Efficient and Accurate Object Classification in Wireless Multimedia Sensor Networks

Hakan Öztarak*, Turgay Yilmaz†, Kemal Akkaya‡ and Adnan Yazıcı†

*ASELSAN Inc., P.K. 1, Yenimahalle, Ankara, Turkey 06172

Email: hoztarak@aselsan.com.tr

†Department of Computer Engineering, Middle East Technical University, Ankara 06531, TURKEY

Email: {turgay—yazici}@ceng.metu.edu.tr

‡Department of Computer Science, Southern Illinois University Carbondale, IL 62901, USA

Email: kemal@cs.siu.edu

***Abstract*—Object classification from video frames has become more challenging in the context of Wireless Multimedia Sensor Networks (WMSNs). This is mainly due to the fact that these networks are severely resource constrained in terms of the deployed camera sensors. The resources refer to battery, processor, memory and storage of the camera sensor. Limited resources mandates the need for efficient classification techniques in terms of energy consumption, space usage and processing power. In this paper, we propose an efficient yet accurate classification algorithm for WMSNs using a genetic algorithm-based classifier. The efficiency of the algorithm is achieved by extracting two simple but effective features of the objects from the video frames, namely shape of the minimum bounding box of the object and the speed of the object in the monitored region. The accuracy of the classification, on the other hand, is provided through using a genetic algorithm whose space/memory requirements are minimal. The training of this genetic algorithm based classifier is done offline and it is stored at each camera in advance to perform online classification during surveillance missions. The experiments indicate that a promising classification accuracy can be achieved without introducing a major energy and storage overhead on camera sensors.**

I. Introduction

Wireless Multimedia Sensor Networks (WMSNs) have received a lot of attention very recently due to their potential to be deployed flexibly in various outdoor applications with lower costs [1]. Such networks deploy a large number of image/video sensors with different capabilities and can collect/process multimedia data [2] [3]. Typical applications of WMSNs include remote surveillance, target tracking, habitat monitoring, intrusion detection and health care delivery [1].

Considering these applications, current WMSN research has focused on the issues regarding energy-efficient video data routing, quality of service (QoS) and intelligent camera actuation for increased network lifetime, which are mostly related to networking aspects of the application [1] [4]. However, WMSNs are multi-camera systems and they also deal with traditional problems of image/video processing to perform object localization, object/event detection and classification. Such processing is associated with networking issues as it may provide in-network processing to improve the network lifetime and reduce the bandwidth requirements. This relationship indicates the need for WMSNs to be able to perform image/video data processing with limited resources in an energy-efficient manner. In this way, the gap between networking and data processing research can be filled so as to realize the deployment of WMSNs in a wide variety of real-life applications.

To this end, our focus in this paper is to tackle the object classification problem which may involve some image/video processing along with data transmission in WMSNs. Deploying a large number of battery-operated image/video camera sensors in the surveillance region helps to alleviate issues regarding possible obstacles (blocking cameras) and thus provides great convenience in terms of object/event coverage. However, accurate classification of the detected objects regardless of the WMSN size and area coverage percentage is still a challenge. If the detected objects are classified correctly on site, then the central decision unit, i.e., the sink, may be alarmed based on the received object information. This is very crucial given that surveillance applications are geared for security and safety. For instance, in a power-plant surveillance application, only human intruders who are strangers should trigger an alarm for the guards. In case of a detected animal or an employee, no alarm is necessary.

In order to perform an accurate object classification, an effective set of features should be selected and a robust classifier should be constructed [5] [6] [7]. Nevertheless, limited resources of WMSNs restricts the options for chosing the features and the classifier. Specifically, features and classifiers which are lightweight in terms of processing, energy consumption and storage are needed. In addition, real-time applicability of the classifier is crucial considering the fact that the classification process is performed at camera sensors when an object is detected. Finally, the flexibility of the classifier for adding new features and object classes and making it applicable for other domains is also a big plus.

In this paper, we choose two simple but effective features from the video frames: $Shape\_Ratio$ and $Speed$ of the

Turgay Yilmaz is currently with the Institute of Industrial Science, University of Tokyo, Japan.

