# **SUSPICIOUS ACTIVITY DETECTION**

**Using OpenCV**

Developed By: Bhawna Priya, Bhavana V, Deeksha Udasi, Kavita Hegde, Reshma B

Smart Bridge – Remote Summer Internship Program 2020

**ABSTRACT**

We present an efficient method for detecting anomalies in videos. Recent applications of convolutional neural networks have shown promises of convolutional layers for object detection and recognition, especially in images. However, convolutional neural networks are supervised and require labels as learning signals. We propose an architecture for anomaly detection in videos including crowded scenes. Our architecture includes two main anomaly detection, one is robbery and the other is vandalism. Experimental results on Avenue, Subway and UCSD benchmarks confirm that the detection accuracy of our method is comparable to state-of-the-art methods at a considerable speed of up to 140 fps.

**1. INTRODUCTION**

With the rapid growth of video data, there is an increasing need not only for recognition of objects and their behaviour, but in particular for detecting the rare, interesting occurrences of unusual objects or suspicious behaviour in the large body of ordinary data. Finding such abnormalities in videos is crucial for applications ranging from automatic quality control to visual surveillance.

Meaningful events that are of interest in long video sequences, such as surveillance footage, often have an extremely low probability of occurring. As such, manually detecting such events, or anomalies, is a very meticulous job that often requires more manpower than is generally available. This has prompted the need for automated detection and segmentation of sequences of interest. However, present technology requires an enormous amount of configuration efforts on each video stream prior to the deployment of the video analysis process, even with that, those events are based on some predefined heuristics, which makes the detection model difficult to generalize to different surveillance scenes.

Video data is challenging to represent and model due to its high dimensionality, noise, and a huge variety of events and interactions. Anomalies are also highly contextual, for example, running in a restaurant would be an anomaly, but running at a park would be normal. Moreover, the definition of anomaly can be ambiguous and often vaguely defined. A person may think walking around on a subway platform is normal, but some may think it should be flagged as an anomaly since it could be suspicious. These challenges have made it difficult for machine learning methods to identify video patterns that produce anomalies in

real-world applications.

There are many successful cases in the related field of action recognition. So we propose our model which can recognize few suspicious activities in real-time just using CNN.

Our proposed method is domain free (i.e., not related to any specific task, no domain expert required), does not require any additional human effort, and can be easily applied to different scenes. To prove the effectiveness of the proposed method we apply the method to real-world datasets and show that our method consistently outperforms similar methods while maintaining a short running time.

**2 LITERATURE SURVEY**

Most of the abnormal instances are beforehand unknown, as this would require predicting all the ways something could happen out of the norm. It is therefore simply impossible to learn a model for all that is abnormal or irregular. But how can we find an anomaly without what to look for?

The majority of the work on anomaly detection relies on the extraction of local features from videos that are then used to train the model. After training, the model can predict similar anomalies from the surveillance video in real-time.

**3 Flow chart**

INPUT VIDEO

EXTRACTION OF FRAMES

IMAGE PREPROCESSING AND LABELLING

AUGMENTATION PROCESS

NEURAL NETWORK TRAINING

TESTING

REAL TIME PREDICTION

**Block Diagram**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Data set  Collection | |  | |  | | --- | | Video Pre-processing | |  | |  | | --- | | Extracting Frames | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Building CNN Model | |  | |  | | --- | | Train And Test Split Data | |  | |  | | --- | | Image Pre-processing | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Testing Model | |  | |  | | --- | | Real- Time Prediction | |  |  |

**4 Experimental Investigation**

Over recent years, surveillance camera is attracting attention due to its wide range of applications in suspicious activity detection. Current surveillance system focuses on analysing past incidents. This paper proposes an intelligent system for real-time monitoring with added functionality of anticipating the outcome through various Image processing techniques. As this is a sensitive matter, human decisions are given priority, still facilitating limited logical intervention of human resource. This framework detects risk in the area under surveillance. One such dangerous circumstance is implemented, like a person with a knife. Here the prediction is that in the firm places like ATM, Banks, Offices etc. a person possessing knife is unusual and likely to cause harmful activities like threatening, injuring and stabbing. The experiment demonstrates the effectiveness of the technique on training dataset collected from distinct environments. An interface is developed to notify concerned authority that boosts reliability and overall accuracy.

