to the j-th class. Equation (3) shows the y_{ij}^* values for different cases:

$$y_{ij}^* = \begin{cases} 1 & \text{if } \; y_i = j \\ 0 & \text{if } \; y_i \neq j \end{cases} \tag{3}$$

In this paper, the specific version of logitboost with embedded feature selection is used. This version is called SimpleLogistic [21]. Output of the SimpleLogistic algorithm is the value of the parameter β for each class.

3.3 Localizing and selecting the four regions and training the classifiers

In the proposed method we assume that the whole body region of pedestrians are detected in the earlier stage. The used dataset was marked manually with nine points corresponding



```
LogitBoost ($J$ classes)
1. Start with weights $w_{ij} = 1/n,\ i = 1, \ldots, n,\ j = 1, \ldots, J,\ F_j(x) = 0$ and $p_j(x) = 1/J \ \ \forall j$
2. Repeat for $m = 1, \ldots, M$:
   (a) Repeat for $j = 1, \ldots, J$:
       i. Compute working responses and weights in the $j$th class
          $z_{ij} = \frac{y_{ij}^* - p_j(x_i)}{p_j(x_i)(1 - p_j(x_i))}$
          $w_{ij} = p_j(x_i)(1 - p_j(x_i))$
       ii. Fit the function $f_{mj}(x)$ by a weighted least-squares regression of $z_{ij}$ to $x_i$ with weights $w_{ij}$
   (b) Set $f_{mj}(x) \leftarrow \frac{J-1}{J}\left(f_{mj}(x) - \frac{1}{J}\sum_{k=1}^{J} f_{mk}(x)\right)$, $F_j(x) \leftarrow F_j(x) + f_{mj}(x)$
   (c) Update $p_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}$
3. Output the classifier $\underset{j}{\operatorname{argmax}} F_j(x)$
```

Fig. 2 LogitBoost algorithm

to different body parts for each pedestrian image. Firstly, we cropped the body region from each dataset image in the scale of 1 : 2 (width:height). Then the cropped regions are scaled to 200 pixels height by 100 pixels width. The dataset is divided into eight directions from 0° to 315° with steps 45°. Two types of regions are used for feature extraction. The first type is a static selection that consists of three rectangular regions: the whole body region (W), the upper half of the image (UH) and the lower half of the image (LH).

The second type is a dynamic selection localizes a rectangular region around the head and shoulders (HS) for each of the dataset image. In training phase, HS regions are localized using the head and basin marks in dataset. Two types of problems affect the HOG and consequently the final classification results. The first is displacement or geometric changes of the object (human in our work), and the second is cluttered background. According to the HOG algorithm, displacement or geometric changes can lead to changes in histogram of gradient in cells and blocks if these changes exceeds a threshold. Cluttered background also degrades the gradient information in different areas of the image and consequently affects the HOG feature vector. In this research, displacement change can be caused by the inaccuracy of the human detection algorithm. Also geometric changes can occur due to specific body poses or partial occlusion with items such as bag and back packs. The impact of these factors on the above mentioned regions are different. These issues convinced us to combine the results of the four region related classifiers. Experimental results show that an ensemble of these classifiers is more accurate than those from each classifier seprately.

In the next step we train the four classifiers for each region (W,UH,LH,HS). As mentioned earlier, the logistic regression is used for classification in this research. The main reasons for choosing this classifier is simplicity and low complexity of the algorithm during the test phase, and a good generality for unseen data [14]. For the first type of regions (W,UH,LH), after extracting the HOG feature vector, we use optionaly principle component analysis (PCA) for dimensionality reduction. We select 200 eigenvectors with the highest eigenvalues from the PCA mapping. By using the PCA, the noise influence is reduced; and the training speed and complexity (caused by mapping procedure) is increased. Figure 3 presents the summary steps of training phase in the proposed method. Another major contribution of the proposed method is localization of the HS region. The HS location is not a fixed place and may vary in different images. This region, unlike the three previous regions, is a dynamic region with a focus information on head and shoulder areas. Angle information of this region can have a positive impact on the overall result.

For localizing the HS region, we propose a fast algorithm which uses the HOG feature vector that is computed once for each body region in the images. For this reason, we slide a window on the image and examine whether the region is HS or not. For examining a region, we trained a logistic regression classifier with two classes: HS region and non-HS region classes. In our experiments the accuracy of the logistic regression classifier by using only the W region is 50 % for the eight directions, and 70 % with a tolerance of ±45°. We use this fact to achieve a greater accuracy in HS region localization. For this purpose, we trained eight binary classifiers in each of the eight directions. In training phase, positive samples for each classifier (angle = $i°$) are gathered from dataset image in three angles ($(i - 45)°$, $i°$, $(i + 45)°$). HS regions as positive samples are localized using the head and basin marks. These regions are cropped with the size of 48 × 40. HOG descriptor for this region is a cell array with the size of 6 × 5. Negative samples are cropped around the HS region in each image with area intersection between 10 % and 40 % of the HS area. In addition, one multinomial logistic regression classifier is trained on HS regions for final classification of the localized HS region. Figure 4 shows the training classifiers that is used for localization and classification of the HS region.

