Part-based RDF for Direction Classification of Pedestrians, and a Benchmark

Junli Tao and Reinhard Klette

The .enpeda.. Project, Tamaki Campus
The University of Auckland, Auckland, New Zealand

Abstract. This paper proposes a new benchmark dataset for pedestrian body-direction classification, proposes a new framework for intra-class classification by directly aiming at pedestrian body-direction classification, shows that the proposed framework outperforms a state-of-the-art method, and it also proposes the use of DCT-HOG features (by combining a discrete cosine transform with the histogram of oriented gradients) as a novel approach for defining a random decision forest.

1 Introduction

Human beings are the most important objects for image sequence analysis for *advanced diver-assistance systems* (ADAS) or surveillance applications. Their study in video data attracted extensive research [19]. The appearance of a human in a single frame contains information about body pose, head pose, head direction, body direction, and so forth. Algorithms for pose-estimation tasks typically require high-resolution images as input. Low-resolution cameras, or humans recorded far away from the camera, still support the estimation of global information about the direction of a person expressed by the recorded pose.

Information about body direction helps to improve path predictions in sequences; see [12]. In the ADAS area, pedestrians are the most vulnerable road users. A pedestrian may change a walking path abruptly; motion information acquired in previous frames does not necessarily define an accurate prediction of the future path. For example, Fig. 1 shows a sample from the pedestrian path-prediction benchmark dataset proposed in [22]; the dataset is available on [13]. In video surveillance, a person's direction offers clues for solving specific tasks such as behavior recognition, group detection, or interaction analysis; see [3, 16]. In [16], the head direction is noted for analysing the interaction between two persons in a film clip.

Our first contribution in this paper is a proposal of a new *Pedestrian Direction Classification* (PDC) dataset; thus responding to an obvious demand in this area for more benchmark data. The task of discrete body-direction classification is not yet studied as intensively as a generic pedestrian-detection task, and the lack of benchmark datasets might be one reason for this situation. The proposed PDC dataset has been generated based on the *Daimler Mono Pedestrian Classification Benchmark*. There are already two existing datasets which provide ground truth on pedestrian directions, the *TUD Multiple View Pedestrian* (TUD) dataset prosed by [1], and the *Human Orientation Classification*

(HOC) dataset introduced by [7]. We show that the the newly introduced PDC dataset outperforms both the TUD and the HOC dataset with respect to defined criteria. A detailed comparison of the TUD, HOC, and the proposed PDC dataset is given in Sections 4 and 5.

The second contribution in this paper is the proposal of an efficient framework (PRDF) for pedestrian direction classification. In order to deal with the multiple intra-class (i.e. different body direction classes in one pedestrian class) classification task, see [1, 3, 14, 18, 26], previously proposed methods learn multiple classifiers, one for each direction. This excludes sharing of information among classes in the training or classification process, and is (thus) more time consuming for both processes. A random decision forest (RDF) is adopted in [2, 24] for direction classification. In both publications, features of a bounding box are used as input for the classifier. Each splitting node in a tree of the RDF, however, selects only one component in the used high-dimensional feature vector. In this paper we propose an PRDF for automatically learning the discrimination of selected body parts for classifying body directions. Experimental results show that the proposed PRDF framework performs better and faster than state-of-the-art methods.

As a third contribution we are proposing a novel feature for direction classification. Features of the histogram of oriented gradients (HOG) are extensively used for pedestrian detection [6]. As reported in [5], complex splitting nodes lead to over-fitting issues. In [24], one or two feature elements are adopted for defining a splitting function. In this paper we propose to perform a discrete cosine transform (DCT) over the HOG feature vector to obtain a more global descriptor before selecting feature elements. Thus, each element contains global frequency information instead of just some local gradient-orientation information.

To summarise the contributions in this paper, we (1) propose a new pedestrian body-direction classification benchmark dataset (PDC), (2) propose a new framework (PRDF) for intra-class classification by directly aiming at pedestrian body-direction classification, (3) show that the proposed framework outperforms a state-of-the-art method, and we (4) propose the use of DCT-HOG features.

Fig. 1. Frames of a body-bending sequence of the Daimler pedestrian path prediction benchmark dataset [13]

2 Related Work

We briefly review body-direction classification algorithms and classifiers based on random decision forests, given a bounding box containing a pedestrian. For solving the body-direction classification task, as an intra-class classification problem, researchers train multiple two-class classifiers [1, 3, 14, 18, 26] or adopt a multi-class classifier [24, 7]. Both approaches use features and classifiers as previously known for pedestrian detection. For example, HOG features and support vector machine (SVM) classifiers are extensively employed for body-direction classification [1, 3, 14].