This work is supported in part by a research grant from Turkish National Science Foundation TÜBİTAK EEEAG with grant number 109E014.

detected objects. The idea is to reduce the cost of the extraction of the features to save energy and improve the WMSN lifetime. These two features are obtained by only extracting the minimum bounding rectangle (MBR) of the detected objects in their video frames. Assuming that the objects' locations can be known by using a localization teachnique, the speed of an object can be calculated by using location information of the object at different times.

For the classifier, we employ a genetic algorithm based classifier proposed in [8] for the best accuracy. This classifier is designed as an instance based classifier which provides a lightweight solution with its low time complexity. In addition, the classifier utilizes the Class-Specific Features (CSF) [9] in order to relate the prototype classes with the most representative and discriminative features for them. The training of the classifier is done offline with a set of sample objects and then the classifier is stored at each camera sensor before the data collection starts. Once the object is classified at a camera sensor online, this class information is sent to the sink node along with the speed and location of the object to extract certain events. The experiments show that the classifier can classify the most usual object types such as human or animal accurately with lower costs in terms of energy, time and storage.

This paper is organized as follows: In the next section, we summarize the related work. Section III states our considered system model and assumptions. In Section IV, we present the details of the feature selection, classifier and our approach. Section V includes performance evaluation of the proposed approach. Finally, Section VI concludes the paper.

II. Related Work

A. Wireless Multimedia Sensor Networks

Recent works on WMSNs focused on processing and compression of the raw video data sent to the sink in order to increase network lifetime, providing Quality of Service (QoS) for data communication and placement of cameras for desired coverage [1], [4], [10]–[12]. In addition to these studies, there are surveillance specific tasks such as object detection, localization or tracking which have also received some interest [13]–[17] by considering the resource constraints of WMSNs.

Object classification is also part of surveillance applications which can be performed in conjunction with object detection and tracking. The main challenges with object classification stay with video processing, data elimination and data reduction so as to minimize communication and coordination needs of the involved camera sensors for saving energy. However, currently, there is not much research focusing solely on this aspect of WMSNs which indirectly affects the networking/communication costs. The works in resource constraint WSNs (such as [18]) do not apply as they work only on scalar sensors not cameras. Our work is one of the first to address this issue in the context of WMSNs at resource constraint cameras rather than the resource rich base-station or sink node. Given that there is no work in WMSNs on classification, next we look at the related literature on object classification and feature selection in general surveillance applications without any resource constraints.

B. Object Classification

Until now, a significant amount of research has been done on classification problem in general. A detailed discussion on several types of classifiers and an extensive comparison of them can be found in [7] and [19]. Below, we present a broad categorization of classifiers according to their decision units. Also, we analyze their applicability to WMSNs according to their complexity and flexibility.

The classifiers in literature can be grouped into three different approaches: Instance-based, probabilistic and decision boundary-based [7]. In an instance-based classifier, objects are classified according to the similarities with the prototypes. The classifier model holds the prototype classes, prototype objects in each class and their features. In order to classify a given object (i.e., query object), similarity of the object with each prototype class is calculated. Thus, the complexity of the classifier in terms of processing, energy and time depends on the similarity function and the features chosen. If reasonable distance functions with some simple features are utilized, this approach provides a lightweight solution. Moreover, considering that the raw features are hold in the classifier model, it is easy to extend the classifier by including new features and objects classes. However, this approach suffers from the risk of low accuracy if its model contains a few number of prototypes for each class and they do not represent the classes well or noise exists in the prototypes. There may be also a high storage requirement if the model is constructed with a large number of prototypes in order to represent the prototype classes better. Besides, the probabilistic and the decision boundary-based approaches perform better than the instance-based approaches at the expense of more complex operations. Some of the well-known classifiers of this type are as follows: Template matching classifier, Nearest mean classifier, Subspace method, 1-nearest neighbor rule [20].

Probabilistic classifiers perform classification by using the generative probability models obtained via the distributions of object classes over the feature space. The classifier model does not hold the features, but holds the probability models. Thus, the storage requirement is less for this type. However, it may not be possible to obtain the flexibility as in the instance-based approaches since the probability models should be recalculated when new features and classes are included. The accuracy of probabilistic classifiers are reasonably high considering the low complexity of the classifier model, but it is sensitive to the outliers and class density estimation errors. Some of the well-known classifiers of this type are: Naive Bayes classifier, K-nearest neighbor rule, Logistic classifier, Parzen classifier [20].

