IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 7, 2019, accepted December 21, 2019, date of publication December 24, 2019, date of current version January 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962100

Automatic Dataset Expansion With Structured Feature Learning for Human Lying Pose Detection

DAOXUN XIA[1,2], LINGJIN ZHAO[1], FANG GUO[1], AND XI CHEN[1]

[1]School of Big Data and Computer Science, Guizhou Normal University, Guiyang 550025, China
[2]Engineering Laboratory for Applied Technology of Big Data in Education, Guizhou Normal University, Guiyang 550025, China

Corresponding author: Daoxun Xia (dxxia@gznu.edu.cn)

This work was supported in part by the Nature Science Foundation of China under Grant 61762023 and Grant 61762022, in part by the Sprouts Come Special Project of Guizhou Department of Science and Technology under Grant QKHPTRC [2017]5726, and in part by the Startup Project of Doctoral Research of Guizhou Normal University (2017).

ABSTRACT In this study, we developed a framework to localize human lying poses by a camera positioned above. Our framework is motivated by the fact that detecting lying poses is fundamentally more difficult than detecting pedestrians or localizing nondeformable objects such as cars, roads, and buildings due to the large number of poses, orientations, and scales that a human lying on the ground can take. An important problem with lying pose detection is the training dataset, which hardly accounts for each possible body configuration. As a solution, we propose a geometric expansion procedure that uses a virtual camera to increase the number of training images. We also use a Gibbs sampler to generate more training samples in the feature space on which the system can train its model. Once the training is completed, detection is performed on a multiscale and multirotational space. Because our framework accommodates a variety of object detection systems, we report the results for the Faster R-CNN, FPN, and RefineDet models. The results show that using automatic dataset expansion models systematically improves the results.

INDEX TERMS Human lying pose detection, automatic dataset expansion, perspective transformation, gibbs sampling, deep learning.

I. INTRODUCTION

Object detection is a fundamental task in modern computer vision systems. Although this task has long been studied [1], [2], only recently have scientific breakthroughs combined with more processing power led to market-ready solutions [19]. Among the most frequently used object detection systems are those that focus on pedestrians [3], [20] and face detection [4], [24]. Despite these successful attempts, object detection remains an open problem for numerous applications. One unresolved issue is the detection of persons lying on the ground when viewed from a top-down perspective. Detecting lying poses is more challenging than detecting pedestrians and human faces because of the large variations in the pose and orientation that a body can take when lying on the ground [13]–[15]; thus, this task requires a specific solution.

Lying pose detection has important uses in numerous applications [14]. One such application is fall detection for elders and persons with disabilities living in smart homes [5], [7], [22]. A 2012 World Health Organization report revealed that falls are the second-leading cause of accidental-injury deaths worldwide and that every year, no fewer than 37 million falls are severe enough to require medical attention.[1] Consequently, efficient visual fall detection algorithms are key to ensuring the safety of elders and persons with disabilities staying at home. Lying pose detection can also be used in conjunction with unmanned aerial vehicles (UAVs) for rescue missions [8], [9], [25]. With an increasing number of UAVs worldwide, robust and rotation-invariant object detection methods are becoming increasingly important.

To date, a limited number of studies have focused on lying pose detection; state-of-the-art human-shape detectors have

The associate editor coordinating the review of this manuscript and approving it for publication was Chunbo Xiu.

[1]World Health Organization, www.who.int/mediacentre/factsheets/fs344/en/.

TABLE 1. Specific aspects of pedestrian detection and lying pose detection.

	Pedestrian detection	Human lying pose detection	Challenges with lying pose detection
Datasets	INRIA, Caltech, TUD, ETH	XMULP	very few datasets
Viewing angle	front, back and side views	all angles	perspective distortion
Pose	upright	any orientation and pose	richer pose and in-plane rotation
Target search space	scale space	scale-rotation space	more time consuming
Common models	DPM, R-CNN, FPN, RefineDet, and many more [2]	none	no code available

FIGURE 1. Left: pedestrians from the INRIA dataset. Right: people lying on the ground from the XMULP (XiaMen University lying pose) dataset. Lying pose detection is more challenging than detecting pedestrians due to the large number of other poses and orientations that a person lying on the ground may take.

focused more on pedestrian detection. One can use a common sliding-window-based pedestrian detector to detect people lying on the ground. However, such a method is likely to fail because people lying on the ground are rarely in an upright position. Furthermore, depending on the camera location, human shapes can suffer from severe perspective distortions, as shown in Fig. 1. Furthermore, no datasets that contain images of people lying on the ground have been released because people do not normally lay on the ground. The main differences between detecting pedestrians and people lying on the ground are listed in Table 1.

In this study, we aim to enhance the performance of human lying pose detection using pose clustering and dataset expansion. We also propose developing a new dataset called XMULP (XiaMen University lying pose dataset), as well as a framework to overcome the challenges associated with detecting people lying on the ground. Our dataset is fully annotated and contains a bounding box and a 15-joint skeleton for each person. The number of images in the dataset is increased by applying a series of perspective transformations. This approach simulates the effect of a camera moving around the persons lying on the ground. In this way, persons with similar poses are grouped together. Then, a D-dimensional feature vector is extracted from each body image of each class. Given these feature vectors, a new series of D-dimensional points are generated with a Gibbs sampler. Since these newly generated points have the same distribution as the original points, they can be considered new body poses, thus increasing the richness of the dataset. Then, human lying pose detectors are trained.

We test three state-of-the-art detectors, namely, the faster region-based convolutional neural network (Faster R-CNN) proposed by Ren *et al.* [10], the feature pyramid network (FPN) proposed by Lin *et al.* [11], and the refinement neural network (RefineDet) proposed by Zhang *et al.* [12]. These detectors are used to locate bodies lying on the ground following a sliding window strategy or feature learning on a multiscale and multirotational space.

This study provides the following contributions:

1) We propose a new and fully annotated dataset called *XMULP* that contains 1, 316 images of 2, 030 persons lying on the ground.
2) We propose an automatic data expansion procedure that increases the size of the training dataset. The experimental results reveal that methods trained on our extended dataset are up to two times more accurate.
3) We propose a structured feature learning method based on a human lying pose 15-joint skeleton such that the feature channels at a body joint can well receive information from other joints.
4) Unlike other methods designed for fall detection [5], [22], our approach works on a single image and does not need a video feed.

The remainder of this paper is organized as follows. Section II presents previous works related to lying pose human detection. Our framework is introduced in Section III, which includes the methods of geometric expansion, pose clustering, and feature space expansion with Gibbs sampling. Section IV presents the experimental results, and Section IV-C presents the conclusions.

