# VIDEO ANOMALY DETECTION SYSTEM

Project report submitted in partial fulfillment of the Requirements for the

Award of the Degree of B. Tech in Information Technology

BY

**Yash Pandey 2013747**

**Ankit Panwar 2013749**

**Kuldeep Nautiyal 2013754**

**Tanu Kumari 2013755**

Under the Guidance of

**Dr. Manoj Diwakar Supervisor**



# VIDEO ANOMALY DETECTION SYSTEM

Project report submitted in partial fulfillment of the Requirements for the

Award of the Degree of B. Tech in Information Technology

BY

**Yash Pandey 2013747**

**Ankit Panwar 2013749**

**Kuldeep Nautiyal 2013754**

**Tanu Kumari 2013755**

Under the Guidance of

**Dr. Manoj Diwakar Supervisor**



**CERTIFICATE**

This is to certify that the project report entitled VIDEO ANOMALYDETECTION SYSTEM being submitted by

|  |  |
| --- | --- |
| Yash Pandey | 2013747 |
| Ankit Panwar | 2013749 |
| Kuldeep Nautiyal | 2013754 |
| Tanu Kumari | 2013755 |

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Graphic Era Deemed to be University is a record of bonafied work carried out under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Dr. Manoj Diwakar Associate Professor Date :

Head of the Department

v

## ACKNOWLEDGMENT

It is my proud privilege to express my gratitude to the entire management of **Graphic Era Deemed to be University** and the faculties of the institute to provide me with the opportunity to avail the excellent facilities and infrastructure. The acknowledgment and values inculcated have proved to be of immense help at a very start of my career.

I would like to express my sincere thanks to my project guide Dr. Manoj Diwakar from **Graphic Era Deemed to be University** for guiding me through the project.

## ABSTRACT

Surveillance videos are able to capture a variety of realistic anomalies. In this project, we propose to learn anomalies by exploiting both normal and abnormal videos. To avoid annotating the anomalous segments of clips in training videos, which is very time consuming, we propose to learn anomaly through the deep multiple instance ranking framework by leveraging weakly labeled training videos, i.e. the training labels (anomalous or normal) are at video level instead of clip- level. In our approach, we consider normal and anomalous videos as bags and video segments as instances in multiple instance learning. And automatically learn a deep anomaly ranking model tha predicts high anomaly scores for anomalous video segments.

## Table Of Content

|  |  |  |
| --- | --- | --- |
| **S.no.** | **Name** | **Page No.** |
| 1 | Introduction | 1 |
| 2 | Analysis | 2-3 |
| 3 | Design | 4-7 |
| 4 | Software and Hardware Requirement | 8 |
| 5 | Code Template | 9-11 |
| 6 | Testing And Output Screens | 12-13 |
| 7 | Conclusion | 14 |

### Introduction

Anomaly detection is of major attention to many different areas within deep learning, statistics, machine learning. The main objective to detect and identify those areas of the region whose pattern or behaviours do not conform to expected in that dataset. These unexpected events which are remarkably different from an observation data set, these are called anomalies. Unobserved pattern or novelty fluctuation in data is also an anomaly. In spite of this, still, there is a no solid standard definition of this concept. An abnormality is likewise alluded to as a special case like an exception, outlier, variation conflicting object and it's by and large relying upon various applications. To detect or identify outliers or unexpected patterns are significant in numerous domains including business intelligence, machine learning, deep learning, decision making, network transmission, transport, production.

Video Anomaly Detection is the process of detecting abnormal events in the Video. Video event detection is the process of detecting an event in the video. Video Anomaly detection is the subclass of Video Event Detection. There is always a necessity for this kind of software that can detect movement and record in a video. There are 3 different types of video Anomaly Detection techniques, they are 1.Manual 2.semi automated 3.Fully automated. Manual Anomaly Detection is the method in which a person is specially designated for the process of monitoring the live video. Data collected by this process nevertheless is hard to monitor. Indeed this is not conceivable to record all along because the hard disk will be full as the software is to be run for a significant number of days. The Semi-automated process is the one in which there is no requirement of an employed guy. When the machine detects face or motion in a video the machine starts recording from that instance and sends the recorded video for analysing. Fully automated is similar to semi-automated but there is no involvement of video analyser to analyse the anomaly video. This can be done in two ways 1. Supervised 2.Unsupervised, Supervised have a training phase, whereas Unsupervised doesn't have any explicit training phase. The algorithm detects an abnormal event in the Supervised method if it finds an event that has no similarity with the trained events. Videos are able to capture a variety of realistic anomalies. In this paper, we propose to learn anomalies by exploiting both normal and anomalous videos. There is a wide variety of abnormal events that might take place even in a single location, and the definition of abnormal event differs from location to another and from time to time.

