# Anomaly Detection in Surveillance Videos

## 1. Introduction

To improve public safety, surveillance cameras are being utilized more frequently in public spaces like roadways, junctions, banks, and shopping centers. Law enforcement organizations' monitoring capacities, however, have not kept up. As a result, there is a conspicuous deficit in the use of security cameras and an inefficient camera-to-monitor ratio. Identifying abnormal events, such as traffic accidents, crimes, or unlawful activity, is a crucial duty in video surveillance. As opposed to usual activity, anomalous events typically happen far less frequently. Therefore, there is an urgent need to create computer vision algorithms for automatic video anomaly identification to reduce labor and time loss. A practical anomaly detection system's objective is to promptly report behavior that deviates from expected patterns and pinpoint the time window in which the anomaly is occurring. As a result, anomaly detection can be thought of as a basic form of video understanding that separates anomalies from expected patterns.

## 2.Executive summary:

It is a problem of anomaly detection in surveillance videos. The workflow for this project starts from collecting anomaly and normal videos, processing and extracting the features from the videos. For example, let’s consider an anomaly video assume that we have three anomalies in that video, we initially start converting the video into frames, from the frames we apply featurization techniques say covolution3D or Inflated3D on it. We send the feature to training module where it has sequence of convolution, pooling, dropout and dense layer at the end in case of C3D or I3D. in the later stage we get some anomaly scores marked on each frame. In C3D we used 2 epochs with 20000 iterations to extract the features. Whereas in the case of I3D we used 7335 epochs. With C3D we got an AUC score of 0.68 but with I3D it is 0.962. here I3D is more efficient in extracting the features from the data and so the training is more efficient.so the accuracy is high comparatively.

## 3. Problem Statement:

Surveillance videos act as proof of evidence, it does not act as a live interacting system. What if a surveillance camera can act as a threat reporting system. The goal of the project is to implement a deep learning technique that can take advantage of surveillance videos and report any unusual activities. This technique is known as anomaly detection.

Once an anomaly is found, it can then be classified using classification algorithms into one of the specific tasks. Creating algorithms to identify a particular anomalous occurrence, such as a violence detector or a traffic accident detector, is a first step in tackling anomaly detection. However, such methods cannot be applied to identify further anomalous events, therefore in reality, their practical utility is quite restricted.

## 4. Techniques

A wide range of believable anomalies can be seen in surveillance videos. In this study, we suggest using both regular and anomalous films to learn abnormalities. We suggest learning anomaly through the deep multiple instances ranking framework by utilizing weakly labeled training videos, where the training labels (anomalous or normal) are at the video level rather than the clip-level, to avoid spending a lot of time annotating the anomalous segments or clips in training videos. Our method uses multiple instance learning (MIL), where video segments are instances and normal and abnormal movies are bags. This method automatically learns a deep anomaly ranking model that forecasts high anomaly scores for abnormal video segments. To locate anomaly more accurately during training, we also include sparsity and temporal smoothness restrictions in the ranking loss function. By requiring consistent optical flow between predicted frames and ground truth frames, we add a motion (temporal) constraint to video prediction in addition to the usual appearance (spatial) constraints on intensity and gradient. The video prediction task is the first to incorporate a temporal constraint in this work.

Used methods:

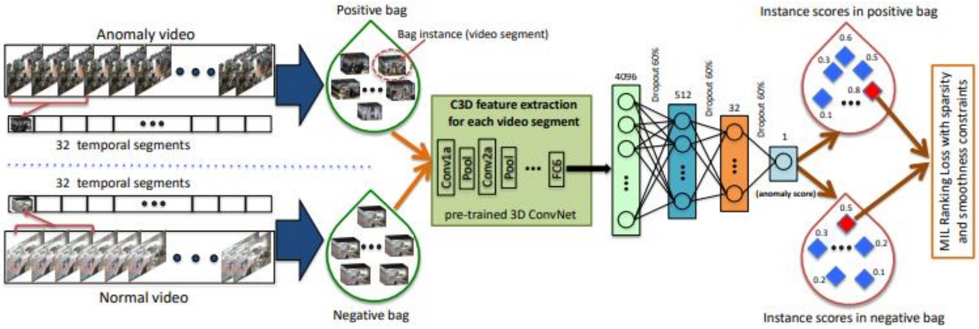
1. Methods based on hand-crafted features

