Animal Detection From Highly Cluttered Natural Scenes Using Spatiotemporal Object Region Proposals and Patch Verification

Zhi Zhang, Zhihai He, Guitao Cao, and Wenming Cao

***Abstract*—In this paper, we consider the animal object detection and segmentation from wildlife monitoring videos captured by motion-triggered cameras, called camera-traps. For these types of videos, existing approaches often suffer from low detection rates due to low contrast between the foreground animals and the cluttered background, as well as high false positive rates due to the dynamic background. To address this issue, we first develop a new approach to generate animal object region proposals using multilevel graph cut in the spatiotemporal domain. We then develop a cross-frame temporal patch verification method to determine if these region proposals are true animals or background patches. We construct an efficient feature description for animal detection using joint deep learning and histogram of oriented gradient features encoded with Fisher vectors. Our extensive experimental results and performance comparisons over a diverse set of challenging camera-trap data demonstrate that the proposed spatiotemporal object proposal and patch verification framework outperforms the state-of-the-art methods, including the recent Faster-RCNN method, on animal object detection accuracy by up to 4.5%.**

***Index Terms*—Background modeling, camera-trap images, graph cut, object proposal, object verification.**

I. Introduction

WILDLIFE monitoring with camera-trap networks, especially with the collaborative efforts of citizen scientists, enable us to collect wildlife activity data at large space and time scales and to study the impact of climate change, habitat modification and human disturbance on species richness and biodiversity along the dimensions of scale, region, season, and species [1]. Camera-traps are stationary camera-sensor systems attached to trees in the field. Triggered by animal motion, they record short image sequences of the animal appearance and activities associated with other sensor data, such as light level, moisture, temperature, and GPS sensor data. They are an important visual sensor for wildlife that can record animal appearance without disturbance. Due to their relatively low cost, rapid deployment, and easy maintenance, camera traps are now being extensively used in wildlife monitoring, with the potential to be deployed at large scales in space and time. From camera-trap images, we can extract a rich set of information about animal appearance, biometric features, species, behaviors, their resource selection, as well as important environmental features about the surrounding habitats [2]. During the past several years, a vast amount of camera-trap data has been collected, far exceeding the capability of manual image processing and annotation by human. There is an urgent need to develop animal detection, segmentation, tracking, and biometric feature extraction tools for automated processing of these massive camera-trap datasets. In this work, we focus on accurate and reliable animal object detection and segmentation from camera-trap images.

Detecting and segmenting moving objects from the background is an important and enabling step in intelligent video analysis [3], [4]. There is a significant body of research conducted during the past two decades on background modeling and foreground object detection [5]–[7]. However, the availability of methods that are robust and generic enough to handle the complexities of natural dynamic scenes is still very limited [8]. Videos captured in natural environments represent a large class of challenging scenes that have not been sufficiently addressed in the literature [4]. These types of scenes are often highly cluttered and dynamic with swaying trees, rippling water, moving shadows, sun spots, rain, etc. It is getting more complicated when natural animal camouflage added extra complexity to the analysis of these scenes. Fig. 1 shows some examples of image sequences captured by camera-traps at days (with color images) and nights (with infrared images). Here, each column represents a camera-trap image sequence triggered by animal motion. The key challenge here is how to establish effective models to capture the complex background motion and texture dynamics while maintaining sufficient discriminative power to detect and segment the foreground animals. Traditional motion-based techniques are not suitable here since the background is highly dynamic.

Recently, approaches based on deep neural networks, such as RCNN [9] and its variations Fast-RCNN [10] and Faster-RCNN [11], are achieving the state-of-the-art performance in object detection. Typically, these methods have two major components: 1) object region proposal which scans the whole image to generate a set of candidate image regions (or bounding boxes) at

Manuscript received February 19, 2016; revised May 30, 2016 and July 13, 2016; accepted July 14, 2016. Date of publication July 27, 2016; date of current version September 15, 2016. This work was supported in part by the National Science Foundation under Grant CyberSEES-1539389 and Grant CPS-1544794. The work of W. Cao was supported in part by the National Science Foundation of China under Grant 61375015. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alessandro Piva.

Z. Zhang and Z. He are with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211 USA (e-mail: zzbhf@mail.missouri.edu; hezhi@missouri.edu).

G. Cao is with the School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China (e-mail: gtcao@sei.ecnu.edu.cn).

W. Cao is with the College of Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: caom@shenzhenu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2016.2594138

Fig. 1. Samples of camera-trap images. Each column represents a camera-trap image sequence triggered by animal motions.

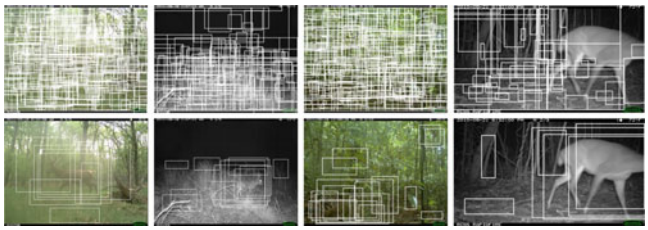


Fig. 2. Examples of bounding box proposals from spatial only and spatial-temporal methods. Each bounding box denotes a candidate object region. Top: selective search algorithm. Bottom: proposed IEC. The columns represent different scenes.

different locations and scales that could possibly contain the target objects, and 2) image classification which determines if these proposed regions are truly the objects or not. We observe that, within the context of animal detection from camera-trap images, these methods suffer from two major issues: speed and accuracy. First, the natural scenes in camera-trap images are highly cluttered. Existing object region proposal methods [12], [13] often generate a large number (thousands) of candidate object regions. We know that the deep convolutional neural network (DCNN) for region classification is computationally intensive and, more importantly, it needs to be performed thousands of times for each of these proposed object regions. Therefore, it is critical to consider the unique characteristics of camera-trap images in the spatiotemporal domain and design a new and efficient object region proposal method which can generate a small number of animal object proposals. To this end, we develop an iterative embedded graph cut (IEC) method with different foreground/background cut-off energy levels to create an embedded group of objects regions for the camera-trap image sequences. The examples in Fig. 2 show that IEC could significantly reduce the number of proposals while maintaining a sufficiently high object coverage rate. Second, we find that the direct application of DCNN to object regions in a single image is not efficient for animal-background classification. The performance can be significantly improved by extending the classification into the temporal domain using our proposed cross-frame patch verification method. Furthermore, for efficient animal object region classification, we find that a combination of DCNN and handcrafted features achieves better classification performance. Our extensive experimental results demonstrate that the proposed method significantly improves the performance while maintaining low computational complexity.

The *major contributions* of this paper can be summarized as follows: 1) We have developed a new and efficient animal object region proposal method using IEC which jointly consider the animal motion and spatial context in the spatiotemporal domain. 2) We propose a cross-frame image verification method for accurate animal-background classification. 3) We have found that, for camera-trap images, the DCNN image features and handcrafted histogram of oriented gradient (HOG) image features encoded with Fisher Vectors (FV) are able to enhance the classification performance for each other. 4) We have established a large dataset of camera-trap images which has been made available for the research community for developing efficient algorithms of object detection from highly cluttered natural scenes.

The remainder of the paper is organized as follows. We provide an overview of the proposed system in Section III. In Section IV, we present our animal object proposal method using iterative embedded graph cut. Section V explains the proposed cross-frame verification method. Experimental results are presented in Section VI. Section VII concludes the paper.

II. Related Work

This work is closely related to foreground-background segmentation, image verification, object region proposal, object detection and image classification. In the following, we provide a review of related work on these topics.

A. Foreground-Background Segmentation

Early work on background subtraction often operated on the assumption of stationary background. Several methods model the background explicitly, assuming a bootstrapping phase where the algorithm is presented with frames containing only the background [14], [15]. The use of multiple hypotheses to describe the behavior of an evolving scene at the pixel level significantly improves the performance of background modeling and subtraction [15]. Elgammal *et al.* [16] used a nonparametric background model to achieve better accuracy under the same constraints as the mixture of Gaussians. Sheikh and Shah incorporate the temporal and spatial consistencies into a single model [3]. Oliver *et al.* [17] focused on global statistics rather than local constraints to create a small number of eigenbackgrounds to capture the dominant variability of background. Considering spatial context and neighborhood constraints, graph cut optimization has achieved fairly good performance in image segmentation [7]. Iterated graph cut is used in [6] to search over a nonlocal parameter space. Background cut is proposed

in [18] which combines background subtraction and color or contrast-based models.