### 1

1. **Analysis**

### Problem Description

The goal of anomaly detection comes from the idea of detecting any abnormal behaviour in real time. As humans, we have always tried to use machine to replicate human functioning for achieving high levels of efficiency. An important trait of human mind is to detect any abnormal activities happening around us. For example, in a crowded hallway or a corridor a healthy human can easily detect any out of normal or malicious activity, the main aim of this project is to emulate the similar kind of processing with a better efficiency. Generally, anomalous events rarely occur as compared to normal activities. Therefore, to alleviate the waste of labour and time, developing intelligent computer vision algorithms for automatic video anomaly detection is a pressing need. The goal of a practical anomaly detection system is to timely signal an activity that deviates normal patterns and identify the time window of the occurring anomaly.

### Convolutional Neural Networks ( CNNs / ConvNets)

Convolutional Neural Networks are very similar to ordinary Neural Networks .They are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they have a loss function (e.g. SVM) on the last (fully-connected) layer. ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network.

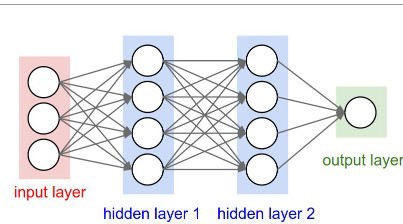


Fig 2.1 : A regular 3-layer neural network

### Autoencoders

Autoencoders are neural networks that aim to copy their inputs to their outputs. They work by compressing the input into a latent space representation .

This kind of network is composed of two parts :

Encoder: This is the part of network that compresses the input into a latent-space representation. It can be represented by an encoding function *h=f(x)* .

Decoder: This part aims to reconstruct the input from latent-space representation. It can be represented by a decoding function *r=g(h)*.

The difference between input representation and output representation is known as reconstruction error(error between input vector and output vector). One of the predominant use cases of the Autoencoder is anomaly detection .

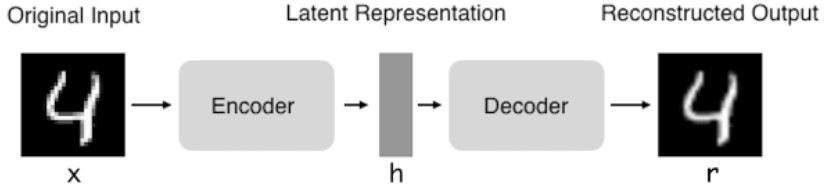


Fig 2.2 – Architecture of an Autoencoder

### Software Design

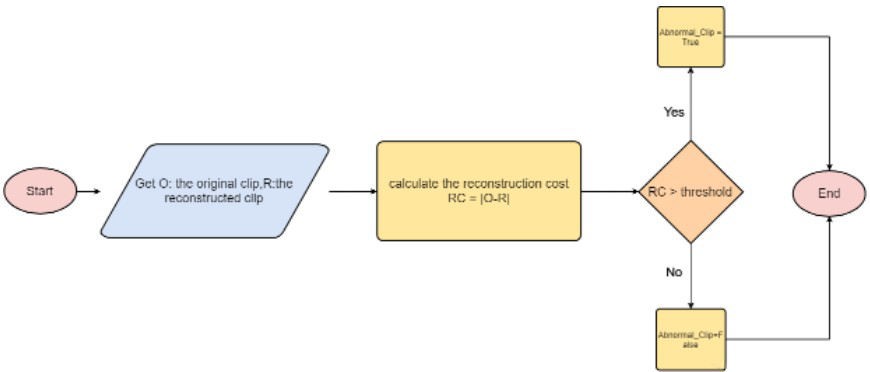
If we want to treat the problem as a binary classification problem , we need labeled data and in this case , collecting labeled data is hard because of the following reasons :

* Abnormal events are challenging to obtain due to their rarity.
* There is a massive variety of abnormal events, and manually detecting and labeling such events is a difficult task that requires much manpower.