**4.1 Data sets**

* In this paper, the dataset we used is derived from <https://www.kaggle.com/mateohervas/dcsass-dataset>.
* Suspicious activity dataset contains videos based on the following 2 classes: Robbery and Vandalism.
* There are a total of 394 videos of each class.
* We are extracting the image frames from these videos at regular intervals to create a dataset consisting of enough images to train our model***.***

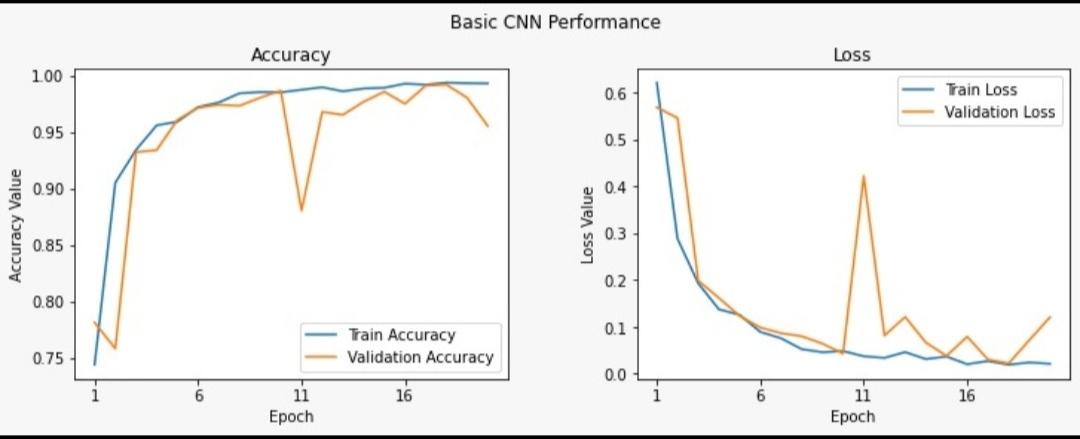
**4.2 Model Parameters**

* We train the model by minimizing the reconstruction error of the input volume.
* We use Adam optimizer to allow it taking the role of setting the learning rate automatically based on the models weight update history.
* We trained our model with 20 epochs.
* We have used softmax activation function for output layer.

**4.3 Results and Analysis**

In this project, Convolutional Neural Network is used to predict the suspicious activity, Normal, Robbery and Vandalism videos. We have added 3 convolutional layers as well as 3 pooling layers to increase the performance.

From the graph below we can see that as we increase the no. of epochs, the accuracy and loss. We got 99.3% accuracy for our model.



# **5. ADVANTAGES AND DISADVANTAGES**

**Advantages:**

* It can easily identify the Robbery and Vandalism happening.
* It can detect suspicious activities from images of respective categories.
* It is composed using the Python, CNN and message API for it’s usage in real time.
* It works in real time and predict as soon as something suspicious happens in front of the camera and predict the result.
* It provides an Email and SMS alert as soon as something suspicious happens and provides more security for the users.

**Disadvantages:**

* Our model can detect only few types of suspicious activities.
* The calculation becomes incompatible because the images of the given activities are not perfect.
* It requires huge dataset for training.
* It has high computational costs. Requires lot of time during training of the model.

**6.** **APPLICATIONS**

* Deep learning based model can detect suspicious activity from given images with accuracy of 99% while requiring just few floating point operations.
* Suspicious Activity Detection using OpenCV, which can detect suspicious activities like vandalism and robbery only from providing similar images as dataset can help us a lot in real life.
* For practical validation of model efficiency, we have deployed the real-time model for Image processing operations and video analysis for object detection for a better output.
* So we use Machine Learning Algorithms using Python to analyse the data and propose suspicious activity detection.