To handle background motion, various dynamic background texture models have been developed [15], [19]. Principal component analysis and autoregressive models are used in [17]. Wiener filters are used to predict the expected pixel value based on the past K samples. To reduce the computational complexity, Kahl *et al.* [20] demonstrated that using eigen-background on patches in an image is sufficient to capture the variance in dynamic scenes. In [21], for each pixel, it builds a codebook. Samples at each pixel are clustered into the set of codewords based on a color distortion metric. Gregorio and Giordano [22] use a weightless neural network to model the change in background. St-Charles and Bilodeau *et al.* [23] introduce a new strategy to tackle the problem of non-stationary background with pixel-level feedback loops to balance the local segmentation sensitivity automatically.

We recognize that, for accurate and robust video object detection and segmentation in dynamic scenes, background modeling of the dynamic pixel process at the image patch level, spatial context analysis and graph cut optimization at the region-level, and embedded foreground-background classification at the sequence level should be jointly considered. In this work, we propose to establish a new framework which tightly integrates these three important components for accurate and robust video object cut in highly dynamic scenes.

B. Region Proposals and Object Detection Using DCNN Methods

Recent studies [24], [25] have shown the extraordinary performance of DCNNs on image classification, object detection and recognition. To speed up the DCNN-based object detection process and avoid scanning of the whole image, object proposal methods have been recently developed for predicting object bounding boxes [11], [26]–[29]. Szegedy *et al.* [26] used a deep neural network as a regression model to predict the object bounding box. Sermanet *et al.* [28] developed a fully connected layer that is trained to predict the box coordinates for the localization task that assumes a single object. The fully connected layer is then turned into a convolutional layer for detecting multiple class-specific objects, which won the ILSVRC2013 localization competition. The original work on MultiBox [27] also used deep neural networks. Instead of producing bounding boxes, the MultiBox approach generates region proposals from a network whose last layer simultaneously predicts multiple class-agnostic boxes.

C. Image Verification

This work is also related to image verification. Image verification, in our particular problem, is regarded as a two-class classification problem: to verify if a proposed object image patch is an animal or belongs to the background scene. Classic learning-based image verification often involves two major steps: *feature representation* and *distance or metric learning*. Features used for image verification include colors, HOG, Haar-like descriptors, SIFT or SURF key point descriptors, maximally stable color regions, texture filters, differential local information, co-occurrence matrices, etc [30]. Statistics of low-level features, such as bag of words (BoW) descriptors, are also used for image verification to handle spatial variations. Recently, FVs [31] are developed which provides a better model to encode the local features. A number of methods built upon this FV approach [32], [33] have shown outstanding performance in image representation.

In this work, we propose to develop an effective cross-frame image verification method to determine if an image patch belongs to the background or not. This problem becomes very challenging since the background is highly dynamic and cluttered. In this work, we will demonstrate that a combination of DCNN features and hand-crafted image features specifically designed for camera-trap data is able to achieve significantly improved performance in animal image patch verification.

III. Algorithm Overview

We recognize that accurate and efficient animal detection from highly cluttered natural scenes in camera-trap images is a challenging task. To achieve accurate and fine-grain animal detection from the background, we need to perform image analysis at the pixel or small block level. However, with the low-contrast between the foreground animal and the cluttered background, it is often very difficult to determine if a pixel or a pixel block belongs to the animal or background based on local neighborhood information only, unless we resort to global image feature analysis. For example, pixels on the deer body might be very similar to the background vegetation. In this case, it is difficult for us to determine if these pixels belong to the deer based on local neighborhood information only until we see the deer head and legs, which involves global image analysis.

To address this issue, in this work, we propose a new animal-background detection framework which tightly couples object proposal using local image segmentation with global image region verification, as illustrated in Fig. 3. Specifically, it has two major components: 1) IEC for animal object proposal and 2) cross–frame patch-level object verification. The first component of IEC analyzes local image features and operates at the level of pixels or small blocks of pixels so as to maintain low computational complexity and achieve multi-level image segmentation in order to generate candidate regions for animal objects. To achieve high detection rate and ensure animals are all detected and covered in the foreground regions, we need to use a series of energy levels for the IEC so as to create an embedded set of regional proposals. Certainly, besides the target animal objects, the proposed regions will also contain regions or image patches from the background. The second component of image verification performs global comparison between foreground regions and background images across multiple frames. It extracts global features from the whole image patch, learns an image verification model to determine whether an image patch is similar to the background or not. In the following sections, we will explain these two components in more detail.

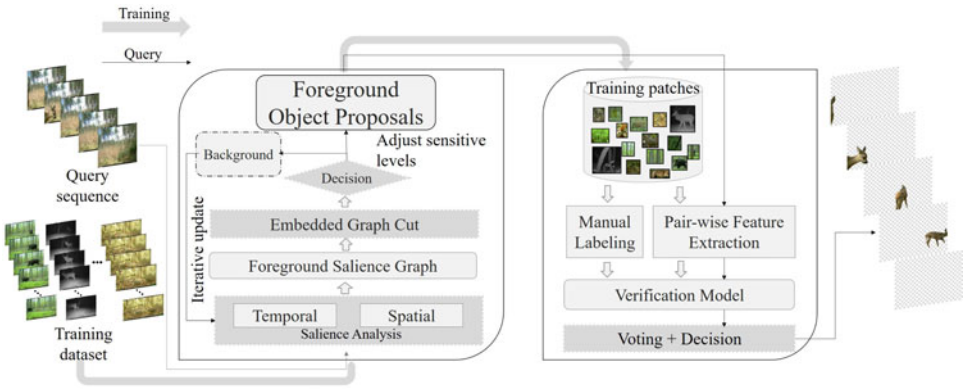


Fig. 3. Overview of the object detection system using the embedded video object cut.

IV. Animal Object Proposal Using IEC

In this work, we propose to develop a new and efficient animal object proposal method using IEC which exploit animal motion and temporal correlation in the camera-trap image sequence. Using a sequence of energy levels, the proposed IEC method is also able to address the over-proposal issue in existing methods [12]: generating too many object proposals in highly cluttered scenes. In our previous work [34], we have developed video object graph cut method, which achieved the state-of-the-art performance on object segmentation from videos with dynamic scenes and outperformed existing methods by up to 12% in segmentation accuracy on the 2012 Change Detection Challenge dataset [34]. In this work, built upon this method, we propose to develop an embedded graph cut method for animal object proposal.

A. Video Object Graph Cut

The video object graph cut algorithm developed in our previous work [34] constructs background models using HOG and BoW features. It constructs a foreground salience graph (FSG) to characterize the salience of an image patch in the spatio-temporal domain. It then formulates the object segmentation as an energy minimization problem which can be solved using the graph cut method. The FSG consists of two components: temporal salience and spatial salience. The temporal salience measures the dis-similarity $d(P^{(x,y)}, P_k^{(x,y)})$ between the current image patch $P^{(x,y)}$ at location (x, y) and the background model. The spatial salience measure between the dis-similarity between the current patch and its neighboring patches. Based on the temporal and spatial salience measures, we construct the FSG. We represent the image by an 8-connectivity undirected graph. A graph cut minimization procedure is used to find the segmented foreground object.

B. IEC for Animal Object Proposals

We recognize that, in cluttered scenes of camera-trap images, the initial segmentation often yields incorrect segmentation results and object contours. For example, in the *Camera-trap* dataset, we find that some parts of the animal body are well segmented in some video frames but poorly segmented in other frames since the foreground object has moved to different background regions. Motivated by this observation, we propose to propagate the foreground-background segmentation information across frames, helping each other to refine the segmentation in an iterative manner. More specifically, from the existing foreground-background segmentation results of all frames (typically about 10 frames in each camera-trap sequence), we estimate the foreground-background probability map for each frame. Specifically, for each pixel x, its foreground probability is defined as

$$p(x) = 0.5 + \frac{1}{\pi}\tan^{-1}(\beta \cdot s \cdot d[x]) \tag{1}$$

where $d[x]$ is the minimum distance from the pixel x to the boundary of the foreground regions, s is 1 if x is a foreground pixel, otherwise s is -1. β is a constant controlling the transition range. In this work, we set β to be 0.5. For each image patch P at location x_P, we use this probability map to modulate the FSG for graph cut of the new iteration.

