Multimedia Systems (2017) 23:515–526
DOI 10.1007/s00530-016-0518-5

REGULAR PAPER

Learning rich features from objectness estimation for human lying-pose detection

Dao-Xun Xia[1,2] · Song-Zhi Su[1] · Li-Chuan Geng[3] · Guo-Xi Wu[4] · Shao-Zi Li[1]

Received: 30 July 2015 / Accepted: 28 April 2016 / Published online: 17 May 2016

Abstract Lying-pose human detection is an active research field of computer vision in recent years. It has a good theoretical significance and furthermore many applications, such as victim detection or home service robot. But the study on lying-pose human detection in low-altitude overlooking images have many unsolved problems owing to multiple poses, arbitrary orientation, in-plane rotation, perspective distortion, and time-consuming. In this paper, the proposed framework of human lying-pose detection is optimization and machine learning algorithms inspired by processes of neurobiology suggest and human vision system to select possible object locations. First, the proposed model effectively utilizes binarized normed gradient features to obtain the objectness rapidly based on the vision saliency. Further, deep-learning techniques based on the convolution neural network are trained for learning rich feature hierarchies, in order to obtain the object of lying-pose human from objectness estimation, unlike the classical sliding-window algorithm. Eventually, employed pyramid mean-shift algorithm and rotation-angle recovery method to find position and direction of human lying-pose. The experimental results show that our method is rapid and efficient, and that it achieves state-of-the-art results with our XMULP dataset.

Keywords Human lying-pose detection · Deep model · Objectness estimation · Rich features learning · Saliency detection

Communicated by S. Kopf.

✉ Shao-Zi Li
szlig@xmu.edu.cn

Dao-Xun Xia
dxxia@gznu.edu.cn

[1] School of Information Science and Technology, Xiamen University, Xiamen 361005, Fujian, People's Republic of China

[2] School of Big Data and Computer Science, Guizhou Normal University, Guiyang 550001, Guizhou, People's Republic of China

[3] School of Urban Planning and Landscaping, Xuchang University, Xuchang 461000, Hebei, People's Republic of China

[4] The UAVLARS Collaborative Innovation Center, Xuchang University, Xuchang 461000, Hebei, People's Republic of China

1 Introduction

Recent years have witnessed rapidly increasing interest in object detection [1], a fundamental component in modern computer-vision systems. This interest is motivated by the importance of object detection in applications such as pedestrian detection [2–6], human lying-pose detection [7–9], fall detection [7, 8, 10], and object segmentation [11, 12]. But despite these successes, object detection is still an open problem for numerous applications. One unsolved issue is the detection of persons lying on the ground when pictured from a top–down perspective. Lying-pose detection is much more challenging than detecting pedestrians and human faces, mainly because of the large variations of pose and orientation a body can take when lying on the ground. In fact, detecting people lying on the ground is challenging enough to require a specific solution. Lying-pose detection is a glaring issue for numerous applications. One such application is fall detection for elders and disabled people living in smart homes [7, 13]. A 2012 report from the World Health Organization revealed that falls are the second leading cause of accidental-injury deaths worldwide and that every year, no less than 37

Table 1 Specific aspects of pedestrian detection, fall detection and lying-pose detection

	Datasets	Viewing angle	Pose	Object search space	Common models	Scenarios
Pedestrian detection	INRIA, Caltech, TUD, ETH	Front, back and side views	Up-right	Scale space	HOG + SVM, ISM, DPM, and many more [3, 5]	Indoor or outdoor
Fall detection	SDUFall, EDF, OCCU	One-two	Eight or different directions	Scale space	BoCSS, 3D-based	Indoor
Lying-pose detection	XMULP	All angles	Any orientation and pose	Scale-rotation space	None	Indoor or outdoor
Challenges with lying-pose detection	Very few datasets	Perspective distortion	Richer pose and in-plane rotation	More time-consuming	No code available	Diversification

million falls are severe enough to require medical attention [14–16]. As result, efficient visual fall detection algorithms are a key component to ensure the safety of elders and disabled staying at home [7]. Unfortunately, the experimental scenarios of fall detection focus on home or inside building, and these scenarios were simulated, such as SDUFall,[1] EDF[2] and OCCU.[3] This paper aims to establish a research framework for detection people lying on the ground based on the outdoor scenarios. Lying-pose detection can also be used in conjunction with unmanned aerial vehicles (UAVs) for rescue missions [9, 14, 17]. With a rising number of UAVs worldwide, the need for robust and rotation-invariant object detection methods is becoming an important issue.

So far, a limited number of papers focused on lying-pose detection, state-of-the-art human-shape detectors being more focused toward pedestrian detection. Of course, one can use a common sliding-window-based pedestrian detector [18] to detect people lying on the ground. However, such a method is likely to fail since people lying on the ground are rarely in upright position. Furthermore, depending on the camera standpoint, human shapes can suffer from severe perspective distortions. Lying-pose detection is more challenging than face and pedestrian detection, and the available study is still in its infancy [7, 9, 19]. The main differences between detecting pedestrians and people lying on the ground are listed in Table 1. Bodies lying on the ground have many unsolved problems owing to multiple poses, arbitrary orientation, in-plane rotation, perspective distortion, and time-consuming.

There is relatively little ongoing research in machine-vision approaches to human lying-pose detection, and there are various evaluation protocols and a lack of datasets. Of course, one can use any number of common object-detection frameworks to detect people lying on the ground, for which there are two key processes: (1) finding a descriptor to encode the object, such as a histogram of oriented gradient (HOG), a Fourier histogram of oriented gradient (F-HOG), a local binary pattern (LBP), or the aggregated channel features (ACF); and (2) choosing a localization strategy, such as a sliding window, jumping window, or a bound and branch search, to determine whether the object exists. However, these methods are prone to fragility and slow calculations. In particular, 120 s is needed for a max pooling strategy across all $K(K = 10)$ detectors [9]. The reason for this is that, unlike pedestrians, people lying on the ground are not in an upright position. Rather, they have an arbitrary orientation and body configuration. Furthermore, depending on the camera's standpoint, human shapes can have a severely distorted perspective.