The above reasons promoted the need to use unsupervised or semi-supervised methods like dictionary learning, Spatio-temporal features, and autoencoders. Unlike supervised methods, these methods only require unlabeled video footages that contain little or no abnormal events that are easy to obtain in real-world applications.

### Approach

It is all about reconstruction error. The difference between input representation and output representation is known as reconstruction error(error between input vector and output vector). One of the predominant use cases of the Autoencoder is anomaly detection. We use an autoencoder to learn regularity in video sequences. The intuition is that trained autoencoder will reconstruct regular video sequences with low error but will not accurately reconstruct motions in irregular video sequences.



### Fig 3.1 – Anomaly detection with autoencoder

* 1. **Building and Training the Model**

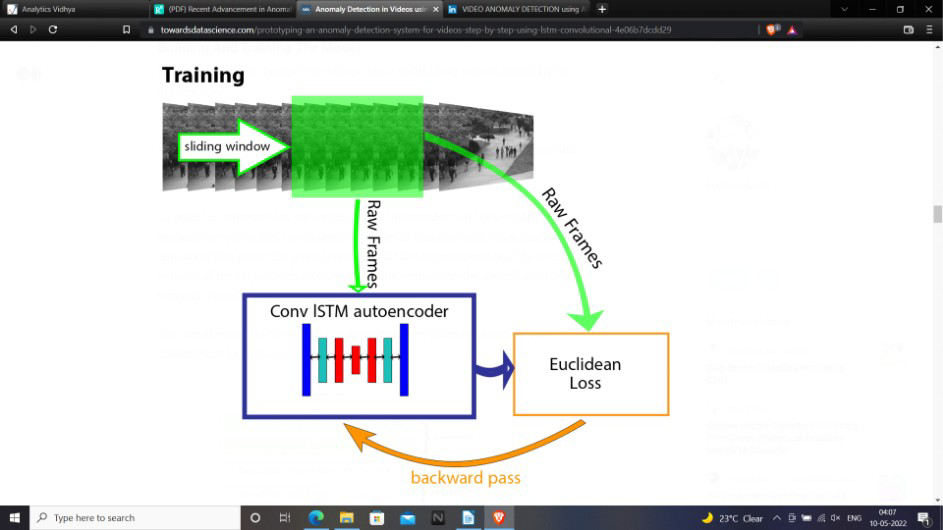
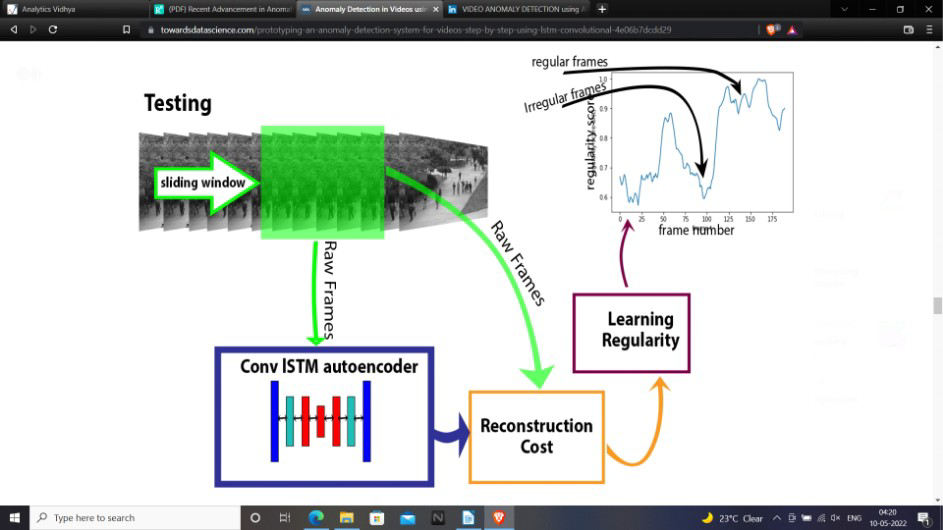


Fig 3.2 – Training Process

To build the autoencoder, we should define the encoder and the decoder. The encoder accepts as input a sequence of frames in chronological order, and it consists of two parts: the spatial encoder and the temporal encoder. The encoded features of the sequence that comes out of the spatial encoder are fed into the temporal encoder for motion encoding. The decoder mirrors the encoder to reconstruct the video sequence, so our autoencoder looks like a sandwich.

