Smart Surveillance Tool

Anirudh RV
Undergrad Student
Computer Science and Engineering
BMS College of Engineering
anirudh.rv1234@gmail.com

Brundha R
Undergrad Student
Computer Science and Engineering
BMS College of Engineering
brundha.r.reddy@gmail.com

Eleanor Prashamshini
Undergrad Student
Computer Science and Engineering
BMS College of Engineering
prashamshini@gmail.com

Myna P
Undergrad Student
Computer Science and Engineering
BMS College of Engineering
myna.pk3@gmail.com

Dr. Jyothi S Nayak
Professor
Computer Science and Engineering
BMS College of Engineering
jyothinayak.cse@bmsce.ac.in

***Abstract*— Most video surveillance systems currently available require a human to monitor the footage 24/7 in order to discover abnormalities and crisis situations; this is labour intensive and inefficient. Through this survey, viable technologies and methods that can augment the existing surveillance system will be discussed. There are three topics primarily discussed in this paper - (a) anomaly detection, (b) object detection and (c) deployment and scalability.**

Keywords— surveillance system, anomaly detection, object detection

I. Introduction

Video surveillance is a widely adopted system and continues to be a prominent crisis detection methodology. Due to the requirement of constant human monitoring, it is prone to suffer from inefficiency and is viewed to be labour intensive approach to surveillance. An automated version of surveillance has a demand in the commercial, law enforcement and military applications. This paper aims to discuss a surveillance tool that needs minimal human monitoring to raise alerts about atypical behaviour and can be easily deployed.

The prime focus of this paper is discussing methodologies for object detection, and understanding how anomalies can be detected. An efficient deployment and scalability strategy has also been discussed to create an overview of an end-to-end product.

The observed disadvantages associated with the existing surveillance system (shown in Table 1).

Table 1. Advantages and disadvantages of existing surveillance systems

Serial Number	Existing Systems	Advantages	Disadvantages
1	Simple video based Surveillance	Easy to install and operate	Requires human monitoring
2	IoT based Surveillance Systems[7]	Comprehensive and customizable	Difficult to scale and requires multiple sensors
3	Military object recognition systems	Highly trained and specialised functions	Access restricted for public and research needs

II. Literature survey

A. *Object Detection Background*

The Viola Jones Object Detector[1] was arguably the first object detector. It was released in 2001 and was designed for real-time facial detection. With the advent of Deep Learning, models like Overfeat Network[2] came into the picture. They mostly used Convolutional Neural Networks (CNNs) along with a sliding window approach to classify each part of an image as an object or non-object and then combined the results.

The time taken to process and accuracy of image processing has reached 30ms and 43mAP(mixed average precision) from 47s and 29mAP. Object Detector design has improved enough to allow them to run on low-memory low-power devices like Mobile Phones now.

B. *Object Detection Models: Two step models*

The first step, Region Proposal step[3], generates regions in the image that have a high probability of being an object. The second step, Object Detection step[3], takes the generated regions as inputs and preforms classification. Two step models generally beat their one step counterparts in accuracy.

a) *CNN*: A CNN for object detection passes a 2D array of pixel RGB values through 5 convolutional layers. It applies filter to get output feature maps. Filters are essentially preset matrices that are multiplied to each section of the 2D array, using a sliding window approach. ReLU[4] is used to perform multiple convolutions in parallel.
CNN is not suitable for detecting multiple objects in a single image. It is also unable to detect overlapping objects or recognise images from different backgrounds.

b) *R-CNN:* The R-CNN[5] model here uses Selective Search[6]. It finds a large number of regions that most probably contain an image, resizes these to 7x7 pixel images and feeds them to the Object Detector Model. Around 2000 region proposals are generated and execution time is about 27 seconds. The Object Detector Model takes the 7x7 pixel images as input and classifies objects by calculating Intersection Over Union (IOU) with respect to the ground truth. Then, a list of class probabilities and offset coordinates for each region is generated and the vectors are run through a pre-trained AlexNet[7]. Finally, the vector is run through 2 SVM networks[8] - the classification network head and the regression network head. The classification head is responsible for predicting the class of object that the region belongs to and the regression head is responsible for predicting the offsets to the coordinates of the region to better fit the object.

It takes about 47 seconds to process a single image and it needs hand engineering to tweak certain parameters. Training is a cumbersome process here and thus, this model was not chosen.

c) Fast R-CNN: There are popular claims that Fast RCNN 9 times faster than previous R-CNN[9]. Fast RCNN also reduces the memory footprint as features no longer need to be stored on disk.