**7. CONCLUSION**

The initial problem of classifying 2 suspicion categories was a challenging multi-class classification problem, and there was not enough predictability in our initial data-set to obtain very high accuracy on it.

We found that a more meaningful approach was to collapse the categories into fewer, larger groups, in order to find structure in the data. We got high accuracy and precision on Prediction.

If proper dataset is available, we can extend this model to predict few more or all kind of suspicious activities which happen in real world.

Other areas to work on include implementing a more accurate multi-class classifier, and exploring better ways to visualize our results. We looked at easy to build open-source techniques leveraging AI which can give us state-of-the-art accuracy in detecting suspicious activity thus enabling AI for social good. Let’s hope for more adoption of open-source AI capabilities across the globe making it cheaper and accessible for everyone across the world!

**8. FUTURE SCOPE**

In future the Convolutional Neural Network algorithm can be applied on other large data sets available for Suspicious Activity Detection to further investigate its accuracy. Various other algorithms can also be used in future to investigate the power machine learning algorithms for suspicious activity detection using OpenCV. For future work, we will investigate how to improve the result of suspicious activity detection by active learning having human feedback to update the learned model for better detection and reduced false alarms. One idea is to add a supervised module to the current system, which the supervised module works only on the video segments iterated by our proposed method, then train a discriminative model to classify anomalies when enough video data has been acquired. In further study, we will try to conduct experiments on larger data sets or try to tune the model so as to achieve the state-of-art performance of the model and a great UI support system making it complete web application model.

**9. BIBLIOGRAPHY**

1. Nosato, H., Sakanashi, H., Takahashi, E., Murakawa, M.: Method of retrieving multi-scale objects from optical colonoscopy images based on image-recognition techniques. In: 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 1–4. IEEE (2015)
2. Li, J., Ye, D.H., Chung, T., Kolsch, M., Wachs, J., Bouman, C.: Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs). In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4992–4997. IEEE (2016) Vaidehi, K., Subashini, T.: Automatic classiﬁcation and retrieval of mammographic tissue density using texture features. In: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), pp. 1–6. IEEE (2015)
3. Rao, R.S., Ali, S.T.: A computer vision technique to detect phishing attacks. In: 2015 Fifth International Conference on Communication Systems and Network Technologies (CSNT), pp. 596–601. IEEE (2015)
4. Herrmann, C., Beyerer, J.: Face retrieval on large-scale video data. In: 2015 12th Conference on Computer and Robot Vision (CRV), pp. 192–199. IEEE (2015)
5. Sidhu, R.S., Sharad, M.: Smart surveillance system for detecting interpersonal crime. In: 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 2003–2007. IEEE (2016)
6. Vallet, A., Sakamoto, H.: Convolutional recurrent neural networks for better image understanding. In: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–7. IEEE (2016)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](http://arxiv.org/abs/1409.1556) (2014)
8. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
9. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). doi:[10.1007/978-3-319-10602-1 48](http://dx.doi.org/10.1007/978-3-319-10602-1_48)
10. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recog- nition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classiﬁcation with deep con- volutional neural networks. In: Advances in Neural Information Processing Sys- tems, pp. 1097–1105 (2012)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classiﬁcation with deep con- volutional neural networks. In: Advances in Neural Information Processing Sys- tems, pp. 1097–1105 (2012)
13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
14. Razavian, A.S., Sullivan, J., Carlsson, S., Maki, A.: Visual instance retrieval with deep convolutional networks. arXiv preprint [arXiv:1412.6574](http://arxiv.org/abs/1412.6574) (2014)
15. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 584–599. Springer, Cham (2014). doi:[10.1007/ 978-3-319-10590-1 38](http://dx.doi.org/10.1007/978-3-319-10590-1_38)
16. Fidalgo, E., Alegre, E., Gonz´alez-Castro, V., Ferna´ndez-Robles, L.: Compass radius estimation for improved image classiﬁcation using edge-sift. Neurocomputing **197**, 119–135 (2016)
17. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616. ACM (2009)
18. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accu- rate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
19. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303– 338 (2010)
20. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 346–361. Springer, Cham (2014). doi:[10. 1007/978-3-319-10578-9 23](http://dx.doi.org/10.1007/978-3-319-10578-9_23)
21. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
22. Fern´andez-Robles, L., Castej´on-Limas, M., Alfonso-Cendo´n, J., Alegre, E.: Eval- uation of clustering conﬁgurations for object retrieval using SIFT features. In: Ayuso Mun˜oz, J.L., Yagu¨e Blanco, J.L., Capuz-Rizo, S.F. (eds.) Project Manage- ment and Engineering. LNMIE, pp. 279–291. Springer, Cham (2015). doi:[10.1007/ 978-3-319-12754-5 21](http://dx.doi.org/10.1007/978-3-319-12754-5_21)