We use Adam as an optimizer with a learning rate set to 0.0001, we reduce it when training loss stops decreasing by using a decay of 0.00001, and we set the epsilon value to 0.000001.For initialization, we use the Xavier algorithm, which prevents the signal from becoming too tiny or too massive to be useful as it goes through each layer.

### Testing

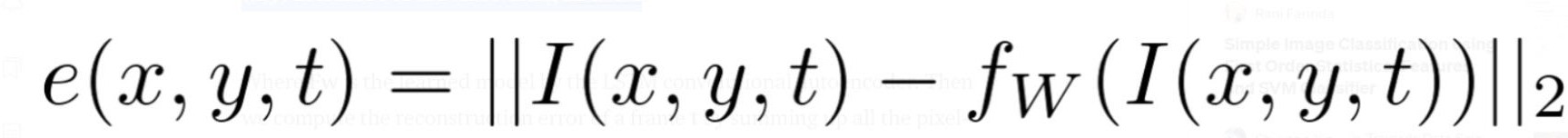


**Fig 3.3: Testing Process**

We use sliding window technique to get all consecutive 10-frame sequences. It means for each t between 0 and 190,we calculate regularity score Sr(t) of the sequence that starts at frame t and ends at frame (t+9).

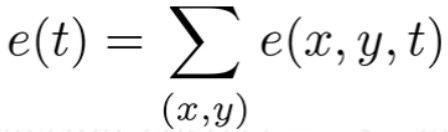
### Regularity Score

We compute the reconstruction error of a pixel’s intensity value I at the location (x,y) in frame t of the video using [L2 norm](https://machinelearningmastery.com/vector-norms-machine-learning/) **:**

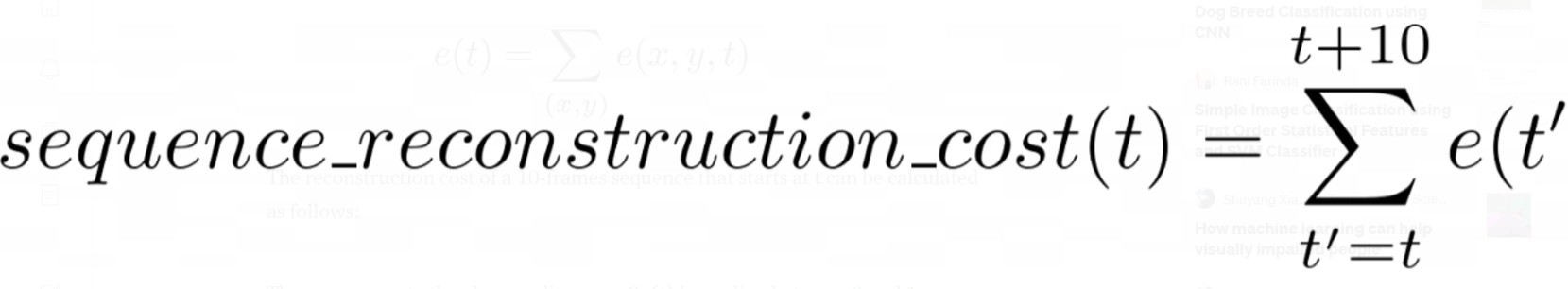


where Fw is the learned model by LSTM convolutional autoencoder.

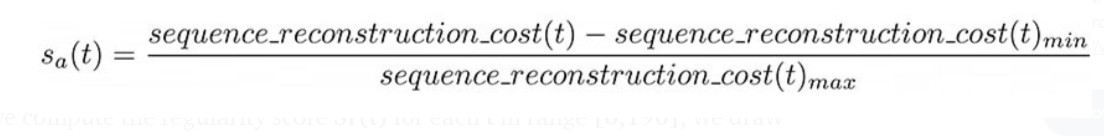
Then we compute the reconstruction error of a frame t by summing up all of the pixel wise errors :



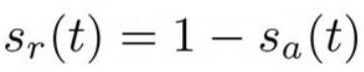
The reconstruction cost of a 10-frame sequence that starts at t can be calculated as follows :



Then we compute the abnormality score Sa(t) by scaling between 0 and 1.