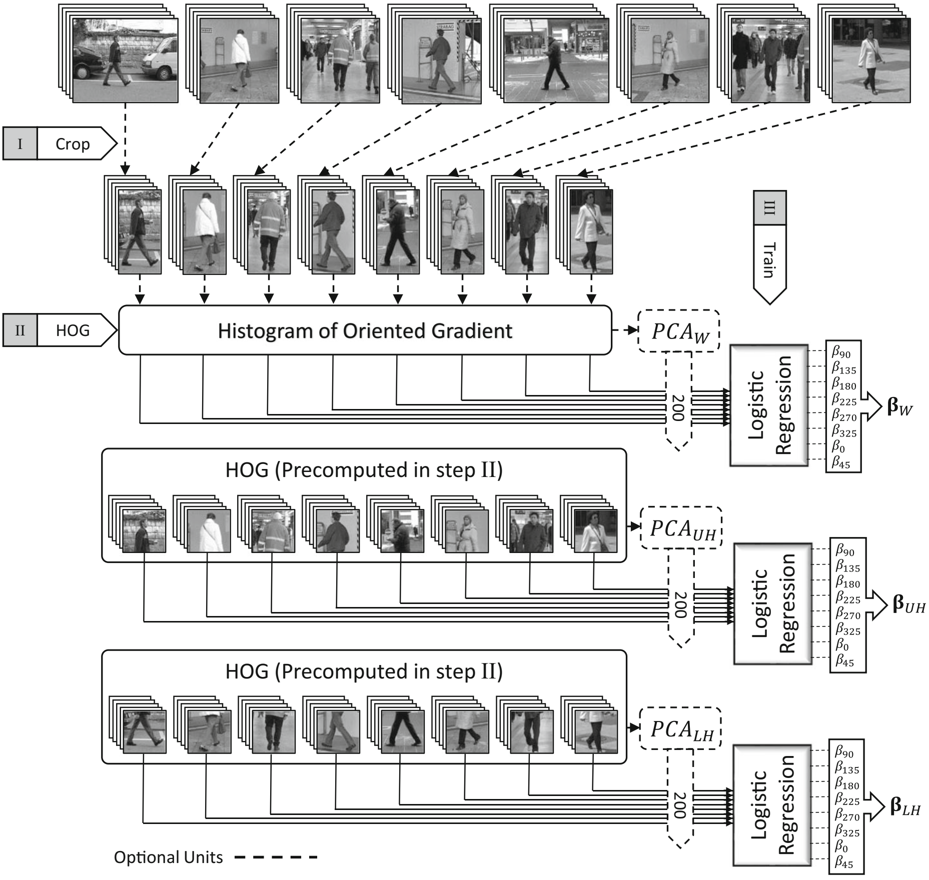


Fig. 3 Training the W,UH,LH region of interests

In the test phase, when a sample image is given, we estimate the initial direction ($\hat{I}$) by using W region classifier. Then, HS region is localized by using $\hat{I}$ classifier. As mentioned before, classification of a test sample in logistic regression classifier is performed through inner product of extended feature vector $< 1, x_1, x_2, \ldots, x_n >$ with vector β. So, sliding a window and testing with this classifier is like convolving the image with the vector β. In order to reduce time complexity and to avoide redundant calculations, we slide the window with a step size of 8 pixels (width of the HOG cells). As a result, HOG descriptor for the sliding window is precomputed. Since the HS region is located in the upper part of the image, we search its location in the top 10 cell rows of the image. Result from the previous step is a probability map that determines the chances of the HS occurrence at any locations. On this map, we set the values to zero if the probability value is less than 50 %. Suppose that a location in a 2D map is a discrete multivariate random variable. This random vector has two components that includes x and y. Therefor a location in the map is an observation of this random vector. According to the classification result, each location can take a probability value which represents the probability of the HS occurrence. So, expectation of this random vector is a precise location of the HS region. Another interpretation of the