It should be noted that, when performing the video object graph cut, there is an important parameter α which controls the level of penalty for foreground-background classification errors.

$$E_p(x_p) = \begin{cases} \frac{D^t(p)}{\sigma}, & x_p = 0 \\ \alpha(1 - \frac{D^t(p)}{20\sigma}), & x_p = 1 \end{cases} \tag{2}$$

Here E_p is the T-link graph-cut energy term, $D^t(p)$ is the temporal saliency at position x_p, as described in [34]. Intuitively, α serves as a foreground energy level parameter, it is simply the weight controlling of how many positive cut error are counted towards the objective function, for example, if $\alpha = 0.5$, then only half of graph-cut energy is generated. From the animal segmentation example in Fig. 4, we can see that, as α decreases, the graph-cut produces more and larger patches, classifying more pixels into the foreground and thus increasing the probability for the animals to be covered by these foreground regions. From our experiments, we recognize that it is difficult to determine a fixed optimal setting for the value of α since different camera-trap image sequences will require different α to achieve the best segmentation results. More importantly, for the same α, the segmentation performance is also region-dependent: some foreground regions are over segmented while others are under segmented.

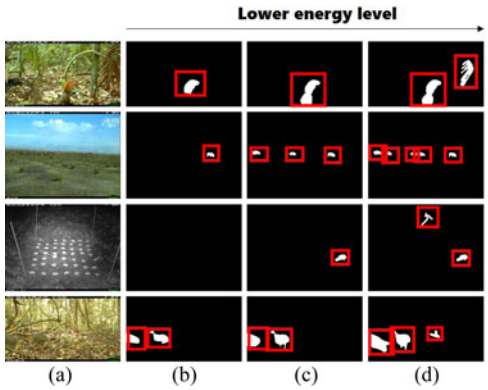


Fig. 4. Initial foreground segmentation with different energy level parameters. (a) Original image. (b)-(d) Foreground maps with lowering energy levels α.

In this work, we propose to use α as the control parameter to generate an embedded set of region proposals for animal detection. More specifically, in our proposed IEC scheme, we perform the video object graph cut with multiple values of the control parameter α, denoted by $\{\alpha^n | 1 \leq n \leq N\}$. N is typically 3 to 10 controlling the number of different α. We choose $N = 5$ in the experiment. Let $\mathbf{M}^n$ be the corresponding foreground regions. We have

$$\mathbf{M}^1 \subseteq \mathbf{M}^2 \cdots \subseteq \mathbf{M}^N. \tag{3}$$

We define patch $\mathbf{P}^1 = \mathbf{M}^1$, and $\mathbf{P}^n = \mathbf{M}^n - \mathbf{M}^{n-1}$ for $n = 2, \ldots, N$. To increase the granularity of the proposed regions, we select those large patches from $\mathbf{P} = \{\mathbf{P}^n | 1 \leq n \leq N\}$

$$\mathbf{P}_L = \{\mathbf{P}^k | \, S(\mathbf{P}^k) > \Delta, \mathbf{P}^k \in \mathbf{P}\} \tag{4}$$

where $S(\mathbf{P}^k)$ represents the size or number of pixels in the patch, and Δ is a threshold parameter controlling the granularity levels of the region proposals. We further segment each patch $\mathbf{P}^k$ in $\mathbf{P}_L$ into smaller ones $\{\mathbf{P}_j^k\}$ of size less than Δ using a k-mean clustering algorithm on its pixels inside with location and color features. We always use 2 clusters in an iterative fashion, which is enough for finer segmentation. We denote all small patches $\{\mathbf{P}_j^k | \mathbf{P}^k \in \mathbf{P}\}$, plus those small patches in the original set $\mathbf{P}_L$, by $\mathbf{p} = \{p_l | 1 \leq l \leq L\}$. Similar to other region proposal approaches [12], we will use these small patches to generate region proposals using a neighborhood grouping procedure. Due to the bounding box constraints, we use rectangle template matching for region proposals. We denote the set of region proposals, or candidate object bounding boxes, by $\mathbf{B} = \{\mathbf{B}_i | 1 \leq i \leq I\}$.

V. Cross-Frame Animal-Background Verification

In the previous section, using IEC, we have generated a set of animal object proposals $\mathbf{B}$. In this section, we will develop a cross-frame animal-background verification method to score each bounding box or image patch in $\mathbf{B}$, determining if it is an animal or not. We observe that false positive foreground patches generated by the animal object proposal often have significantly different characteristics from true animal image patches. Fig. 5(a)–(d) illustrate four examples of false positives:

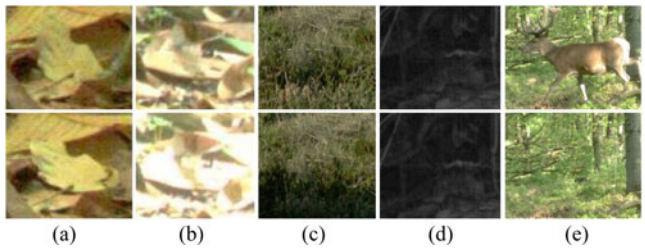


Fig. 5. Pairs of image patches for verification. (a)-(d) are examples of false positive: (a) displacement of leaves; (b) illumination change; (c) shadow; and (d) sensor noises in low light conditions. (e) is an example of one true animal object.

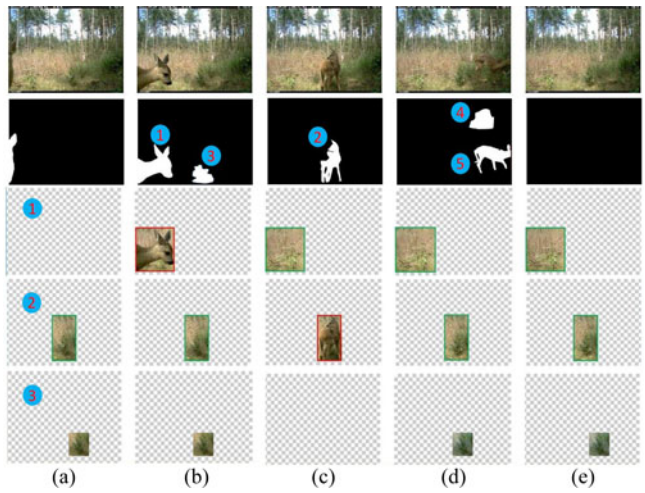


Fig. 6. Work flow of animal verification: (a) original camera-trap images; (b) foreground segmentation using ensemble graph cuts; and (c)-(d) cross-frame patch verification for foreground object 1–3, respectively. To save space, foreground objects 4 and 5 are ignored in this figure.

(a) displacement of leaves and branches caused by wind, (b) dramatic change of sun lighting, (c) moving shadow, and (d) sensor noises in low light conditions. Fig. 5(e) shows one example of true animal. Here, the top and bottom rows are from two different frames with interval of 3 s. The task of image verification here is to determine if the image patch in the top row is the same as the one in the bottom row.

A. Cross-Frame Animal Object Verification

In Fig. 6, we use one example to show the work flow of our cross-frame animal-background verification. The first row (a) shows the original camera-trap images where a deer passes by. Row (b) shows the foreground-background segmentation results obtained by the ensemble graph cut method explained in Section IV. We can see that image patches 1 and 2 are animals while patch 3 is wrongly detected due to a moving shadow. To perform cross-frame animal-background verification, for each foreground patch $B_i \in \mathbf{B}$, we extract co-located image patches across the entire sequence at the same location of the same size. We exclude those patches which have significant overlap with foreground regions in the same frame. Here, we set the max overlap ratio to be 0.25, which is the ratio between the intersection part and the size of B_i. We denote the remaining patches

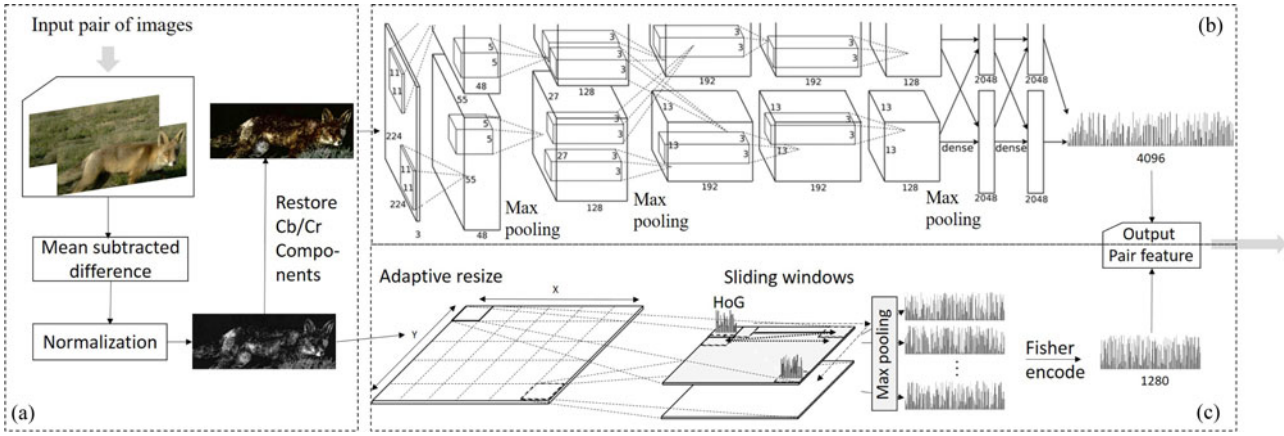


Fig. 7. Proposed feature extraction module: (a) pre-processing; (b) deep CNN feature extraction; and (c) adaptive size HOG FV descriptor extraction.