In recent years, the accuracy rate of object detection has greatly improved as a result of simplifying and extending object-detection algorithms with deep learning. In 2012, Krizhevsky et al. [19] rekindled interest in convolutional neural networks (CNNs) by showing a substantial improvement in the accuracy of image classification for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012). Later, Girshick et al. [20] proposed a simple and scalable detection algorithm that improves the average precision (*AP*) by more than 30 % relative to the previous best result on PASCAL VOC 2012, achieving a *AP* of 53.3 %. This approach of region proposals with CNNs (R-CNN) [21] combines two key insights: bottom-up objectness and CNNs. It is worth noting that these models are very slow and they cannot achieve the speed necessary for real-time processing. For example, the R-CNN's processing speed is 53 s per image on a CPU [21]. Most models for object detection are subject to a tradeoff between accuracy and speed. On the one hand, when the accuracy

[1] http://www.sucro.org/homepage/wanghaibo/SDUFall.html.

[2] https://sites.google.com/site/kinectfalldetection/.

[3] http://sites.google.com/site/occlusiondataset.

rate improves, the speed is significantly reduced, as for instance, with Benenson et al. [2], who presented a pedestrian-detection method at 100 frames per second. On the other hand, when speed improves, the method tends to be less accurate.

In this paper, we provide a diffusion viewpoint to understand the neurobiological mechanism, and we investigate the physical nature of human lying-pose detection. Our first contribution is novel and reliable objectness estimation. Our work is motivated by the things-versus-stuff distinction [22–24]. The proposed estimation involves training a generic objectness measure to produce a small set of candidate object windows, unlike the classic sliding-window algorithm. Therefore, objectness is a property of stand-alone things with well-defined closed boundaries and centers. Our second contribution is a principled framework for salient objectness using CNNs as a global optimization problem. We focus on two problems related to localizing objects with a deep network and training a high-capacity model with only a small amount of annotated detection data. The first is to apply high-capacity convolutional neural networks to estimate objectness in order to localize and segment objects. The second is a paradigm for training large CNNs when labeled training data is scarce. Our third contribution is a rotation-angle recovery method based on the confidence map of an object window using principal component analysis (PCA) to obtain the orientation of the human in a lying pose. Figure 1 shows the strategy of our algorithm for human lying-pose detection. First, given sparse coding and the feedback mechanism from a biological visual model, the salient regions in an image are detected. Then, rich features can be learned with CNNs by combining the objectness with the search for the orientation of a human lying down. The red dashed rectangle in Fig. 1 indicates the salient region, and the yellow rectangle is the detection result from the deep-learning model on the right. Thus, we present a novel online and high-performance method for human lying-pose detection.

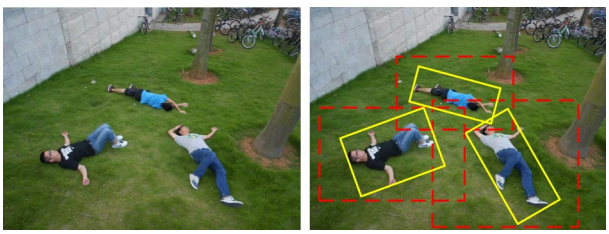

Fig. 1 Strategy for human lying-pose detection. The *red dashed rectangle* denotes the objectness, and the *yellow bounding-box* is the detection result from the deep-learning model on the *right*

This paper is organized as follows. In Sect. 2, we discuss the related works about pedestrian detection, human lying-pose detection and fall detection. In Sect. 3, we discuss detection methodologies in detail, including rapid estimation objectness, learning feature representations with CNNs, and the recovery of the rotation angle. In Sect. 4, we report the results from a performance evaluation on our XMULP (Xiamen University Lying-Pose) dataset. We conclude with a discussion of the state-of-the-art in lying-pose detection in Sect. 5.

2 Related works

As we have covered before, most papers published on the topic of human-shape detection focus on pedestrian detection. Furthermore, pedestrian detection methods can be divided into four categories: single-model detectors, part-based detectors, patch-based detectors and deep model detectors. For more complete pedestrian detection surveys, please refer to [3, 5]. Lying-pose human detection is a specific detection technology which is often regarded as a natural extension in pedestrian detection with advanced algorithm and ideology. Fortunately, Xia et al. [9, 25] proposed a multi-model classifier using pose clustering of lying-pose human with a 15-joint skeleton, and proposed rotation angle recovery for rotation-invariant detector. But, these methods have not been satisfactory for the accuracy of detectors.

2.1 Single-model detection methods

These methods treat each human shape as a whole (without considering body parts) and usually make the assumption that every human shape are in upright position with roughly the same pose. In general, single-model methods extract image features from a scanning window without seeking for body parts. Some methods use global features such as edge template, others use local features such as Haar-like features [18], histogram of oriented gradients [26], local binary patterns [27], and channel features [28] to name a few. More recently, some people used machine learning to learn optimal features to detect pedestrians. For example, as for Ren et al. [29], they use dictionaries of features learned through K-singular value decomposition (K-SVD) which they aggregate into so-called histograms of sparse codes (HSC). Dollar et al. [30] proposed a feature mining strategy to explore large feature spaces to train a boosted classifier. Costea et al. [31] present a pedestrian detection approach that uses the same classifier for all pedestrian scales based on image features computed for a single scale.

2.2 Part-based detection methods

These methods have been proposed to detect people whose body configuration is more complex than that of pedestrians. It typically models a person as a set of connected parts such as head, torso, legs, and arms. For example, Mohan et al. [32] trained four distinct part detectors to find heads, legs, left arms and right arms. The detectors scores are then fed to a classifier to ensure that the components have a correct anatomical configuration. Most part-based detectors need a training dataset with manually annotated body parts. Felzenszwalb et al. [33] proposed a different approach which accommodates with a weakly annotated training dataset, i.e., a dataset with only a rectangular window around each human body. For this method, the position and orientation of body parts are initially unknown, and thus treated as latent variables. These variables are learned and then used to detect humanoid shapes with an SVM framework. Recently, Yan et al. [34] proposed an accelerated version of Felzenszwalb et al's method. Using multi-modal and multi-channel Haar-like features, Zhang et al. [35] employ a statistical model of the up-right human body where the head, the upper body, and the lower body are treated as three distinct components.

2.3 Patch-based methods

They are the third family of human-shape detection methods. One typical patch-based detection method is the implicit shape model by Leibe et al. [36]. With this method, a codebook of local appearance is learned by clustering patches during the training phase. K-Means or a Hough Forest [37] is one of the typical representatives. During the detection phase, local features are matched to the codebook entries. Since human shapes are associated to features that match several codebook entries, human bodies are identified by cumulating match counts. In this way, sections of images with a large number of match counts are likely to contain a human shape.