II. RELATED WORKS

A. PEDESTRIAN DETECTION

Most studies on human-shape detection focus on pedestrian detection methods, which can be divided into three categories: single-model detectors, part-based detectors, patch-based detectors, and deep learning (for more complete pedestrian detection surveys, please refer to [3], [20], [21]).

Single-model detection methods treat each human shape as a whole without considering the body parts. These methods assume that the human shape is in an upright position with roughly the same pose. These methods typically extract image features from a scanning window without seeking body parts. Some methods use global features such as

edge templates, while others use local features such as Haar-like features, histograms of oriented gradients, channel features, and local binary patterns. Recently, some researchers have used machine learning to learn the optimal features for detecting pedestrians. Dollár *et al.* [27] proposed a feature mining strategy to explore a large feature space to train a boosted classifier. Sermanet *et al.* [28] used a convolutional neural network to learn pedestrian-specific multistage features. Ren *et al.* [29] used dictionaries of the features learned through K-SVD and aggregated them into so-called ''histograms of sparse codes (HSC)''.

Part-based detection methods have been proposed to detect people whose body configuration is more complex than that of pedestrians. Such methods typically model a person as a set of connected parts, such as legs, torso, arms, and head. Mohan et al. trained four distinct part detectors to locate the head, legs, left arm, and right arm. The detectors' scores were 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.* [30] proposed a different approach that accommodates a weakly annotated training dataset, *i.e.,* a dataset with only a rectangular window around each human body. In this method, the position and orientation of the body parts are initially unknown and thus treated as latent variables. These variables are learned and then used to detect the humanoid shapes with an SVM framework. Recently, Yan *et al.* [31] proposed an accelerated version of Felzenszwalb et al.'s method. Bar-Hillel *et al.* [32] proposed a scheme for synthesizing and combining a family of part-based features in an SVM framework. Such a scheme can process up to 10 fps when using kd-ferns [33]. Ghiasi et al. [46] described a hierarchical deformable part model for face detection and keypoint localization that explicitly accounts for occlusions. The proposed model structure enables augmenting positive training data with large numbers of synthetically occluded instances.

The third family of human-shape detection methods is patch-based methods. One typical patch-based detection method is the implicit shape model by Leibe et al. With this method, a codebook of local appearance is learned by clustering the patches (typically with k-means or a Hough forest) during the training phase. During the detection phase, local features are matched to the codebook entries. Since human shapes are associated with features that match several codebook entries, the human body is identified by accumulating the match counts. In this manner, sections of an image with a large number of match counts are likely to contain a human shape.

Deep learning has recently been used to detect pedestrians and has achieved promising results [2], [18], [42], [43]. A discriminative deep model was used by Ouyang *et al.* [42] for learning the visibility relationship among overlapping parts at multiple layers. This approach estimates the visibility of pedestrian parts at multiple layers and learns their relationship with a discriminative deep model. This method was further expanded using a so-called joint deep learning [24], [43], whose goal is to jointly learn the pedestrian parts to maximize their strengths through cooperation. Sermanet *et al.* [47] presented an integrated convolutional network (convnet) framework for classifying, localizing, and detecting human shapes with a novel approach for learning to predict the object boundaries. Xianjie *et al.* [45] presented a method for estimating the articulated human pose from a single static image based on a graphical model with pairwise relations that make adaptive use of local image measurements and use convnets to learn the conditional probabilities for the presence of parts and their spatial relationships. Ren *et al.* [10] merged RPN and Fast R-CNN into a single network by sharing their convolutional features using the popular approach of neural networks with ''attention'' mechanisms; the RPN component tells the unified network where to look.

B. LYING POSE DETECTION

Lying pose detection is more challenging than basic human (pedestrian) detection [13], [14]. Bodies lying on the ground may have arbitrary positions and configurations and may suffer from severe perspective distortions. Papers published in this area have mainly focused on two applications: fall detection [5]–[7], [15] and victim localization for rescue missions [25]. Mirmahboub *et al.* [22] proposed a low-cost and easy-to-implement video-based system for human fall detection. With a background subtraction method, they analyzed the temporal variations in a human silhouette and argued that a sudden increase in the size of the silhouette is a strong indication that the person just fell on the ground. However, they did not explain how their system accommodates multiple people and partial occlusions. Su *et al.* [5] proposed a method based on spatiotemporal interest points (STIPs) from multiple views. The number of local STIP clusters was designed to indicate the degree of the impact shock and body vibrations.

The main inconvenience with fall detection methods is their need for a 30-fps video feed from a fixed camera. These methods are thus inappropriate for videos with a very low frame rate and/or videos taken by a moving camera, such as on UAVs. As a solution, some authors have proposed single-image lying pose detection methods. Andriluka *et al.* [25] evaluated four state-of-the-art pedestrian detectors, *i.e.* , HOG+SVM, deformable part model (DPM) [26], pictorial structure (PS), and a poselet-based detector. These methods were tested in the context of UAV search and rescue missions. The evaluation results revealed that DPM is the top-performing detector and that combining visual detectors with contextual information such as the UAV height, orientation, and position helps improve the performance. Wang *et al.* [23] proposed an extension to DPM [26]. Additional robustness was achieved by combining a viewpoint-specific foreground segmentation into the detection and body pose estimation stages. Although their system showed promising results, their testing dataset was

rather limited, containing images taken by an indoor UAV flying at a height of 1.2 to 1.6 m looking down with a 15-deg angle. Another approach is to consider the fall detection problem as an activity recognition problem. From that perspective, Qian *et al.* [34] proposed a method that 1) detects moving blobs, 2) extracts local and global features from every moving blob, and 3) uses a multiclass SVM to recognize the activities of people, including persons falling on the ground.

C. DATASETS

Most of the datasets used in human-shape detection models focus on pedestrians. The INRIA human dataset contains 1, 208 images for training and 566 images for testing. The Caltech pedestrian dataset [21] contains 250, 000 images with 2, 300 unique pedestrians. The NICTA dataset contains 18, 700 training images and 6, 900 testing images. Since these datasets contain pedestrians, none can be used to train a lying pose detection system. One of the few datasets adapted for lying pose detection was proposed by Andriluka *et al.* [25]. However, this dataset contains only 220 indoor-images (all taken by a quadrotor UAV flying at a height of 1.5 to 2.5 m) and is far too limited to train a good detector. The multiple cameras fall dataset[2] contains 24 scenarios recorded with 8 IP video cameras. The first 22 scenarios contain fall and confounding events, and the last 2 scenarios contain only confounding events.