Fast RCNN reduces the effort of running each region proposal though a convolutional base by running the convolutional base over the entire image just once to generate a feature map. Regions are taken from the so generated feature map instead of the image. This feature vector cropping is done via Region of Interest (ROI) pooling[10].

Also, single step training pipeline (where the classification and regression subnetworks can be trained together) and multitask loss were introduced to reduce training time and memory. The object detector model in the Fast RCNN is very similar to the detector used in the RCNN. The loss function used here is easier to train and does not suffer gradient explosion problem.

The region proposal step slowed down the entire model and created a bottleneck.

d) Faster R-CNN: The Faster RCNN introduced the Region Proposal Network (RPN)[11] to replace Selective Search. The RPN needed to have the capability of predicting regions of multiple scales. RPN uses anchors which are basically rectangular crops of an image. Various combinations of scales and aspect ratios are used to crop anchors in a sliding window fashion.

The rectangular offsets are calculated in a similar fashion as RCNN and anchor shape is modified as training progresses.

The architecture of the convolutional base, convolutional subnetwork and regression subnetwork is similar to that in RCNN's object detection model, the only difference is that object detection model's input is generated from RPN instead of Selective Search.

C. *Object Detection Models: One step models*

Single step models combine classification and locating into a single step and show higher speeds and memory efficiency, despite their simplicity[3].

a) *YOLO:* Yolo[12] is an object detection system targeted for real-time processing; it uses regression[13] and compacts the whole detection pipeline to one network. With Yolo, the rectangular area in which an object of a certain classification exists, along with an objectless score can be obtained. The version YOLOv3 is surveyed here. Logistic regression and multilevel classifiers with sum of squares errors and binary cross-entropy loss [15] respectively are used. YOLOv3 predicts boxes at 3 different scales by upsampling data from previous layers and concatenating it to the data at a particular level. This helps to get more semantic information from the upsampled features and finer-grained information from the earlier feature map. Then k-means clustering is used to determine bounding box priors.

Yolo is almost on par with RetinaNet and far above the SSD variants. However, performance drops significantly as the IOU threshold increases indicating YOLOv3 struggles to get the boxes perfectly aligned with the object. For this application which is real-time and involves abundant noise and varying backgrounds, YOLOv3 is a good fit.

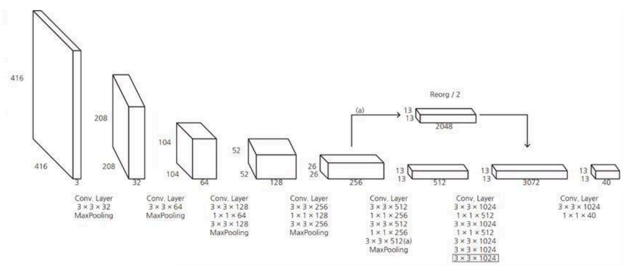
416
416
3
208
208
32
104
104
64
52
52
128
26
26
256
(a)
Reorg / 2
13
13
2048
13
13
512
13
13
3072
13
13
40
Conv. Layer 3 × 3 × 32 MaxPooling
Conv. Layer 3 × 3 × 64 MaxPooling
Conv. Layer 3 × 3 × 128 1 × 1 × 64 3 × 3 × 128 MaxPooling
Conv. Layer 3 × 3 × 256 1 × 1 × 128 3 × 3 × 256 MaxPooling
Conv. Layer 3 × 3 × 512 1 × 1 × 256 3 × 3 × 512 1 × 1 × 256 3 × 3 × 512(a) MaxPooling
Conv. Layer 3 × 3 × 1024 1 × 1 × 512 3 × 3 × 1024 1 × 1 × 512 3 × 3 × 1024 3 × 3 × 1024 3 × 3 × 1024
Conv. Layer 3 × 3 × 1024 1 × 1 × 40

Fig. 1 YOLOv3 architecture [14]

b) SSD: In Single Shot MultiBox Detector Model[16] , a single pass is sufficient for localization and classification. Anchors are used to initialise default regions and then predict classes and location offsets. A convolutional base and a multitask loss is used to train the network.

SSD uses a feature pyramid, and this is unique to it. The network uses a concept of aspect ratios and scales, each cell on the feature map generates 6 types of anchors, similar to the Faster RCNN. These anchors vary in aspect ratio and the scale is captured by the multiple feature maps, in a similar fashion as the FPN[17]. SSD uses this feature pyramid to achieve a high accuracy, while still being incredibly fast.