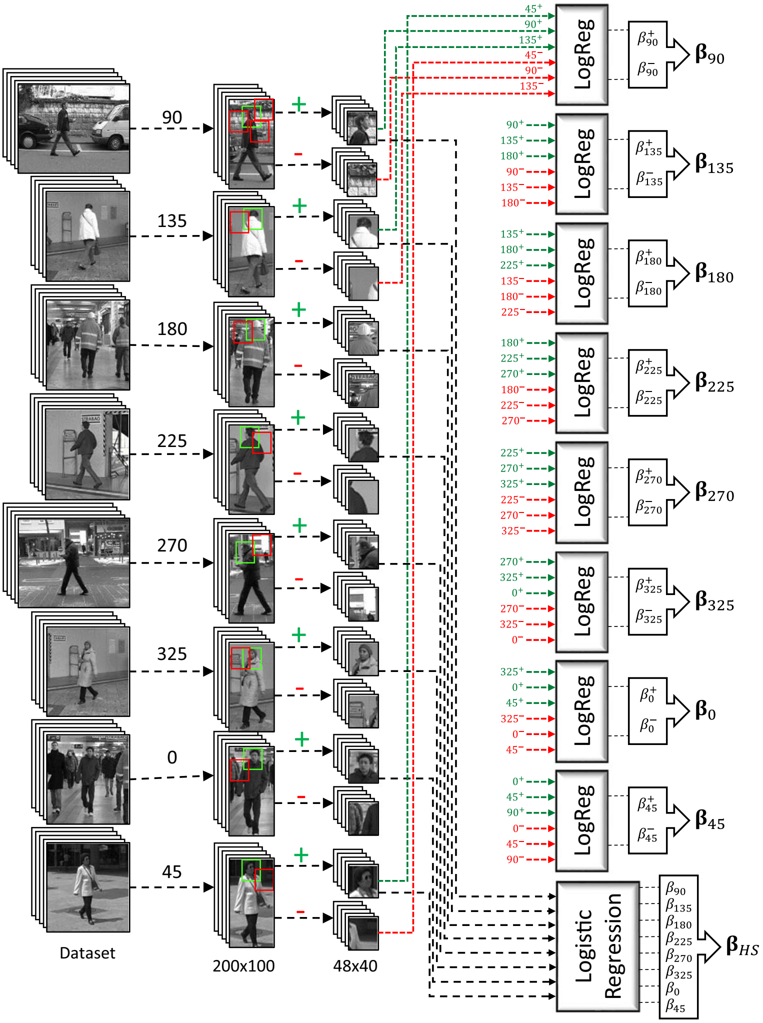


Fig. 4 Training classifiers that is used for localization and classification of the HS region

expected value in this context is a weighted average of the observations. Equation (4) shows calculation of the expected value.

$$\begin{aligned}
&E[X] = \tfrac{(p_1x_1+p_2x_2+\cdots+p_nx_n)}{p_1+p_2+\cdots+p_n}\\
&x_i =< a, b >\\
&a \in \{1, 8, 16, 24, \ldots, 80\}, \quad \text{10 cell rows}\\
&b \in \{1, 8, 16, 24, \ldots, 88\}, \quad \text{all columns}
\end{aligned} \tag{4}$$

In order to measure the accuracy of the localization algorithm, we localize the HS regions in all the test samples. Then, distribution of the test samples over the percentage of ground

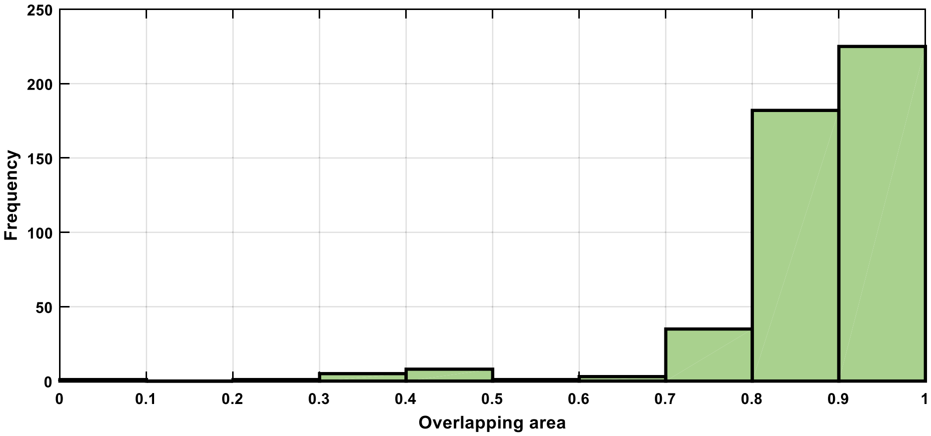


Fig. 5 Distribution of the test samples over the percentage of ground truth, covered by the localized HS region

truth, covered by the localized region is calculated and shown in Fig. 5. This figure shows that in majority of the test samples, 90 % of area in the ground truth is covered. In this level of localization accuracy, HOG descriptor for localized region and ground-truth are similar.

After localizing the HS region, HOG descriptor of the HS region is classified with the HS region classifier (see Fig. 4).

3.4 Weighting and ensembling the classification results

Ensemble methods combine multiple classifiers to form a better classifier. Final classifier is more accurate and more stable in unseen data. As mentioned in Section 3.3, we have four classifiers corresponding to different regions. The result of each classifier is an 8-component vector that predicts the probability distribution over a set of classes. These probabilistic values can be useful for combining classifiers. For improvement in the final combination, we use the weighting scheme that is based on our problem definition. In many classification problems, samples of different classes are not similar to each other. For example in object recognition, we have a number of object classes such as human, cats, airplane. that are completely irrelevant in shape and other characteristics. But in our classification problem, we have eight classes that are related to each other. For example, human images in 90° orientation are similar to the human images in 45° and 135° and even to the opposite orientation (270°). We use this idea to weighting the result of classifiers before combining them. For this reason, we consider two constants for weighting; one for neighborhood effects (N) and another one for opposition effects (O). These parameters are estimated using cross validation. Figure 6 shows an example of weighting scheme for a sample of eight-class membership and the parameter values ($N = 0.4$, $O = 0.1$).