by $O_{i,m}$, $1 \leq m \leq M_i$. We will learn a pairwise verification model $\mathcal{V}$ to determine if a pair of image patches B_i and $O_{i,m}$ are similar to each other up to a certain amount of background changes, including texture dynamics, shadows, noise, lighting changes, etc. Specifically, $\mathcal{V}(B_i, O_{i,m}) = 1$ implies that B_i and $O_{i,m}$ are similar to each other. Otherwise, $\mathcal{V}(B_i, O_{i,m}) = -1$. To form a joint decision for B_i, we use the following weighted summation:

$$S = \sum_{m=1}^{M} w_m \times \mathcal{V}(B_i, O_{i,m}) \tag{5}$$

where w_m is the normalized weight of the mth background patch. We introduce w_m to accommodate the background transition, especially in a long sequence. In this work, we choose w_m as the inverse of the frame distance between B_i and $O_{i,m}$

$$w_m = \frac{1}{|N[B_i] - N[O_{i,m}]|^{\epsilon}}. \tag{6}$$

Here, $N[\cdot]$ represents the frame index of the image patch. $\epsilon \in (0, 1]$ is a parameter controlling the amount of contribution from each frame. After normalization, we then use $\{w_m\}$ as the weights to compute the verification score S. If $S > 0$, we determine that image patch B_i belongs to the background. Otherwise, it is classified as the animal.

B. Learning the Animal-Background Verification Model

In this section, we discuss how to train the verification model $\mathcal{V}(B_i, O_{i,m})$ to determine if two collocated image patches from the camera-trap image sequence are similar to each other. The major challenges are: 1) this verification model should be able to accommodate large background variations between B_i and $O_{i,m}$, which include background texture dynamics, displace of objects (such as tree leaves or branches), changes of lighting conditions, moving shadows or sun spots, etc. This requires that the image feature description should be invariant under these types of changes. 2) Unlike other image verification tasks, such as face verification, our animal-background verification needs to deal with image patches with a wide ranges of sizes and aspect ratios since they are directly generated by the ensemble graph cuts.

To address this challenge, we propose the following image processing and feature representation schemes for animal-background verification. As is shown in Fig. 7, our scheme has three major steps: 1) background modulation, 2) fine-tuned DCNN features; 3) adaptive HOG extraction and FV encoding.

1) Background Modulation: We recognize that, in camera-trap images, the background is highly cluttered and has large variations across frames due to background motion. To minimize this background interference and improve the animal-background verification performance, we propose to modulate the original image patch using the background model. We call this pre-processing step as mean subtracted difference enhancement (MSDE). In MSDE, the pair of image patches are first converted to the YCbCr color space. We then perform mean centering (scale normalization) to the absolute difference between foreground and reference background images on Y (luminance) components with all negative values clamped to zero. Note that this truncation must be coupled with mean subtracted difference image because we will normalize the entire image to 8-bit range [0, 255] to stretch the contrast between object and background. Fig. 8 shows three examples of the MSDE images.

2) DCNN Features Learned From the Camera-Trap Images: Recent results on image classification, scene recognition, fine-grain object recognition and detection, and image retrieval have demonstrated that the generic image descriptors learned from DCNN are very effective [35]. A DCNN consists of multiple convolutional layers and then followed by one or more fully connected layers. Pooling methods are often used in DCNN to achieve translation invariant properties for DCNN features. In this work, we follow the architecture described by Krizhevsky *et al.* [35] using the caffe [36] implementation. We use the pre-trained DCNNs model by Girshick *et al.* [37] on large auxiliary ILSVRC 2012 dataset and perform fine-tuning on our annotated camera-trap images. During fine-tuning, we replace the 21-way soft max output classification layer with a two-way animal-background layer. The rest five convolutional and three fully-connected layers remain the same. The input image is resized to

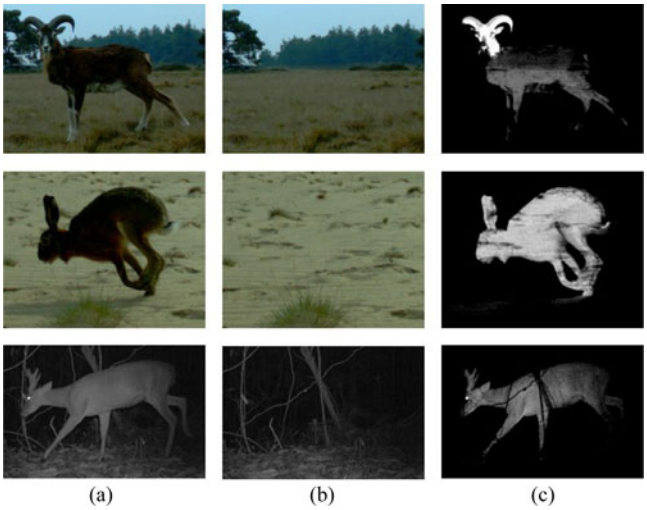


Fig. 8. Background modulation using MSDE. (a) Animal patches. (b) Reference backgrounds. (c) Processed patches using MSDE.

227×227 regardless of its original size and aspect ratio to fit in the CNN input layer. We recognize that this will change the aspect ratios of the input images and create image distortion. However, since both image patches, P_k and O_{km} under the verification task are experiencing the same distortion, the impact on the verification performance by this distortion is very limited. The output DCNN feature is a 4096-dimensional vector.

3) HOG With FV Encoding: DCNN features, self-learned from training data, provide an efficient and generic low to mid-level features that are rich enough to describe image details such as edge, shape, color distribution, etc. However, it often requires a large amount of data for training and fine-tuning to achieve desired performance levels. In practice, the amount of available training data is often limited. In this work, we explore a mixture of self-learned DCNN features and hand-crafted image features, aiming to enhance the performance of DCNN features with specially designed features for our animal-background verification. Specifically, we propose to use FV-encoded HOG (histogram of oriented gradients) features. We use HOG since it is insensitive to illumination change and is able to capture local textures. We partition the input image into the 20×20 blocks without overlapping (the input patches to our animal-background verification module will have different sizes and aspect ratios). Within each block, we extract its HOG feature.

We observe that the original HOG feature are not able to accommodate large background variations and texture dynamics. To address this issue, we propose to encode the HOG features using FV [38] to extract high-level invariant statistical features. The FV encoding aggregates a large set of HOG vectors into a high-dimensional vector representation by fitting a parametric generative model, e.g. the Gaussian Mixture Model to these features, and then encoding the derivatives of the log-likelihood of the model with respect to its parameter [38]. Unlike the BoW model which approximates a feature using a pre-trained codebook, FV successfully retains

Fig. 9. Example grount-truth images from the Camera_Trap dataset.

extra information by soft assignment to the Gaussian components. Let $\gamma_t(k)$ be the soft assignment of the descriptor d_t to the Gaussian component k where w_k, μ_k and σ_k are respectively the mixture weight, mean vector and covariance matrix of Gaussian k

$$\gamma_t(k) = \frac{w_k \mu_k(d_t)}{\sum_{j=1}^{K} w_j \mu_j(d_t)}. \tag{7}$$

$G^M_{\mu,k}$ and $G^M_{\sigma,k}$ are descriptor gradients with respect to μ_k and σ_k of the component k. $G^M_{\mu,k}$ and $G^M_{\sigma,k}$ can be computed as

$$G^M_{\mu,k} = \frac{1}{T\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \Big(\frac{m_t - \mu_k}{\sigma_k} \Big), \tag{8}$$

$$G^M_{\sigma,k} = \frac{1}{T\sqrt{2w_k}} \sum_{t=1}^{T} \gamma_t(k) \left[\frac{(m_t - \mu_k)^2}{\sigma_k^2} - 1 \right]. \tag{9}$$

We concatenate $G^M_{\mu,k}$ and $G^M_{\sigma,k}$ vectors for $k = 1, 2, \ldots K$, resulting in a $2 \times 16 \times K$ dimensional feature, where 16 is the dimensionality of our local descriptor.