2.Methods Using Deep Learning

The underlying assumption of the proposed method is that the network can automatically learn to predict the location of the anomaly in a video when presented with many positive and negative videos labeled at the video level. During training iterations, the network should acquire the ability to produce high scores for anomalous video segments to achieve this objective.

**Technique 1:**

This technique is to separate each of the positive (contain anomaly anywhere) and negative (contain no anomaly) videos into several temporal video chunks. Each temporal segment then represents an instance in the bag, which is represented by each video as a bag. We train a fully connected neural network using a unique ranking loss function after extracting C3D features for video segments. This function calculates the ranking loss between the top scored occurrences (shown in red) in the positive bag and the negative bag.



Although current methods show effective detection performance, their recognition of the positive instances, i.e., rare abnormal snippets in the abnormal videos, is largely biased by the dominant negative instances, especially when the abnormal events are subtle anomalies that exhibit only small differences compared with normal events.

In standard supervised classification problems using support vector machine, the labels of all positive and negative examples are available, and the classifier is learned using the following optimization function

$$\min_{\mathbf{w}} \left[\frac{1}{k}\sum_{i=1}^{k}\overbrace{max(0, 1 - y_i(\mathbf{w}.\phi(x) - b))}^{\textcircled{1}}\right] + \frac{1}{2}\left\|\mathbf{w}\right\|^2 ,$$

where k is the total number of training examples, w is the classifier to be learnt, yi represents the label of each case, x symbolizes feature representation of an image patch or video segment, b is a bias, and 1 is the hinge loss. Accurate annotations of positive and negative instances are necessary to develop a robust classifier. A classifier needs temporal annotations for each video segment in the context of supervised anomaly detection. However, getting temporal annotations for videos is a tedious and time-consuming process. The requirement of providing these precise temporal annotations is relaxed by MIL. MIL does not know the exact temporal positions of abnormal events in videos. Instead, all that is required are video-level labels showing the presence of an abnormality throughout the entire video. Positive labels are applied to videos having abnormalities, and negative labels are applied to videos without any anomalies. Then, a positive video is represented as a positive bag Ba, where different temporal segments create distinct instances in the bag, (p1, p2,..., pm), where m is the total number of instances in the bag. We presume that the anomaly is present in at least one of these situations. Similar to this, a negative bag, Bn, is used to represent the negative video, where temporal segments generate negative instances (n1, n2,.. nm). None of the examples in the negative bag have an oddity. One can optimize the objective function since the precise information (i.e., instance-level label) of the positive examples is unknown

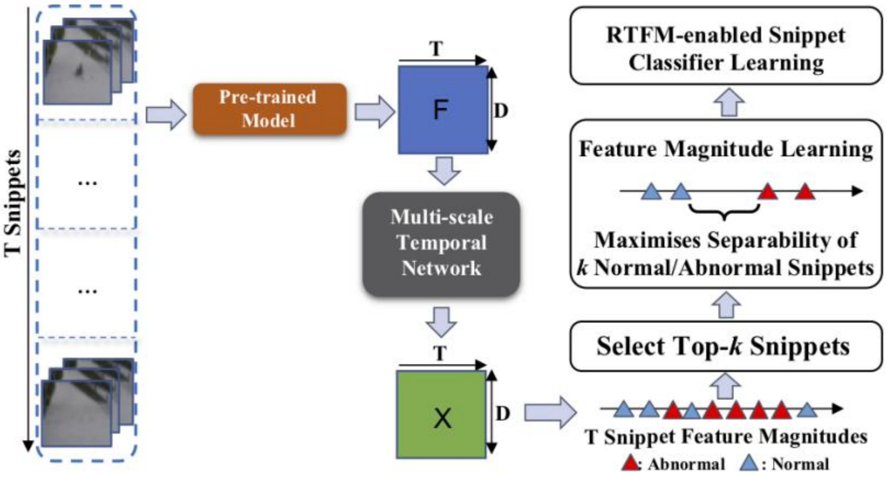
$$\min_{\mathbf{w}} \left[\frac{1}{z}\sum_{j=1}^{z} max(0, 1 - Y_{\mathcal{B}_j}(\max_{i \in \mathcal{B}_j}(\mathbf{w}.\phi(x_i)) - b))\right] + \|\mathbf{w}\|^2$$