The authors of [7] propose a weighted array of covariances (WARCO) for deriving features for classifying body- and head-direction. The values of 13 combined feature channels are taken as defining a manifold in feature space; those 13 channels are composed of eight difference-of-offset-Gaussian filter channels, three color channels, gradient magnitude, and a gradient-direction channel. A method based on silhouettes is presented in [18]; used shape descriptors limit the range of body directions to the interval $[0°, 180°]$. The estimation of pedestrian direction is performed in [23] more robustly by selecting a recognition result based on multiple still images, rather than by using just a single image. Multiple random-tree classifiers are trained in [2] and compared with trained SVM classifiers; outputs are integrated using a *mixture of approximated wrapped Gaussians* (MAWG). Using calculated probabilities of multiple outputs obtained from all participating classifiers, the final direction is obtained by maximising the mixed probability of the WAWG.

Direction recognition is difficult as head pose, torso, and body might point into different directions; but their poses are interrelated to each other. For example, body direction is estimated in [3, 4] by considering location and head pose, and assuming that tracks are available.

There are also several methods proposed for classifying pedestrians against a background together with their body directions [14, 24, 7]. The authors of [24] propose to modify the objective function of each split node in the RDF for simultaneously handling both tasks, pedestrian detection and direction classification, a single, or two HOG elements are compared against a randomly generated threshold, and results are selected for optimising a combined objective function. [14] presents a three stage process; Stages 1 and 2 adopt different HoG where blocks are either overlapping or not, to reject non-pedestrian boxes; at Stage 3, four SVM classifiers are trained for the four directions separately using pedestrian samples only. A unified Bayesian model is used in [9], based on shape and motion cues; the proposed method classifies pedestrians with recognizing one of four possible directions.

RDFs are extensively applied for many detection or categorisation subjects, including object detection [11, 15], action recognition [25], image labelling [17], or edge detection [8]. The structure of trees in an RDF depends on the training process, which may vary for different subjects. In [15], an RDF is structured for doing pedestrian detection. Instead of using simple algebraic splitting functions, a two-class SVM is adopted for each splitting node. The authors of [20] propose

alternating decision forests; instead of independently learning each tree in a forest, a gradient boosting theory is introduced for concurrently training a forest. For each depth level, a significance distribution of training samples is updated based on the performance of the current forest. Miss-classified samples receive more attention when training split nodes at the next depth level. The authors of [21] propose corresponding alternating regression forests.

3 Proposed Algorithm

We detail the proposed algorithm. We start with introducing the used notation following a general RDF framework.

3.1 Random Decision Forest

An RDF acts for a given categorisation problem as a (strong) classifier, defined by a set of trees, each acting as a weak classifier. Let T_t, for $t \in \{1, \ldots, N\}$, be a set of randomly trained decision trees which defines an RDF. In each tree, a classification problem is splitted by answering subsequently "simple questions" defined by split functions. In other words, such a decision tree consists of a set of split functions hierarchically arranged into a tree structure.

A decision tree has internal (or split) and terminal (or leaf) nodes. We assign a split function to each split node which has two out-edges connected to two nodes, being either split or leaf nodes. The assigned split function $h_\phi(\cdot)$ decides which of the two nodes comes next. Let $\mathcal{I}$ denote the set of inputs and

$$\mathcal{I}_L(\phi) = \{I \in \mathcal{I} | h_\phi(I) = 0\} \quad \text{and} \quad \mathcal{I}_R(\phi) = \{I \in \mathcal{I} | h_\phi(I) = 1\} \tag{1}$$

Later we specify split function $h_\phi(\cdot)$ and its parameters ϕ.

A set $\mathcal{I}^{tr} \subset \mathcal{I}$ of labelled pedestrian are used in the training process for expanding the trees of an RDF. Samples $I \in \mathcal{I}^{tr}$ split along internal nodes and end up in leaf nodes of trained trees.

A decision tree is trained by growing subsequently internal nodes, starting at a root node. Suitable functions $h_\phi(\cdot)$ are selected with respect to a predefined target function. Trees of an RDF grow randomly and independently to each-other. Randomness when training a tree is important to ensure some variety in the forest (i.e. trees need to be uncorrelated to ensure that the forest can investigate samples from "different perspectives"). For the assembled forest we intentionally avoid to grow "similar" trees.

A stop criterion defines when a leaf node L is created. The distribution of classes in a leaf node is obtained with respect to those samples $I \in \mathcal{I}^{tr}$ which reach this leaf node. According to this the distribution, the leaf node assigns probabilities $p(d|L_t)$, for $d \in \{N, E, S, W\}$, where N, E, S, W denote body directions.