2.4 Deep model methods

The deep learning has been applied to pedestrian detection and achieved promising results [20, 38–40] for the past few years. The deep model is especially appropriate for this task, because it can organize these components into different layers and jointly optimize them through back-propagation. A discriminative deep model is used by Ouyang et al. [38] for learning the visibility relationship among overlapping parts at multiple layers. It effectively estimates the visibility of parts at multiple layers and learns their relationship with the proposed discriminative deep model. This method was further expanded using joint deep learning [39], they should be jointly learned in order to maximize their strengths through cooperation. Later, Luo et al. [40] propose a Switchable Deep Network (SDN) for pedestrian detection. We should note that Girshick et al. [20] propose a simple and scalable detection algorithm. This algorithm combines objectness with convolutional neural networks and was renamed R-CNN. It improves average precision (AP) by more than 30 % relative to the previous best result on PASCAL VOC 2012 achieving a AP of 53.3 %.

In fact, several such methods focus on the more specific problem of fall detection. For example, Wang et al. [7] proposed a deformable part-based model for indoor applications. Amin et al. [41] describe the signal processing algorithms and techniques involved in elderly fall detection using radar. Zhang et al. [42] proposed a new privacy preserving automatic fall detection method to facilitate the independence of older adults living in the community, reduce risks, and enhance the quality of life at home activities of daily living (ADLs) by using RGBD cameras. Other research methods, such as based on human characteristic matrix and SVM [43], based on surveillance videos [44, 45], based on using k-nearest neighbor classifier [46] and other assistive environments [47–50].

Most ambient device-based approaches use pressure sensors for object detection and tracking [8]. It is very cost effective and less intrusive for the implementation of surveillance systems. However, it has a big disadvantage of sensing pressure of everything in and around the object and generating false alarms in the case of fall detection, which leads to a low detection accuracy. Most of the existing vision-based approaches lack flexibility (e.g., in home) and privacy issues (e.g., in a bathroom) [8, 10]. These approaches are often case specific and dependent on different scenarios. There is a need for a reliable and robust generic fall detection algorithm. Both ambience and sensor-based approaches share a common disadvantage, generally, of object data not being visually verified by the control or care service provider for accuracy.

As previously stated, human lying-pose detection is more challenging than basic human detection including pedestrian detection. Bodies lying on the ground may have multiple poses, arbitrary orientation, in-plane rotation, and perspective distortion. The importance of human lying-pose detection in applications such as fall detection for smart-homes [7, 8] and victim localization for search and rescue missions [14, 17, 51] must be supported scheduling with real-time capabilities for the human lying-pose detection and they need greater than 30 fps video feed obtained by a fixed camera. As a solution, some authors proposed a single-image lying-pose detection method. Andriluka

et al. [51] evaluated four state-of-the-art pedestrian detectors, i.e., HOG + SVM [26], deformable part model (DPM) [52], pictorial structure (PS) [53], and a poselet-based detector [54]. Wang et al. [7] proposed an extension of the deformable part model [52]. Additional robustness was achieved by combining viewpoint-specific foreground segmentation into the detection and body pose estimation stages. Although their system got promising results, detection result is never being satisfied with both on speed and on the accuracy. Thus a new on-line and high-performance detecting method for the human lying-pose detection is presented.

In our case, how to improve detection speed and accuracy of detector is the most urgent problem to solve. The effectiveness and robustness of algorithm detector were proposed, it based on the human reaction time that is observed and the biological signal transmission time that is estimated, human attention theories hypothesize that the human vision system processes only parts of an image in detail, while leaving others nearly unprocessed. There are simple mechanisms in the human vision system to select possible object locations. In order to reduce the number of windows each classifier needs to consider, training an objectness measure which is generic over categories has recently becomes important, in order to improve detecting speed, unlike the classical sliding-window algorithm. The saliency algorithms and deep-learning algorithms are fused to obtain a new detection algorithm framework of human lying-pose, in order to improve the speed and accuracy of it. We also use a Gibbs sampler to increase the number of training feature vectors.

3 Methodology

We employed two superior descriptor algorithms used for the study of novel detector framework of lying-pose human in this section, these are BING features and rich feature hierarchies. The proposed model effectively utilizes binarized normed gradients (BINGs) [24] to obtain the objectness rapidly based on the vision saliency of human. Meanwhile, the deep learning based on the convolution neural network is trained for rich feature hierarchies, in order to obtain the object of lying-pose human from objectness estimation that speed up the classical sliding window object detection paradigm.

We first introduce a variant of the BING feature for efficiently capturing the objectness of an image saliency window and its CNN features (Sect. 3.1). Further, we focus on combining a cascade support vector machine (SVM) classifier to generate the objectness with a deep model classifier [20] (see Sect. 3.2). Thus, we can exploit the unique benefits originating from the objectness using the cascade

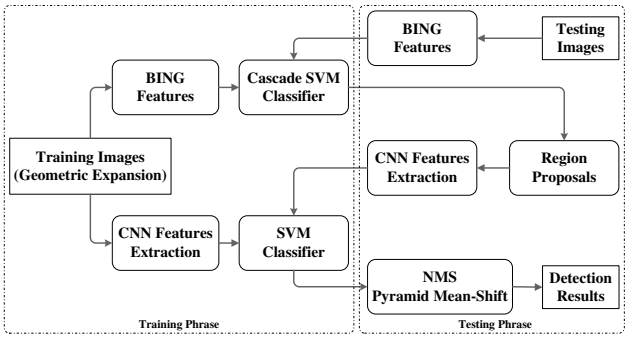

Fig. 2 Human lying-pose detection framework

SVM classifier, unlike the sliding-window method (see Sect. 3.3). Finally, we merge all the detected windows with the detection results using the pyramid mean-shift to obtain the object windows, and a rotation-angle recovery method based on the confidence map of the object windows using principal component analysis (PCA) for the orientation of the lying pose (see Sect. 3.4). Figure 2 shows the overview for the proposed human lying-pose detection system.

3.1 Feature construction

There are two types of features in the proposed system: BING features [24] for estimating the objectness rapidly, and CNN features to train the deep model. These features are motivated by saliency optimization and the R-CNN model [20]. In addition, they represent the fundamental features for building the classifier models.

Many researchers have proposed methods for generating category-independent objectness. Our algorithm is agnostic to any objectness method, however. We use an improved BING for a controlled comparison with prior detection work in [24]. BINGs are an accelerated version of normed gradients, designed to speed up the feature extraction and testing processes. In order to efficiently quantify the objectness of an image window, we resize it to 8×8, using the normed gradients as a simple 64D feature. Furthermore, we use the top normed-gradient binary bits from the BYTE values to approximate the normed gradient values. Thus, a 64D normed-gradient feature g_l can be approximated by the BING features as follows:

$$g_l = \sum_{k=1}^{N_g} 2^{8-k} b_{k,l}, \tag{1}$$

where l denotes the location, $l = (i, x, y)$, and i denotes the size. What is remarkable about these BING features is that they have different weights, depending on their corresponding bit position in the BYTE values. In the RGB

color space, we calculate the image gradients (g_x and g_y) using the $1-D$ mask $[-1, 0, 1]$ in both the horizontal and vertical directions. We use min $(|g_x| + |g_y|, 255)$ to calculate the normed gradients and save them in BYTE values. In the experiment, the POPCNT SSE instructions and the OPENMP options were enabled.