**10. APPENDIX**

CODE FOR EXTRACTING FRAMES FROM VIDEOS

from glob import glob

from tqdm import tqdm

import sys

import cv2

import os

videos = glob("Dataset/Training/Vandalism/\*.mp4")

prediction=[]

for i in tqdm(range(len(videos))):

count = 0

vidcap = cv2.VideoCapture(videos[i])

success,image = vidcap.read()

success = True

while success:

vidcap.set(cv2.CAP\_PROP\_POS\_MSEC,(count\*500)) # added this line

success,image = vidcap.read()

if not success:

break

# print ('Read a new frame: ', success)

cv2.imwrite( 'Dataset/ImageDataset/Training/Vandalism/' +videos[i].split('/')[2].split('\\')[1] + "frame%d.jpg" % count, image) # save frame as JPEG file

count = count + 1

print("SUCCESS")

videos = glob("Dataset/Training/Robbery/\*.mp4")

prediction=[]

for i in tqdm(range(len(videos))):

count = 0

vidcap = cv2.VideoCapture(videos[i])

success,image = vidcap.read()

success = True

while success:

vidcap.set(cv2.CAP\_PROP\_POS\_MSEC,(count\*500)) # added this line

success,image = vidcap.read()

if not success:

break

#print ('Read a new frame: ', success)

cv2.imwrite( 'Dataset/ImageDataset/Training/Robbery/' +videos[i].split('/')[2].split('\\')[1] + "frame%d.jpg" % count, image) # save frame as JPEG file

count = count + 1

print("SUCCESS")

videos = glob("Dataset/Testing/Vandalism/\*.mp4")

prediction=[]

for i in tqdm(range(len(videos))):

count = 0

vidcap = cv2.VideoCapture(videos[i])

success,image = vidcap.read()

success = True

while success:

vidcap.set(cv2.CAP\_PROP\_POS\_MSEC,(count\*500))

success,image = vidcap.read()

if not success:

break

# print ('Read a new frame: ', success)

cv2.imwrite( 'Dataset/ImageDataset/Testing/Vandalism/' +videos[i].split('/')[2].split('\\')[1] + "frame%d.jpg" % count, image) # save frame as JPEG file

count = count + 1

print("SUCCESS")

videos = glob("Dataset/Testing/Robbery/\*.mp4")

prediction=[]

for i in tqdm(range(len(videos))):

count = 0

vidcap = cv2.VideoCapture(videos[i])

success,image = vidcap.read()

success = True

while success:

vidcap.set(cv2.CAP\_PROP\_POS\_MSEC,(count\*500))

success,image = vidcap.read()

if not success:

break

# print ('Read a new frame: ', success)

cv2.imwrite( 'Dataset/ImageDataset/Testing/Robbery/' +videos[i].split('/')[2].split('\\')[1] + "frame%d.jpg" % count, image) # save frame as JPEG file

count = count + 1

print("SUCCESS")