The decision boundary-based classifiers construct decision boundaries by optimizing some certain error criteria and perform classification based on these boundaries. Decision boundary approach is the most complex one among the three approaches with respect to the processing. However, the accuracy of this approach is usually better. As in the case of the probabilistic approaches, it is not easy to include new features and classes. The decision boundary should be recalculated. Some of the well-known classifiers of this type are as follows: Fisher linear discriminant, Perceptron, Binary decision tree, Neural networks, Support vector classifier [20].

Considering these three approaches, the third one is the least convenient for WMSNs because of its complexity and inflexibility though it provides the best accuracy. Thus, we prefer to use a classifier that benefits from the advantages of both the instance-based classifiers and the probabilistic classifiers, as described in [8]. The classifier is first constructed like an instance based classifier by respecting the low-complexity requirements of the WMSNs. Yet, the classifier model is constructed with a large number of prototypes. Then, by applying genetic algorithm operations, the prototypes are regulated according to their fitness and the probability knowledge present in the training data is absorbed so that the classifier is enhanced.

C. Feature Selection

One important issue for classification of objects is the representation of images. There exists an excessive number of visual features proposed for the representation of images. The features can be classified as global and local features. The former include color [21], shape [22] and texture [21] descriptors on the whole image while the latter is based on some special local points in the image (SIFT [23], SURF [24]). However, most of these features are low-level features. Extraction and utilization of such low-level features impose a high processing complexity as far as the WMSN applications are concerned.

As a result, we prefer utilizing more semantic features which can be easily extracted during normal video processing for tracking purposes (e.g., object detection or localization). The goal is to reduce the burden on camera sensors when extracting the features. However, there is a tradeoff here regarding the number and type of the features used and the classification accuracy. Our features of object shape and object speed are unique and novel in the sense that they do not require additional processing while they can still represent the objects for accurate classification.

III. Preliminaries

A. Assumptions

We assume a WMSN consisting of a set of camera sensors (e.g., CmuCam3 [25]), placed randomly in a power-plant surveillance application to monitor multiple moving objects. In our application, we assume three general object classes that may be captured: human, animal and vehicle. The resolution and storage features of CmuCam3 camera sensors are [26]: 1) Data format of Common Intermediate Format with resolution (352x288) on RGB color sensor; 2) For storage, MMC/SD flash slot (e.g., PQI 128MB MMC, SanDisk 2GB/1GB/512MB SD cards, SanDisk 512MB MMC cards).

Each camera sensor has a certain Field-of-View(FoV) β and Depth-of-Field(DoF) d which are the angle and the distance respectively, where the camera sensor can capture an accurate image/video as seen in Figure 1. The camera sensors are assumed to have a fixed random position and orientation. We assume that all camera sensors know their locations [27] and can communicate with one another independent of the type of camera sensor as long as they are within the transmission range of each other.

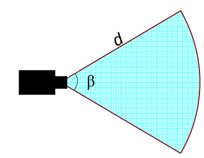

Fig. 1. A camera sensor with its FoV (β) and DoF (d)

B. Problem Definition

Our problem can be formally defined as follows: "Given a video camera sensor which is generating video frames within a WMSN,determine the class of a detected object by such a camera with maximum accuracy and minimum overhead in terms of energy, time and storage". Our goal is to minimize the need for human intervention on the WMSN applications with automated and accurate object classification.

IV. Moving Object Classification Using a Genetic Algorithm Based Classifier

A. Feature Selection & Extraction

In visual object classification task, extracting features from images and performing classification with these features are usually complex and costly operations. In order to provide a lightweight classification mechanism in terms of processing and energy consumption, it is crucial to adapt an energy-efficient and effective solution for the choice of features and the classification process. Therefore, in this paper, we propose to use two simple visual features that are related to the objects rather than the whole image: $Shape\_Ratio$ and $Speed$. The $Shape\_Ratio$ feature is the ratio of width and height of the detected object's minimum bounding rectangle (MBR), whereas $Speed$ feature gives the speed of the object. Note that while these features are easy to extract, the classification using such limited features (i.e., in terms of the number and the characteristics of features) will be challenging. We will address this challenge when we employ a suitable classifier as will be detailed in the next subsection.