We can derive regularity score Sr(t) by subtracting abnormality scores from 1



After we compute the regularity score Sr(t) for each t in range [0,190] , we draw Sr(t).

### Software and Hardware Requirement

* 1. **GPU**

A graphics processing unit (GPU) is a computer chip that renders graphics and images by performing rapid mathematical calculations. GPUs are used for both professional and personal computing. Traditionally, GPUs are responsible for the rendering of 2D and [3D](https://www.techtarget.com/whatis/definition/3-D-three-dimensions-or-three-dimensional) images, animations and video even though, now, they have a wider use range. Because of the presence of a huge number of parameters we’re using GPU.

In the early days of computing, the central processing unit ([CPU](https://www.techtarget.com/whatis/definition/processor)) performed these calculations. As more graphics-intensive applications were developed, however, their demands put a strain on the CPU and decreased performance. GPUs were developed as a way to offload those tasks from CPUs and to improve the rendering of 3D graphics. GPUs work by using a method called [parallel processing](https://www.techtarget.com/searchdatacenter/definition/parallel-processing), where multiple processors handle separate parts of the same task.

### Python

It is a general purpose coding language which means that unlike HTML, CSS and JavaScript , it can be used for other types of programming and software development besides web development. That includes back end development , software development , data science and writing system scripts among other things.

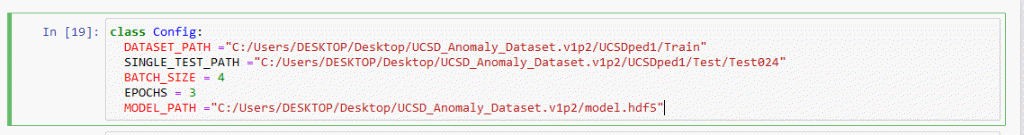
### Keras

It is a open source software library that provides python interface for artificial neural networks. It acts as an interface for TensorFlow library.

### TensorFlow

It is an open source library developed primarily for deep learning applications. It also supports traditional Machine Learning. It was originally developed for large numeric computations without keeping deep learning in mind.

1. **Code Templates**



Defining the path of Dataset , Data model and Test data.

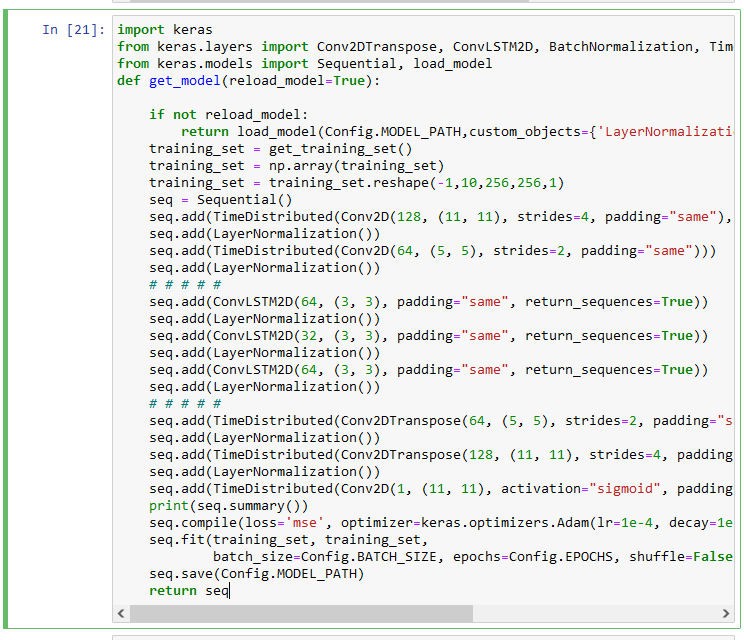


The training set consists of sequences of regular video frames; the model will be trained to reconstruct these sequences. So, let’s get the data ready to feed our model by following these three steps:

* Divide the training video frames into temporal sequences, each of size 10 using the sliding window technique.
* Resize each frame to 256 × 256 to ensure that input images have the same resolution.
* Scale the pixels values between 0 and 1 by dividing each pixel by 256.