CNN features are important for the deep model. A total of eight layers were designed in our deep model. The convolution layers are the top five layers, while the remaining are fully connected layers. The first layer is a convolution layer (conv1), with the response-normalization layer following closely behind (viz., norm1). The second convolution layer (conv2) is similar in nature. After each convolution layer and fully connected layer, there is a loss function. In the experiment, the model uses the soft-plus function $f(x) = \log(1 + e^x)$ directly as its loss function. Max pooling is added after the top two layers of the response-normalization, and after the fifth convolution layer. The dropout algorithm is executed in the last two fully connected layers. Owing to the uncertainty of the angle direction in humans lying down, the samples of lying-posed humans (i.e., the annotation results from the training images) or objectness (i.e., from the salience detection by the cascade SVM classifier.) were resized to 250×250 resolution. Then, a 4096-dimensional feature vector was extracted from each sample of the lying-posed human or objectness using the CNN. Features are computed by forward propagating a mean-subtracted 250×250 RGB image through five convolution layers and three fully connected layers.

3.2 Model training

Next, we consider training models from images labeled with bounding boxes around humans in a lying pose. This is the type of data available in the XMULP dataset. The dataset contains thousands of images, and each image has annotations specifying a bounding box and a class label as the positive samples. Outside the bounding box of each image, there are random samples and a large number of negative samples. To process this dataset, we used two classification models: the cascade SVM classifier and the deep model [19, 21].

Cascade SVM classifier We train a detector merely to regain objectness, rather than modifying the distribution of negative windows for all sliding windows with which the classifier is trained. We adopted a strategy to conduct training in two stages with a cascade SVM [55] learning the objectness measure with image windows. During the first stage, the classifier learns a single model W for $s_l = \langle \mathbf{w}, g_l \rangle$ using a linear SVM, where $l = (i, x, y)$, s_l, g_l, l, i, and (x, y) denote the filter score, normed gradient feature, location, size, and position of a window, respectively. Here, g_l is derived from Eq. (1). The linear model $W \in R^{64}$ can be defined as follows:

$$\langle \mathbf{w}, b \rangle \approx \sum_{j=1}^{N_\omega} \beta_j \Big(2\langle a_j^+, b \rangle - |b| \Big), \tag{2}$$

where N_ω denotes the number of basis vectors, $a_j \in \{0, 1\}^{64}$ denotes a basis vector, $a_j = a_j^+ - \overline{a_j^+}$, and $a_j^+ \in \{0, 1\}^{64}$ denotes the corresponding coefficient. The filter score s_l for an image window can be tested efficiently as follows:

$$s_l \approx \sum_{j=1}^{N_\omega} \beta_j \sum_{k=1}^{N_g} 2^{8-k} \Big(2\langle a_j^+, b_{k,l} \rangle - \big| b_{k,l} \big| \Big), \tag{3}$$

where $b_{k,l}$ denotes the BING features. We use non-maximum suppression (NMS) to select a small set of objectness from each size i. Some sizes are less likely than others to contain an instance of an object. Thus we define the objectness score as $r_l = \upsilon_i \cdot s_l + t_i$, where $\upsilon_i, t_i \in R$ are separately learned coefficients with a bias term for each quantized size i.

During the second stage, the two sets of parameters υ_i, t_i must be learned using the linear SVM. We use Eq. (1) at size i for training images and the selected NMS objectness for training samples. Further, we use their filter scores as 1-D features, and check the labeling using the training-image annotations.

Deep model Here, we describe some of the unique features to our network architecture. Before the training phase, data processing is an important step with the deep model. Data processing involves expanding the training samples and enhancing the images. We employ two obvious forms of data expanding, and the first form of data augmentation consists of generating image translations and horizontal reflections. We do this by extracting random 224×224 patches from the 250×250 images and training our network with these extracted patches. This increases the size of our training set by a factor of 2048. The second form of data expanding consists of altering the intensities of the RGB channels using PCA in training images. PCA transformation is employed in order to alter the intensities of the RGB channels and remove noise in training images for the purpose of image enhancement. At the same time, a random scale factor is added in order to guarantee the diversity of images in the eigenvalue. This process is beneficial in terms of reducing the complexity of the network structure and preventing overfitting. The architecture of our network contains five convolutional and three fully connected layers. It is worth noting that local-response normalization is employed in the model. We found that following a local normalization scheme facilitated the generalization process when some training samples resulted in a positive input to a

rectified linear unit (ReLU) (i.e., learning will nevertheless occur in that neuron). Here, $a^i_{x,y}$ denotes the activity of a neuron computed by applying the kernel i at position (x, y) and then applying the ReLU nonlinearity. The response-normalized activity $b^i_{x,y}$ is given by this expression, and its computation formula is as follows:

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} \left(\alpha^j_{x,y}\right)^2 \right)^{\beta}, \tag{4}$$

where the sum runs over n adjacent kernel maps at the same spatial position, and N is the total number of kernels in the layer. The constants k, n, a, and β are hyper-parameters whose values are determined using a validation set. In the experiment, we used $k = 2$, $n = 5$, $a = 10^{-4}$, and $\beta = 0.75$. We executed this normalization process after applying the ReLU nonlinearity in certain layers. A dropout strategy is employed to improve the generalization ability of the system in the last two fully connected layers. To adapt our proposed CNN to this new task and domain, we continued a stochastic gradient descent (SGD) training of the CNN parameters using only warped objectness. The ultimate human lying-pose classifier is performed with a histogram intersection kernel SVM, rather than simply using the outputs from the final softmax layer of the fine-tuned CNN. In summary, we pretrained a large set of labeled images for the sake of image classification, and then fine-tuned it on a much smaller detection dataset for the sake of pose detection.