where YBj denotes bag-level label, z is the total number of bags.

During training, this strategy begins by dividing surveillance videos into predetermined segments. A bag is made up of these segments. The proposed deep MIL ranking loss is used to train the anomaly detection model with positive (anomalous) and negative (normal) bags.

**Technique 2**:

RTFM methodology developed on top of MIL approach, and it is accepted at ICCV 2021. t trains a feature magnitude learning function to effectively recognize the positive instances, substantially improving the robustness of the MIL approach to the negative instances from abnormal videos. RTFM also adapts dilated convolutions and self-attention mechanisms to capture long- and short-range temporal dependencies to learn the feature magnitude more faithfully.


T Snippets
...
...
Pre-trained Model
T
F
D
Multi-scale Temporal Network
T
X
D
RTFM-enabled Snippet Classifier Learning
Feature Magnitude Learning
Maximises Separability of k Normal/Abnormal Snippets
Select Top-k Snippets
T Snippet Feature Magnitudes
▲: Abnormal ▲: Normal

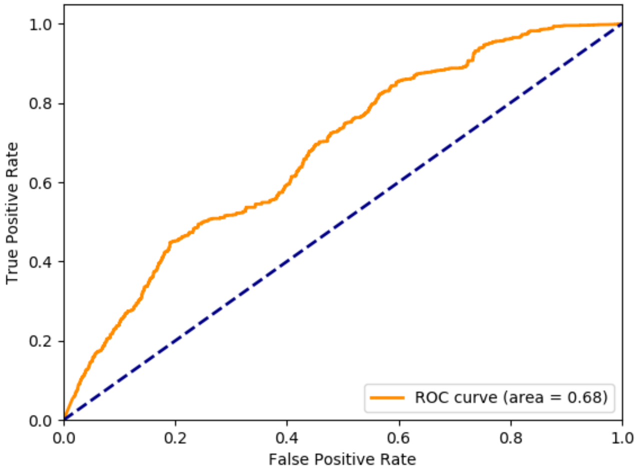
RTFM receives a T × D feature matrix F extracted from a video containing T snippets. Then, MTN captures the long and short-range temporal dependencies between snippet features to produce X = sθ(F). Next, we maximize the separability between abnormal and normal video features and train a snippet classifier using the top-k largest magnitude feature snippets from abnormal and normal videos.

