AN UNSUPERVISED ANOMALOUS EVENT DETECTION FRAMEWORK WITH CLASS AWARE SOURCE SEPARATION

Burhan A. Mudassar *Jong Hwan Ko* *Saibal Mukhopadhyay*

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA
{burhan.mudassar, jonghwan.ko}@gatech.edu, saibal.mukhopadhyay@ece.gatech.edu

ABSTRACT

This paper presents a novel problem of detection and localization of anomalous events due to a certain class of objects in video data with applications to smart surveillance. A baseline system is proposed that uses a convolutional neural network (CNN) to generate pixel level masks corresponding to objects of a class of interest. A Restricted Boltzmann Machine (RBM) is then trained on the mask to learn patterns of normal behavior. The free energy of the RBM is used to detect the presence of an anomaly while the reconstruction error is used to localize the anomaly. Our approach is scalable to a low power and energy constrained setting with 1930.48 ms of latency and 4826 mJ energy consumed per frame on a mGPU.

Index Terms— Intelligent Video Surveillance, Anomaly Detection, IoT, RBM, CNN

1. INTRODUCTION

Increasing number of IoTs (Internet of Things) has led to a sharp increase in the amount of data generating nodes thus increasing the burden of processing at the host. Consider for example a surveillance camera network which can consist of multiple cameras transmitting video feeds to the host. It is impossible for a human operator to monitor each of these feeds in an efficient manner. This not only poses a security risk but also stretches the transmission bandwidth to its limit.

The end goal of intelligent video surveillance is to detect interesting/anomalous parts of the video. Automated anomalous event detection can be applied to detect and send important parts of a video [1]. For example, a car can be categorized as anomalous if its stopped on a highway full of regular moving traffic.

Existing video anomaly detection frameworks have two limitations. First, they adopt a semi-supervised training approach to the problem which requires generating a dataset of normal events. This does not scale well to multiple sensor nodes due to scene variations and complexity of the training process. For example, Xu et. al train deep autoencoders using 50 million image patches [2]. Second, they lack the capability of separation of anomalies based on objects of a certain class.

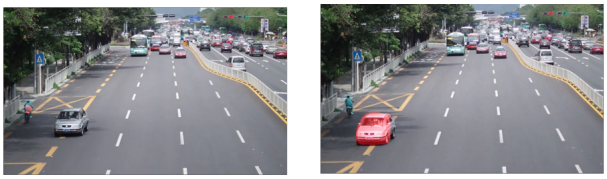


Fig. 1: Separation of Anomalous Events based on Target Class of Interest

This capability can be motivated by the fact that the focus of the observer depends highly on the scene and may shift as the scene changes e.g. a traffic camera on a highway needs to focus only on the vehicles. Another example is presented in Fig 1.

We solve the above-mentioned problems by developing an anomalous event detection framework that is (a) trainable in an unsupervised manner to account for scene variabilities and scene progression, (b) has low train and test complexity allowing an edge level implementation and (c) provides reconfigurable target class based separability. Our system is composed of a pre-trained segmentation CNN that generates pixel level masks for objects of a certain class. These masks are then used to train a RBM, an energy based model. The energy of the RBM is used to detect the presence of anomalies.

We validate our approach on some anomaly detection datasets. On the UCSD Ped 2 Dataset we can detect 11 out of the 12 anomalies present and are able to perform source separation based on target class. On a mobile GPU we achieve 1930.48/4826 ms/mJ (CNN + RBM) of inference latency/energy and 8.95/22 ms/mJ of train latency/energy (RBM only). We validate the unsupervised training capability of our system in a qualitative manner using examples from the AVSS 07 Parked Vehicle Dataset and the UMN Dataset.

Acknowledgement: This work was supported by the *Office of Naval Research Young Investigator Program (2012)*

2. RELATED WORK

In prior work on anomaly detection, the processing pipeline is divided into a feature extraction module followed by clustering or reconstruction based on the extracted features. Semi-supervised approaches have been the most popular due to the rare nature of abnormal data. In a semi-supervised system, a training set of normal examples are provided, and a model is learned. Any examples that deviate from the learned model are classified as anomalies.

For feature extraction, hand-crafted features such as HOG [3], HOF [3], 3D gradients [4] etc. have been the norm until the rise of deep learning based feature extraction methods such as autoencoders [2, 5, 6], LSTMs [5], 3D-CNN [7]. For anomaly detection, clustering based approaches construct a decision boundary such as one-class SVMs [2]. Reconstruction cost based approaches include LSTMs [5], autoencoders [6] and sparse coding based approaches [3, 8, 9, 4].