3.3 Estimating objectness based on saliency detection

Essentially, objectness detection methods are a classification process with multiple categories based on the cascade SVM classification. Estimating objectness is based on low-level image features, which are used to generate candidate windows, similar to detecting points of interest in order to mitigate computational expense when only particular points of interest are important. Given these low-level features, the method quickly decides whether a window should be considered for detection. To evaluate repeatability, we regain the objectness from one image in another slightly modified image. Among the traditional methods used for classifying specific features, cascade methods can also be applied to very large sets of categories (e.g., ImageNet). High levels of accuracy and fast calculations are especially important for detecting pedestrians and humans in a lying-down pose. Thus, BINGs [24], which require only a few atomic operations, are employed to estimate objectness effectively.

The BING method is the only method that is not based on segmentation. BINGs provide adequate approximations using a fast and class-agnostic detector. The test set consisted of 313 images with bounding-box annotations for the objects. A considerable number of humans are seen in a lying-down pose in this test set, along with the objects' position, occlusion, viewpoint, scale, illumination, and pose. This makes the dataset especially suitable for our evaluation, because we want to find all of the objects in the images. We captured the windows of each test image with 2^n $(n = 3, 4\ldots)$ in high and wide separately, and we resized the image windows to 8×8. The gradient (g_x and g_y) computation for each windows see Sect. 3.1, and the score for the objectness is defined using $o_l = \upsilon_i \cdot s_l + t_i$.

Note that only the scores are required when re-ranking a small set of final objectness, and this is needed in order to propose a pool of candidates for downstream processing—that is, for candidate ranking. However, each category is equally important in determining the objectness. Thus, a standard metric is needed to measure the coverage of ground-truth regions as a function of the number of objectness. The coverage is measured for K region candidates by

$$coverage(K) = \frac{1}{C} \sum_{i=1}^{C} \left(\frac{1}{N_i} \left(\sum_{j=1}^{N_i} \max_{k \in [1\ldots K]} O\left(R_k^{l(i,j)}, I_j^i\right) \right) \right), \tag{5}$$

where N_i denotes the number of instances of category i, C denotes the number of categories, $O(a, b)$ denotes the intersection of the union between regions a and b, I^i_j denotes the region corresponding to the jth instance of class i, $l(i, j)$ denotes the image that contains the jth instance of class i, and R^l_k denotes the kth ranked region in image l.

3.4 Detector design

We focus on computer-vision algorithms for detecting humans in a lying-down pose in individual monocular overlooking images. For the sake of simplicity, these are referred to simply as human lying-pose detectors. The unique benefits originating from the objectness found using the cascade SVM classifier differ significantly from the sliding-window method [9] and the R-CNN method [20], insofar as a selective search is run on each image in "val1", "val2", and "test". The detector algorithm begins with a set of bounding boxes for the objects in an image, using the cascade SVM classifier to extract the BING features. Rich features are learned with regard to the objects using a convolutional neural network, and classifies each object as either the target lying pose or not, using a linear SVM as shown in Algorithm 1.

Algorithm 1. Detector algorithm for human lying-pose detection.

Input:
(a) Image I
(b) Cascade SVM classifier C_c and deep model classifier C_d.
(c) Indexes of the active size N_s.
Output: Bounding boxes for object detection.

for $ir = N_s$ to 1
  $NewImg \leftarrow$ Resize I using Δs.
  $imgI \leftarrow GImg\,(NewImg)$ using min($|g_x| + |g_y|$,255).
  Initialize BING matrix: $B_{W\times H} = 0$ and $R_{W\times H} = 0$.
  for each position (x, y) in scan-line order **do**
    $R_{x,y} = (R_{x-1,y} \ll 1) \mid B_{x,y}$
    $B_{x,y} = (B_{x,y-1} \ll 8) \mid R_{x,y}$
  end for
  Matching score map: $S_m = matchTemplate\,(B_{W\times H})$
  Matching cost: $Set_{Ms} = nonMaxSup\,(S_m)$
  for $ib = 1$ to $Set_{Ms}.size()$
    $Set_{BBs} = Set_{BBs} \bigcup \langle x, y, w, h\rangle$
  end for
end for
for $ic = 1$ to $Set_{BBs}.size()$
  $\langle x, y, w, h, score^1\rangle \leftarrow$ apply C_c to $\langle x, y, w, h\rangle$
end for
$Reorder\,(Set_{BBs})$ by $score_1$
for $id = 1$ to $Set_{BBs}.size()$
  $ImgDeep \leftarrow$ Resize $Set_{BBs}\,[i]$ to $w_\theta \times h_\theta$.
  $D_F \leftarrow$ Translate $ImgDeep$ into CNN feature.
  $\langle x, y, w, h, score_1, score_2\rangle \leftarrow$ apply C_d to D_F
  $Set_{BBsLp} = Set_{BBsLp} \bigcup \langle x, y, w, h, score_1, score_2\rangle$
end for
Return results from the pyramid mean-shift applied to the detections and recovers the angle of rotation.

Let C_c denote the cascade SVM classifier and C_d denote the deep model classifier. The input image I is provided, along with its active size N_s (e.g., $N_s = 12$). The input image I is resized to 8×8 using scale Δs and the normed gradients are applied as a simple 64D feature to describe it. Further, this feature is encoded in BINGs because they require only a few atomic operations. The set of scored and ordered objects in an image is discovered using the functions *matchTemplate* and *nonMaxSup*, and the classifier C_c. Next, each object is translated into CNN features and classified. Ultimately, the algorithm returns the results of the pyramid mean-shift that was applied to the candidate bounding boxes.

The gradient is calculated by the function *GImg*. For an image of width W and height H, with a gradient magnitude map, the function *matchTemplate* is used to determine the $W - 7$ by $H - 7$ float-matching score map. Function *nonMaxSup* searches for points of interest from this matching score map S_m. In $\langle x, y, w, h, score_1, score_2\rangle$, x and y denote the position of the input image, w denotes the width of the object, and h denotes the height of the object. In $B_{W\times H}$ or $R_{W\times H}$, the values W and H are increased by 1 based on the columns and rows of the feature map *imgI*. This is done to expand the original size to avoid dealing with boundary conditions. Here, $w_\theta \times h_\theta$ denotes the size of input image for the deep model (e.g., 250×250).

4 Experimental results

In this section, we compare our detector to other state-of-the-art detectors using the XMULP dataset. Other state-of-the-art detectors include the HOG SVM [26], the deformable parts model (DPM) [52], the accelerated channel feature (ACF) detector [30], and the R-CNN [21]. We used the same experimental protocol as [3], and we evaluated the performance in terms of the false positives per image (FPPIs), the miss rate curves (MRCs), and the average precision (AP). Our detector was implemented on an *Intel Core*^{TM} *i*7 − 4770 *CPU* @3.40 GHz *RAM* @8 GB, and it was not implemented in GPU mode.