The SSD was rejected as it failed in more generalized object detection with noise and distractions in the background. Such scenes with heavy noise and distractions are to be expected in the video footage of surveillance systems.

c) RetinaNet: While training the above networks, class imbalance problems[18] occurred as somewhat equal number of samples for all classes was not available. RetinaNet solved the problem using Focal Loss[19] functions. Focal Loss function prevents a vast number of easy negatives from overwhelming the model.

The use of the focal loss function has empowered RetinaNet to be the first single step detector that gives better accuracy than two step detectors, while still maintaining the speed advantage of single step detectors.

Though RetinaNet can boast of high speeds and accuracy, it currently has only 75 labels and this pales in comparison to YOLO 9000 which has 9000 object class labels. Also, RetinaNet is a fairly new model with limited support available in case of usability and incompatibility with other problems.

Table 2. Average Precision for Object Detection Models

Model	Backbone	Average Precision
Two-stage models		
Fast RCNN	VGG-16	19.7
Faster RCNN	VGG- 16	21.9
One-stage models		
YOLOv3	DarkNet- 53	33.0
SSD300	VGG- 16	25.1
RetinaNet	ResNeXt-101-FPN	40.8

D. *Metric Used for Quantitative Evaluation*

This paper uses average precision to compare models trained on COCO dataset. Average precision is calculated using the top 11 predictions made by the model for an object. For results where IOU is above a threshold, a prediction is said to be correct. For results with recall values between 0 and 1, maximum precision values are calculated. Finally, all the maximum precision values are averaged. Values[20][21] of average precision for most models surveyed in this paper are presented in Table 2.

E. *Anomaly Detection - Abnormal Behaviour :*

a) Modelling frameworks and classification methods [22]: A comparison of various frameworks and methods is presented in Table 3.

b) *Scene density and moving object interaction:* A comparison of various interactions considered in this paper is presented in Table 4.

F. *Anomaly Detection - Text Recognition*

a) Textboxes++: The aim of using text recognition in anomaly detection is to identify text in images. Textboxes++ is a word-based, multi-oriented text detector. The pipeline consists of an end-to-end training of a fully convolutional neural network to detect arbitrary-oriented text. The fundamental idea was inspired by an object detection algorithm SSD[12] proposed. It is highly stable and efficient to generate word proposals against cluttered backgrounds. Using a regression model by quadrilateral representation, it can be enabled to predict bounding boxes for oriented words as well. [23].

It consists of a fully convolutional network including 13 layers from VGG-16 followed by 10 extra convolutional layers, and 6 Text-box layers connected to 6 intermediate convolutional layers. Each location of a text-box layer predicts an n-dimensional vector for each default box consisting of the text presence scores (2 dimensions), horizontal bounding rectangles offsets (4 dimensions), and rotated rectangle bounding box offsets (5 dimensions) or quadrilateral bounding box offsets (8 dimensions). A non-maximum suppression is applied during test phase to merge the results of all 6 text-box layers.

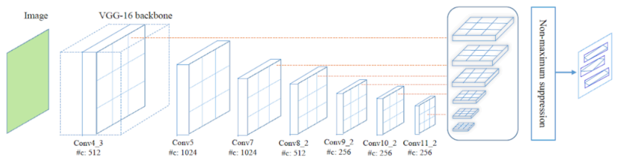

Fig: The architecture of TextBoxes++[23] (Note that "#c" stands for the number of channels.)

To help gain a comparative understanding of the performance of Textboxes to other word-spotting technologies, the performance of all the methods on the ICDAR 2015 Incidental Text dataset (IC15) were analysed. It is composed of 1000 training images and 500 testing images, which relatively low resolutions Google glasses captured. Classical evaluation measures - precision (P), recall (R), and f-measure (F) - were used to compare various models, using equations 1, 2 and 3. Results of the comparison are shown in Table 5.

$$P = TP/(TP + FN) \qquad \text{(eq. 1)}$$

$$R = TP/(TP + FN) \qquad \text{(eq. 2)}$$

$$F = 2x(PxR)/(P + R) \qquad \text{(eq. 3)}$$

b) *Tesseract:* It is an Optical Character Recognition engine that was developed at HP between 1984 and 1994. It was developed in secrecy and was rarely ever available for examination. But since it has been made open-source, the architecture and algorithms are available to us as well. It has been sponsored by Google since 2006, and has grown to become a highly accurate OCR. Its line finding algorithm is created such that a skewed page can be recognized without having to de-skew it. This in turn helps in saving loss of image quality. [24]