D. DATASET EXPANSION

Machine learning methods have long been facing the problem of limited training data. One solution is dataset expansion (also called *data augmentation*) to create fake data that can be used to train the model. Simply flipping, rotating, and cropping images can provide a positive impact for several applications [35]. As will be shown in Section III-B, our geometric expansion method is somewhat similar to that method but is adapted to the context of top-down lying pose detection.

Another increasingly popular solution to limited training datasets is transfer learning [36], which trains a model on another [yet richer] dataset and then refines the model on a smaller application-specific dataset. The first dataset can be a generic one such as ImageNet or an easy-to-generate synthetic dataset. Recent studies have shown that when synthetic images are sufficiently photorealistic, they can be used alone to train the model [37], [38].

Note that in our case, we also use a Gibbs sampler to increase the number of training feature vectors.

III. PROPOSED METHOD

A. OVERVIEW

The total number of body configurations is eminently large. Thus, building a complete training dataset that would include each body configuration is very challenging. Therefore, rather than constructing such a large dataset, we start from a smaller dataset (here, XMULP) and increase its size with two procedures, which are presented in Sections III-B and III-D. The first procedure, which we call geometric expansion, increases the number of images in the training dataset by simulating a camera moving around persons lying on the ground. The second procedure focuses on increasing the number of body poses with the help of a Gibbs sampler.

The training stage of our method implements the following four steps: (1) increase the number of images in the training dataset with a geometric expansion method; (2) structured feature learning based on their 15-joint skeleton; (3) increase the number of poses with Gibbs sampling; and (4) train human-shape classifiers on the newly expanded dataset. These four steps are illustrated in Fig. 2.

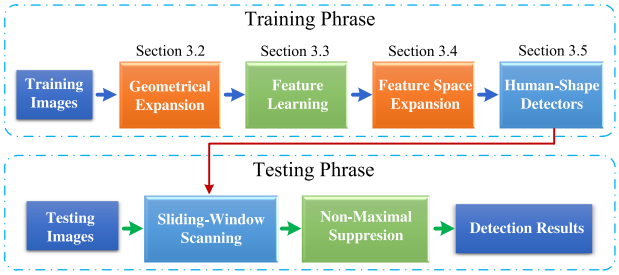

FIGURE 2. Overview of our proposed method.

In the testing stage, which is shown at the bottom of Fig. 2, a sliding window is used to localize the lying bodies. Since bodies have an arbitrary orientation and scale, we slide the window at different scales and different orientations of the image. We call this operation a rotation-scale space scanning procedure. At the end, all detected windows are merged with a pyramid mean-shift nonmaximum suppression.

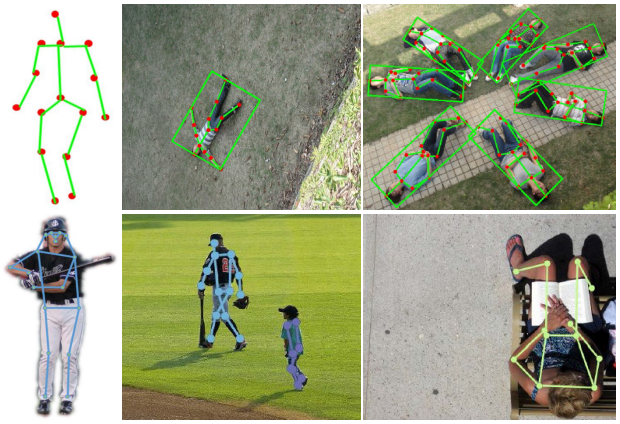

FIGURE 3. Fifteen-joint skeleton obtained after manually outlying each human body in our dataset (in the first row), and seventeen types of keypoints in MS COCO (person, in the second row), similarly.

B. GEOMETRIC EXPANSION

Our training method starts with a dataset in which each lying body has been cropped and manually outlined with a 15-joint skeleton (c.f., Fig. 3). The first step of our method

[2]http://www.iro.umontreal.ca/ labimage/Dataset/

increases the number of training images (and skeletons). For this purpose, we simulate a synthetic camera that moves around each training image (and skeleton) and generate new images showing different perspective transformations. New body configurations can also be obtained by tweaking the Kinect-like 3D skeletons [41]. The main reason we did not do so is the absence of affordable range scanners that can work on a 30 m range (a Kinect is not effective at more than 5 m).

This synthetic camera-reprojection procedure is inspired by Cai et al. [39]. According to their method, each training image is placed on the XY plane of a three-dimensional (3D) coordinate system (X, Y, Z). A synthetic camera is then positioned at (X', Y', Z') and oriented toward the origin of the world. With that configuration, each pixel of the image has a 3D position $w = (x, y, z)$ with $z = 0$ because the image is on the XY plane. Each pixel can then be reprojected onto the camera image plane following the projection equation.

$$p = K[R|t]w \tag{1}$$

where K, R, t are the intrinsic matrix, rotation matrix, and translation vector, respectively. Cai et al. [39] show that this projection procedure can be obtained with a warping homography matrix H between the pixels $[x, y, 1]^{\mathrm{T}}$ of the input image to those of the simulated image $[x', y', 1]^{\mathrm{T}}$, *i.e.,*

$$H = \begin{bmatrix} -f\cos\kappa & -f\sin\kappa & 0 \\ f\cos\varphi\sin\kappa & -f\cos\varphi\cos\kappa & 0 \\ \sin\varphi\sin\kappa & -\sin\varphi\cos\kappa & -r \end{bmatrix} \tag{2}$$

where r is the distance between the optical center and the reference image plane center, f is the camera focal length, and (φ, κ) are the yaw and pitch angles of the camera, respectively.

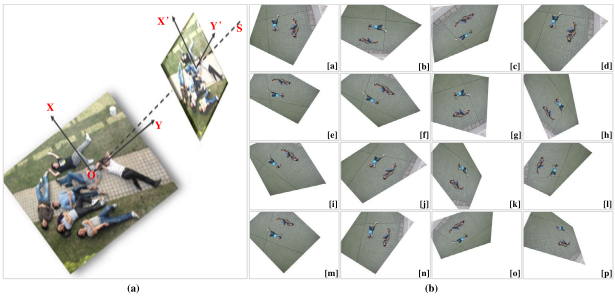

FIGURE 4. Perspective camera model. (a) The reference image on the reference image plane is projected onto the simulated image plane of camera S. (b) Resulting images from 16 different perspective transformations.

Using different combinations of φ and κ, we obtain various viewpoints of the input images. Fig. 4(b) illustrates our geometric expansion procedure. Given an input image as in Fig. 4 (a), the geometric expansion procedure generates a series of perspectively warped images, as shown in (b).