For testing of an RDF we use a set of input bounding boxes denoted by $\mathcal{I}^{ts} \subset \mathcal{I}$. Any sample $I^{ts} \in \mathcal{I}^{ts}$ is passed through the N trees T_t of the trained

forest. Sample I^{ts} ends up in a leaf node L_t in tree T_t. This way we assign N distributions to one test box. The simple rule

$$d^* = \arg\max_d \sum_{t=1}^{N} p(d|L_t) \tag{2}$$

defines a maximum-likelihood decision for classifying the direction of a pedestrian.

3.2 Split and Objective Function

Let V denote a feature vector, and i be the index of a feature element. A split function is then defined as follows:

$$h_\phi(I) = \begin{cases} 0 & \text{if} \quad V(i) > \tau \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

The goal is to split the training samples uniformly for maximising the information gain at each internal node. More specific, for each internal node, parameters $\phi = \{i, \tau\}$ are learned with respect to maximizing a predefined objective function. It is important to choose an appropriate objective function for obtaining "good" split functions during the training process. This is supported by Shannon's entropy-based objective function. Let $E_d(\mathcal{I})$ denote the entropy of direction classes. We use

$$\begin{aligned} o_d(\phi, \mathcal{I}) &= E_d(\mathcal{I}) - \sum_{k \in \{L,R\}} \omega_k E_d(\mathcal{I}_k(\phi)) \\ &\text{with} \quad E_d(\mathcal{I}) = -\sum_d p(d|\mathcal{I}) \log(p(d|\mathcal{I})) \\ &\text{and} \quad \omega_k = |\mathcal{I}_k(\phi)|/|\mathcal{I}(\phi)| \end{aligned} \tag{4}$$

The ω-values are the weights for balancing the bias caused by varying numbers of samples, going either to the left or right child node. By allowing different objective functions, split nodes can be generated individually.

3.3 DCT-HOG Feature

The HOG was introduced in [6] for pedestrian detection. The steps of HOG calculation can be summarised as follows:

- An input gray-level image is partitioned into cells of equal sizes (e.g. of 8×8 pixels).
- For each cell, the gradient magnitude at each pixel in the cell votes to "its" discrete phase bin (e.g. nine bins, $[0^o, 20^o]$, $[20^o, 40^o]$,...,$[160^o, 180^o]$). Thus, each cell contains nine elements, where each element corresponds to the sum of gradient magnitudes in those discrete phase ranges.

– To obtain a feature vector, blocks of identical size (e.g. each 2×2 cells) slide through the cell matrix. The cell elements are normalized within the block and combined into one vector (either column or row wise).
– The vectors from all those blocks are now augmented to generate the *HOG feature vector* V_{hog}.

HOG feature vectors are extensively used in object detection because of their positive performance compared to other features (e.g. local binary patterns, or histogram of optical flow). For splitting training samples at a reached node, one feature element is adopted at a time. One feature element in HOG contains specifically local information for one phase bin of one cell. In order to employ more global information in a split node, we propose the DCT-HOG feature $V_{dct-hog}$. It is obtained by applying the discrete cosine transform (DCT) first over the HOG feature vector V_{hog} taken as a 1-dimensional discrete signal:

$$V_{dct-hog} = C^{(|V_{hog}|)} \cdot V_{hog} \tag{5}$$

$$c_{jk}^{(|V_{hog}|)} = \sqrt{\alpha_j / |V_{hog}|} \cdot \cos\left(\frac{\pi(2k+1)j}{2|V_{hog}|}\right) \tag{6}$$

where $|V|$ denotes the number of elements in a vector V, $c_{jk}^{(|V_{hog}|)}$ is an element in the orthogonal matrix $C^{(|V_{hog}|)}$ of dimension $|V_{hog}| \times |V_{hog}|$, and $\alpha_0 = 1$, $\alpha_j = 2$. The elements in DCT-HOG contains global frequency information. Thus, global information is adopted in a split node, when splitting based on the DCT-HOG feature elements.

3.4 Part Based Random Decision Forest

A conventional RDF learning procedure is based on feature vectors of the whole object - in our case, of a person. Following the classifier's internal structure, introduced in Section 3.2, a randomly selected element from HOG is not

Fig. 2. Selected parts for several trees. *Bottom, right*: All the selected parts for a forest

a discriminative representation of the sample. Thus, we propose to apply the forest to deduct discriminative information from image patches, i.e. from parts of the human body.

Instead of mixing randomly selected local patches from random locations, we use the same location and size of a patch in each training sample as a tree-training set. In this way, the location of body parts is kind of "encoded" for training. The tree-training procedure focuses on the appearance of the body parts.