## 5. Conclusions

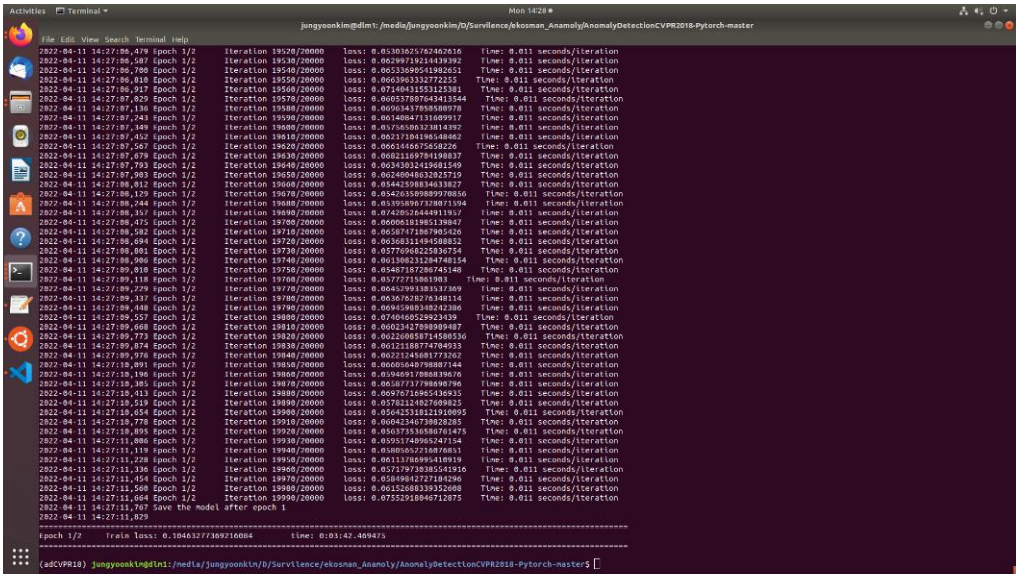
We use the frame-level area under the ROC curve (AUC) as the evaluation measure for all data sets. Moreover, following we also use average precision (AP) as the evaluation measure Larger AUC and AP values indicate better performance. Some recent studies recommend using the region-based detection criterion (RBDC) and the track-based detection criterion (TBDC) to complement the AUC measure, but these two measures are inapplicable in the weakly supervised setting. Thus, we focus on the AUC and AP measures

**Results using C3D Feature extraction**:

**ROC _AUC**:



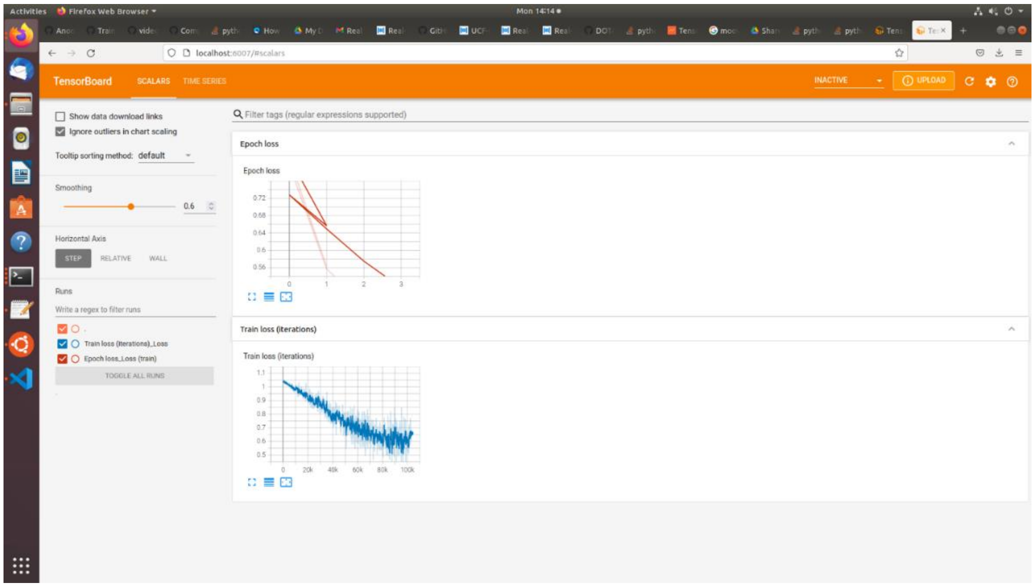
Training images for 2 epochs:


Activities
Terminal
Mon 14:28
jungyoonkim@dlm1: /media/jungyoonkim/D/Survilence/ekosman_Anamoly/AnomalyDetectionCVPR2018-Pytorch-master
File
Edit
View
Search
Terminal
Help

```
2022-04-11 14:27:06,479 Epoch 1/2       Iteration 19520/20000    loss: 0.05303625762462616    Time: 0.011 seconds/iteration
2022-04-11 14:27:06,587 Epoch 1/2       Iteration 19530/20000    loss: 0.06299719214439392    Time: 0.011 seconds/iteration
2022-04-11 14:27:06,700 Epoch 1/2       Iteration 19540/20000    loss: 0.06533690541982651    Time: 0.011 seconds/iteration
2022-04-11 14:27:06,810 Epoch 1/2       Iteration 19550/20000    loss: 0.0663963332772255    Time: 0.011 seconds/iteration
2022-04-11 14:27:06,917 Epoch 1/2       Iteration 19560/20000    loss: 0.07140431553125381    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,029 Epoch 1/2       Iteration 19570/20000    loss: 0.060537807643413544    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,136 Epoch 1/2       Iteration 19580/20000    loss: 0.06963437058580978    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,243 Epoch 1/2       Iteration 19590/20000    loss: 0.06140847131609917    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,349 Epoch 1/2       Iteration 19600/20000    loss: 0.05756506323814392    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,452 Epoch 1/2       Iteration 19610/20000    loss: 0.06217104196548462    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,567 Epoch 1/2       Iteration 19620/20000    loss: 0.0661446675658226    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,679 Epoch 1/2       Iteration 19630/20000    loss: 0.06821169704198837    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,793 Epoch 1/2       Iteration 19640/20000    loss: 0.06343032419681549    Time: 0.011 seconds/iteration
2022-04-11 14:27:07,903 Epoch 1/2       Iteration 19650/20000    loss: 0.06240048632025719    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,012 Epoch 1/2       Iteration 19660/20000    loss: 0.05442598834633827    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,129 Epoch 1/2       Iteration 19670/20000    loss: 0.054263509809970856    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,244 Epoch 1/2       Iteration 19680/20000    loss: 0.053958967328071594    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,357 Epoch 1/2       Iteration 19690/20000    loss: 0.07420526444911957    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,475 Epoch 1/2       Iteration 19700/20000    loss: 0.06006181985139847    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,582 Epoch 1/2       Iteration 19710/20000    loss: 0.06587471067905426    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,694 Epoch 1/2       Iteration 19720/20000    loss: 0.06368311494588852    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,801 Epoch 1/2       Iteration 19730/20000    loss: 0.05776968225836754    Time: 0.011 seconds/iteration