$Shape\_Ratio$ can be easily obtained from a detected object. To do this, first the object is extracted from the video frames using frame differencing at individual camera sensors. MBR on the image can then be easily calculated. This process does not require a lot of video processing as it only depends on the frame differencing which was shown to be energy-efficient in [17]. MBR has the height and width of the object which are used to calculate $Shape\_Ratio = Width/Height$

The speed can also be calculated based on some pre-processing using camera properties and camera locations. Specifically, once the object is extracted, the distance between the object and the camera sensor is calculated. The object location is then calculated by using the camera's location and direction, the object position on the frame and the distance between the object and the camera. At the end, previous frame's results for localization is used to calculate the speed of the object in the area under surveillance. This process has also been shown to be feasible and cost-efficient in [17] for object tracking applications.

B. Object Classification using GA based Classifier

In this paper, an instance-based classification mechanism is adopted. The instance-based mechanism requires some prototype images for each class as the classifier model and makes a decision by using the dissimilarities of the prototypes with the query object. To avoid the large space requirements of such a classifier model (i.e., it can be used on resource constrained cameras), a pre-defined number of prototypes are selected by using a Genetic Algorithm (GA) based approach [8] on the training data. In addition to the GA support, the classifier utilizes CSF approach [9] in order to define prototype objects with the most representative and discriminative features.

The classifier model is constructed offline and can be deployed to the camera sensors to be used in classification process. Note that when a new and improved classifier is available, this classifier can also be deployed to the camera sensors to replace the existing classifier.

1) Classifier Model Construction During Training: Before starting the actual classification, a training phase is performed to construct the classifier model. The goal in this phase is to train the algorithm by using a set of prototype images. In order to do this, two things need to be performed: 1) Objects features that best represent each class need to be identified by using Class Specific Feature (CSF) indices; and 2) A genetic algorithm (GA) should be used to classify candidate prototype objects from all images available. Thus, the classifier model contains two major parts: CSF Model and GA Model. Note that the object features used here are different than the features that will be used in the classification of the objects after the training is done.

As introduced in [9], the CSF model is built on the idea that different object types can be represented better by different visual features. For example, a 'car' object can be represented better by its 'shape', whereas a 'sea' object can be represented with its 'color'. In this study, the class-specific feature selection mechanism, which is proposed in [9], is utilized. The method finds out the representative and discriminative features for each image class. The representative characteristics of features are calculated according to the dissimilarities of images within the same class, and discriminative characteristics are calculated according to the dissimilarities of images between different image classes. Using these representative and discriminative characteristics, the weight values of features for each image class are calculated. These weight values are referred to as CSF indices and are used during the decision making in the classification process.

GA model contains the prototype images for each class. These prototype images are obtained by using the GA based approach in [8]. Based on the GA notations, each class is represented with a set of chromosomes (prototypes) which are representative images of the class. Each chromosome holds an *Effectiveness Value* that is used as a weight during the decision making. In addition, each chromosome is represented with a set of genes which are the representative features for that class in correspondence with the CSF model.

To utilize a GA based approach and perform genetic operations, the training phase is divided into two parts: one training phase for obtaining the definitions of the training objects (First-Training) and one for making improvements on the definitions by using genetic operations (Second-Training). The initial population of genetic algorithm is generated by using the First-Training data. In other words, the classes in the classifier model gains chromosomes by identifying new objects in the First-Training phase. Then, the genetic operations are applied to the system during Second-Training. These genetic operations are: *Effectiveness Correction*, *Crossover* and *Mutation*. By applying these operations, "survival of the fittest" principle of genetic algorithm is provided and the fittest chromosomes (prototypes) are selected as the classifier model.