3. SYSTEM ARCHITECTURE

We propose a system that can separate out the target class of choice and learn the activity pattern of that specific class. An object segmentation/detection module (CNN) is used to generate pixel level masks/bounding boxes for a class of interest. The class-specific mask is then fed into a RBM which generates a scalar value called the free energy score.

The free energy is used to determine whether an anomalous event has occurred based on past examples. In addition to the free energy, the RBM is also used to generate a reconstruction. A combination of both measures is used to detect and localize anomalies. The system block diagram is presented in Fig 2.

***CNN*:** We use a segmentation CNN from [10] with a VGG-16 backbone [11]. The CNN is pre-trained on ImageNet and then fine-tuned on the PASCAL VOC/MS COCO segmentation dataset. Fully convolutional layers generate a score map corresponding to the position of classes of interest. The score map is up-sampled and combined with the output of the pooling layers to generate a segmentation map equal in dimensions to the input image.

***RBM*:** The Restricted Boltzmann Machine is a generative stochastic neural network that is described by an energy function and can be represented in the form of a bi-partite graph with undirected edges and no intra-layer connections. For an input v and hidden vector h, the energy of the RBM is given by (1) where W is the weight matrix, b is the hidden layer bias and c is the visible layer bias.

$$E(v,h) = -c^T v - b^T h - v^T W h \tag{1}$$

The joint probability of the configuration can be expressed in terms of the energy function by (2) where Z is the partition function. As Z is the sum of the energies of all possible configurations of the RBM, it is intractable to compute.

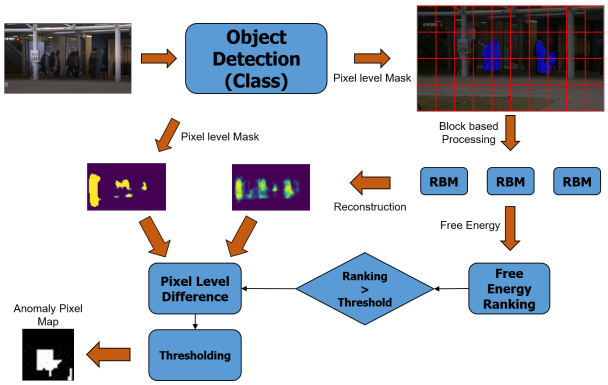


Fig. 2: System Level Block Diagram for Class-Aware Anomaly Detection

$$P(v,h) = \frac{e^{-E(v,h)}}{Z} \tag{2}$$

The marginal probability of the visible layer vector can be found by summing over all the possible configurations of the hidden layer vector. Since there are no intra-layer connections, the conditional probability of the hidden layer given the visible layer and vice versa factorizes into a product of the individual probabilities of the neurons. For a binary-binary RBM, the conditional probabilities are expressed by (3) and (4) where σ is the sigmoid function. The free energy for the binary case is then expressed by (5).

$$P(h = 1 \mid v) = \sigma(Wv + b) \tag{3}$$

$$P(v = 1 \mid h) = \sigma(W^T h + c) \tag{4}$$

$$F(v) = -c^T v - \sum_h log(1 + e^{Wv+b}) \tag{5}$$

The RBM is trained to maximize the likelihood of the parameters for a given set of data. This is analogous to increasing the energy of the model for unseen examples and decreasing the energy of the model for seen examples. Since Z is intractable, approximations to the likelihood are needed. A fast approximate algorithm, Contrastive Divergence (CD), for training the RBM is described by Hinton et. al [12].

In CD-1, the visible units are initialized to the training examples. A forward pass generates activation values for the hidden units. The activation values are then used to sample from a Bernoulli distribution. A backward pass on the hidden layer states generates the reconstruction. The loss function is defined as the differences in energies of the input and reconstruction.

At test time, the energies are ranked and compared against past values. If the free energy is high, then the sample is anomalous. A pixel-level difference of the reconstruction and the input sample is taken to localize the anomalous object.