4.1 Dataset

The XMULP dataset is divided into two groups—one for training and one for testing. The training dataset contains 1003 images with between 1 and 7 humans per image, 1487 human bodies, and 3764 negative examples(image background and no people). The testing dataset contains 313 images and 532 human bodies. The dataset contains both indoor and outdoor(major part) images taken in buildings, in parking lots, on beaches, and in various grassland areas, and contains 30 degraded images and 21 occluded images. The cameras of UAVs or aerial cameras were positioned at a height of 2–20 m with different viewing angles and orientations, and obtained artificially static images with people lying on the ground from videos. As opposed to other datasets—such as INRIA and Caltech, which merely provide the bounding boxes—our dataset contains a 15-joint skeleton for each human body.

We employed a geometric expansion (GE) procedure using a virtual camera to increase the number of training images (see our previous work [9]). Our training method begins with a dataset for which each lying body has been cropped and manually outlined with a 15-joint skeleton. The first step in our method is to increase the number of training images (and skeletons). To do so, we simulated a synthetic camera that moves around each training image

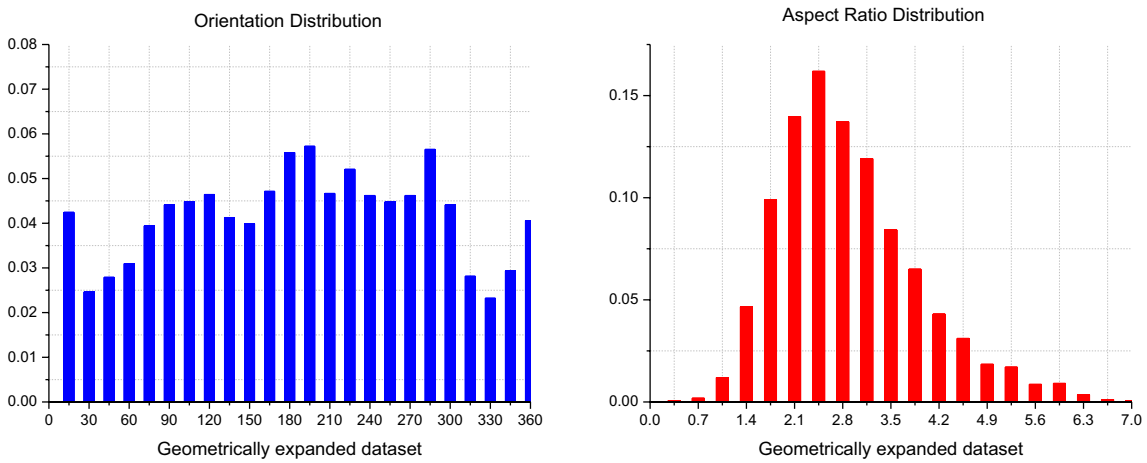

Fig. 3 Aspect ratio and angular distribution of the images of bodies in the XMULP dataset

(and skeleton) and generates new images showing a different perspective. Note that with geometric expansion, the number of training bodies increased to 23,792.

In order to illustrate the richness of our dataset, we plotted the orientation and aspect-ratio distribution for all of the human bodies. The bar plots in Fig. 3 represent the original dataset. As one can see, our dataset contains human shapes in all orientations, following a distribution that is almost uniform, whereas the aspect ratios follow a Gaussian distribution centered on 2.5. These facts speak volumes for the overall coherence of the dataset.

4.2 Implementation details

To optimize our detector, we analyzed the influence of different parameter settings. Here, we present our experimental results on the XMULP dataset.

Features During the saliency detection phase, we resized the image windows to 8×8 and used normed gradients as a simple $64D$ feature for learning the generic objectness measure. This was done in order to efficiently quantify the objectness of an image window. We further calculated the binarized version of the $64D$ feature—that is, we calculated the BING [24]. During the human lying-pose detection phase, we adopt our proposed CNN, which automatically learns hierarchical features, salience maps, and mixture representations of different body parts. We followed the CNN procedure from the Caffe CNN library [21]. This CNN contains eight learned layers: five convolutional layers and three fully connected layers. To configure the CNN with this structure, we extracted random patches that were $256 \times 3 \times 224 \times 224$ (number $\times$ K channel $\times$ height $\times$ width) from the $256 \times 3 \times 250 \times 250$ images, and trained our network with these extracted patches. The weight of the first convolutional layer was $96 \times 3 \times 11 \times 11$ (N output $\times$ K input $\times$ height $\times$ width) with a bias of $96 \times 1 \times 1 \times 1$. The fully connected layers contained 4096 neurons each. Specifically, the second layer (conv2) filters are shown in Fig. 4. In total, there were 256 filters, each with dimensions of $5 \times 5 \times 48$. In the figure, we depict only a few filters, with each channel shown separately, and where each filter is a row.

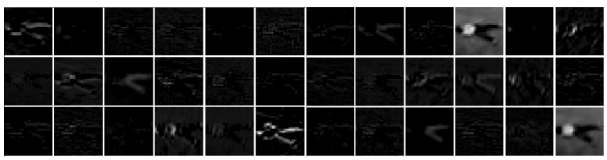

Fig. 4 Feature visualization of the second-layer (conv2) filters

Classifiers First, the generic objectness measure was learned in a cascaded SVM framework using a $64D$ feature. Their filter scores are $1D$ features, and the labels were verified using the training-image annotations. Second, the linear SVMs were trained for detection, and we computed features from the fully connected layer $7(fc7)$ [21]. We fixed the positive examples such that they were from the ground-truth boxes for the target class, and the negative examples were defined as boxes having less than a 0.3 intersection over the union with the ground-truth instances from that class. Training was done on the training set with SVM hyper-parameters $C = 0.002, B = 10, w1 = 2.1$ using liblinear.

Detection We obtained a 95.9 % object-detection rate (DR-#WIN) [24] with 1000 objects during the saliency detection phase, and resized each object to 250×250. We scored each extracted feature vector using the SVM trained for that object. Given all the scored bounding boxes in an image, we applied a greedy non-maximum suppression

(a pyramid mean-shift) that rejects a bounding box, provided that it has an intersection-over-union (IoU) overlap with a higher score for a selected object that is larger than a learned threshold. Then, our previously proposed rotation-angle recovery method based on a confidence map with PCA was used for the target bounding box (see our previous work [25]).