In the next step, the weighted results are combined with an ensemble method. There are many ensemble methods for combining the multiple classifiers or experts. Some of these methods require a learning phase. In this research, we use an algebraic ensemble combination rule. This combiner is a non-trainable combiner that combines results of the classifiers through an algebraic expression. This expression is the product of the outputs of

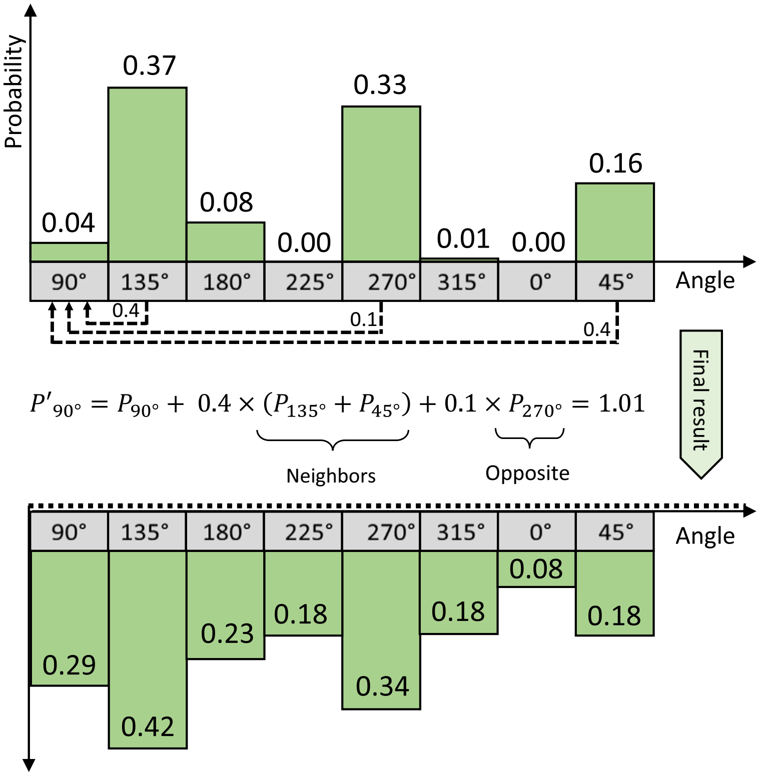


Fig. 6 Weighting scheme for a sample classifier result

classifiers which is calculated as (5). The final decision is class j that has the largest value in the production expression.

$$\begin{aligned} h_{final}(x) &= \arg\max_j \mu_j(x) \\ \mu_j(x) &= \prod_{r \in R} p_{r,j}(x) \\ R &= \{W, UH, LH, HS\} \end{aligned} \tag{5}$$

In this equation, $p_{r,j}$ stands for class j membership value corespounding to region r classfier. Figure 7 presents a summary of the overall proposed method. The used modules and parameters in this figure were described in the previous sections.

4 Experimental results

In this section, we evaluate the classification performance of the proposed approach on a dataset which are labeled for eight orientations. We compare our proposed method against the other existing approaches in the application of pedestrian orientation estimation [2, 3, 8, 11, 19, 22, 23]. Some of these approaches are publicly available in source code such as [23] and the rest were implemented in our research.

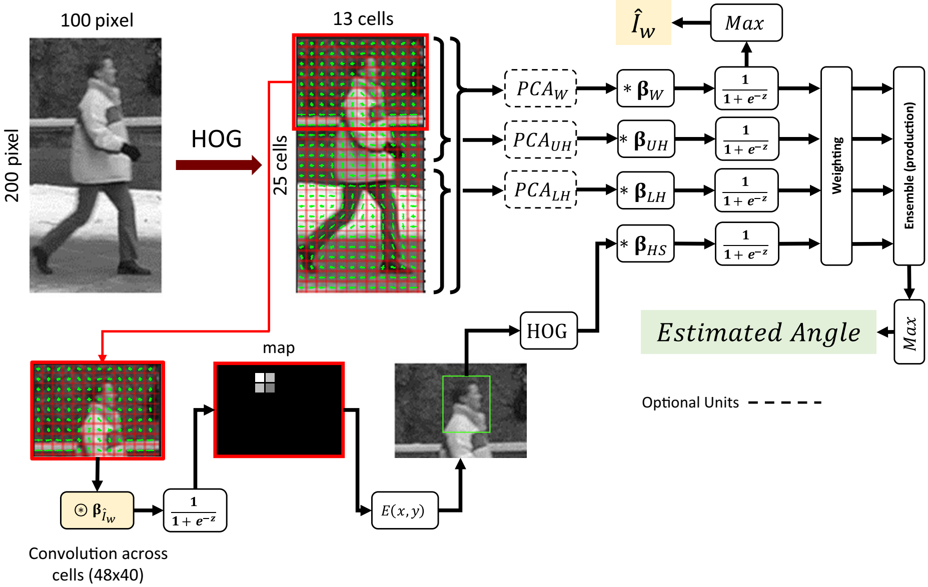


Fig. 7 Overall diagram of the proposed method

4.1 Dataset

We tested our method on the TUD Multiview Pedestrians dataset that is freely available [2]. This dataset containes 4730 images for training, 248 images for validation, and 248 images for testing. In order to have a larger test set we merged test and validation sets (496 samples in the new set) and used a two-fold cross validation for tuning the parameters and calculating the classification accuracy. All images in the dataset were annotated with many useful information such as: bounding box of a pedestrian, viewpoint of the pedestrian, nine markers corresponding to different parts of the body. This dataset is in eight viewpoints as shown in Fig. 3. Firstly, we convert all the RGB images to grayscale intensity images. Then we crop the bounding box of the pedestrian in all images.