2) Classification Process on the Camera Sensors: Our proposed classification method performs multiple labeling on a query object by providing a fuzzy membership decision for each class in the range [0,1]. For instance, if there are two classes as human and animal, an object's fuzzy membership value can be [0.7, 0.3] which indicates that the object is classified as a human with 0.7 fuzzy membership value and as an animal with 0.3 fuzzy membership value. In order to obtain the membership value of each class, decisions of chromosomes that are contained in that class are combined by using a weighted sum. The weights of the chromosomes are the effectiveness values of chromosomes that are updated during the genetic operations according to the fitness of the chromosomes. The decision of each chromosome is also found by a weighted sum. To obtain a chromosome decision, decisions of genes in the chromosome are combined. This time, the weights are the CSF indices for the decision-giving class. Lastly, decision of each gene is calculated by using the similarity of the query object with the corresponding prototype object to which the feature belongs.

As can be seen from the explanations above, the calculation of the fuzzy membership value does not bring a significant processing overhead to the camera sensors as the calculations are mostly finding the weighted sums.

V. Algorithm Analysis

The steps that are followed before the actual classification algorithm is run are listed below. These are the steps to extract the features that we use in the classification of the object.

1) Detect the moving object on the frame
2) Calculate the distance between the moving object and the camera sensor
3) Locate and calculate the speed of the moving object

We can assume that the features that we are using in the classification will be available as part of object detection and tracking in surveillance. Therefore, we focus on the analysis of the classification algorithm only. We analyze the time, energy and storage complexity of the algorithm to see its applicability to WMSNs.

A. Time Complexity Analysis

As mentioned in Section IV-B2, the object classification process requires calculating the similarity between the query object and each of the prototype objects in the classifier model for each feature. Considering that preferred features are one-dimensional, performing an Euclidian-distance calculation in order to find the similarities requires a constant time, which is $O(1)$. Thus, assuming that m and n denote the feature count and the total number of prototypes respectively, the total classification process is performed in $O(mn)$ time.

Note that m is the feature count that cannot increase dramatically (e.g., it is 2 in our case). As a result, n is a more dominant value than m. Hence, we can simplify the complexity to $O(n)$ which is linearly dependent to n.

B. Energy Analysis

The energy consumption of classification algorithm depends on calculations made on each video frame for the classification process. In our algorithm, in order to calculate the similarity between the query object and each of the prototype objects in the classifier model, we perform Euclidian-distance calculation for each feature and it requires $O(mn)$ simple subtractions where m refers the feature count and n refers to number of prototypes. In order to calculate the total weight for the query object, we perform $O(mn)$ additions and multiplications. Then, for each class, we perform $O(1)$ division to find decision value for the prototype, $O(1)$ addition to find effectiveness for the prototype, and $O(1)$ multiplication and addition to find the total sum for the prototype. Lastly we perform $O(1)$ more division to find the decision of the class for the query object. Hence, if we have c classes, totally we perform $O(cmn)$ subtractions, $O(cmn + c)$ additions, $O(cmn + c)$ multiplications and $O(c)$ divisions. Since typically c is small value, total operation count is linearly dependent on mn. Therefore, total energy cost at each camera sensor for data processing would be the cost of performing $O(mn)$ multiplications, additions and subtractions. Hence, we can again simplify the complexity to $O(n)$ which is linearly dependent to n.

C. Storage Analysis

As stated in Section IV-B1, the classifier model contains two models: CSF model and the genetic algorithm model. CSF model includes the weights of each feature for each class. Thus, the complexity of the function for calculating the bytes needed for storage of CSF is $O(mc)$, where m and c denotes the number of features and classes respectively. Besides, genetic algorithm model holds features of each prototype and weights of the prototypes. As a result, the complexity of the storage required for the genetic algorithm model is $O(n(m + 1))$.

The overall complexity of the storage needed to store the classifier is $O(mc + n(m + 1))$ which is linearly dependent on the number of features, classes and prototypes. Since n is larger compared to c and m values (i.e., generally c and m are constants), we can simplify the complexity to $O(n)$. If we emphasize having limited storage on camera nodes, linear dependency is the best we can achieve for that purpose.