Table 3. Comparison of modelling frameworks and classification methods

Method	**Description**	**Advantages**	**Limitations**
Supervised	Building a model of normal and abnormal behaviours vis labeled data.	-Good results in detecting known abnormal behaviours -Easy to run and to understand	- Designed to detect only specific behaviours, e.g. fighting, loitering or falling. - Strongly depends on training data. - Unknown anomalies cannot be detected. - Learning all abnormal behaviours is not practical
Semi-supervised: rule based method	- Rule construction using normal patterns. - Any new sample that does not fit the rule is an anomaly	- Easy to perform and to interpret. - Very expressive. - Close to human reasoning.	High memory and computational complexities
Semi-supervised: model based method	- Build model representing the normal behaviours. -Any new sample that does not respect the model is an anomaly.	Model is easy to generate and to understand. - A new instance is rapidly classified.	- Sensitive to multiple parameters. - Unknown normal data can be identified as abnormal (false alarm).
Unsupervised	Learning using statistical properties extracted from unlabeled data.	- Fast and easy to perform. - No prior knowledge required.	- Result interpretation is time consuming. - Based on the assumption that abnormal behaviours are rare.

Table 4. Comparison of scene density and moving object interaction

Main Topic	**Type of interaction**	**Type of the scene**	**References**
People monitoring / Falling detection	No Interaction	Uncrowded	Thi-Lan & Thanh-Hai, 2015; Rezaee et al., 2015; Bian et al., 2015; Aslan et al., 2015; Stone & Skubic, 2015; Ye et al., 2014; Nguyen et al., 2014
Suspicious behavior detection (e.g. chasing, following)	No Interaction	Uncrowded	Takai, 2015; Chundi et al., 2015; Ouivirach et al., 2013; Arroyo et al., 2015; Huang et al., 2014; Tran et al., 2014; Wiliem et al., 2012
Loitering	No Interaction	Uncrowded	(Gomez et al., 2015) (Ko & Yoo, 2013)
Being in the wrong place	No Interaction	Uncrowded	Delgado et al., 2014
Violence in the elevator	One to One Interaction	Uncrowded	Yujie & Zengfu, 2016; Guang et al., 2014
Aggressive behaviors (kicking, punching, fighting, etc)	One to One Interaction	Crowded, Uncrowded	D´eniz et al., 2014; Arroyo et al., 2015; Bermejo et al., 2011; Gao et al., 2016
Escape panic / crowd anomaly	Group Interaction	Crowded	Wang et al., 2016a,b; Nannan et al., 2015; Yogameena & Priya, 2015; Wang & Xu, 2015; Santhiya et al., 2014; Gu et al., 2014; Wang et al., 2014; Huang & Chen, 2014; Cong et al., 2013; Hassner et al., 2012a; Ramin et al., 2009
Special areas monitoring (pedestrians walkway, subway station, roads, etc)	Group Interaction	Crowded	Chaker et al., 2017; Chen et al., 2015; Kai-Wen et al., 2015a; Cho & Kang, 2014; Alvar et al., 2014; Liu et al., 2014; Leach et al., 2014b,a; Hajananth et al., 2014; Aravinda et al., 2014; Huo et al., 2014; Li et al., 2014; Yang et al., 2013; Cho & Kang, 2012

Table 5. Average Precision for Object Detection Models

Model	**R**	**P**	**F**	**References**
CNN MSER	0.34	0.35	0.35	D. Karatzas et al., 2015
AJOU	0.47	0.47	0.47	H. I. Koo et al., 2013
NJU	0.36	0.70	0.48	D. Karatzas et al., 2015
StradVision1	0.46	0.53	0.50	D. Karatzas et al., 2015
StradVision2	0.37	0.77	0.50	D. Karatzas et al., 2015
Zhang et al.	0.43	0.71	0.54	Z. Zhang et al., 2016
Tian et al.	0.52	0.74	0.61	Z. Tian et al., 2016
Yao et al.	0.59	0.72	0.65	C. Yao et al., 2016
Liu et al	0.682	0.732	0.706	Y. Lui et al., 2017
Shi et al.	0.768	0.731	0.750	B. Shi et al., 2017
EAST PVANET2x RBOX	0.735	0.836	0.782	X. Zhou et al., 2017
EAST PVANET2x RBOX MS	0.783	0.833	0.807	X. Zhou et al., 2017
TextBoxes++	0.767	0.872	0.817	