4.2 Results

All the implemented algorithms are compared in terms of: classification accuracy and time complexity. In the first case, we employed confusion matrix and overall accuracy to compare the results. In the second case, we counted the number of required operations (multiplication, accumilation, indexing) as time complexity in the test phase.

4.2.1 Classification performance

We conducted two experiments to evaluate the overall accuracy. In the first experiment, the average performance of the exact classification are considered. In the second experiment, the classification of an instance is taken as correct if it is classified to ±45° in adjacent of the true direction. Table 1 compares the results of the proposed method with the other methods in terms of accuracy. In [22] orientation estimation and pedestrian classification are

Table 1 Comparison between the overall accuracy of the proposed method and the other existing methods

Method	Accuracy	
	0° (Exact)	±45° (nonExact)
Proposed Method	**57.9%**	**83.7%**
ERT+MoAWG [3]	53.0%	81.5%
HOG+SVM+PCA [8,11]	53.2%	78.8%
HOG+SVM(1-vs-all)+PCA[2]	51.6%	79.0%
HOG+SVM-adj(1-vs-all)+PCA[2]	38.5%	79.6%
HOG+SVM [8,11]	37.0%	64.3%
Wavelet+SVM+PCA [19]	26.0%	51.6%
CovDescriptor [23]	25.8%	45.0%

performed simultaneously in four directions. The experiments in [22] for eight orientations classification are in mixture mode (exact classification for angles of 0°, 90°, 180°, 270° and non-exact classification for angles of 45°, 135°, 225°, 315°). With this assumption, the overall accuracy of our proposed method is 73 % and the one reported in [22] is 69 %.

Figure 8 presents the classification results in confusion matrix for the proposed method and the method presented in [3] as the leading method. As you can see, the proposed method is more accurate than the method in [3] for five out of eight directions. These results indicate that our method outperforms the other methods in terms of classification accuracy.

4.2.2 Time complexity

In this subsection we count the number of required operations (multiplication,accumilation,camparison,indexing) to estimate the time complexity in the methods under comparison.

- Logistic Regression

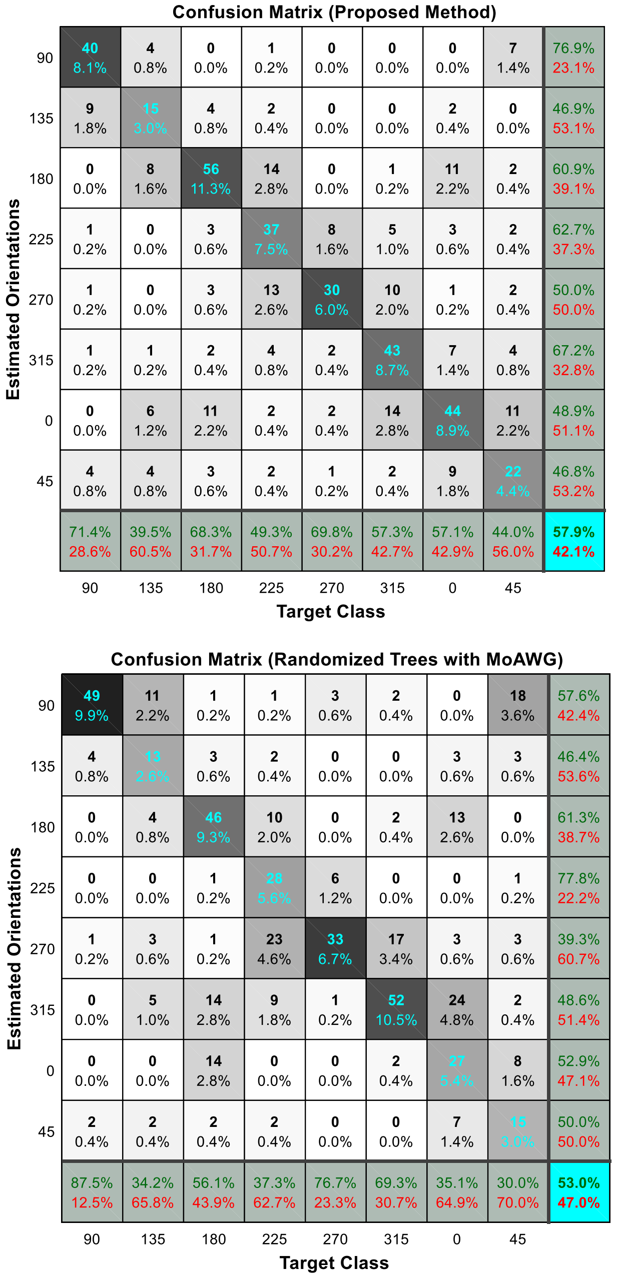
  According to (2) the inner product of β and x requires $\|\beta\|_0$[1] multiplications and $\|\beta\|_0$ accumulation operations. Also, for every multiplication and accumulation operations we need two indexing operations for the elements of β and x. We group together one multiplication, one accumulation and two indexing operations, and call the group MA2I computing unit.
- Decision Tree