Using such localized regions, simple split functions yield better performance compared to an application of the whole bounding box for tree training; see

Algorithm 1 (Training)
Input: All training samples $\mathcal{I}$
Output: Trained trees T_t, for $t = 1, 2, \ldots, N$

1: randomly select the location and size for an image patch, identified by top-left coordinates (row_t, col_t) and patch size $(width_t, height_t)$.
2: calculate feature vector V of each training patch; for different experiments the V stands either for V_{hog}, $V_{dct-hog}$, or V_{comb}.
3: let $T_t = \emptyset$, $num = |\mathcal{I}|$, $dep = 0$, stop criterion $t_{num} = 20$, $t_{dep} = 15$, temporal data store variables $temp_{od1} = 0$, $temp_{od2} = 0$.
4: **if** $num < t_{num}$ || $dep > t_{dep}$ **then**
5: calculate $p(d|L)$ with $\mathcal{I}$, according to Equ. (8);
6: add leaf L to the tree: $T_t = T_t \cup L$
7: return T_t.
8: **else**
9: $dep = dep + 1$;
10: **for** $s = 1, \ldots, 1000$ **do**
11: randomly select a feature element index i_s;
12: find range $[\tau_{min}, \tau_{max}]$ of V_{i_s} with current node samples;
13: **for** $h = 1, \ldots, 10$ **do**
14: randomly select $\tau_h \in [\tau_{min}, \tau_{max}]$;
15: split $\mathcal{I}$ into $\mathcal{I}_{Lh}, \mathcal{I}_{Rh}$ according to Equ. (3);
16: calculate $o_d(\{i_s, \tau_h\}, \mathcal{I})$ with Equ. (4);
17: **if** $o_d(\{i_s, \tau_h\}, \mathcal{I}) > temp_{od2}$ **then**
18: $temp_{od2} = o_d(\phi_s, \mathcal{I})$;
19: $\tau_s = \tau_h$, $\phi_s = \{i_s, \tau_s\}$;
20: **end if**
21: **end for**
22: **if** $temp_{od2} > temp_{od1}$ **then**
23: $\phi^* = \phi_s$;
24: **end if**
25: **end for**
26: expand tree by new split node: $T_t = T_t \cup \phi^*$;
27: split $\mathcal{I}$ into $\mathcal{I}_L$ and $\mathcal{I}_R$;
28: $num = |\mathcal{I}_L|$, $\mathcal{I} = \mathcal{I}_L$, and go to Line 4;
29: $num = |\mathcal{I}_R|$, $\mathcal{I} = \mathcal{I}_R$, and go to Line 4;
30: **end if**

results in Section 5. Figure 2 illustrates selected discriminative parts for a tree and a forest. For the considered intra-class classification task, the global appearance of a person is similar to some degree for the whole pedestrian class; the distinctive information among classes actually lies in local body parts.

3.5 Implementation

For the used training and testing algorithms, see Algorithms 1 and 2, respectively. We apply the whole training set when training a tree of the RDF. As reported in [5], randomness is significant for the performance of a forest. Instead of bagging, we introduce randomness by randomly selecting patches and feature elements. The stop criteria parameters, including the tree's depth and the minimum number of samples, is set according to [24].

Because of the different cardinalities of training samples for the different classes, a sample-bias compensation is necessary for calculating probabilities at a leaf node. This is achieved by using a balancing factor r_d, defined as follows:

$$p(d|L) = |\mathcal{I}_d^L| \cdot r_d / \sum_d (|\mathcal{I}_d^L| \cdot r_d) \quad \text{with} \quad r_d = |\mathcal{I}^{tr}|/|\mathcal{I}_d^{tr}| \tag{7}$$

where $|\mathcal{I}^{tr}|$ denotes the cardinality of the training samples for each tree; set $\mathcal{I}_d^{tr} \subset \mathcal{I}^{tr}$ contains the training samples for direction d in $\mathcal{I}^{tr}$, and set $\mathcal{I}^L \subset \mathcal{I}^{tr}$ contains all the samples arriving at leaf node L. In our experiments, we set $N = 120$.

During testing, for each tree T_t, the corresponding patch I_{patch}, specified by location (row_t, col_t) and size $(width_t, height_t)$, from a test image I is adopted to calculate the feature vector V, and then passed through the tree. See Algorithm 2 for details.

Algorithm 2 (Testing)
Input: Test bounding box I, trained trees T_t, with $t = 1, 2, \ldots, N$.
Output: Class label d^*.

1: **for** $t = 1, \ldots, N$ **do**
2: extract corresponding patch location (row_t, col_t) and size $(width_t, height_t)$ from tree t.
3: calculate feature vector V for the image patch I_{patch}.
4: pass V through T_t until reaching a leaf node L_t, obtain distribution $p(d|L_t)$.
5: **end for**
6: obtain d^* with Equ. (2);
7: return d^*.