C. Animal-Background Verification Model Training

The FV-encoded HOG feature is a 1280-dimension vector according to our experimental setup. After normalization, we concatenate it with the 4096-dimension DCNN feature, resulting in a 5376-dimension feature vector for the image pair under verification. To generate the training data, we have manually labeled 800 camera-trap image sequences (about 8000 images in total), placing bounding boxes around animals. For each animal patch B_i^+ inside the bounding box, we find their collocated image patches in the same sequence $\{O_{i,m}^+\}$ and make sure that all $O_{i,m}^+$ are from the background. Each pair of B_i^+ and $O_{i,m}^+$ will constitute a positive sample for training our verification model. To construct negative samples, we randomly generate bounding boxes within the background regions. For each background patch B_i^-, we extract it collocated patches $\{O_{i,m}^-\}$ within the background regions. Each pair of B_i^- and $O_{i,m}^-$ will constitute a negative sample. The next step is to combine Fisher HOG feature with DCNN feature, followed by a classifier. Here one option is to implement a customized adaptive HOG layer in neural network, a concatenate layer to combine features with original DCNN feature, followed by a softmax loss layer. Considering support vector machine (SVM) and softmax layer provide very close performance as classifier [9], and Fisher-HOG are not highly parallelizable, we propose to use external concatenation

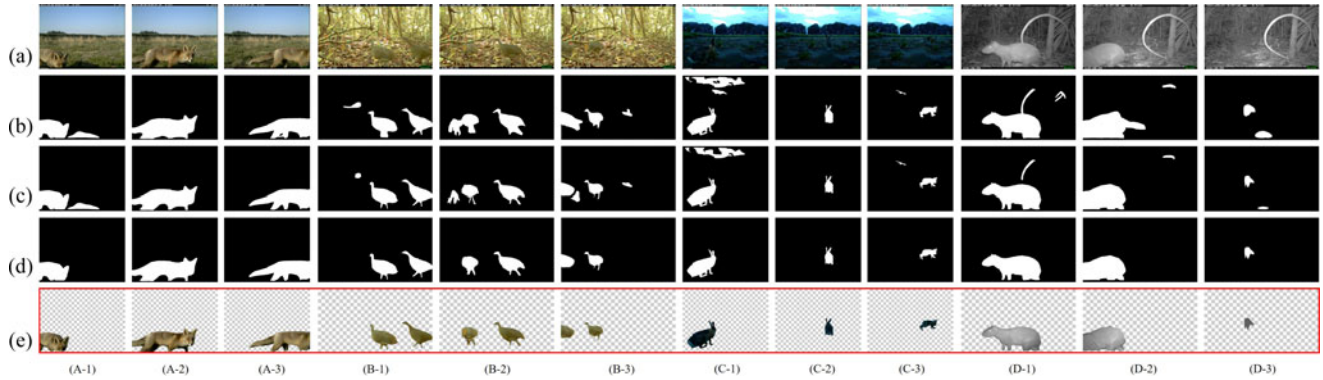


Fig. 10. Examples of segmentation and verification results on the *Camera_Trap* dataset with four (A-D) sequences: (a) original image in sequences; (b) initial segmentation foreground map; (c) iterative ensemble object cut results; (d) joint object verification results; and (e) animal segmentation results using the proposed method. The required output is animal bounding boxes; however, the foreground masks are still available from graph-cut results.

to train the pairwise verification model $V(B_i, O_{i,m})$ using linear SVM.

VI. Experimental Results

A. Datasets

In this work, we use our Camera_trap and popular CDnet dataset for evaluation purposes. We establish the Camera-Trap dataset for performance evaluation of animal detection from highly cluttered natural scenes. We made this dataset publicly available for peer researchers: videonet.ece.missouri.edu/cameratrap/videonet.ece.missouri.edu/cameratrap/. In our ongoing work on automated large-scale wildlife monitoring, we have collected over 1 million camera-trap images of wildlife species. This dataset consists of 800 camera-trap image sequences with 23 animal species, in both daytime color and nighttime infrared formats. These are very challenging videos with highly cluttered and dynamic natural scenes. Spatial resolutions of the images vary from 1920×1080 to 2048×1536, and are considered as very high resolution images. The length of each sequence varies from 10 to more than 300 frames depending on the duration of the animal activities. For each image, we have manually labeled the bounding box of each animal in the scene. An overall of 6493 ground-truth foreground bounding-boxes are recorded and archived in the dataset. Ground-truth labels are stored in a plain text file, each row corresponding to one image. Fig. 9 shows some example images from the dataset.

We also include Change Detection dataset as a reference dataset since its popular in motion detection community. More details will be covered in Section VI-C.

For performance evaluations, we plan to use the following performance metrics as in the CDNet [39] dataset: 1) Recall $TP/(TP+FN)$; 2) Precision $TP/(TP+FP)$; and 3) F-measure

$$F = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2TP}{2TP + FN + FP} \quad (10)$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. True detections are those which have intersection-over-union (IoU) ≥ 0.5 with ground-truth bounding boxes.

Fig. 11. Examples of final animal object detection and segmentation results.

TABLE I
Performance Comparison on Object Proposals in the Camera_Trap Dataset With Other Methods

Method	Avg. # Proposals		
	80% Coverage	90% Coverage	Best Coverage (Maximum %)
Selective Search [12]	2829.7	5903.5	13882 (**99.8%**)
GOP [41]	2489.1	3984.6	9874 (98.2%)
MOP [42]	335.8	482.3	891.7 (96.7%)
FCOP [43]	132.8	384.2	**393.1*** (90.4%)
STODP [44]	146.0	352.8	418.5 (91.9%)
Proposed	**95.4**	**237.3**	626.9 (93.1%)

B. Experimental Setup

For background modeling with HOG and BoW, we use a patch size of 16×16. The codebook size is set to be 128. All

TABLE II
PERFORMANCE COMPARISON ON CAMERA_TRAP DATASET

	EC Best	YOLO[45]	Fast-RCNN[10]	Faster-RCNN[11]	IEC+DCNN	Proposed
Train-set Finetune-set		voc07+voc12 camera-trap	voc07+voc12 camera-trap	voc07+voc12 camera-trap	camera-trap	camera-trap
Agouti	0.7382	0.7239	0.8088	0.8105	0.8218	**0.8364**
Collared Peccary	0.8436	0.8516	0.8838	0.8865	0.9049	**0.9202**
Paca	0.7799	0.7658	0.797	0.8055	0.7946	**0.8226**
Red Brocket Deer	0.7772	0.7905	0.8492	**0.8794**	0.8587	0.8723
White-nosed Coati	0.8221	0.8016	0.8739	0.8883	0.8893	**0.8993**
Spiny Rat	0.6908	0.7016	0.7729	0.7924	0.789	**0.8092**
Ocelot	0.7935	0.7893	0.8592	0.8796	0.8732	**0.8855**
Red Squirrel	0.7978	0.7761	0.8682	0.8901	0.8839	**0.8914**
Common Opossum	0.7395	0.7582	0.8187	0.8456	0.8263	**0.8623**
Bird spec	0.5505	0.4968	0.6083	0.6188	0.6515	**0.6717**
Great Tinamou	0.6964	0.7247	0.8282	0.8473	0.8546	**0.8699**
White-tailed Deer	0.7847	0.8165	0.8251	0.8549	0.8403	**0.8611**
Mouflon	0.7788	0.7743	0.8197	0.8395	0.8429	**0.8782**
Red Deer	0.8555	0.8642	0.8792	**0.9052**	0.898	0.9008
Roe Deer	0.8353	0.8548	0.8853	0.8968	0.8956	**0.9076**
Wile Boar	0.8013	0.8553	0.8732	0.9018	0.8922	**0.907**
Red Fox	0.676	0.6548	0.7538	0.7682	0.7765	**0.7933**
European Hare	0.6695	0.6561	0.7862	0.7892	0.7983	**0.8283**
Wood Mouse	0.7176	0.6815	0.7972	0.8136	0.8098	**0.8357**
Coiban Agouti	0.6678	0.6915	0.7982	0.8046	0.8121	**0.8221**
Average	0.7587	0.7674	0.8251	0.8493	0.8417	**0.8597**

Metrics showing average **Recalls**.