In our case, we consider 16 different camera positions. We multiply the number of images (and skeletons) in the dataset by 16. Details of the geometric expansion procedure are presented in Algorithm 1.

Algorithm 1 Geometric Expansion

Input: image I
r: distance between the optical center and the origin of the world
f: the camera focal length
$\Delta\varphi$, $\Delta\kappa$ the step size of the camera angles.
Output: $I_1, I_2, \ldots, I_m$

for $i = 1$ to m
  for $\kappa = 0, \kappa < 2\pi, \kappa = \kappa + \Delta\kappa$
    for $\varphi = 0, \varphi < \pi/2, \varphi = \varphi + \Delta\varphi$
      Compute homography matrix H according to Eq. (2)
      for e ach pixel (x, y) in I
        compute (x', y') as follows:

$$\begin{cases} x' = \frac{(-f\cos\kappa)x - (f\sin\kappa)y}{(\sin\varphi\sin\kappa)x - (\sin\varphi\cos\kappa)y - r} \\ y' = \frac{(f\cos\varphi\sin\kappa)x - (f\cos\varphi\cos\kappa)y}{(\sin\varphi\sin\kappa)x - (\sin\varphi\cos\kappa)y - r} \end{cases}$$

        $I_i(x', y') = I(x, y)$;
      end for
    end for
  end for
  i++;
end for

Note that Algorithm 1 samples φ and κ but not r and f. Sampling r and f would lead to the same images but with a different scaling factor, which in our case would be redundant with respect to the upcoming human shape detection method (Algorithm 2), which already accounts for multiple scales.

C. STRUCTURED FEATURE LEARNING

In feature expansion, the effective extraction of human contours is a prerequisite for subsequent processing, including feature extraction, feature expression, object detection and object recognition. The structured feature learning framework is used to calculate the correlations among body joints at the feature level [16]–[18], and helps maintain the integrity of these correlations throughout the human body. The correlations among the feature maps of the body joints are modeled for human lying pose detection, unlike the existing approaches of modeling the structures on score maps or predicted labels. Feature-level information passing delivers more detailed descriptions of body joints than score maps. The relationships between the feature maps of joints can easily be detected with a convolution layer and geometrical transform kernels. We proposed a bidirectional tree-structured model for feature channels at a body joint that can well receive information from other joints.

Hierarchical feature representations of the input images are learned by convnets with multiple layers. Features capture low-level information in lower layers. More abstract concepts can be represented using high-level information. In this work, we use the fully convolutional faster R-CNN [10], FPN [11],

Algorithm 2 Pose-Specific Detectors for Lying Pose Detection

Input:
(a) input image I
(b) The classifier C.
(c) Scale step Δs and rotation step $\Delta\theta$.
Output: Bounding boxes of object detections.

detections $= \varnothing$
foreach classifier C.
for $s = 0.7$ to 1.1, $s = s \times \Delta s$
for $\theta = 0$ to 360, $\theta = \theta + \Delta\theta$.
$I_{s\theta} \leftarrow$ rescale and rotate image I
$\langle x, y, s, \theta, score\rangle \leftarrow$apply C_k to $I_{s\theta}$.
detections = detections $\cup\langle x, y, s, \theta, score\rangle$
end
end
end
return results of pyramid mean shift applied on "detections".

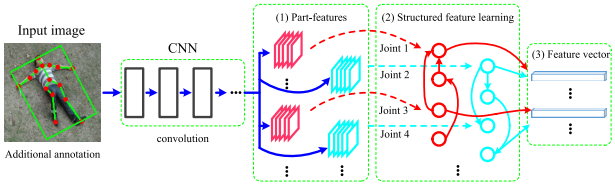

FIGURE 5. Structured feature learning based on human lying pose 15-joint skeleton.

and RefineDet [12] as the base models and extract feature maps in the fully convolutional neural networks (fcn7) that replaced the fully connected (fc) layers. All the joints share lower layers up to the fcn6 layer, and each body joint has a separate set of 128 feature maps. Suppose that symbol $h_{fcn6}(x, y)$ represents the feature vector obtained at location (x, y) in the fcn6 layer, which has a 4096-dimensional feature vector. Body joint k at (x, y) in the fcn7 layer is computed as

$$h^k_{fcn7}(x, y) = f(h_{fcn6}(x, y) \otimes w^k_{fcn7} + b_{fcn6}) \tag{3}$$

where h^k_{fcn7} is the feature tensor containing 128 feature maps for joint k; f is a nonlinear function; $\otimes$ denotes convolution; w^k_{fcn7} is the filter bank for joint k, which includes 128 filters; and b_{fcn6} is the bias.

In this model, the spatial distribution and semantic meaning of the feature maps obtained at different joints can effectively improve the features learned at each joint. The rich information and detailed descriptions of the human lying pose are contained within the feature maps of the body joints and show that under a fully convolutional neural network, messages can be passed between the feature maps through the introduced geometrical transform kernels and combine the features at multiple scales. Feature channels for each joint have different semantic meanings, and the relationships between the feature maps of the neighboring joints are part-specific. In this section, the model ultimately needs to capture the feature vectors.

D. FEATURE EXPANSION

After K different poses are identified, a D-dimensional feature vector $\vec{x}_i \in \boldsymbol{R}^D$ is assigned to each training image i. This leads to a collection of N training samples $X = \{\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_N\}$. Depending on the human-shape detection method, different types of features, such as HOG, Haar-based features, or features learned with a deep learning method, can be used.

Regardless of the type of feature vector used, the goal of the current step is to increase the number of training samples associated with each body pose recovered in the previous step. One simplistic way of doing so is by adding random noise to each training feature, $\vec{x}'_i \leftarrow \vec{x}_i + \vec{\epsilon}$ ($\vec{\epsilon}$ being a random vector), and consider $\vec{x}'_i$ to be a new training sample. However, there is no easy way to determine the correct magnitude of $\vec{\epsilon}$; a magnitude that is too large would lead to a noisy dataset, whereas a magnitude that is too low would lead to samples aggregated around the original training features.

A better way to increase the amount of training data is by sampling the data distribution $P(\vec{x})$. Given training set X whose elements are *iid* from $P(\vec{x})$, the goal is to generate a new set of points X' such that the distribution $P(\vec{x}')$ of the newly generated samples is close to $P(\vec{x})$. This task can be performed with rejection sampling [25]. Rejection sampling generates a series of samples *iid* of a pdf $P(\vec{x})$ given a second pdf $Q(\vec{x})$ that is easier to sample (e.g., a uniform distribution). Unfortunately, since samples are randomly sampled from $Q(\vec{x})$ and not from $P(\vec{x})$, rejection sampling is known to be prohibitively slow in a high-dimensional space.