4.3 Results

We also entered our model in the XMULP competition. The results from our evaluation are shown in Fig. 5 and Table 2. Our method outperformed other state-of-the-art methods when BING features were used, and it was significantly faster. The plots in Fig. 5 show the FPPIs vs. MRCs for five human-shape detectors. For an FPPI of 10^{-1}, the MRC for the proposed BING–CNN was 34.9 %, which is 0.1 % lower than the MRC from the R-CNN. However, our proposed BING–CNN resulted in an AP that was 2.5 % higher than the AP of the R-CNN and significantly faster. The MRC was 11.9 % lower than that of the ACF, 12.3 % lower than that of the DPM, and 30.1 % lower than that of the HOG–SVM. The time consumed (in seconds) by each method is shown in the last column.

In summary, the results in Fig. 5 demonstrate the benefits of BING features coupled with the CNN model for human lying-pose detection. The results obtained with our proposed BING–CNN detector are shown in Fig. 6. The green bounding boxes show the results obtained with our entire framework. It is clear from the last row in this figure that our framework is not without limitations. Nevertheless, our framework's overall results are more accurate than without it.

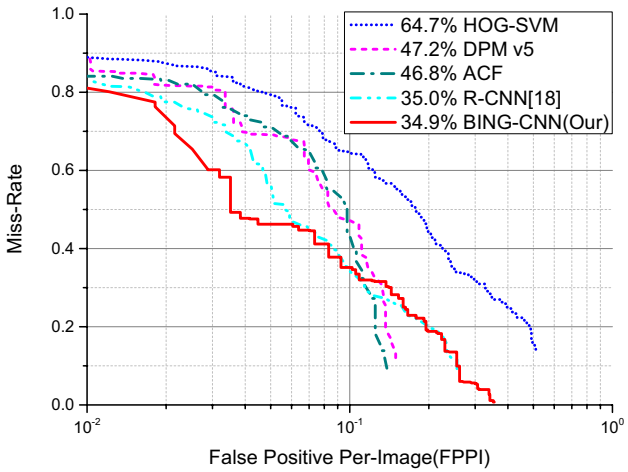

Fig. 5 FPPIs vs. MRCs with five human-shape detection methods

Table 2 Bounding-box detection AP (%) with the XiaMen University Lying-pose test set

No.	Methods	AP	Time consumed (spf)
1	HOG–SVM	32.5	79.56
2	DPM	39.6	2.87
3	ACF	40.2	0.08
4	R-CNN [20]	51.3	37.13
5	BING–CNN (proposed)	53.8	15.92

The time consumed [seconds per frame (spf)] is shown in the last column

5 Conclusion

In this paper, the proposed framework of human lying-pose detection is optimization, and machine learning algorithms inspired by processes of neurobiology suggestion and human vision system in order to select possible object locations. Two superior descriptor algorithms were employed for the study of novel detector framework of lying-pose human, these are binarized normed gradients and rich feature hierarchies. The BING features obtain the objectness rapidly based on the vision saliency of human. The convolution neural networks are trained for rich feature hierarchies in order to obtain the object of lying-pose human from objectness estimation, unlike the classical sliding-window algorithm. Finally, find position and direction of human lying-pose using pyramid mean-shift algorithm and rotation-angle recovery method (see our previous work [25]).

Our experimental results show that by coupling BING features with a deep CNN, a state-of-the-art performance can be achieved when detecting humans in a lying-down pose. We demonstrated that using an objectness framework for learning feature representations with CNNs can outperform other frameworks. Our proposed human lying-pose detection system achieved an average precision of 53.8 % and a miss rate of 34.9 % for an FPPI of 10^{-1}. It is also worth noting that the proposed BING–CNN is faster than the R-CNN. This is because our method focuses on the task of recovering the instance angle of the human in a lying pose.

In future work, we shall investigate whether the resulting algorithm can compute CNN features as accurately as previous methods, and at an order of magnitude faster. We plan to implement our algorithm on a GPU to generate CNN features in real-time. We also plan to explore how the insights of this work can be exploited by more general detection architectures, such as CNNs. We believe advances such as the ones represented in this paper will facilitate the use of perception in fields like robotics.

Fig. 6 Detection results with the proposed BING–CNN. False positives are shown in the *last row*

Acknowledgments This work was supported by the Nature Science Foundation of China (No. 61202143), the Collaborative Innovation Special Foundation of Xuchang University (No. XCUXT2014-08), and the Natural Science Foundation of Guizhou Province (No. QKHLHZi [2015] 7784).

References

1. Andreopoulos, A., John, K.T.: 50 years of object recognition: directions forward. Comput. Vis. Image Underst. **8**(117), 827–891 (2013)
2. Benenson, R., Mathias, M., Timofte, R., Gool, L.V.: Pedestrian detection at 100 frames per second. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2903–2910 (2012)
3. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2012)
4. Song-zhi, S., Shao-zi, L., Shu-yuan, C., Guo-rong, C., Yundong, W.: A survey on pedestrian detection. Dianzi Xuebao (Acta Electronica Sinica) **40**(4), 814–820 (2012)
5. Geronimo, D., Lopez, A.M.: Vision-Based Pedestrian Protection Systems for Intelligent Vehicles, Springer Briefs in Computer Science. Springer, Berlin (2014)
6. Wang, X.G., Wang, M., Li, W.: Scene-specific pedestrian detection for static video surveillance. IEEE Trans. Pattern Anal. Mach. Intell. **36**(2), 361–374 (2014)
7. Wang, S.M.: Lying pose recognition for elderly fall detection. Robot. Sci. Syst. VII **1**, 345–353 (2012)
8. Muhammad, M., Ling, S., Luke, S.: A survey on fall detection: principles and approaches. Neurocomputing **100**, 144–152 (2013)
9. Xia, D.X., Su, S.Z., Li, S.Z., Pierre-Marc, J.: Pose-specific lying human detection with samples expanding. In: IEEE International Conference on Image Processing, Oct. 2014
10. Zhang, Z., Conly, C., Athitsos, V.: A survey on vision-based fall detection. In: 8th ACM Int. Conf. on PErvasive Technologies Related to Assistive Environments, pp. 1–7, July 2015
11. Yang, D., Kriegman, M.H., Ahuja, N.: Detecting faces in images: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **24**(1), 34–58 (2002)
12. Fan, H. et al.: Learning deep face representation. In: CoRR (2014). arXiv:abs/1403.2802
13. Mirmahboub, B., Samavi, S., Karimi, N., Shirani, S.: Automatic monocular system for human fall detection based on variations in silhouette area. IEEE Trans. Biomed. Eng. **60**(2), 427–436 (2013)
14. Rudol, P., Doherty, P.: Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery. In: IEEE Aerospace Conference, pp. 1–8 (2008)
15. Diraco, G., Leone, A., Siciliano, P.: An active vision system for fall detection and posture recognition in elderly healthcare. In: Europe Conference and Exhibition on Design, Automation and Test, pp. 1536–1541, Mar. 2010
16. Yu, X.G.: Approaches and principles of fall detection for elderly and patient. In: 10th Int. Conf. on e-health Networking, Applications and Services, pp. 42–47, July 2008
17. Doherty, P., Rudol, P.: A UAV search and rescue scenario with human body detection and geolocalization. In: Advances in Artificial Intelligence, pp. 1–13. Springer, Berlin (2007)