  In a binary decision tree the classification label is determined in the leaf nodes. To traverse a binary tree, three operators are used: two comparison operators (comparing with the key value to check if a node is a leaf node or not), one indexing operator (continue to the left or right child). We call this sets of operations as I2C unit.
- SVM

  Computational complexity of SVM in testing phase is related to the number of produced support vectors in the learning phase. Equation (6) shows the kernel-SVM decision boundary equation for a binary classification problem:

$$\hat{y} = \operatorname{sgn}\left(\sum_{i=1}^{n} w_i y_i k(x_i, x')\right), \tag{6}$$

[1] $\|.\|_0$ (norm 0), means the number of none-zero elements of the argument vector



Confusion Matrix (Proposed Method)

Estimated Orientations

90	40 8.1%	4 0.8%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	7 1.4%	76.9% 23.1%
135	9 1.8%	15 3.0%	4 0.8%	2 0.4%	0 0.0%	0 0.0%	2 0.4%	0 0.0%	46.9% 53.1%
180	0 0.0%	8 1.6%	56 11.3%	14 2.8%	0 0.0%	1 0.2%	11 2.2%	2 0.4%	60.9% 39.1%
225	1 0.2%	0 0.0%	3 0.6%	37 7.5%	8 1.6%	5 1.0%	3 0.6%	2 0.4%	62.7% 37.3%
270	1 0.2%	0 0.0%	3 0.6%	13 2.6%	30 6.0%	10 2.0%	1 0.2%	2 0.4%	50.0% 50.0%
315	1 0.2%	1 0.2%	2 0.4%	4 0.8%	2 0.4%	43 8.7%	7 1.4%	4 0.8%	67.2% 32.8%
0	0 0.0%	6 1.2%	11 2.2%	2 0.4%	2 0.4%	14 2.8%	44 8.9%	11 2.2%	48.9% 51.1%
45	4 0.8%	4 0.8%	3 0.6%	2 0.4%	1 0.2%	2 0.4%	9 1.8%	22 4.4%	46.8% 53.2%
	71.4% 28.6%	39.5% 60.5%	68.3% 31.7%	49.3% 50.7%	69.8% 30.2%	57.3% 42.7%	57.1% 42.9%	44.0% 56.0%	57.9% 42.1%
	90	135	180	225	270	315	0	45	

Target Class

Confusion Matrix (Randomized Trees with MoAWG)

Estimated Orientations

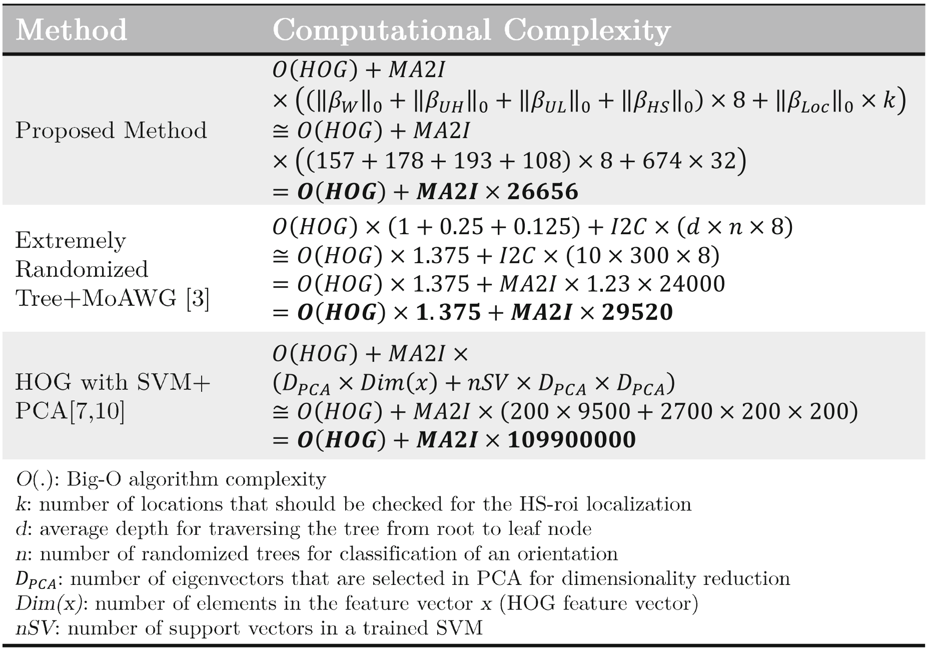
90	49 9.9%	11 2.2%	1 0.2%	1 0.2%	3 0.6%	2 0.4%	0 0.0%	18 3.6%	57.6% 42.4%
135	4 0.8%	13 2.6%	3 0.6%	2 0.4%	0 0.0%	0 0.0%	3 0.6%	3 0.6%	46.4% 53.6%
180	0 0.0%	4 0.8%	46 9.3%	10 2.0%	0 0.0%	2 0.4%	13 2.6%	0 0.0%	61.3% 38.7%
225	0 0.0%	0 0.0%	1 0.2%	28 5.6%	6 1.2%	0 0.0%	0 0.0%	1 0.2%	77.8% 22.2%
270	1 0.2%	3 0.6%	1 0.2%	23 4.6%	33 6.7%	17 3.4%	3 0.6%	3 0.6%	39.3% 60.7%
315	0 0.0%	5 1.0%	14 2.8%	9 1.8%	1 0.2%	52 10.5%	24 4.8%	2 0.4%	48.6% 51.4%
0	0 0.0%	0 0.0%	14 2.8%	0 0.0%	0 0.0%	2 0.4%	27 5.4%	8 1.6%	52.9% 47.1%
45	2 0.4%	2 0.4%	2 0.4%	2 0.4%	0 0.0%	0 0.0%	7 1.4%	15 3.0%	50.0% 50.0%
	87.5% 12.5%	34.2% 65.8%	56.1% 43.9%	37.3% 62.7%	76.7% 23.3%	69.3% 30.7%	35.1% 64.9%	30.0% 70.0%	53.0% 47.0%
	90	135	180	225	270	315	0	45	