To our knowledge, the best way to sample a high-dimensional distribution $P(\vec{x})$ is via the Markov Chain Monte Carlo (MCMC) method. The MCMC method performs a random walk in the space of $\vec{x}$ such that the fraction of samples $\vec{x}^{[0]}, \vec{x}^{[1]}, \ldots, \vec{x}^{[n]}$ generated from the random walk within an area Ω is always proportional to $\int_\Omega P(\vec{x})$.

A random walk is defined as

$$\vec{x}^{[i+1]} = \vec{x}^{[i]} + \vec{\epsilon} \tag{4}$$

where $\vec{\epsilon}$ is a random vector sampled from a kernel distribution. In our case, $\vec{\epsilon}$ is sampled from a zero-centered Gaussian distribution, and $\vec{x}^{[0]}$ is a vector taken from the training dataset X. In this way, $\vec{x}^{[i+1]}$ can be considered a random variable sampled from a *transition* probability $q(\vec{x}^{[i+1]}|\vec{x}^{[i]})$, which in our case is a Gaussian distribution centered at $\vec{x}^{[i]}$. If we choose a transition probability of the form $q(\vec{x}^{i+1}|\vec{x}^{i}) = q(\vec{x}^{i+1})$, each new sample will be independent of the previous one, leading to a method that is similar to rejection sampling.

To force the successive samples to follow the $P(\vec{x})$ distribution, we need to accept (or reject) each newly generated sample $x^{[i+1]}$ according to some criteria. These criteria must ensure that the fraction of the time spent in some area Ω

is proportional to $\int_{\Omega} P(\vec{x})$, the density of $P(\vec{x})$ in that area. According to the Metropolis-Hastings algorithm, the acceptance probability of $\vec{x}^{[i+1]}$ is given by

$$\alpha(\vec{x}^{[i+1]}|\vec{x}^{[i]}) = \min\left(1, \frac{p(\vec{x}^{[i+1]})q(\vec{x}^{[i]}|\vec{x}^{[i+1]})}{p(\vec{x}^{[i]})q(\vec{x}^{[i+1]}|\vec{x}^{[i]})}\right) \quad (5)$$

where $q(a, b)$ is a Gaussian distribution centered at b and evaluated at position a in our case. Since $P(\vec{x})$ is unknown in our case, we approximate it with a Parzen window distribution.

$$P(\vec{x}) \approx \frac{1}{N}\sum_{\vec{x}_i \in X} \mathcal{N}(\vec{x}, \vec{x}_i, \Sigma) \quad (6)$$

where $\mathcal{N}(.)$ is a Gaussian distribution centered at $\vec{x}_i$ with variance Σ. The experimental results reveal that a simple identity variance-covariance matrix Σ works well.

Since the dimensionality of the feature samples $\vec{x}$ can be very large, we implemented a Gibbs sampler, which is a special case of the Metropolis-Hastings algorithm. Rather than generating a new sample $\vec{x}^{[i+1]}$ at once, each dimension of $\vec{x}^{[i+1]}$ is generated in turn. This leads to the following transition probability:

$$q(\vec{x}^{[i+1]}|\vec{x}^{[i]}) = \prod_{j}^{D} P(\vec{x}_{(j)}^{[i+1]}|\vec{x}_{(1)}^{[i+1]}, \ldots, \vec{x}_{(j-1)}^{[i+1]}, \vec{x}_{(j+1)}^{[i]}, \vec{x}_{(D)}^{[i]})$$

where D is the total number of dimensions and $\vec{x}_{(j)}^{[i]}$ is the value at the j^{th} dimension of vector $\vec{x}^{[i]}$.

As mentioned in [49], it is often necessary to discard the first T samples until the Markov chain has *burned in* or has entered a stationary distribution. In our case, we discarded the first $T = 1,500$ samples and then retained the subsequent 4 samples, *i.e.,* $x^{[T+1]}, x^{[T+2]}, x^{[T+3]}, x^{[T+4]}$ (the value of $1,500$ was selected empirically). Since the Gibbs sampler was launched on each training sample of X, we multiplied the total number of training samples by 5.

Finding the best values of hyperparameter Σ is challenging considering the dimensionality of the data. If the dimensionality had been low, we could have used a diagonal matrix whose values could be estimated with a grid search algorithm. However, it is widely accepted that the grid search processing cost increases exponentially with the number of dimensions. Consequently, we had to simplify the problem by assuming that the data follow an isotropic Gaussian distribution. We then tested a few variance values and found that a variance of 1 (the identity matrix) works well in our case because the average variance along each feature dimension is 1.09. Since Σ is used within a Parzen window PDF estimator, changing its value does not significantly affect the results.

E. HUMAN LYING POSE DETECTION

The last step is to train the classifiers, one for each body pose recovered at step 2. The negative examples used for training consist of non-human images. In this study, we tested four state-of-the-art detectors: faster R-CNN [10], FPN [11], and RefineDet [12]. A brief overview of these detectors is presented below.

Faster R-CNN: is composed of two modules: a deep fully convolutional network and a fast R-CNN detector. The deep fully convolutional network proposes regions, and the fast R-CNN detector uses the proposed regions. The model further merges RPN and fast R-CNN into a single network by sharing their convolutional features. The entire system of the group of items forms a single, unified network for object detection.

FPN (Feature Pyramid Network): exploits the inherent multiscale, pyramidal hierarchy of deep convolutional networks such that the marginal extra cost constructs feature pyramids. Feature pyramids are a basic component in recognition systems for detecting objects at different scales. A top-down architecture with lateral connections is developed for building high-level semantic feature maps at all scales. It shows significant improvement as a generic feature extractor in several applications.

RefineDet: consists of two interconnected modules, an anchor refinement module and an object detection module, and it is a single-shot refinement neural-network-based detector. The entire network is trained in an end-to-end fashion with multitask loss and introduces the attention mechanism in RefineDet to further improve the performance.

Our detection phase implements a scanning window procedure. The three major challenges of human lying pose detection are scale, in-plane rotation, and pose. The challenge in pose estimation is to perform the estimation by using structured feature learning, one for each pose. The scale and orientation challenges are overcome by scanning the image at different scales and different orientations.

Let C be the classifier, I be the input image, and $I_{s\theta}$ be the image at scale s and rotation angle θ. As shown in Algorithm 2, classifier C scans the $I_{s\theta}$ images with a sliding window. Once each classifier has scanned each image $I_{s\theta}$, the windows activated by the classifier are fused to obtain the final detection results.