VI. Experimental Evaluation

A. Experiment Setup and Performance Metrics

For the experiments, we assume a power plant surveillance application scenario. In this scenario, when an intrusion occurs in the area under surveillance, the detected objects are classified at the camera sensors. The classification is performed as a multi-class choice with three classes: $Human$, $Vehicle$ and $Animal$. For the camera sensor experiment data, the Caltech 101 image dataset [28] (for $Vehicle$ and $Animal$ classes) and search results from Google Image Search (for $Human$ class) are used by formatting them into the CmuCam3 [25] output format. The CalTech101 dataset does not contain $Vehicle$ and $Animal$ classes. Therefore, images from several different classes in Caltech 101 dataset are re-grouped according to these classes. The dataset is divided into three sets: First-Training, Second-Training and Test. The number of images is determined as 10 for each class in each of the training sets and 20 for each class in the test set.

The $Shape\_Ratio$ feature of each sample is extracted by simply dividing the width of the image by the height value. Besides, $Speed$ values are randomly generated, considering that it is not possible to have speed values in an image dataset. For the random generation, speed values between $[1, 10]$, $[5, 25]$ and $[10, 100]$ meter/sec are used for classes $Human$, $Animal$, $Vehicle$, respectively.

As mentioned in Section IV-B, the CSF mechanism [9] is applied in order to find representative and discriminative features for each object class. CSF mechanism gives weights of each feature for each class. Acquired weights are given in Table I. According to these weights, it has been observed that $Speed$ is the dominant feature for all classes. However, the effect of it is more for $Animal$ class than the other two classes.

TABLE I
CSF Weights

	$Shape\_Ratio$	$Speed$
$Human$	0.371051	0.628949
$Vehicle$	0.342217	0.657783
$Animal$	0.130904	0.869096

We have considered three metrics to assess the performance of the classification:

- *Classification Accuracy*: This metric shows the accuracy in estimating the class of the intruder.
- *Energy Overhead*: This constitutes total energy in processing and transmitting the frames (if needed). Our goal is to minimize this overhead in order to maximize the lifetime of the cameras.
- *Occupied Space*: This metric shows the required space for classifier model at the camera sensor. The occupied

should be minimum, considering that the sensor has a limited memory.

B. Performance Results

1) Classification Accuracy: Under given test setup, the classification results in Table II and Table III are obtained. As mentioned in Section IV-B2, the classifier performs multiple labeling by providing fuzzy membership values in the range [0,1] for each class. Thus, in order to measure the precision values, the class with the highest membership value is taken as the classification result. The system performs a high accuracy ratio in which only a total of 1 instance is classified wrong among the 60 test instances. The only instance is a $Vehicle$ instance and classified as $Animal$ due to its very low $Speed$ value. As seen in Table I, $Speed$ is a more effective feature for $Animal$ class. Thus, having $Speed$ values very close to those of the $Animal$ class caused such a classification.

TABLE II
CONFUSION MATRIX FOR GA-BASED CLASSIFICATION

		Prediction		
		$Human$	$Vehicle$	$Animal$
Actual	$Human$	20	0	0
	$Vehicle$	0	19	1
	$Animal$	0	0	20

TABLE III
CLASS ACCURACIES FOR GA-BASED CLASSIFICATION

Class	**Accuracy**
$Human$	1.00
$Vehicle$	0.95
$Animal$	1.00
Total	**0.98**

In addition, we compared our classification results with a baseline approach which uses a fuzzy membership set defined by an expert user to do the classification. The results shown in Table IV and Table V indicate that overall classification accuracy is increased to 0.98 by using genetic algorithm based classifier from 0.92, where an expert user prepares fuzzy membership functions to be used in classification process. In addition, the whole classification process is automated and human intervention is removed.

Finally, we repeated the same experiments by introducing a noise of 20% to the randomly-generated $Shape\_Ratio$ & $Speed$ values that may happen due to errors in localization or obstacles in the environment. We observe that the results do not change as the proposed approach can handle such errors perfectly.

2) Energy Overhead: We also performed experiments to assess the energy consumption of our approach at camera sensors. We used the AVR Simulation and Analysis Framework (AVRORA) to calculate energy costs [29] at camera sensors. AVRORA is an emulator which can provide realistic results as if the approach is run on a typical CMOS sensor. It has built in functions that can compute the processing and communication costs.