One last point is that since the number of parameters in this model is huge, we need a large amount of training data, so we perform data augmentation in the temporal dimension. To generate more training sequences, we concatenate frames with various skipping strides. For example, the first stride-1 sequence is made up of frames (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), whereas the first stride-2 sequence consists of frames (1, 3, 5, 7, 9,

11, 13, 15, 17, 19).



Training the model using Conv ISTM autoencoder , Euclidean Loss and backward pass.

We will test each testing video individually. UCSD dataset provides 34 testing videos, the value of Config.SINGLE\_TEST\_PATH determines which one will be used.

### 10



finally plotting the graph for acquired result.

1. **Testing And Output Screens**

let’s take a look at test 32 of UCSDped1. At the beginning of the video, there is a bicycle on the walkway, which explains the low regularity score. After the bicycle left, the regularity score starts to increase. At frame 60, another bicycle enters, the regularity score decreases again and increases right after it left.

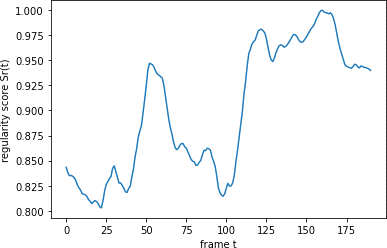


Fig 6.1: Test Case 1

Test 004 of UCSDped1 dataset shows a skater entering the walkway at the beginning of the video, and someone walks on the grass at frame 140, which explains the two drops in the regularity score.

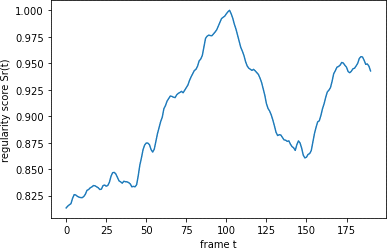
 

Fig 6.2 : Test Case 004

Test 024 of UCSDped1 dataset shows a small cart crossing the walkway, causing a drop in the regularity score. The regularity score returns to the normal state after the cart left.

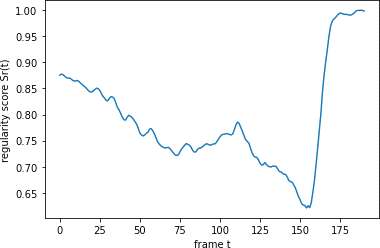


Fig 6.3 : Test Case 024

1. **Conclusion**

We propose a deep learning approach to detect anomalies in surveillance videos. Due to the complexity of these realistic anomalies, using only normal data alone may not be optimal for anomaly detection. We attempt to exploit both normal and anomalous surveillance videos. To avoid labour intensive temporal annotations of anomalous segments in training videos, we learn a general model of anomaly detection using our method with weakly labelled data.

We have achieved a great deal of technological advancements. Everyday a new advancement is made and we are creating new technology every single day. We are surrounded by thousands of cameras around us . There is a wide variety of abnormal events that might take place at a single location . Using Automated systems to detect these abnormal events is highly desirable. Thus through our project we have tried to achieve that.

We can use multiple datasets or even gather our own data using a surveillance camera or a small camera in our room . The training data is easy to collect since it consists of videos that contain only regular event .

1. **References**
2. Yong Shean Chong, Abnormal Event Detection in Videos using Spatiotemporal Autoencoder (2017),[arXiv:1701.01546](http://arxiv.org/abs/1701.01546).
3. Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K. Roy-Chowdhury, Learning Temporal Regularity in Video Sequences(2016),[arXiv:1604.04574](http://arxiv.org/abs/1604.04574).
4. A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixedlocation monitors. TPAMI, 2008.
5. S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In NIPS, pages 577–584, Cambridge, MA, USA, 2002.
6. R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic.´ NetVLAD: CNN architecture for weakly supervised place recognition. In CVPR, 2016.
7. A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly .
8. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM
9. S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. Pattern Recognition, 48(10):2993–3003, 2015.
10. Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu. Violence detection using oriented violent flows. Image and Vision Computing, 2016.
11. M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In CVPR, June 2016.
12. T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In ICCV, 2009.
13. R. Hou, C. Chen, and M. Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In ICCV, 2017.
14. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In CVPR, 2014.