TABLE III
PERFORMANCE COMPARISON ON CAMERA_TRAP DATASET

	EC Best	YOLO[45]	Fast-RCNN[10]	Faster-RCNN[11]	IEC+DCNN	Proposed
Train-set Finetune-set		voc07+voc12 camera-trap	voc07+voc12 camera-trap	voc07+voc12 camera-trap	camera-trap	camera-trap
Agouti	0.7632	0.7593	0.742	0.7514	0.7875	**0.8244**
Collared Peccary	0.8209	0.8359	0.8015	0.8094	0.7682	**0.8152**
Paca	0.7969	0.8169	0.8039	0.8289	0.8122	**0.8333**
Red Brocket Deer	0.8563	0.8915	0.8517	**0.8879**	0.8658	0.8867
White-nosed Coati	0.8059	0.8314	0.7899	0.7952	0.803	**0.8221**
Spiny Rat	0.7539	0.7642	0.7193	0.7314	0.7604	**0.7756**
Ocelot	0.7918	0.8192	0.7726	0.7952	0.8011	**0.8154**
Red Squirrel	0.7345	0.7682	0.7328	0.7437	0.7638	**0.7727**
Common Opossum	0.7816	0.8164	0.7951	0.8155	0.8023	**0.8205**
Bird spec	0.6527	0.7465	0.6412	0.6619	0.6898	**0.7228**
Great Tinamou	0.789	0.8349	0.8035	0.8148	0.8313	**0.8441**
White-tailed Deer	0.8218	0.8432	0.8303	0.8792	0.8551	**0.8671**
Mouflon	0.7594	0.8448	0.7692	0.7846	0.7922	**0.8107**
Red Deer	0.7947	0.8214	0.7963	0.7991	0.8234	**0.8391**
Roe Deer	0.7969	0.8391	0.7793	0.7925	0.8022	**0.8218**
Wile Boar	0.7863	0.8417	0.7965	0.805	0.8131	**0.8282**
Red Fox	0.6471	0.7349	0.6752	0.6849	0.7056	**0.7358**
European Hare	0.7156	0.7514	0.7391	0.7485	0.753	**0.772**
Wood Mouse	0.7094	0.7539	0.7293	0.7336	0.7493	**0.7632**
Coiban Agouti	0.7316	0.7815	0.749	0.7598	0.7732	**0.7778**
Average	0.7824	0.8315	0.7801	0.7886	0.8017	**0.8209**

Metrics showing average **Precisions**.

region candidates with $\geq$ 0.5 IoU overlap with ground-truth boxes are treated as true positive patches, the rest are regarded as false positives. However, to avoid mixing possible animal object patch with negative samples, in practice, we only select IoU $= 0$ patches as negatives for training. Thanks to pre-trained object classification model, we can easily achieve more than 97% training accuracy in less than 12 hours on an Nvidia GTX Titan X GPU. We use 50% (400 sequences) as training data and the rest for testing in the following experiments. We also use Pascal VOC 2007 and 2012 detection dataset [40] as auxiliary data to train a generic object model to be used on CDnet 2014 dataset [39] since it contains objects such as cars and boats. To use Pascal dataset, we randomly sample background regions with zero overlapping to any foreground region, marked as 0.

TABLE IV
PERFORMANCE COMPARISON ON CAMERA_TRAP DATASET

	EC Best	YOLO[45]	Fast-RCNN[10]	Faster-RCNN[11]	IEC+DCNN	Proposed
Train-set Finetune-set		voc07+voc12 camera-trap	voc07+voc12 camera-trap	voc07+voc12 camera-trap	camera-trap	camera-trap
Agouti	0.7505	0.7436	0.7783	0.7825	0.8043	**0.8303**
Collared Peccary	0.8321	0.8246	0.8455	0.8546	0.831	**0.8646**
Paca	0.7883	0.7816	0.8004	0.8145	0.8035	**0.828**
Red Brocket Deer	0.8148	0.8241	0.8568	**0.8803**	0.8622	0.8795
White-nosed Coati	0.814	0.8348	0.8398	0.8415	0.8439	**0.859**
Spiny Rat	0.721	0.7282	0.7485	0.7503	0.7745	**0.7921**
Ocelot	0.7926	0.7844	0.8048	0.8117	0.8356	**0.849**
Red Squirrel	0.7648	0.7486	0.7892	0.7962	0.8194	**0.8278**
Common Opossum	0.76	0.7782	0.8071	0.8286	0.8142	**0.8409**
Bird spec	0.5973	0.5543	0.6367	0.6488	0.6701	**0.6963**
Great Tinamou	0.7398	0.7581	0.8143	0.8185	0.8428	**0.8568**
White-tailed Deer	0.8028	0.8147	0.847	**0.8672**	0.8476	0.8641
Mouflon	0.769	0.7498	0.7962	0.8067	0.8168	**0.8431**
Red Deer	0.824	0.8345	0.8397	0.8416	0.8591	**0.8689**
Roe Deer	0.8157	0.8354	0.8254	0.8435	0.8463	**0.8626**
Wile Boar	0.7937	0.8491	0.8312	0.8477	0.8508	**0.8658**
Red Fox	0.6612	0.6814	0.7162	0.7211	0.7394	**0.7634**
European Hare	0.6918	0.6815	0.7573	0.7604	0.775	**0.7992**
Wood Mouse	0.7135	0.6981	0.7681	0.7692	0.7784	**0.7978**
Coiban Agouti	0.6982	0.7204	0.7582	0.7685	0.7921	**0.7993**
Average	0.7703	0.7515	0.7937	0.8043	0.8212	**0.8398**

Metrics showing average **F-scores**.

And we use 20 original classes objects as positive samples, marked as 1. For all verification models in experiments, we finetune a 2-way classification model, with a batch size 128, learning rate 0.01, momentum 0.9 and a decay of 0.0005. We continue training with a maximum iteration 40 000 on camera-trap, and 80 000 on Pascal VOC, respectively. When training accuracy is higher than 98% and stopped climbing for a while, we stop the training to prevent over-fitting.

C. Experimental Results

1) Qualitative Evaluations: Fig. 10 shows some example experimental results on four sequences from different species, which are *Red Fox*, *Great Tinamou*, *European Hare* and *Paca*, respectively. Row (a) is the original image from the camera-trap. Due to space limitations, we only include 3 out 10 images here. Row (b) shows the segmentation results after the video object graph cut. The animal body boundaries (shapes) are not very accurate and there a significant amount of incorrect segmentation results. After several iterations of cross-frame information fusion and graph cuts by utilizing existing background information, better results are achieved in row (c). We can see that false positive patches caused by background variations, such as shadows, waving leaves, and moving clouds, are still in the segmentation results. Row (d) shows the final results after animal-background verification. These false positive patches have been successfully removed. Row (e) shows the animal pixels with row (d) as the mask. We can see that the proposed method is able to achieve very accurate and reliable segmentation of the foreground animals in dynamic scenes by preserving true positives and filtering out false positives. Fig. 11 shows some examples of animal segmentation from highly cluttered and dynamic natural scenes.

2) Quantitative Results: We first compare the performance of our animal object proposal method using IEC with the following state-of-the-art object proposal methods: Spatial-Temporal Object Detection Proposals (STODP) [44], Fully Connected Object Proposals for Video Segmentation (FCOP) [43], and Learning to Segment Moving Object in Videos (MOP) [42]. For reference purposes, we also compare the performances of single frame object proposal techniques, as proposed in Geodesic Object Proposals [41] and Selective Search [12]. The latter is very popular and is used in RCNN and fast-RCNN as the enabling proposal method. Table I provides the average number of proposal bounding box required by each method in order to cover 80%, 90% and the most ground-truth animal objects, respectively. Here, coverage is the percentage of ground-truth bounding boxes which have $IoU \geq 0.5$ with any box in detection proposal list. The best coverage rate indicates the capability of detecting all objects in every frame. Improving the coverage is hard and costly, which often results in a massive amount of proposal detections. We can see that our proposed method is much more efficient than existing methods at the coverage rates of 80% and 90%, and find a good trade-off between the number of proposals (which affects the subsequent verification time) and the coverage rate. The single-frame based methods, such as the Selective search and GOP often produce a large amount of proposals. The limitation of proposed IEC method is its weakness in detecting slow moving object, which could be neglected in motion triggered dataset such as camera_trap. In return, IEC is exceptionally good at filtering out non-candidate object proposals, which is a crucial for accurate animal detection.

Tables II, III, and IV provides quantitative recall, precision and F-score comparisons on our Camera_trap dataset, respectively. We compare our proposed method with the following

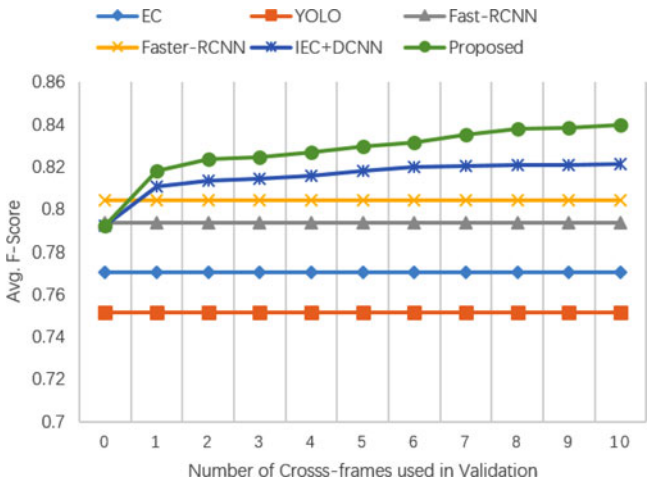


Fig. 12. Effect of number of frames used in validation on the overall F-score performance.