We used a baseline approach which processes the frames at a base-station and determines the location and classification

TABLE IV
CONFUSION MATRIX FOR FUZZY-MEMBERSHIP-BASED CLASSIFICATION

		Prediction		
		$Human$	$Vehicle$	$Animal$
Actual	$Human$	20	0	0
	$Vehicle$	0	19	1
	$Animal$	1	3	16

TABLE V
CLASS ACCURACIES FOR FUZZY-MEMBERSHIP-BASED CLASSIFICATION

Class	**Accuracy**
$Human$	1.00
$Vehicle$	0.95
$Animal$	0.80
Total	**0.92**

of the objects at the base-station. In that case, the frames travel through multiple hops (i.e., k) to reach the base station. This approach is referred to as '*Traditional Method*' in the graphs. Note that our approach performs the localization and classification on site and does not send any data to the base-station. It only sends alarms whenever needed. The results are given in Table VI and Table VII.

TABLE VI
ENERGY COSTS FOR DIFFERENT TASKS IN BOTH APPROACHES

Task	**Cost in Joule**
C: One-time CPU cost to process the frame to extract and classify the moving object (our approach)	0.0220
M: Transmission cost of the whole frame for 1 hop (traditional approach)	0.0700
T: Transmission cost of the alarm for 1 hop (our approach)	0.0007
Taking the video data	Same for both cases

TABLE VII
TOTAL ENERGY COST COMPARISON OF BOTH APPROACHES (IN JOULES)

Process	**Traditional Method**	**Proposed Method**
For 1 Hop	$M = 0.0700$	$C + T = 0.0227$
For k Hops	$M * k = 0.0700 * k$	$C + T * k =$ 0.022 + 0.0007*k

The results for varying k (*Hop Count*) values are depicted in Figure 2. As can be seen from this figure, energy overhead for our approach is constant and significantly less than the traditional method. We would like to note that, in this experiment, every moving object detection event is sent as an alarm to the sink. If we define some alarm criteria for the proposed method, the energy consumption would be further reduced (i.e., alarms are only sent when needed).

3) Occupied Space on Camera Sensors: As mentioned in Section V-C, the sensor holds two types of models for classifier. The actual required space for each of these in our experiments are given in Table VIII. As can be seen, the space requirements are not significant compared to the available space on CMUCam3 sensors (e.g., 128MB).

VII. CONCLUSION

To realize the remote surveillance with WMSNs, the detected objects should be accurately classified at camera nodes

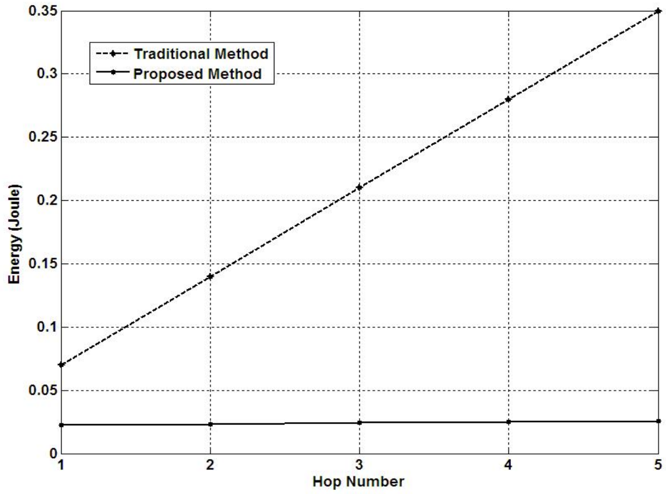

Fig. 2. Energy Costs of Two Different Methods

TABLE VIII
SPACE OCCUPIED BY CLASSIFIER MODELS ON THE SENSOR

Model	Occupied Space
CSF Model	$24 Bytes$
GA Model	$360 Bytes$

with limited processing, so that the lifetime of the network can be extended. In this paper, we present a lightweight and accurate object classification approach which can work on-site at a camera sensor with limited information. The approach utilizes genetic algorithm to built classifiers on top of two simple but effective features, which are the $Shape\_Ratio$ & $Speed$ of the detected objects.

The experimental evaluation reveals that, our approach can effectively classify typical objects, i.e., human, animal and vehicles, in a power-plant surveillance application with an error rate of at most 2% overall. We also assess the energy overhead of our approach on the individual camera sensors. The energy consumption is significantly reduced compared to the cases where the classification is performed at the base-station due to reduction in the communication energy cost.