III. Implementation

A. *Architecture Style*

The architecture is broadly classified into two types, namely - Monolithic and Microservice Architecture.

a) *Monolithic Architecture:* Monolithic architecture is equivalent to a big container wherein all the software components of an application are assembled together and tightly packaged.
Disadvantages of this architecture include that it is not easily scalable and it blocks continuous development. Thus, many features of the application cannot be built and deployed at the same time.

b) *Microservice Architecture:* In microservice architecture, each service is self-contained and implements a single business capability. Microservices can be implemented as fully independent deployable services.
Microservices support agile development and thus, new feature can be quickly developed and discarded again. Also, services within a system are largely decoupled, so the application as a whole can be easily built, altered, and scaled.

Thus, due to the scalability of microservices, we have opted for a microservice based architectural style.

B. Deployment

The following is proposed to be used for deployment:

a) *Dockerization:* The complete application will be containerised and deployed on Docker. This will enable total automation. Dockerizing the application will make it highly scalable and ensures high availability.

b) AWS Technologies to be used:

i) *AWS Simple Storage Service (S3):* Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.
The React JS web application, along with all the static images, will be deployed in separate S3 buckets. One can store an infinite amount of data in a bucket.

ii) *AWS Elastic Container Service(ECS):* Docker is to be run in a container on AWS ECS.
Legacy Linux or Windows applications can be migrated from on-premise to the cloud and run as containerized applications using Amazon ECS.
Thus, the entire system will extensively use AWS cloud technology for elastic scalability and fault tolerance.

IV. Conclusion

From the outline of this project, it can be concluded that surveillance is a segment that can be automated to achieve great efficiency and accuracy. This would reduce human labour and provide an automated service. This would save the effort put into employing a human workforce. Also, the proposed architecture ensures that the system is parallel and scalable.

IV. Acknowledgment

We would like to thank Dr. Jyothi S Nayak for her excellent guidance and constant support. We are also grateful to BMS College Of Engineering for providing all the facilities and the materials needed for the completion of this survey paper.

IV. References

1. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization, and detection using convolutional networks. In ICLR, 2014.
2. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In CVPR, 2001.
3. Chahal, Karanbir Singh, and Kuntal Dey. "A survey of modern object detection literature using deep learning." arXiv preprint arXiv:1808.07256 (2018).
4. Agarap, Abien Fred. "Deep learning using rectified linear units (relu)." arXiv preprint arXiv:1803.08375 (2018).
5.] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014
6. J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. IJCV, 2013
7. Zheng-Wu Yuan and Jun Zhang "Feature extraction and image retrieval based on AlexNet", Proc. SPIE 10033, Eighth International Conference on Digital Image Processing (ICDIP 2016), 100330E (29 August 2016)
8. Reyna, Roberto A., et al. "Implementation of the SVM neural network generalization function for image processing." Proceedings Fifth IEEE International Workshop on Computer Architectures for Machine Perception. IEEE, 2000.
9. Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.
10. Qin, Yuanyuan, et al. "RoI pooling based fast multi-domain convolutional neural networks for visual tracking." 2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016). Atlantis Press, 2016.
11. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.
12. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
13. Szegedy, Christian, Alexander Toshev, and Dumitru Erhan. "Deep neural networks for object detection." Advances in neural information processing systems. 2013.
14. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv: 1804.02767* (2018).
15. Zhang, Zhilu, and Mert Sabuncu. "Generalized cross entropy loss for training deep neural networks with noisy labels." *Advances in neural information processing systems*. 2018.
16. Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.
17. Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
18. Japkowicz, Nathalie. "The class imbalance problem: Significance and strategies." Proc. of the Int'l Conf. on Artificial Intelligence. 2000
19. Lin, Tsung-Yi, et al. "Focal loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017.
20. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv: 1804.02767 (2018).
21. Zhang, Shifeng, et al. "Single-Shot Refinement Neural Network for Object Detection," 10.1109/CVPR. 2018.00442. 2018.
22. Mabrouk, Amira Ben, and Ezzeddine Zagrouba. "Abnormal behavior recognition for intelligent video surveillance systems: A review." Expert Systems with Applications 91 (2018): 480-491.
23. Liao, Minghui, Baoguang Shi, and Xiang Bai. "Textboxes++: A single-shot oriented scene text detector." IEEE transactions on image processing 27.8 (2018): 3676-3690.
24. Smith, Ray. "An overview of the Tesseract OCR engine." Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). Vol. 2. IEEE, 2007.