4 Proposed PDC Benchmark Dataset

We introduce our *pedestrian direction classification* (PDC) benchmark dataset.[1] We compare our dataset with two other available datasets, TUD and HOC. Sample images are shown in Fig. 3, top.

Fig. 3. *Top row:* PDC samples illustrating ambiguity between *N*, *NE*, and *E*. *Middle and bottom rows:* Sample images from the HOC (*middle*) and TUD (*bottom*) datasets

Besides very advanced research in the driver-assistance area, there is not yet any publicly available pedestrian body-direction classification dataset available in a driving context. Thus, we use one popular pedestrian-classification dataset from Daimler, and manually classified the 12,000 pedestrian bounding boxes (sized 48×96) into 8 directions (namely *N*, *NE*, *E*, *SE*, *S*, *SW*, *W*, or *NW*). Even for human beings, it is difficult to classify which direction a pedestrian should be assigned, e.g. among *N*, *NE*, and *E*, *E*, *SE*, and *S*, *S*, *SW*, and *W*, and *W*, *NW*, and *N*. Due to the ambiguity, we classified a person, for example, to *NE* only if the person is facing into diagonal direction.

As the PDC pedestrians are shown in various scenes, an often cluttered background enables that the PDC dataset gives more general information when used

[1] See ccv.wordpress.fos.auckland.ac.nz/data/object-detection/.

Table 1. PDC, HOC, and TUD datasets summary

	Number of samples	Image channel	Directions
HOC	11,881	color	*N*, *S*, *E*, *W*
TUD	5,183	color	*N*, *NE*, *E*, *SE*, *S*, *SW*, *W*, *NW*
PDC	12,000	gray	*N*, *NE*, *E*, *SE*, *S*, *SW*, *W*, *NW*

for training, and it is more challenging when used for testing; see experimental results in Section 5.

The HOC dataset [7] contains 11,881 bounding boxes, including 6,860 training samples and 5,021 test samples. The original image size is 62×132. The bounding boxes are extracted from ETHZ data, see [10]. The sequences are taken from two cameras mounted on top of a trolley. Four directions, *N*, *S*, *E*, *W*, are labelled for each sample. The TUD dataset [1] contains 5,183 bounding boxes, including 4,935 training samples and 248 test samples. This dataset labels pedestrians for 8 directions, as in our PDC dataset. Sample images from the HOC and TUD datasets are shown in Fig. 3, bottom. Information for all three datasets is summarized in Table 1.

5 Experiments

We report about several sets of experiments for illustrating the performance of the proposed PRDF framework for all datasets, the merits of the new proposed PDC dataset for both training and testing tasks, the performance of the proposed feature DCT-HOG, and when augmenting DCT-HOG with a HOG feature. Finally, the proposed algorithm is compared with state-of-the-art methods as reported in [1–3, 24, 7]. We also provide the mean processing time for testing one image.

5.1 RDF versus PRDF

To compare the proposed *part-based random decision forest* (PRDF) with the conventional RDF, two algorithms are trained with HOG feature on the three datasets TUD, HOC, and PDC respectively. As the testing set of the TUD dataset only contains 248 images, it is not quite sufficient to evaluate the performance; we use PDC as the test set when using TUD for training, and, correspondingly, TUD as the testing set after training on PDC. The confusion matrices are given in Tables 2 and 3. The average error, depending from the number of applied trees, is shown in Fig. 4.

Figure 4 shows curves for the results of RDFs using different colors for different training sets. For results of PRDFs we use squares in the colors of the used training set. Obviously, the PRDFs outperform the corresponding RDFs (e.g. PRDF-HOC versus RDF-HOC) for all datasets. Thus, for later experiments,