TABLE V
AVERAGE PROCESSING TIME PER IMAGE IN SECONDS WITH VARIOUS EXPERIMENTAL CHOICES

	Experimental choices					
Use Selective Search	✓	✓	✓			
Use Iterative Graph-Cut				✓	✓	✓
Run CNN feature extractor on GPU		✓	✓		✓	✓
Run CNN feature extractor on CPU	✓			✓		
Run CNN using large batch			✓			✓
Proposal generation	0.75	0.75	0.75	1.03	1.03	1.03
Verification	8.94	3.68	2.92	4.19	1.01	0.54
Total	9.69	4.43	3.67	5.22	2.04	1.57

methods: (1) EC, Ensemble graph Cuts developed in our previous work [34]; (2) YOLO, a fast unified object detection algorithm which detect objects in single inference [45]; (3) Fast-RCNN [10]; (4) Faster-RCNN [11], an improved version of Fast-RCNN which propose candidates using fully convolutional network; (5) the IEC+DCNN method which performs animal-background verification using the DCNN features only; and (6) the proposed method that uses joint DCNN and FV-HOG features. Note that models in (2)–(4) are trained on Pascal VOC dataset (convolutional layers are pre-trained on imagenet [46] for better convergence) and fine-tuned on camera-trap dataset. The last column shows the performance of the method developed in this work which uses combined FV-HOG and DCNN features for animal-background verification. We can see that the animal-background verification significantly improves the performance, either with FV-HOG or DCNN features. The verification module can significantly reduce the false positive rates while maintaining high true-positive rates. The average F-Score is 83.98%, almost 7% higher than the original 77.03%. The average precision is also improved by a large margin from 78.24% to 82.09%. We also conclude that it is more efficient to combine FV-HOG and DCNN features and they can enhance the performance of each other. The proposed framework outperforms the fine-tuned Faster-RCNN [11] by up to 4%. With limited training data and highly dynamic and cluttered background, it is easy to explain that cross-frame information fusion could compensate for the inevitable noise introduced during learning and fine-tuning, thus improves the performance of learned models.

In order to evaluate the impact of cross-frame validation, we adjust the number of pairs, i.e., M in (5) to be used in the verification phase. A special case is when M $= 0$, all cross-frame validation is disabled, and features are only extracted on a single patch without MSDE. From Fig. 12, we can see that, with more frames being included in the verification, the overall F-Score is being improved significantly. Our proposed method outperforms Fast-RCNN and Faster-RCNN using only one pair, which proved the effectiveness of cross-frame validation in improving the final performance.

We also monitored the processing time of our system with different experimental choices, as shown in Table V. The hardware choice is Intel Xeon E5-1603v3 @2.8GHz (4C4T) and Nvidia Titan X. The results indicate that even with longer processing time of our iterative graph-cut method, it significantly reduced the number of verification candidates, thus in return not only benefit to overall performance but also ran faster.

3) Performance Evaluation as Segmentation Task: We also evaluate our proposed method on Change Detection 2014 dataset [39], using three categories: dynamic background, low frame-rate, and intermittent object motion. These selected categories best fit the complex scenes in the wild, thus are ideal for comparison with other approaches. Note that CDnet dataset is purely for motion detection and does not provide any training data. In order to make our patch verification model to recognize objects such as cars and boats in CDnet, we need extra training data. In experiments, we evaluate the models trained on camera-trap, Pascal VOC 2007+2012 [40] and both, respectively. Table VI shows the performance comparisons on CDnet 2014 dataset. By comparing with state-of-art approaches, we observed that no single algorithm could handle all scenes well enough especially in different categories. Our method, though require off-line processing and auxiliary training data, is very robust in maintaining relatively good recall while preserve very high precision. For intermittent object motion category, our method achieved weaker performance compared with the other two categories due to the incapability to detect object candidates at the first stage, i.e., iterative graph-cut, however, our verification model still achieved high precision by rejecting most non-object detections. As is shown in Fig. 13, our method successfully addressed the false alarms that are inevitably spotted by various algorithms due to the complex scenes. By comparing columns (d) to (i), we can see the two-step propose and reject scheme can alleviate the trade-off problem between recall and precision. The generic object based verification model is highly robust, effective, and self-adaptive to various complex scenes. We also noticed that mixing camera-trap with Pascal VOC dataset generates slightly inferior results. Considering the entirely different types of scenes between camera-trap and Pascal VOC, and the large volume of Pascal VOC dataset itself, we recognize that using Pascal VOC dataset is good enough to train

TABLE VI
PERFORMANCE COMPARISON ON CHANGE-DETECTION 2014 DATASET USING THREE CATEGORIES

		Dynamic Background			Low Frame-rate			Intermittent Object Motion		
	Training data	avg Re	avg Prec	avg F	avg Re	avg Prec	avg F	avg Re	avg Prec	avg F
AMBER [47]		**0.9177**	0.799	0.8436	0.5226	0.5937	0.4689	0.7617	0.753	0.7211
EFIC [48]		0.6667	0.6849	0.5779	0.7694	0.7232	0.6632	0.7416	0.5634	0.5783
CEFIC [49]		0.6556	0.6993	0.5627	0.8077	0.7135	0.6806	**0.8107**	0.5823	0.6229
CwisarDRP [50]		0.8291	0.8723	0.8487	0.7718	0.7045	0.6858	0.4614	**0.8543**	0.5626
SSBS		0.7875	0.9185	0.8391	0.7446	0.6816	0.691	0.484	0.8255	0.54
FTSG [51]		0.8691	0.9129	0.8792	0.7517	0.655	0.5259	0.7813	0.8512	**0.7891**
PAWCS [52]		0.8868	0.9038	0.8938	0.7732	0.6405	0.6588	0.7487	0.8392	0.7764
IUTIS-3[53]		0.8778	0.8239	0.896	0.8213	0.6995	0.7327	0.6987	0.8146	0.7136
IUTIS-5[53]		0.8636	0.9324	0.9239	**0.8398**	0.7424	0.7743	0.7047	0.8501	0.7296
This work	camera-trap	0.8593	0.9014	0.8615	0.7925	0.7851	0.6248	0.6714	0.7924	0.6624
This work	voc07+voc12	0.8859	**0.9425**	**0.9381**	0.8196	**0.8248**	**0.8016**	0.7036	0.8362	0.6952
This work	trap+voc07+12	0.8764	0.9401	0.9296	0.8049	0.8205	0.7736	0.7099	0.8217	0.6846

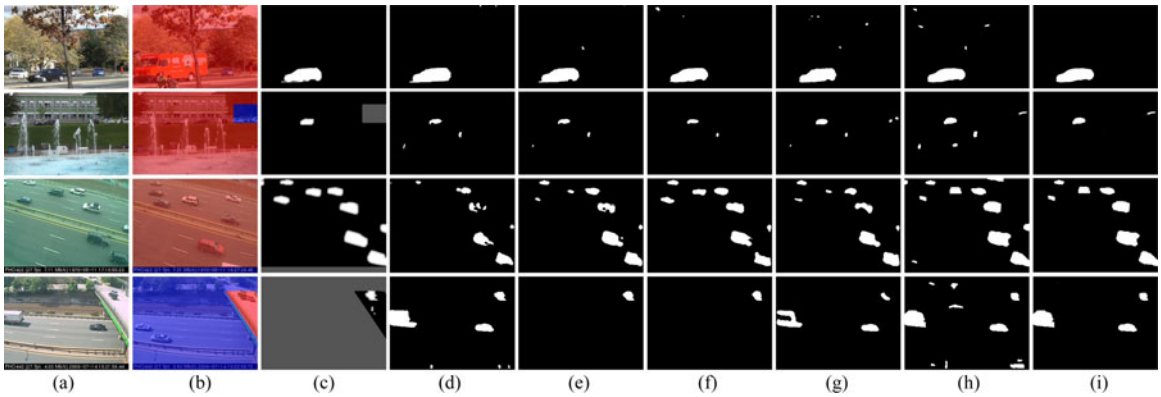


Fig. 13. Segmentation results comparison with state-of-the-art methods on Change-Detection 2014 dataset. (a)-(c) Original frame, region of interest, and ground-truth mask. (d) FTSG [51]. (e) IUTIS-5 [53]. (f) PAWCS [52]. (g) Superpixel strengthen background subtraction. (h) IEC without verification model. (i) Proposed method.

a generic purpose objectness model to be used in CDnet dataset which has similar human surrounded environments.