2022-04-11 14:27:08,906 Epoch 1/2       Iteration 19740/20000    loss: 0.061308231284748154    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,010 Epoch 1/2       Iteration 19750/20000    loss: 0.05487187286745148    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,118 Epoch 1/2       Iteration 19760/20000    loss: 0.05772715061903    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,229 Epoch 1/2       Iteration 19770/20000    loss: 0.06452993303537369    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,337 Epoch 1/2       Iteration 19780/20000    loss: 0.06367628276348114    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,448 Epoch 1/2       Iteration 19790/20000    loss: 0.06945988340242386    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,557 Epoch 1/2       Iteration 19800/20000    loss: 0.0740460529923439    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,668 Epoch 1/2       Iteration 19810/20000    loss: 0.06023427098989487    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,773 Epoch 1/2       Iteration 19820/20000    loss: 0.062260858714580536    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,874 Epoch 1/2       Iteration 19830/20000    loss: 0.06121188774704933    Time: 0.011 seconds/iteration
2022-04-11 14:27:09,976 Epoch 1/2       Iteration 19840/20000    loss: 0.06221245601773262    Time: 0.011 seconds/iteration
2022-04-11 14:27:10,091 Epoch 1/2       Iteration 19850/20000    loss: 0.06605640798807144    Time: 0.011 seconds/iteration
2022-04-11 14:27:10,196 Epoch 1/2       Iteration 19860/20000    loss: 0.05946917086839676    Time: 0.011 seconds/iteration
2022-04-11 14:27:10,305 Epoch 1/2       Iteration 19870/20000    loss: 0.06587737798690796    Time: 0.011 seconds/iteration
2022-04-11 14:27:10,413 Epoch 1/2       Iteration 19880/20000    loss: 0.06976716965436935    Time: 0.011 seconds/iteration
2022-04-11 14:27:10,519 Epoch 1/2       Iteration 19890/20000    loss: 0.05782124027609825    Time: 0.011 seconds/iteration
2022-04-11 14:27:10,654 Epoch 1/2       Iteration 19900/20000    loss: 0.056425318121910095    Time: 0.011 seconds/iteration
2022-04-11 14:27:10,778 Epoch 1/2       Iteration 19910/20000    loss: 0.06042346730828285    Time: 0.011 seconds/iteration
2022-04-11 14:27:10,895 Epoch 1/2       Iteration 19920/20000    loss: 0.056373536586761475    Time: 0.011 seconds/iteration
2022-04-11 14:27:11,006 Epoch 1/2       Iteration 19930/20000    loss: 0.05951740965247154    Time: 0.011 seconds/iteration
2022-04-11 14:27:11,119 Epoch 1/2       Iteration 19940/20000    loss: 0.05805652216076851    Time: 0.011 seconds/iteration
2022-04-11 14:27:11,228 Epoch 1/2       Iteration 19950/20000    loss: 0.06113786995410919    Time: 0.011 seconds/iteration
2022-04-11 14:27:11,336 Epoch 1/2       Iteration 19960/20000    loss: 0.057179730385541916    Time: 0.011 seconds/iteration
2022-04-11 14:27:11,454 Epoch 1/2       Iteration 19970/20000    loss: 0.05849842727184296    Time: 0.011 seconds/iteration
2022-04-11 14:27:11,560 Epoch 1/2       Iteration 19980/20000    loss: 0.06152680339352608    Time: 0.011 seconds/iteration
2022-04-11 14:27:11,664 Epoch 1/2       Iteration 19990/20000    loss: 0.07552918046712875    Time: 0.011 seconds/iteration
2022-04-11 14:27:11,767 Save the model after epoch 1
2022-04-11 14:27:11,829
==========================================================================================================================
Epoch 1/2     Train loss: 0.10463277369216084       time: 0:03:42.469475
==========================================================================================================================

(adCVPR18) jungyoonkim@dlm1:/media/jungyoonkim/D/Survilence/ekosman_Anamoly/AnomalyDetectionCVPR2018-Pytorch-master$ 
```