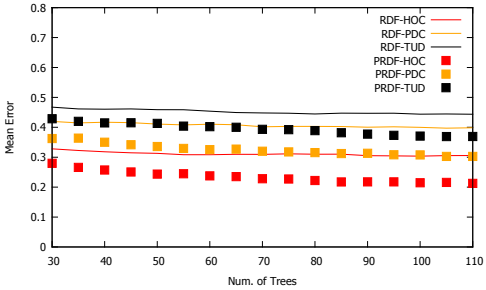

Fig. 4. Mean errors of RDFs and PRDFs for used numbers of trees in the forest

Table 2. RDF confusion matrices. *Left to right*: TUD,HOC,PDC

	N	E	S	W
N	0.67	0.04	0.26	0.04
E	0.17	0.46	0.24	0.13
S	0.45	0.07	0.43	0.04
W	0.15	0.14	0.22	0.49

	N	E	S	W
N	0.77	0.07	0.07	0.10
E	0.10	0.67	0.13	0.10
S	0.09	0.05	0.78	0.07
W	0.11	0.10	0.20	0.60

	N	E	S	W
N	0.63	0.13	0.20	0.04
E	0.06	0.81	0.03	0.09
S	0.33	0.18	0.37	0.12
W	0.08	0.15	0.03	0.74

Table 3. PRDF confusion matrices. *Left to right*: TUD, HOC, PDC

	N	E	S	W
N	0.76	0.02	0.2	0.02
E	0.16	0.55	0.19	0.09
S	0.41	0.04	0.52	0.02
W	0.16	0.10	0.16	0.57

	N	E	S	W
N	0.84	0.03	0.07	0.06
E	0.05	0.76	0.10	0.09
S	0.05	0.02	0.89	0.04
W	0.07	0.06	0.18	0.69

	N	E	S	W
N	0.70	0.07	0.19	0.04
E	0.05	0.90	0.01	0.04
S	0.27	0.13	0.52	0.08
W	0.11	0.09	0.02	0.78

PRDF framework is adopted. The confusion matrices quantify how the PRDFs improve the classification performance for body directions.

5.2 Dataset Comparisons

In order to compare the datasets, the PRDFs trained with HOG feature on one of the three datasets are subsequently tested on the other two training sets respectively. For example, a TUD trained PRDF is tested on HOC and PDC training sets. The test results are summarised in Table 4. The mean error is shown in Fig. 5.

In Fig. 5, the TUD-HOC means PRDF trained with TUD training set, and HOC training set is adopted as test set. Both TUD and PDC trained PRDFs perform better than an HOC trained PRDF when tested on TUD and PDC training sets. The PDC trained PRDF significantly outperforms a TUD trained PRDF over HOC, and the HOC trained PRDF over TUD. Using the HOC

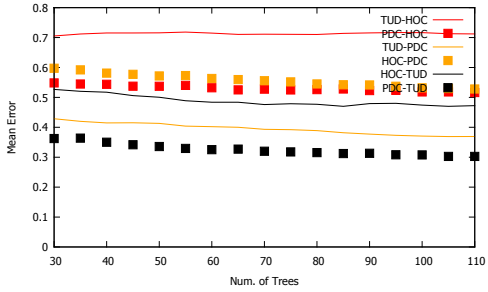

Fig. 5. Mean errors of cross-tested datasets, e.g. TUD-HOC is trained on the TUD training set, but tested on the HOC training set

Table 4. Confusion matrices. *Top, left to right*: TUD-HOC, TUD-PDC, HOC-TUD. *Bottom, left to right*: HOC-PDC, PDC-TUD, PDC-HOC

	N	E	S	W
N	0.10	0.23	0.36	0.31
E	0.12	0.34	0.23	0.31
S	0.10	0.19	0.37	0.34
W	0.11	0.28	0.27	0.34

	N	E	S	W
N	0.76	0.02	0.2	0.02
E	0.16	0.55	0.19	0.09
S	0.41	0.04	0.52	0.02
W	0.16	0.10	0.16	0.57

	N	E	S	W
N	0.62	0.07	0.22	0.10
E	0.04	0.46	0.28	0.22
S	0.30	0.11	0.47	0.12
W	0.08	0.24	0.18	0.50

	N	E	S	W
N	0.53	0.13	0.19	0.16
E	0.07	0.29	0.29	0.35
S	0.22	0.12	0.52	0.13
W	0.09	0.19	0.22	0.50

	N	E	S	W
N	0.70	0.07	0.19	0.04
E	0.05	0.90	0.01	0.04
S	0.27	0.13	0.52	0.08
W	0.11	0.09	0.02	0.78

	N	E	S	W
N	0.75	0.09	0.12	0.04
E	0.36	0.42	0.12	0.10
S	0.31	0.16	0.44	0.09
W	0.39	0.17	0.13	0.31

training set as testing set leads to the largest mean error. The TUD training set is the easiest one according to this experiment. Thus, we conclude that the proposed PDC dataset offers better generalized information for training, and also offers challenges for testing in general.

The confusion matrices show that a TUD-trained PRDF appears to be totally confused with respect to testing on the HOC data, but performs reasonable on the PDC training set. An HOC-trained PRDF performs worse for classifier E direction on both the TUD and PDC training set, while a PDC-trained PRDF performs best for classifying E.

5.3 Feature Comparison

In this section, the PRDF is trained with three different feature settings, HOG, DCT-HOG, or a combined HOG and DCT-HOG, for the three datasets. The performance of nine PRDFs is shown in Fig. 6. The confusion matrices are illustrated in Tables 3, 5, and 6.