Target Class

Fig. 8 The confusion matrix for the proposed method and Extremely Randomized Tree with MoAWG [3]

where $x' \in R^n$ is a test sample, $x_i \in R^n$ is the ith training sample, $y_i \in \{-1, +1\}$ is output for the ith training sample and w_i is the weight of the ith training sample. In our research the kernel function is considered as $tansig(x_i.x') = \frac{2}{1+e^{-2(x_i.x')}} - 1$.

Table 2 The computational complexity comparison between our proposed method and other methods



Method	Computational Complexity
Proposed Method	$O(HOG) + MA2I$ $\times \left((\lVert\beta_W\rVert_0 + \lVert\beta_{UH}\rVert_0 + \lVert\beta_{UL}\rVert_0 + \lVert\beta_{HS}\rVert_0) \times 8 + \lVert\beta_{Loc}\rVert_0 \times k \right)$ $\cong O(HOG) + MA2I$ $\times \left((157 + 178 + 193 + 108) \times 8 + 674 \times 32 \right)$ $= \boldsymbol{O(HOG) + MA2I \times 26656}$
Extremely Randomized Tree+MoAWG [3]	$O(HOG) \times (1 + 0.25 + 0.125) + I2C \times (d \times n \times 8)$ $\cong O(HOG) \times 1.375 + I2C \times (10 \times 300 \times 8)$ $= O(HOG) \times 1.375 + MA2I \times 1.23 \times 24000$ $= \boldsymbol{O(HOG) \times 1.375 + MA2I \times 29520}$
HOG with SVM+ PCA[7,10]	$O(HOG) + MA2I \times$ $(D_{PCA} \times Dim(x) + nSV \times D_{PCA} \times D_{PCA})$ $\cong O(HOG) + MA2I \times (200 \times 9500 + 2700 \times 200 \times 200)$ $= \boldsymbol{O(HOG) + MA2I \times 109900000}$

$O(.)$: Big-O algorithm complexity
k: number of locations that should be checked for the HS-roi localization
d: average depth for traversing the tree from root to leaf node
n: number of randomized trees for classification of an orientation
D_{PCA}: number of eigenvectors that are selected in PCA for dimensionality reduction
$Dim(x)$: number of elements in the feature vector x (HOG feature vector)
nSV: number of support vectors in a trained SVM

In this equation training samples corresponding to support vectors have a non-zero w_i. Therefor we have an inner product between input sample and all support vectors. Each inner product has n number of MA2I computing units.

In order to compare the MA2I and I2C units, we execute these two sets of instructions in a C# language loop control for billions of times. Our results show that $O(\text{I2C}) = 1.23 \times O(\text{MA2I})$. Therefore, time complexity of MA2I unit is less than I2C. Table 2 provides the computational complexity comparison of the proposed method with the other methods. Accordingly, the proposed method has less computational complexity than the extremely randomized tree method. In addition, the proposed method has the ability to full parallelization. In terms of space complexity, the proposed method needs to store only nonzero elements of vectors β (in our experiment there are 1200 floating numbers) and therefore it has significantly outperformed the other methods.

5 Conclusions

In this paper we proposed a novel approach to estimate the body orientation. We combined the results of four classifiers corresponding to the four regions of interest. We used the HOG descriptor as a feature to represent each region. HOG computation is performed only once for a person in this approach. In addition, we used logistic regression as a classifier, which is very simple and efficient. We also proposed a weighting scheme on classifiers outputs to increase the classification accuracy. The experimental results showed that the proposed method outperforms the other methods in terms of classification accuracy and computational complexity.