Loss Images:



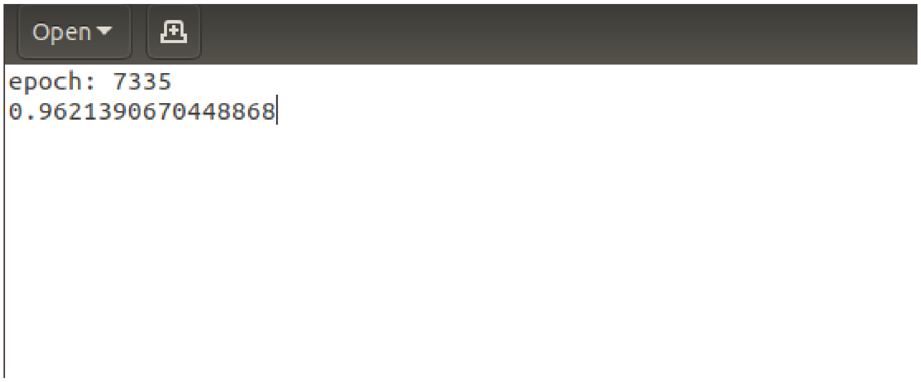
□ Show data download links
☑ Ignore outliers in chart scaling
Tooltip sorting method: default

Smoothing 0.6

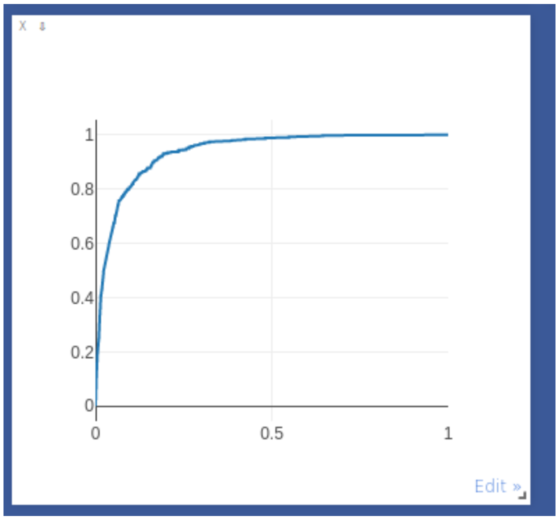
Horizontal Axis: STEP RELATIVE WALL

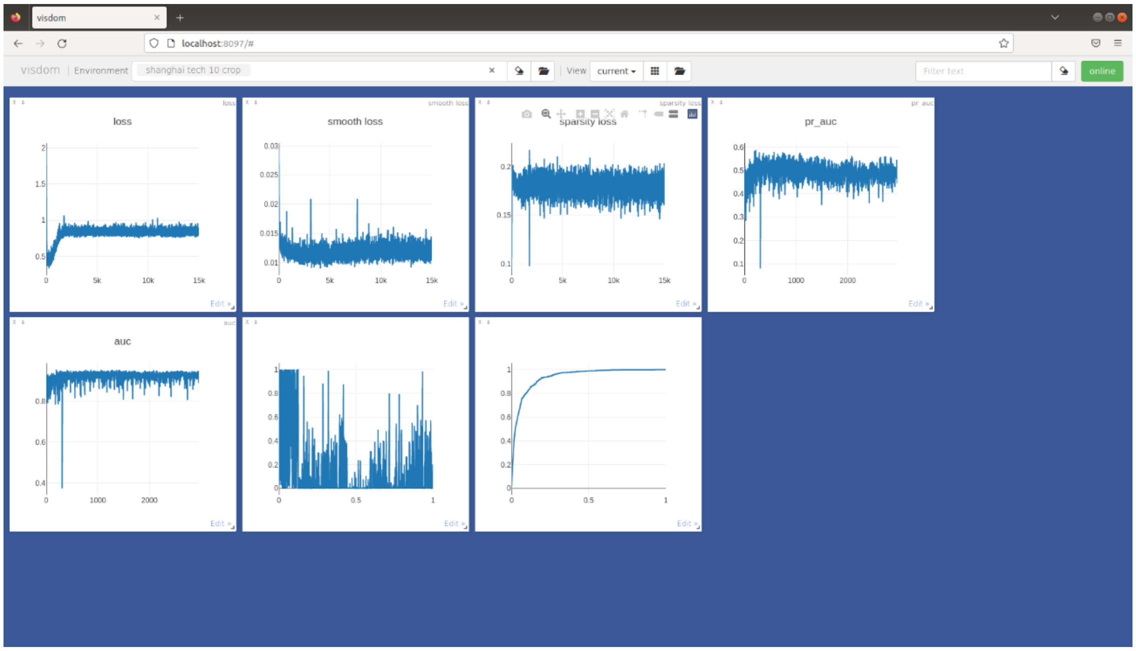
Runs
Write a regex to filter runs
☑ ○ .
☑ ○ Train loss (iterations)_Loss
☑ ○ Epoch loss_Loss (train)