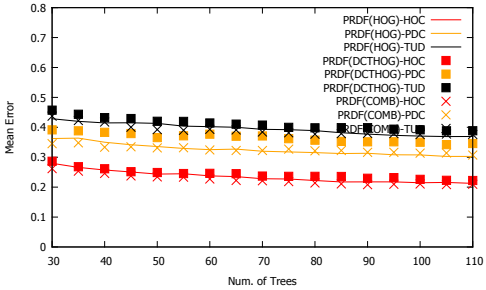

Fig. 6. Mean error of PRDFs for different feature selections. PRDF(HOG)-TUD means trained on the TUD training set using HOG features. PRDF(DCTHOG)-TUD means trained on the TUD training set using DCT-HOG features. PRDF(COMB)-TUD means trained on the TUD training set using combined HOG and DCT-HOG features.

Table 5. PRDF(DCTHOG) confusion matrices. *Left to right*: TUD, HOC, PDC

	N	E	S	W
N	0.82	0.02	0.15	0.02
E	0.20	0.54	0.12	0.13
S	0.59	0.05	0.32	0.04
W	0.21	0.12	0.10	0.57

	N	E	S	W
N	0.83	0.03	0.08	0.06
E	0.06	0.73	0.11	0.10
S	0.06	0.03	0.86	0.05
W	0.08	0.07	0.17	0.67

	N	E	S	W
N	0.73	0.07	0.16	0.03
E	0.06	0.90	0.01	0.03
S	0.43	0.15	0.39	0.04
W	0.12	0.17	0.02	0.69

Table 6. PRDF(COMB) confusion matrices. *Left to right*: TUD, HOC, PDC

	N	E	S	W
N	0.79	0.01	0.19	0.01
E	0.17	0.54	0.16	0.13
S	0.51	0.04	0.41	0.04
W	0.18	0.10	0.12	0.60

	N	E	S	W
N	0.84	0.02	0.07	0.07
E	0.05	0.78	0.11	0.07
S	0.03	0.03	0.90	0.04
W	0.06	0.07	0.18	0.69

	N	E	S	W
N	0.72	0.07	0.16	0.04
E	0.05	0.92	0.01	0.02
S	0.36	0.16	0.44	0.04
W	0.08	0.13	0.01	0.78

Figure 6 illustrates that the HOG feature performs better than DCT-HOG feature in the mean-error sense. But the confusion matrices show that the N direction classification is improved by using the DCT-HOG feature when training on the TUD and HOC datasets. The combined HOG and DCT-HOG feature slightly improves the overall performance. The confusion matrices tell us that the combined feature improves performance for all direction classifications except for direction S.

5.4 Comparison with State-of-the-Art Algorithms

The proposed PRDF(COMB) is compared with two methods proposed in [7], called FEOB, and CBH1. To ensure a fair comparison with the results in [7], PRDF(COMB) is trained with the HOC training set. Three confusion matrices

are shown in Table 7. The proposed PRDF(COMG) performs best on classifying N, E, S, and achieved the highest average accuracy (0.79).

Table 7. Confusion matrices of FEOB [23], CBH1 [23], and PRDF(COMB) on HOC dataset. *Left*: FEOB (average accuracy 0.78189). *Middle*: CBH1 (average accuracy 0.78692). *Right*: PRDF(COMB) (average accuracy 0.79031)

	N	E	S	W
N	0.76	0.11	0.04	0.09
E	0.03	0.76	0.15	0.06
S	0.00	0.04	0.88	0.08
W	0.04	0.08	0.15	0.73

	N	E	S	W
N	0.77	0.10	0.04	0.09
E	0.03	0.77	0.14	0.06
S	0.00	0.04	0.88	0.08
W	0.04	0.08	0.15	0.73

	N	E	S	W
N	0.84	0.02	0.07	0.07
E	0.05	0.78	0.11	0.07
S	0.03	0.03	0.90	0.04
W	0.06	0.07	0.18	0.69

For the sake of completeness of experiments, the results of PRDF, trained on the TUD set, are illustrated in Table 8, along with results reported in [1–3, 24]. Note that the test set contains 248 images only. We do not consider this as being a sufficient test set for drawing general conclusions.

Table 8. Test results for 248 test images from TUD

	N	E	S	W
PRDF	0.85	0.82	0.26	0.71
[24]	0.91	0.85	0.37	0.69
[2]	0.76	0.95	0.64	0.86
[3]	0.71	0.65	0.41	0.70
[1]	0.46	0.54	0.4	0.38

Processing Time. The mean processing time for one test input over 120 trees is 0.28 seconds. As trees are independent classifiers, parallel processing could be applied. Thus, the process time could be reduced to 2-3 milliseconds. The given processing time was measured for a standard desktop PC, with 3.4 GHz CPU, and 8 GB RAM. All the methods are coded with C++, and compiled with Visual Studio 2010.