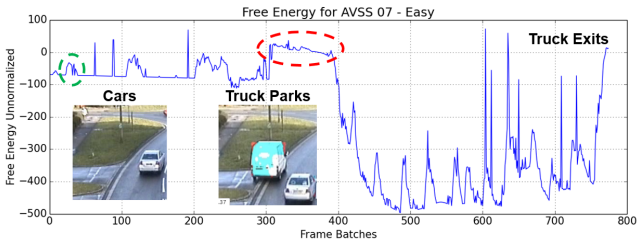


Fig. 3: Free Energy Progression for AVSS 07 PV Dataset. The free energy of the system rises when the truck parks but gradually becomes lower as the truck stays there indicating evolving behavior

Dataset	Anomaly Detection Task		
Name	No. Events	Correct Detection/ False Alarm	
		Ours	State of the Art
UCSD Ped 1	40	20/4	38/6 [6]
UCSD Ped 2	12	11/1	12/1 [6]

Table 1: Detection Results on the UCSD Ped Dataset [13]

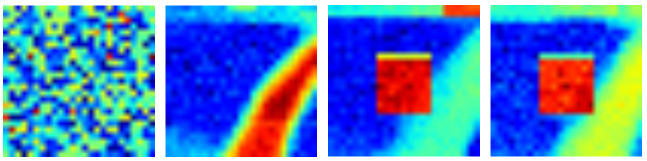
***Semi-Supervised Anomaly Detection*:** In semi-supervised anomaly detection, we are provided with a set of normal examples even though the examples are not annotated. In this case, we train the RBM in an offline manner with examples from the training data. For the test examples we do not update the weights of the RBM.

***Unsupervised Anomaly Detection*:** In a real-world case, training data may not be available for every scene and thus a framework is desired which can evolve with the dynamics of the scene as it progresses. Thus, we propose a streaming algorithm where the RBM is trained with examples as they are sampled and the free energy of the RBM is queried to detect the presence of anomalies.

An example is shown in Fig. 3 where we apply the algorithm to a short, cropped video sequence from the AVSS 07 Parked Vehicle Dataset [14]. The evolution of the free energy is shown as the scene progress. At first, only cars are present followed by a truck that parks on the side. The system first detects the truck as an anomaly but soon learns to model it as a normal behavior. In Fig. 4 the evolution of the weight matrix for one of the RBM neurons is shown. In Fig. 8 we apply the algorithm to the UMN Dataset. The free energy decreases gradually but rises as the panic event happens.

4. EXPERIMENTS

Verification of the semi-supervised algorithm is performed on two real-world datasets i.e. UCSD and Avenue Dataset. The streaming algorithm is applied on the UMN Dataset and the AVSS 07 Dataset. For the semi-supervised version, each RBM is trained for 30 epochs with a learning rate of 0.01 and with a minibatch size of 20. For the unsupervised version, the RBM is trained in an online manner with a learning rate of 0.001 and minibatch size of 5.



(a) Frame 0 (b) Frame 200 (c) Frame 400 (d) Frame 750

Fig. 4: Visualization of filter 0 (a) Random Initialization (b) After frame 200 normal behavior is cars on road (c) After frame 400 truck parked on side is normal behavior (d) After frame 750 as truck exits, strength of road region increases

Dataset	Anomaly Detection	
Name	AUC/EER	
	Ours	State of the Art
CUHK Avenue	77.26/26.9	70.2/25.1 [6]

Table 2: Detection Results on Avenue Dataset [4]

4.1. UCSD Dataset

The UCSD Dataset [13] consists of grayscale videos of a typical pedestrian scene. The dataset is divided into two sets Ped 1 and Ped 2 with their respective train and test splits. Ped 1 contains 34 training clips of 200 frames each with spatial resolution 158 x 238. Ped 2 contains 16 training clips at a resolution of 240 x 360. The test set contains labelled anomalies due to non-pedestrians such as bicycles, skaters, etc. and some pedestrian behaviors such as loitering and uncommon paths.

4 separate RBM networks are created for classes of interest, i.e. person, bicycle, car and skateboard. Performance is affected on Ped 1 due to poor detection performance on objects of bicycle class. Detection results are presented in Table 1. Some detected anomalies are shown in Fig. 6 and 7.

4.2. Avenue Dataset

The Avenue Dataset [4] consists of 16 clips for training and 21 clips for testing. Anomalies in this dataset include loitering, irregular motion, irregular actions such as throwing etc. Some detected anomalies on the Avenue Dataset are presented in Fig. 5. Detection results are presented in Table 2. We achieve an Equal Error Rate (EER) of 26.9% with 77.6 % Area Under the Curve (AUC).

4.3. Class-Aware Separability

Depending on the class of interest, different anomalous events are reported. In Fig. 6 and Fig. 7 we show class aware separability on the UCSD Dataset with separation of anomalies produced due to the bicycle, cart and person class.