In our experiments, the rotation step was set to $\Delta\theta = 20$ deg and the scale step to $\Delta s = 1.05$. The total number of scales is thus $log(1.1/0.7)/log(1.05) = 10$. The results returned by each detector are recorded as follows:

$$\text{detections} = \{\langle x, y, s, \theta, score\rangle\}_{i=1}^{n} \quad (7)$$

where (x, y) is the center of the sliding window, (s, θ) are the scale and orientation of the current image $I_{s,\theta}$, and *score* is the classifier output. This output is similar to that of most sliding-window-based detectors, except rotation. We use the mean shift on the (x, y, s, θ) results to merge the overlapping windows.

IV. EXPERIMENTAL RESULTS

In this section, we compare our running time with those of the baseline methods on the same laptop with an Intel CoreTM i7-4770 CPU @8 GB. We also compare our detector with

other state-of-the-art detectors using the XMULP dataset. The sizes of the images are 560×420 and 612×405, and the size of the sample is 80×160. Other state-of-the-art detectors include Faster R-CNN [10], FPN [11], and RefineDet [12].

A. DATASET

Our dataset includes various challenging scenarios in indoor and outdoor environments, parking lots, beaches, and various grassland areas. The cameras were positioned at a height of 2 to 20 m with different viewing angles and orientations. Nearly 30 volunteers participated in the process of building the dataset. Overall, the dataset contains 1, 316 images with 1 to 7 persons per image for a total of 2, 030 human bodies. In contrast to other datasets such as INRIA and Caltech [21], which only provide bounding boxes, our dataset contains a 15-joint skeleton for each human body (c.f., Fig. 3).

TABLE 2. Details of the Xiamen University lying pose human dataset.

	Training Dataset		Testing Dataset	
	Positive	Negative	Positive	Negative
Images	1,003	3,764	313	-
Human bodies	1,498	-	532	-

To evaluate the performance of the models, the dataset was divided into two groups: training and testing. As shown in Table 2, the training dataset contains 1, 003 images, 1, 498 human bodies, and 3, 764 negative examples (images without a human body), while the testing dataset contains 313 images and 532 human bodies. Note that with geometric expansion, the number of training human bodies increases to 23, 792. With the Gibbs samples, the detection methods train on a total of 118, 960 feature points.

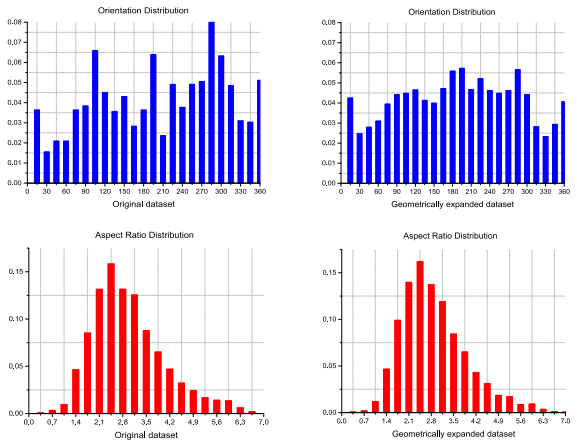

FIGURE 6. The aspect ratio and human body angular distribution of the dataset before (on the left) and after (on the right) geometric expansion.

To illustrate the richness of our dataset, we plotted the orientation and aspect ratio distributions of all human bodies. The bar plots on the left side of Fig. 6 show the original dataset, while those on the right side show the geometrically expanded dataset. Our dataset contains human shapes with all orientations (the distribution being almost uniform), while the aspect ratios follow a Gaussian distribution centered at 2.5. The similarity between the left and right plots underlines the effectiveness of our geometric expansion procedure, which preserves the overall coherence of the dataset. To encourage future work, our dataset is freely available on the web at https://bge.gznu.edu.cn/info/1031/1099.htm.

B. PERFORMANCE EVALUATION

Plots of the false positive per image (FPPI) versus the miss rate are commonly used to evaluate human shape detectors. The curve is obtained by increasing the detection threshold. A detected window is considered a true detection if its bounding box BB_{dt} has a significant overlap with the ground-truth bounding box BB_{gt}:

$$\alpha_0 = \frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} \geq thr. \tag{8}$$

In our case, we use $thr = 0.5$.

The precision and recall curves are also extensively used to gauge the performance of object recognition or detection methods. The average precision (mAP) is the area under the precision and recall curves, namely, $AP = \int_0^1 p(r)dr$, where $p(r)$ is precision p as a function of recall r. Average precision computes the average value of $p(r)$ over the interval from $r = 0$ to $r = 1$. We chose to interpolate the $p(r)$ function to reduce the impact of the "wiggles" in the curve. Just as the PASCAL Visual Object Classes (VOC) and Microsoft COCO[3] challenges, we averaged the precision over a set of evenly spaced recall levels $\{0, 0.1, 0.2, \ldots 1.0\}$.

$$AP = \sum_{r \in \{0, 0.1, \ldots, 1.0\}} p_{interp}(r)\Delta r \tag{9}$$

where $\Delta r = \frac{1}{11}$ and $p_{\text{interp}}(r)$ is an interpolated precision that takes the maximum precision over all recalls greater than r and $p_{\text{interp}}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r})$. An alternative is to derive an analytical $p(r)$ function by assuming a particular parametric distribution for the underlying decision values.

C. RESULTS

Here, we present the experimental results for the four human-shape detection methods: Faster R-CNN, Mask R-CNN, FPN, and RefineDet512+. We tested each method independently without our framework, with geometric expansion (GE), with structured feature learning (SFL), with feature space expansion (FE), and with GE + SFL, and GE + SFL + FE detectors (*i.e.,* our entire framework is shown in Fig. 2). The results are shown in Table3 and Fig. 7.

Table 3 shows that the mean average precision (mAP) ranges from 67.6% for faster R-CNN to 68.5% for faster R-CNN + GE, 70.2% for faster R-CNN + GE + SFL and 71.5% for the full-blown method, and Mask R-CNN has a similar trend. FPN + GE + SFL + FE achieves the

[3]http://cocodataset.org

TABLE 3. Bounding box detection average precision (%) on the XiaMen University Human lying pose test set and MS COCO(person).