6 Conclusions

This paper proposed a new pedestrian-direction classification benchmark dataset and a new framework for solving the pedestrian direction classification task. Experimental results prove that the proposed benchmark dataset outperforms the existing two datasets based on defined criteria. The proposed framework performs better than a state-of-the-art method on the HOC dataset. Results support future work on applying derived direction information while tracking pedestrians in sequences. Pedestrians are assumed to be located in the middle of the given bounding boxes. Localization errors also need to be considered when applying the proposed method. The PRDF framework may be tested with additional features, e.g. PCA-HOG.

References

1. Andriluka, M., Roth, S., Schiele, B.: Monocular 3d pose estimation and tracking by detection. CVPR, (2010) 623–630.
2. Baltieri, D., Vezzani, R., Cucchiara, R.: People orientation recognition by mixtures of wrapped distributions on random trees. ECCV, (2012) 270–283
3. Chen, C., Heili, A., Odobez, J.-M.: Combined estimation of location and body pose in surveillance video. AVSS, (2011) 5–10
4. Chen, C., Heili, A., Odobez, J.-M.: A joint estimation of head and body orientation cues in surveillance video. ICCV Workshops, (2011) 860–867
5. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations Trends Computer Graphics Vision, 7 (2011) 81–227
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. CVPR, (2005) 886–893
7. Diego, T., Spera, M., Cristani, M., Murino, V.: Characterizing humans on riemannian manifolds. IEEE Trans. Pattern Analysis Machine Intelligence, 35(8) (2013) 1972–1984
8. Dollar, P., Zitnick L. C.: Structured forests for fast edge detection. ICCV (2013)
9. Enzweiler M., Gavrila, D. M.: Integrated pedestrian classification and orientation estimation. CVPR, (2010) 982–989
10. Ess, A., Leibe, B., Gool, L.: Depth and appearance for mobile scene analysis, http://www.vision.ee.ethz.ch/ aess/iccv2007/ (2014)
11. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. CVPR, (2009) 1022–1029
12. Gandhi, T., Trivedi, M. M.: Image based estimation of pedestrian orientation for improving path prediction. IEEE IV, (2008) 506–511
13. Gavrila, D. M.: www.gavrila.net/Datasets,(2014)
14. Goto, K., Kidono, K., Kimura, Y., Naito, T.: Pedestrian detection and direction estimation by cascade detector with multi-classifiers utilizing feature interaction descriptor. IEEE IV, (2011) 224–229
15. Marin, J., Vazquez, D., Lopez, A. M., Amores, J., and Leibe, B. Random forests of local experts for pedestrian detection. ICCV, (2013) 2592–2599
16. Marin-Jimenez, M. J., Zisserman, A., Eichner, M., Ferrari, V.: Detecting people looking at each other in videos. Int. J. Computer Vision 106(3) (2014) 282–296.
17. Peter, K., Bulo, S. R., Bischof, H., Pelillo, M.: Structured class-labels in random forests for semantic image labelling. ICCV, (2011) 2190–2197
18. Piérard S., Droogenbroeck, M. Van.: Estimation of human orientation based on silhouettes and machine learning principles. *ICPRAM*, (2012)
19. Rosenhahn, B., Klette, R., Metaxas, D. (eds): Human Motion. Springer, Dordrecht (2008)
20. Samuel, S., Wohlhart, P., Leistner, C., Saffari, A., Roth, P. M., Bischof, H.: Alternating decision forests. CVPR, (2013) 508-515
21. Samuel, S., Leistner, C., Wohlhart, P., Roth, P. M., Bischof, H.: Alternating regression forests for object detection and pose estimation. ICCV, (2013) 417–424
22. Schneider, N., Gavrila, D. M.: Pedestrian path prediction with recursive bayesian filters: a comparative study. Proc. of the German Conference on Pattern Recognition (GCPR), 8142 (2013)
23. Shimizu, H., Poggio, T.: Direction estimation of pedestrian from multiple still images. IEEE IV, (2004) 596–600

24. Tao, J., Klette, R.: Integrated pedestrian and direction classification using a random decision forest. ICCV Workshop, (2013) 230–237
25. Yao, A., Gall, J., Gool, L. V.: A Hough transform-based voting framework for action recognition. CVPR, (2010) 2061–2068
26. Zhao, G., Takafumi, M., Shoji, K., Kenji, M.: Video based estimation of pedestrian walking direction for pedestrian protection system. J. Electronics (China), 29 (2012) 72–81