18. Papageorgiou, C., Poggio, T.: A trainable system for object detection. Int. J. Comput. Vis. **38**(1), 15–33 (2000)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
20. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, June 2014
21. Jia, Y.Q., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia. ACM, pp. 675–678 (2014)
22. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: European Conference on Computer Vision, of Lecture Notes in Computer Science. Springer, Berlin, pp. 30–43 (2008)
23. Bogdan, A., Thomas, D., Vittorio, F.: Measuring the objectness of image windows. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2189–2202 (2012)
24. Cheng, M.M., Ziming, Z., Wen-Yan, L., Philip, T.: BING: binarized normed gradients for objectness estimation at 300fps. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3286–3293 (2014)
25. Xia D.X., Li, S.Z.: Rotation angle recovery for rotation invariant detector in lying pose human body detection. J. Eng. (2015). doi:10.1049/joe.2015.0032
26. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. Int. Conf. Comput. Vis. Pattern Recognit. **2**, 886–893 (2005)
27. Wang, X.Y., Han, T.X., Yan, S.C.: An hog-lbp human detector with partial occlusion handling. In: IEEE 12th Int. Conf. on, Computer Vision, pp. 32–39, Sept. 2009
28. Dollar, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: Proc. British Machine Vision Conference, pp. 1–11, Sept. 2009
29. Ren, X.F., Ramanan, D.: Histograms of sparse codes for object detection. In: IEEE Conf. Computer Vision and Pattern Recognition, pp. 3246–3253, June 2013
30. Dollar, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1532–1545 (2014)
31. Costea, A.D., Nedevschi, S.: Word channel based multiscale pedestrian detection without image resizing and using only one classifier. In: IEEE Conf. Computer Vision and Pattern Recognition, pp.2393–2400, June 2014
32. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE Trans. Pattern Anal. Mach. Intell. **23**(4), 349–361 (2001)
33. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Cascade object detection with deformable part models. In: IEEE Conf. Computer Vision and Pattern Recognition, pp. 2241–2248, June 2010
34. Yan, J.J., Lei, Z., Wen, L.Y., Li, S.Z.: The fastest deformable part model for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2497–2504, June 2014
35. Zhang, S., Bauckhage, C., Cremers, A.B.: Informed Haar-Like features improve pedestrian detection. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 947–954, June 2014
36. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. Int. J. Comput. Vis. **77**(1–3), 259–289 (2008)
37. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: IEEE Conf. Computer Vision and Pattern Recognition, pp. 1022–1029, June 2009
38. Ouyang, W.L., Wang, X.G.: A discriminative deep model for pedestrian detection with occlusion handling. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3258–3265, June 2012
39. Ouyang, W.L., Wang, X.G.: Joint deep learning for pedestrian detection. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 2056–2063, Dec. 2013
40. Luo, P., Tian, Y.L., Wang, X.G., Tang, X.O.: Switchable deep network for pedestrian detection. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 899–906, June 2014
41. Amin, M.G., Zhang, Y.D., Ahmad, F., Ho, K.C.D.: Radar signal processing for elderly fall detection: the future for in-home monitoring. IEEE Signal Process. Mag. **33**(2), 71–80 (2016)
42. Zhang, C.Y., Tian, Y.L., Capezuti, E.: Privacy preserving automatic fall detection for elderly using RGBD cameras. In: 13th International Conference on Computers Helping People with Special Needs, no. 9, pp. 625–633, July 2012
43. Wand, R.D., Zhang, Y.L., Dong, L.P., Lu, J.W,. Zhang, Z.Q., He, X.: Fall detection algorithm for the elderly based on human characteristic matrix and SVM. In: 15th Int. Conf. on Control, Automation and Systems, pp. 1190–1195, Oct. 2015
44. Wang, S., Xu, Z.W., Yang, Y., Li, X., Pang, C.Y., Alexander, G.: Fall detection in multi-camera surveillance videos: experimentations and observations. In: 1st ACM Int. Workshop on Multimedia Indexing and Information Retrieval for Healthcare, pp. 33–38, Oct. 2013
45. Feng, W.G., Liu, R., Zhu, M.: Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera, on Signal. Image Video Process. **8**, 1129–1138 (2014)
46. Liu, C.L., Lee, C.H., Lin, P.M.: A fall detection system using k-nearest neighbor classifier. Expert Syst. Appl. **37**(10), 7174–7181 (2010)
47. Tasoulis, S.K., Doukas, C.N., Plagianakos, V.P., Maglogiannis, I.: Statistical data mining of streaming motion data for activity and fall recognition in assistive environments. Neurocomputing **107**, 87–96 (2013)
48. Htike, Z.Z., Egerton, S., Chow, K.Y.: A monocular view-invariant fall detection system for the elderly in assisted home environments. In: 7th Int. Conf. on Intelligent Environments, pp. 40–46, July 2011
49. Yu, M., Yu, Y., Rhuma, A., Naqvi, S.M.R., Wang, L., Chambers, J.A.: An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. IEEE J. Biomed. Health Inform. **17**(6), 1002–1014 (2013)
50. Alazrai, R., Zmily, A., Mowafi, Y.: Fall detection for elderly using anatomical-plane-based representation. In: 36th Annual Int. Conf. of the IEEE on Engineering in Medicine and Biology Society, pp. 5916–5919, Aug. 2014
51. Andriluka, M., Schnitzspan, P., Meyer, J., Kohlbrecher, S., Petersen, K., Stryk, O.V., Roth, S., Schiele, B.: Vision based victim detection from unmanned aerial vehicles. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1740–1747 (2010)
52. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
53. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. Int. J. Comput. Vis. **61**(1), 55–79 (2005)
54. Bourdev, L., Malik, J.: Poselets: body part detectors trained using 3D human pose annotations. In: IEEE 12th Int. Conf. on Computer Vision, pp. 1365–1372, Sept. 2009
55. Zhang, Z.M., Warrell, J., Torr, P.H.S.: Proposal generation for object detection using cascaded ranking SVMs. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1497–1504, June 2011