Method(mAP@IoU=0.5)	XMULP	MS COCO(person)
Faster R-CNN	67.6	52.3 [51]
Faster R-CNN + GE	68.5	53.2
Faster R-CNN + GE + SFL	70.2	54.8
Faster R-CNN + GE + SFL + FE	**71.5**	**55.8**
Mask R-CNN	69.7	54.9 [2]
Mask R-CNN + GE	70.5	55.6
Mask R-CNN + GE + SFL	71.9	56.8
Mask R-CNN + GE + SFL + FE	**73.0**	**57.6**
FPN	71.1	61.0 [51]
FPN + GE	72.3	61.9
FPN + GE + SFL	74.5	63.5
FPN + GE + SFL + FE	**76.1**	**64.4**
RefineDet512+	74.8	63.2 [52]
RefineDet512+ + GE	75.7	63.8
RefineDet512+ + GE + SFL	77.2	65.3
RefineDet512+ + GE + SFL + FE	**79.1**	**66.7**

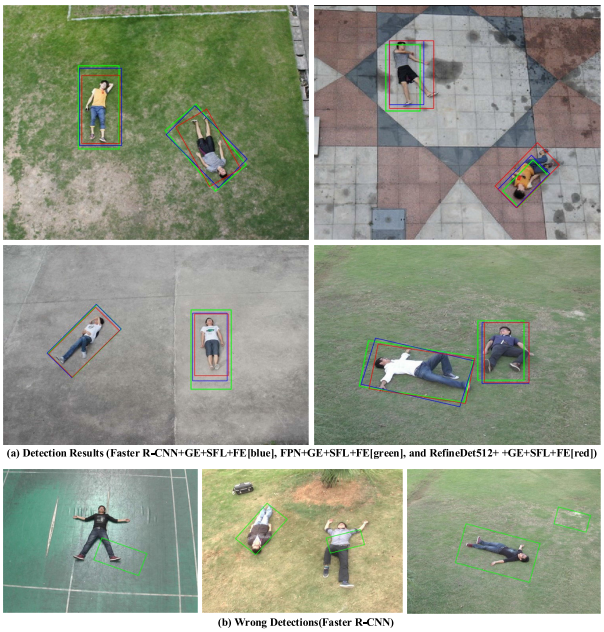

FIGURE 7. Detection results of Faster R-CNN, FPN, and RefineDet512+. False positives of the Faster R-CNN model are shown in the last row.

highest mAP, 76.1%, while FPN, FPN + GE and FPN + GE + SFL have mAPs of 71.1%, 72.3% and 74.5%, respectively (Table3). Our expansion framework also reduces the error rate for RefineDet512+. The mAP of RefineDet512+ GE + SFL + FE is 79.1%, which is the best result (Table3). Note that dataset expansion and feature learning improved the results; thus, the findings also indicated that SFL has the most significant impact on detector performance, and the benefits of FE are independent of the dimensionality of the data.

In summary, the results reported in Table3 show that regardless of which detector is used, geometric expansion, feature space expansion, and pose-specific detectors improve the detection performance. Since detecting people lying on the ground is by nature more difficult than detecting people in an upright position, the performances reported here are lower than those reported in recent surveys [48]. However, our goal is to show that dataset expansion and feature learning are sound solutions for improving human lying pose detection accuracy.

To further prove the reliability of the theory, we evaluate the proposed detection algorithm on an MS COCO dataset that has the person object category (with person keypoints or person skeleton). It should be noted that a graphics processing unit (GPU, NVIDIA TITAN RTX@24GB, two pieces of GPU card) cluster system has been built to accomplish these tasks. We train the specific detector using automatic dataset expansion with structured feature learning on the union set of 64,115 images in the training set (262,465 human bodies) and 2,693 images in the validation set (11,004 human bodies, minival); the validation set was used for the performance tests. In the experiment, our detection algorithm improves the MS COCO-style mAP: an optimized data expansion policy with structured feature learning improves RefineDet512+ detection accuracy by more than +0.6 mAP, +1.5 mAP and +1.4 mAP, and the performance of other detectors also improved under these conditions. See the third column of Table3.

The results obtained by all three detectors are shown in Fig. 7. The bounding boxes show the results obtained using our entire framework, whereas those in blue, green, and red are the submodules of our framework, representing Faster R-CNN + GE + SFL + FE[blue], FPN + GE + SFL + FE[green], and RefineDet512+ + GE + SFL + FE[red]. The human lying poses could be well detected by these methods. These methods can also determine the direction of the human torso series. Although our framework is not without limitations (c.f., last row), the overall results are more accurate than those of the other frameworks.

Because sample expansion is an important parameter, we investigated the sensitivity of the Faster R-CNN, FPN, and RefineDet512+ models with respect to the geometric expansion and feature expansion frameworks. The evaluation of mAP for all three methods and different models of sample expansion is illustrated in Fig. 7. The performance of these detection models reaches a peak with the feature expansion framework. For feature expansion, the number of candidate bounding boxes becomes redundant such that the nonmaximal suppression and the computing rate gradually decrease for each image. This shows that the geometric expansion enriches the dataset and improves the results. The results were obtained using the MatConvNet and VGG libraries.

In this study, we developed a sample expansion framework for human lying pose detection. The main objective is to artificially increase the number of data points on which a human-shape detection method can be trained. Since generating a dataset that includes a large number of body configurations is challenging, our method starts with a small annotated dataset whose size is increased with a geometric expansion procedure and a Gibbs sampling procedure. Moreover, to account for various body poses, we built human lying

pose classifiers based on presegmented skeletons. The experimental results obtained on three state-of-the-art human-shape detection methods show the effectiveness of our expansion procedure.

Our future research will focus on other related topics, including occlusion, self-occlusion, camouflage (when the clothing of people has the same color as the background), poor illumination and motion blur (which often occurs when pictures are taken by a moving drone).

REFERENCES

[1] A. Andreopoulos and K. T. John, "50 years of object recognition: Directions forward," *Comput. Vis. Image Understand.*, vol. 117, no. 8, pp. 827–891, 2013.

[2] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietika, "Deep learning for generic object detection: A survey," 2018, *arXiv:1809.02165*. [Online]. Available: https://arxiv.org/abs/1809.02165

[3] S. Zhang, J. Yang, and B. Schiele, "Occluded pedestrian detection through guided attention in CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6995–7003.

[4] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: A survey," in *Proc. Conf. Graph., Patterns Images (SIBGRAPI)*, 2018, pp. 471–478.

[5] S. Su, S.-S. Wu, S.-Y. Chen, D.-J. Duh, and S. Li, "Multi-view fall detection based on spatio-temporal interest points," *Multimedia Tools Appl.*, vol. 75, no. 14, pp. 8469–8492, 2016.

[6] B. Kwolek and M. Kepski, "Improving fall detection by the use of depth sensor and accelerometer," *Neurocomputing*, vol. 168, pp. 637–645, Nov. 2015.

[7] W. Min, H. Cui, H. Rao, Z. Li, and L. Yao, "Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics," *IEEE Access*, vol. 6, pp. 9324–9335, Jan. 2018.