In the future, we plan to increase the number of features used in classification to further improve the accuracy and be able to work with more classes. Additionally, we will also investigate online training opportunities for the approach.

REFERENCES

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks Elsevier*, vol. 51, no. 4, pp. 921–960, March 2007.

[2] M. Rahimi, R. Baer, O. Iroezi, J. Garcia, J. Warrior, and M. Srivastava, "Cyclops: in situ image sensing and interpretation in wireless sensor networks," *Proc. 3rd ACM Conf. Embedded Networked Sensor Systems (SenSys 2005)*, pp. 193–204, 2006.

[3] S. H. et al., "Mesheye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," Cambridge, MA, April 2007.

[4] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia Hindawi Journal*, 2009.

[5] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age," *MIR '05: ACM Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 253–262, 2005.

[6] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, 2000.

[7] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, 2000.

[8] T. Yilmaz, Y. Yildirim, and A. Yazici, "A genetic algorithms based classifier for object classification in images," in *Computer and Information Sciences II*, E. Gelenbe, R. Lent, and G. Sakellari, Eds. Springer London, 2012, pp. 519–525, 10.1007/978-1-4471-2155-8_66. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-2155-8_66

[9] T. Yilmaz, A. Yazici, and Y. Yildirim, "Exploiting class-specific features in multi-feature dissimilarity space for efficient querying of images," in *FQAS*, ser. Lecture Notes in Computer Science, H. Christiansen, G. D. Tré, A. Yazici, S. Zadrozny, T. Andreasen, and H. L. Larsen, Eds., vol. 7022. Springer, 2011, pp. 149–161.

[10] K. Akkaya, M. Demirbas, and R. S. Aygun, "The impact of data aggregation on the performance of wireless sensor networks," *Wireless Comm. & Mobile Computing Journal*, vol. 8, pp. 171–193, 2008.

[11] A. Newell and K. Akkaya, "Actuation of camera sensor fro redundant data elimination in wireless multimedia sensor networks," *IEEE International Conference on Communications (ICC'09)*, to appear.

[12] U. M. Erdem and S. Sclaroff, "Automated camera layout to satisfy task specific and floor plan-specific coverage requirements." *Computer Vision and Image Understanding*, pp. 156–169, 2006.

[13] J. Kassebaum, N. Bulusu, and W. Feng, "Smart camera network localization using a 3d target," *Proceedings of the Workshop on Applications, Systems & Algorithms for Image Sensing*, 2008.

[14] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, and L. Iftode, "Indoor localization using camera phones," *Proceedings of 7th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'06)*, 2005.

[15] L. Liu, X. Zhang, and H. Ma, "Dynamic node collaboration for mobile target tracking in wireless camera sensor networks," *INFOCOM 2009, IEEE*, pp. 1188–1196, June 2009.

[16] C. Arth, C. Leistner, and H. Bishof, "Object reacquisition and tracking in large-scale smart camera networks," *Proc. of 1st AC/IEEE Int. Conf. on Smart Distributed Cameras*, 2007.

[17] H. Öztarak, K. Akkaya, and A. Yazıcı, "Lightweight object localization with a single camera in wireless multimedia sensor networks," *Proceedings of the GLOBECOM*, pp. 12–13, 2009.

[18] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proceedings of the*, pp. 1163–1171, 2003.

[19] "Machine learning , neural and statistical classification," *Technometrics*, vol. 37, no. 4. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.3615

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2001.

[21] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada, "Color and texture descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 703 –715, jun 2001.

[22] M. Bober, "Mpeg-7 visual shape descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 716 –719, jun 2001.

[23] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, pp. 346–359, June 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1370312.1370556

[25] A. Rowe, A. Goode, D. Goel, and I. Nourbakhsh, "Cmucam3: an open programmable embedded vision sensor," Carnegie Mellon Robotics Institute, Technical Report, RI-TR-07-13, 2007.

[26] "Cmucam3," January 2012. [Online]. Available: http://cmucam.org/wiki

[27] N. Patwari and et. al., "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54–69, 2005.

[28] P. P. L. Fei-Fei, R. Fergus, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, p. 594, 2006.

[29] "Avrora," October 2011. [Online]. Available: http://compilers.cs.ucla.edu/avrora/