References

1. Andriluka M, Roth S, Schiele B (2009) Pictorial structures revisited: People detection and articulated pose estimation. In: IEEE computer society conference on computer vision and pattern recognition (CVPR 2009)
2. Andriluka M, Roth S, Schiele B (2010) Monocular 3d pose estimation and tracking by detection. In: IEEE conference on computer vision and pattern recognition (CVPR), 2010. IEEE, pp 623–630
3. Baltieri D, Vezzani R, Cucchiara R (2012) People orientation recognition by mixtures of wrapped distributions on random trees. In: Computer vision–ECCV 2012. Springer, pp 270–283
4. Baltieri D, Vezzani R, Cucchiara R (2015) Mapping appearance descriptors on 3d body models for people re-identification. Int J Comput Vis 111(3):345–364
5. Bourdev L, Malik J (2009) Poselets: body part detectors trained using 3d human pose annotations. In: IEEE 12th international conference on computer vision, 2009. IEEE, pp 1365–1372
6. Chavel I (2006) Riemannian geometry: a modern introduction, vol 98. Cambridge University Press
7. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005, vol 1. IEEE, pp 886–893
8. Enzweiler M, Gavrila DM (2010) Integrated pedestrian classification and orientation estimation. In: IEEE conference on computer vision and pattern recognition (CVPR), 2010. IEEE, pp 982–989
9. Fanelli G, Gall J, Van Gool L (2011) Real time head pose estimation with random regression forests. In: IEEE conference on computer vision and pattern recognition (CVPR), 2011. IEEE, pp 617–624
10. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645
11. Gandhi T, Trivedi MM (2008) Image based estimation of pedestrian orientation for improving path prediction. In: IEEE intelligent vehicles symposium, 2008. IEEE, pp 506–511
12. Gavrila DM, Munder S (2007) Multi-cue pedestrian detection and tracking from a moving vehicle. Int J Comput Vis 73(1):41–59
13. Huang D, Storer M, De la Torre F, Bischof H (2011) Supervised local subspace learning for continuous head pose estimation. In: IEEE conference on computer vision and pattern recognition (CVPR), 2011. IEEE, pp 2921–2928
14. Komarek P, Moore AW (2005) Making logistic regression a core data mining tool with tr-irls. In: 5th IEEE international conference on data mining. IEEE, pp 4–pp
15. Mu Y, Yan S, Liu Y, Huang T, Zhou B (2008) Discriminative local binary patterns for human detection in personal album. In: IEEE conference on computer vision and pattern recognition, 2008. CVPR 2008. IEEE, pp 1–8
16. Murphy-Chutorian E, Trivedi MM (2009) Head pose estimation in computer vision: a survey. IEEE Trans Pattern Anal Mach Intell 31(4):607–626
17. Nguyen DT, Li W, Ogunbona PO (2016) Human detection from images and videos: a survey. Pattern Recogn 51:148–175
18. Polikar R (2006) Ensemble based systems in decision making. IEEE Circuits Syst Mag 6(3):21–45
19. Shimizu H, Poggio T (2004) Direction estimation of pedestrian from multiple still images. In: IEEE intelligent vehicles symposium, 2004. IEEE, pp 596–600
20. Smith K, Ba SO, Odobez JM, Gatica-Perez D (2008) Tracking the visual focus of attention for a varying number of wandering people. IEEE Trans Pattern Anal Mach Intell 30(7):1212–1229
21. Sumner M, Frank E, Hall M (2005) Speeding up logistic model tree induction. In: 9th European conference on principles and practice of knowledge discovery in databases. Springer, pp 675–683
22. Tao J, Klette R (2013) Integrated pedestrian and direction classification using a random decision forest. In: Proceedings of the IEEE international conference on computer vision workshops, pp 230–237
23. Tosato D, Spera M, Cristani M, Murino V (2013) Characterizing humans on riemannian manifolds. IEEE Trans Pattern Anal Mach Intell 35(8):1972–1984
24. Tran D, Forsyth D (2010) Improved human parsing with a full relational model. In: Computer vision–ECCV 2010. Springer, pp 227–240
25. Tuzel O, Porikli F, Meer P (2008) Pedestrian detection via classification on riemannian manifolds. IEEE Trans Pattern Anal Mach Intell 30(10):1713–1727
26. Zhu Q, Yeh MC, Cheng KT, Avidan S (2006) Fast human detection using a cascade of histograms of oriented gradients. In: IEEE computer society conference on computer vision and pattern recognition, 2006, vol 2. IEEE, pp 1491–1498

Ali Sebti received his B.Sc. degree in software engineering from the School of Computer Engineering, Yazd University, Yazd, Iran, in 2006, and his M.Sc. degree from the Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran, in 2009, majoring in artificial intelligence. He is currently pursuing the Ph.D. degree with the Laboratory of Image Processing and Data Mining, University of Shahrood, Shahrood, Iran. His research interest focuses on people re-identification in video surveillance systems.

Hamid Hassanpour received the B.S. degree in computer engineering from Iran University of Science and Technology, Tehran, Iran, in 1993, the M.S. degree in computer engineering from Amirkabir University of Technology, Tehran, Iran, in 1996, and the Ph.D. from the Queensland University of Technology, Brisbane, Australia, in 2004. He has a professor position in faculty of Computer Engineering & IT at the University of Shahrood, Iran. His research interests include Image Processing, Signal Processing, timefrequency signal processing and analysis.