[8] K. L. Crandall and M. A. Minor, "UAV fall detection from a dynamic perch using Instantaneous Centers of rotation and inertial sensing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4675–4679.

[9] C. Iuga, P. Drăgan, and L. Busoniu, "Fall monitoring and detection for at-risk persons using a UAV," *IFAC-PapersOnLine*, vol. 51, no. 10, pp. 199–204, 2018.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[12] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4203–4212.

[13] D.-X. Xia, S.-Z. Su, S.-Z. Li, and P.-M. Jodoin, "Lying-pose detection with training dataset expansion," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3377–3381.

[14] D.-X. Xia, S.-Z. Su, L.-C. Geng, G.-X. Wu, and S.-Z. Li, "Learning rich features from objectness estimation for human lying-pose detection," *Multimedia Syst.*, vol. 23, no. 4, pp. 515–526, 2017.

[15] Z. Liu, Y. Cao, L. Cui, J. Song, and G. Zhao, "A benchmark database and baseline evaluation for fall detection based on wearable sensors for the Internet of medical things platform," *IEEE Access*, vol. 6, pp. 51286–51296, 2019.

[16] X. Chu, W. Ouyang, H. Li, and X. Wang, "Structured feature learning for pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4715–4723.

[17] H. Haggag, M. Hossny, S. Nahavandi, and O. Haggag, "An adaptable system for RGB-D based human body detection and pose estimation: Incorporating attached props," in *Proc. IEEE Conf. Syst., Man, Cybern.*, Oct. 2016, pp. 1544–1549.

[18] A. Abobakr, D. Nahavandi, J. Iskander, M. Hossny, S. Nahavandi, and M. Smets, "RGB-D human posture analysis for ergonomie studies using deep convolutional neural network," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2017, pp. 2885–2890.

[19] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2014.

[20] Y.-Z. Hsieh and Y.-L. Jeng, "Development of home intelligent fall detection IoT system based on feedback optical flow convolutional neural network," *IEEE Access*, vol. 6, pp. 6048–6057, 2017.

[21] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[22] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani, "Automatic monocular system for human fall detection based on variations in silhouette area," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 2, pp. 427–436, Feb. 2013.

[23] S. Wang, S. Zabir, and B. Leibe, "Lying pose recognition for elderly fall detection," in *Proc. Robot., Sci. Syst.*, 2011, pp. 345–353.

[24] A. Abobakr, M. Hossny, and S. Nahavandi, "A skeleton-free fall detection system from depth images using random decision forest," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2994–3005, Dec. 2018.

[25] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. Von Stryk, S. Roth, and B. Schiele, "Vision based victim detection from unmanned aerial vehicles," in *Proc. Conf. Int. Robots Syst.*, Oct. 2010, pp. 1740–1747.

[26] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[27] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Jan. 2014.

[28] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3626–3633.

[29] X. F. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3246–3253.

[30] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2241–2248.

[31] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2497–2504.

[32] A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg, "Part-based feature synthesis for human detection," in *Proc. Eur. Conf. Comput. Vis.*, vol. 6314, 2010, pp. 127–142.

[33] D. Levi, S. Silberstein, and A. Bar-Hillel, "Fast multiple-part based object detection using kd-ferns," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 947–954.

[34] H. M. Qian, Y. B. Mao, W. B. Xiang, and Z. Q. Wang, "Recognition of human activities using SVM multi-class classifier," *Pattern Recognit. Lett.*, vol. 31, no. 2, pp. 100–111, 2010.

[35] K. K. Chatfield Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf. (BMCV)*, 2014, pp. 1–11.

[36] X. Caoa, Z. Wangb, P. Yanc, and X. Lic, "Transfer learning for pedestrian detection," *Neurocomputing*, vol. 100, no. 16, pp. 51–57, 2013.

[37] H. Hattori, V. N. Bodetti, K. M. Kitani, and T. Kanade, "Learning scene-specific pedestrian detectors without real data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3819–3827.

[38] D. Vázquez, A. M. López, J. Marín, D. Ponsa, and D. G. Gomez, "Virtual and real world adaptation for pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 797–809, Aug. 2014.

[39] G.-R. Cai, P.-M. Jodoin, S.-Z. Li, Y.-D. Wu, S.-Z. Su, and Z.-K. Huang, "Perspective-SIFT: An efficient tool for low-altitude remote sensing image registration," *Signal Process.*, vol. 93, no. 1, pp. 3088–3110, 2013.

[40] R. Appel, T. Fuchs, P. Dollár, and P. Perona, "Quickly boosting decision trees–pruning underachieving features early," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 594–602.

[41] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1297–1304.

[42] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3258–3265.

[43] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2056–2063.

[44] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.
[45] X. Chen and A. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in *Proc. Adv. Neural Inf. Process. Syst. 27 (NIPS)*, 2014, pp. 1736–1744.
[46] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1899–1906.
[47] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–16.
[48] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 613–627.
[49] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Boston, MA, USA: MIT Press, 2009.
[50] Y. Kim, B.-N. Kang, and D. Kim, "SAN: Learning relationship between convolutional features for multi-scale object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 328–343.

DAOXUN XIA received the B.S. degree from the School of Mathematics and Computer Science, Guizhou Normal University, Guiyang, China, in 2004, the M.S. degree from the College of Computer Science and Technology, Guizhou University, China, in 2010, and the Ph.D. degree from the College of Information Science and Engineering, Xiamen University, in 2016. He joined the School of Big Data and Computer Science and the Engineering Laboratory for Applied Technology of Big Data in Education, Guizhou Normal University, in 2016. His main research interests include object detection, object recognition, computer vision, and big data technology.

LINGJIN ZHAO received the B.S. degree from the School of Geographic and Biologic Information, Nanjing University of Posts and Telecommunications, China, in 2012. She is currently pursuing the master's degree with the School of Big Data and Computer Science, Guizhou Normal University. Her main research interests include big data analysis and geographic information systems.

FANG GUO received the B.S. degree from the School of Information Mechanical and Electrical Engineering, Shanghai Normal University, China, in 2013. She is currently pursuing the master's degree with the School of Big Data and Computer Science, Guizhou Normal University. Her main research interests include big data analysis and computer vision.

XI CHEN received the M.S. degree from the College of Polytechnic, Hunan Normal University, Hunan, China, in 2007, and the Ph.D. degree in pattern recognition and biometrics from the Sichuan Province Key Laboratory of Signal and Information Processing, Southwest Jiaotong University, Chengdu, China. He joined the School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China, in 2011. He joined the School of Big Data and Computer Science, Guizhou Normal University, Guiyang, China, in 2016. His current research interests include digital signal processing, pattern recognition, and biometrics.