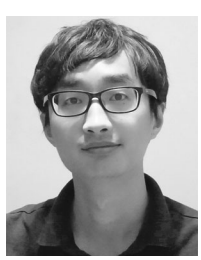
VII. CONCLUSION

In this paper, we have successfully developed an accurate method for animal object detection from highly cluttered natural scenes captured by motion-triggered cameras, called camera-traps. We developed a new approach to generate animal object region proposals using multi-level graph cut in the spatiotemporal domain. We then developed a cross-frame temporal patch verification method to determine if these region proposals are true animals or background patches. We found that the DCNN and FV-HOG features are able to enhance the performance of each other during animal object verification. Our extensive experimental results and performance comparisons over a diverse set of challenging camera-trap data demonstrated that the proposed spatiotemporal object proposal and patch verification framework is sensitive to objects in motion and confident in rejecting false alarms, thus is capable of building the basis of a robust object detection system in dynamic scenes.

REFERENCES

[1] S. Tilak *et al.*, "Monitoring wild animal communities with arrays of motion sensitive camera," *Int. J. Res. Rev. Wireless Sensor Netw.*, vol. 1, pp. 19–29, 2011.
[2] R. Kays *et al.*, "eMammal—Citizen science camera trapping as a solution for broad-scale, long-term monitoring of wildlife populations," in *Proc. North Am. Conservation Biol.*, 2014, pp. 80–86.
[3] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.
[4] T. Ko, S. Soatto, and D. Estrin, "Background subtraction on distributions," *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 276–289.
[5] Y. Ren, C.-S. Chua, and Y.-K. Ho, "Motion detection with nonstationary background," *Mach. Vis. Appl.*, vol. 13, no. 5, pp. 332–343, Mar. 2003.
[6] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, 2004.
[7] Y. Boykov and V. Kolmogorov, "An experimental comparison of mincut max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.
[8] V. Mahadevan and N. Vasconcelos, "Background subtraction in highly dynamic scenes," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–6.
[9] 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 Recog.*, Jun. 2014, pp. 580–587.
[10] R. Girshick, "Fast r-CNN," in *Proc. Int. Conf. Comput. Vis.*, pp. 1440–1448, 2015.

[11] K. H. Shaoqing Ren and J. S. Ross Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[12] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.
[13] M. M. Cheng, Z. Zhang, W. Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 3286–3293.
[14] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th IEEE Int. Conf. Pattern Recog.*, Aug. 2004, vol. 2, pp. 28–31.
[15] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, vol. 2, pp. 1305–1312.
[16] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. 6th Eur. Conf. Comput. Vis.*, 2000, pp. 751–767.
[17] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.
[18] J. Sun, W. Zhang, X. Tang, and H. Y. Shum, "Background cut," *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 628–641.
[19] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, vol. 2, pp. 1236–1242.
[20] F. Kahl, R. Hartley, and V. Hilsenstein, "Novelty detection in image sequences with dynamic background," in *Statistical Methods in Video Processing*. New York, NY, USA: Springer, 2004, pp. 117–128.
[21] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imag.*, vol. 11, no. 3, pp. 167–256, 2005.
[22] M. D. Gregorio and M. Giordano, "Change detection with weightless neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops.*, Jun. 2004, pp. 409–413.
[23] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Flexible background subtraction with self-balanced local sensitivity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2014, pp. 414–419.
[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 1717–1724.
[25] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2014, pp. 512–519.
[26] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2553–2561.
[27] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 2155–2162.
[28] P. Sermanet *et al.*,"Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, Dec. 2013. [Online]. Available: http://adsabs.harvard.edu/abs/2014arXiv1412.1441S.
[29] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," Dec. 2014.
[30] O. Oreifej, R. Mehran, and M. Shah, "Human identity recognition in aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 709–716.
[31] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.
[32] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.
[33] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods," in *Proc. Brit. Mach. Vis. Conf.*, 2011, vol. 2, pp. 76.1–76.12.
[34] X. Ren, T. X. Han, and Z. He, "Ensemble video object cut in highly dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 1947–1954.
[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[36] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[37] 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 Recog.*, Jun. 2014, pp. 580–587.
[38] B. Ma, Y. Su, and F. Jurie, "Local descriptors encoded by fisher vectors for person re-identification," in *Proc. ECCV Workshops*, 2012, pp. 413–422.
[39] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "changedetection.net: A new change detection benchmark dataset," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2012, pp. 1–8.
[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2012 (VOC2012) results," [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
[41] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 725–739.
[42] K. Fragkiadaki, P. Arbeláez, P. Felsen, and J. Malik, "Learning to segment moving objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 4083–4090.
[43] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung, "Fully connected object proposals for video segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3227–3234.
[44] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid, "Spatio-temporal object detection proposals," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 737–752.
[45] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, Jun. 2015. [Online]. Available: http://arxiv.org/abs/1506.02640
[46] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[47] B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2014, pp. 401–404.
[48] G. Allebosch, F. Deboeverie, P. Veelaert, and W. Philips, "Efic: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints," in *Proc. 16th Int. Conf. Adv. Concepts Intell. Vis. Syst.*, 2015, pp. 130–141.
[49] G. Allebosch, D. Van Hamme, F. Deboeverie, P. Veelaert, and W. Philips, "C-efic: Color and edge based foreground background segmentation with interior classification," in *Proc. 10th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2015, pp. 433–454.
[50] M. De Gregorio and M. Giordano, "Background modeling by weightless neural networks," in *Proc. New Trends Image Anal. Process.—ICIAP Workshops*, 2015, pp. 493–501.
[51] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split Gaussian models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2014, pp. 420–424.
[52] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 990–997.
[53] S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?," May 2015. [Online]. Available: http://adsabs.harvard.edu/abs/2015arXiv150502921B.



Zhi Zhang received the B.S. degree in electronic and information technology from Beijing Jiaotong University, Beijing, China, in 2012, the M.S. degree in electrical and computer engineering from the University of Missouri-Columbia, Columbia, MO, USA, in 2014, and is currently working toward the Ph.D. degree in electrical and computer engineering at the University of Missouri-Columbia.

His current research interests include multimedia content retrieval, object recognition and classification, wild animal detection, and person reidentification.

Zhihai He (S'98–M'01–SM'06–F'15) received the B.S. degree in mathematics from Beijing Normal University, Beijing, China, in 1994, the M.S. degree in mathematics from the Institute of Computational Mathematics, Chinese Academy of Sciences, Beijing, China, in 1997, and the Ph.D. degree in electrical engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2001.

In 2001, he joined Sarnoff Corporation, Princeton, NJ, USA, as a Member of Technical Staff. In 2003, he joined the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO, USA, where he is currently a Tenured Full Professor. His current research interests include image/video processing and compression, wireless communication, computer vision, and sensor networks.

Prof. He served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, and the *Journal of Visual Communication and Image Representation*. He was also a Guest Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Special Issue on Video Surveillance. He was the Co-Chair of the 2007 International Symposium on Multimedia over Wireless in Hawaii. He is a Member of the Visual Signal Processing and Communication Technical Committee of the IEEE Circuits and Systems Society, and serves as Technical Program Committee Member or Session Chair of a number of international conferences. He was the recipient of the 2002 IEEE Transactions on Circuits and Systems for Video Technology Best Paper Award and the SPIE VCIP Young Investigator Award in 2004.

Guitao Cao received the M.S. degree from Shandong University, Jinan, China, in 2001, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2006.

She is an Associate Professor with the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China. She has authored or coauthored more than 30 publications, including papers that have appeared in the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING. Her research interests include image processing and pattern recognition, as well as media analysis and understanding.

Wenming Cao received the M.S. degree from the System Engineering Research Institute, Chinese Academy of Sciences, Beijing, China, in 1991, the Ph.D. degree from Southeast University, Nanjing, China, in 2004, and the Postdoctoral degree from the Institute of Automation, Beijing, China, in 2007.

He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China. He is also currently the Director of the Department of Electronic Engineering, Shenzhen University. His research interests include signal processing and communication, media analysis and retrieval, and pattern recognition.

Prof. Cao was the recipient of the First Prize of the Beijing Municipal Science and Technology Progress Award and the Second Prize of the Zhejiang Province Municipal Science and Technology Progress Award in 2005.