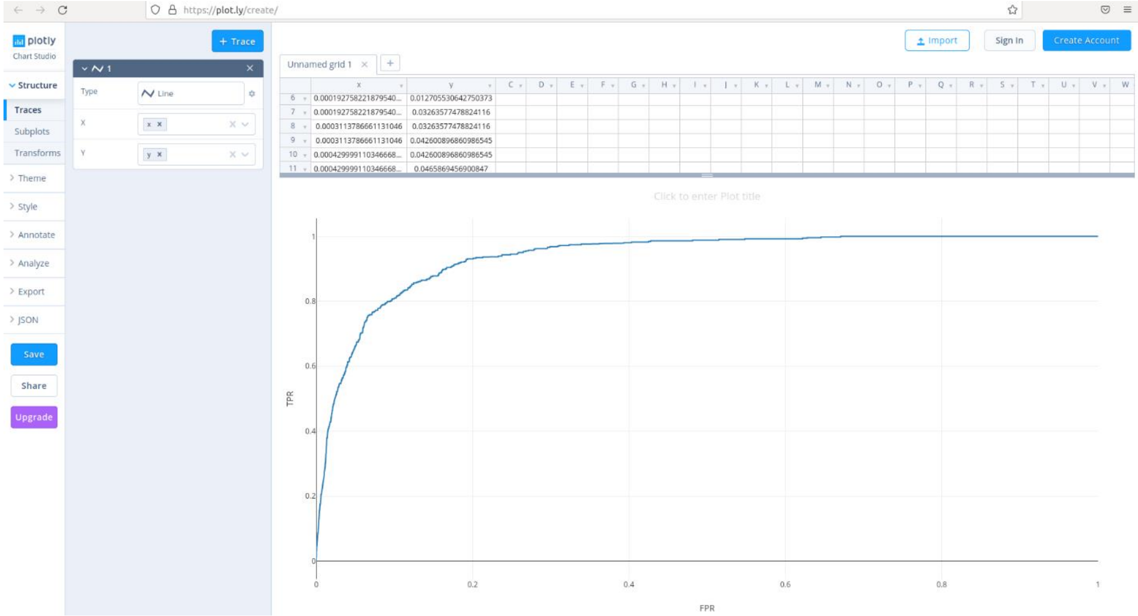
**Results using I3D Feature extraction**:



```
epoch: 7335
0.9621390670448868
```







|  | x | y | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 0.000192758221879540... | 0.012705530642750373 | | | | | | | | | | | | | | | | | | | | | |
| 7 | 0.000192758221879540... | 0.03263577478824116 | | | | | | | | | | | | | | | | | | | | | |
| 8 | 0.0003113786661131046 | 0.03263577478824116 | | | | | | | | | | | | | | | | | | | | | |
| 9 | 0.0003113786661131046 | 0.042600896860986545 | | | | | | | | | | | | | | | | | | | | | |
| 10 | 0.000429999110346668... | 0.042600896860986545 | | | | | | | | | | | | | | | | | | | | | |
| 11 | 0.000429999110346668... | 0.0465869456900847 | | | | | | | | | | | | | | | | | | | | | |

## 6. Contributions

- Anomaly Detection is ongoing research in Computer Vision (Deep Learning) world.

- Practical implementation of the techniques we have learned in the class like convolution network, drop outs, fully connected networks, pooling, regularizers and etc

- Apart from the convolution networks, here I tried implementing the I3D (Inflated 3 Dimensional networks). This is currently being used in the real world for any type of video classification models. The advantageous thing about I3D is that It uses 3D convolution to learn spatiotemporal information directly from videos

## 7. References

[1] Reynold Cheng Dmitri V. Kalashnikov Sunil Prabhakar Department of Computer Science, Purdue University

[2] YuTian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W. Verjans, Gustavo Carneiro – ICCV 2021

[3] R.Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. Technical Report TR 02-026, Department of Computer Science, Purdue University, West Lafayette, IN 47907, Nov. 2002.

[4] R. Cheng, S. Prabhakar, and D. Kalashnikov. Querying imprecise data in moving object environments. In ICDE'03, Proc. of IEEE Int'l Conf. on Data Engineering, Mar 5–8 2003