(a) (b) (c) (d)

Fig. 5: Detected Anomalies on the Avenue Dataset (a) Wrong Direction (b) Loitering (c)(d) False Alarm in GT but detected by our framework as anomaly (Anomalies highlighted in red. Best viewed in Color)

(a) (b) (c) (d)

Fig. 6: Detected Anomalies on UCSD Ped 1 (a) GT (b) SDAE+LSTM [5] (c) Our Framework (Bicycle) (d) Our Framework (Person)

(a) (b)

Fig. 7: Detected anomalies on the UCSD Ped 2 Dataset due to (a) bicycle (b) cart

Module	No. Ops (MFLOPS)		
	Feed-forward	Back-Propagation	Total
CNN (VGG16)	15484	-	15484
RBM (625 → 800) x18	36	36	72

Table 3: Number of Operations per Module

Platform	Inference Latency (ms)	Inference Energy (mJ)	Train Latency (ms)	Train Energy (mJ)
ARM T760	1930.48	4826	8.96	22
EyeRiss	611.63	170	2.84	1
Tegra X1 (FP16)	319.51	1629	1.48	8

Table 4: Latency and Energy Consumption on Different Platforms

4.4. Latency and Energy Consumption

Latency and energy consumption of our framework was calculated by estimating the number of operations per module. Projections are made using reported operational efficiency of

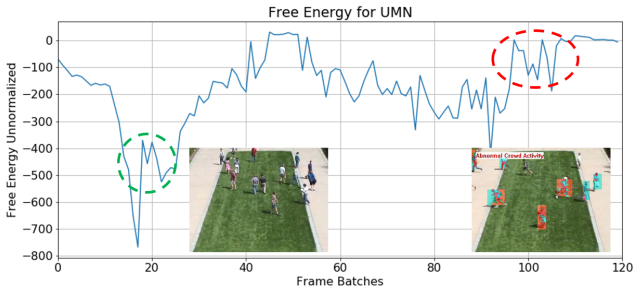


Fig. 8: Evolution of free energy on scene 1 of the UMN Dataset

3 energy efficient platforms i.e. EyeRiss [15], Tegra X1 and mGPU [16]. The estimated operations per module are presented in Table 3. The VGG-16 CNN requires 15.5 GFLOPs for one forward pass over a 224 x 224 image. 18 RBMs with $(625 \rightarrow 800)$ neurons each are used for block processing of CNN output. Training each RBM consists of 2 forward passes and one backward pass. Latency and energy numbers are presented in Table 4. Only the RBM is updated online hence the lower numbers for training.

5. CONCLUSION

We have presented an anomaly detection pipeline that is able to detect and localize class-specific anomalies and can be trained in an unsupervised manner. Our approach is scalable to a low power setting making it ideal for deployment in an IoT. For future work, we intend to quantitatively analyze the effectiveness of our unsupervised algorithm.

6. REFERENCES

[1] Oluwatoyin P Popoola and Kejun Wang, "Video-based abnormal human behavior recognition - A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 865–878, 2012.

[2] D Xu, J Song, Y Yan, E Ricci, N Sebe, et al., "Learning deep representations of appearance and motion for anomalous event detection," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[3] Bin Zhao, Li Fei-Fei, and Eric P Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3313–3320.

[4] Cewu Lu, Jianping Shi, and Jiaya Jia, "Abnormal event detection at 150 fps in MATLAB," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2720–2727.

[5] Yachuang Feng, Yuan Yuan, and Xiaoqiang Lu, "Deep representation for abnormal event detection in crowded scenes," in *Proceedings of the ACM Multimedia Conference*, 2016, pp. 591–595.

[6] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis, "Learning temporal regularity in video sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 733–742.

[7] Shifu Zhou, Wei Shen, Dan Zeng, Mei Fang, Yuanwang Wei, and Zhijiang Zhang, "Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Signal Processing: Image Communication*, vol. 47, pp. 358–368, 2016.

[8] Yang Cong, Junsong Yuan, and Ji Liu, "Sparse reconstruction cost for abnormal event detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3449–3456.

[9] Jayanta K Dutta, Bonny Banerjee, and Chandan K Reddy, "Rods: Rarity based outlier detection in a sparse coding framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 483–495, 2016.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[11] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[12] Geoffrey E Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[13] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos, "Anomaly detection in crowded scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1975–1981.

[14] "i-lids dataset for avss 2007," ftp://motinas.elec.qmul.ac.uk/pub/iLids, Accessed: 2017-10-27.

[15] Yu-Hsin Chen, Tushar Krishna, Joel S Emer, and Vivienne Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2017.

[16] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.