# B.Tech Project I : Anomaly Detection in Video frames

Rohan Bansal, 170070058

December 14, 2020

## 1 INTRODUCTION

Anomaly detection in videos refers to the identification of events that do not conform to expected behavior[1]. It is an important task because of its applications in video surveillance. Detecting abnormal events is a critical task too as it is based on what cameras capture, which is traditionally labor-intensive and requires non-stop human attention. Majority of the existing works employ reconstruction tasks for anomaly detection. In this project, we first develop a baseline based on reconstruction and develop on top of that to achieve novelty. If the reconstruction error, which is the error between the reconstruction and the previous frame is large we say an anomaly has been encountered. Majorly three kinds of losses are used to train the encoder-decoder, which are the flow loss, gradient loss and intensity loss. U-net architecture is used for the encoder as it has extensively been used for various reconstruction tasks. Reconstruction of frames however, has it's drawbacks. It does not consider the diversity of normal patterns explicitly, and the powerful representation capacity of CNNs allows to reconstruct abnormal video frames.

This project focuses on removing this drawback by incorporating a memory module consisting of prototypical features[2]. The task of the prototypes is to map normal events close to each other. Since there can be multiple anomalies on multiple scenes, we need more than one prototypes. These prototypes are like centroids in the normal events feature space. To map normal events as close as possible, we use the reconstruction losses as described above along with feature compactness and feature separateness loss. The datasets that we use are UCSD Ped2 [3] and CUHK Avenue [4].

# 2 LITERATURE REVIEW

Lots of efforts have been made for anomaly detection. Of all these work, the idea of feature reconstruction for normal training data is a commonly used strategy. Based on the features used, all existing methods can be roughly categorized into two categories: i) hand-crafted features based methods and ii) Deep Learning based methods.

Hand-crafted features based methods represent each video with some hand-crafted features including appearance and motion ones. Then a dictionary is learnt to reconstruct normal events with small reconstruction errors. It is expected that the features corresponding to abnormal events would have larger reconstruction errors. But since the dictionary is not trained with abnormal events and it is usually overcomplete, we cannot guarantee the expectation. To overcome the shortcoming of features, low-level spatial-temporal features, such as histogram of oriented gradients (HOG) [6], histogram of oriented flows (HOF) [7] are widely used. Sparse coding or dictionary learning is also a popular approach to encode the normal patterns. [5] suggests a sparsity based anomaly detection approach which implements anomaly detection at speeds of 140-150 FPS. It proposes to discard the sparse constraint and learn multiple dictionaries to encode normal scale-invariant patches. Graph formulation and kernel SVM based approaches have also been used[8]. Undirected graphs are generated using space time interest points which are calculated using eigenvalues of spatio-temporal second-moment matrices. Two interest points would have a graph between them if they have similarity measures above a certain threshold. Local activity and global activity recognition is done using SVM leveraging bag of graphs(BOG) as feature vector.

Deep learning approaches have demonstrated their successes in many computer vision tasks [18][11] as well as anomaly detection [13]. CNNs have allowed remarkable advances in anomaly detection over the last decade. [9] uses CNN to extract features followed by LSTM and finally SVM. Spatial Feature extraction is done using deep CNN of VGG16 architecture. The final fully connected layer is passed as an input to a Bi-LSTM to learn temporal features. Finally, SVM is used for classification. [10] introduces and studies a modified pre-trained convolutional neural network based on AlexNet[11]. (CNN) for detecting and localizing anomalies. The considered CNN is not trained from scratch but “just” fine-tuned.

Most of the approaches are data hungry and have limited generalization abilities. They usually need to be trained on a large number of videos from a target scene to achieve good results in that scene. [12] introduces few shot anomaly detection to address these limitations. It proposes a meta-learning based approach to this problem. During training, a model is learned that can quickly adapt to a new scene by using only a few frames from it. This is accomplished by learning from a set of tasks, where each task mimics the few-shot scene-adaptive anomaly detection scenario using videos from an available scene. [13] extends this problem and introduces Domain Adaptive Few shot Learning. It deals with a problem in addition to Few shot learning, which is that the target classes may not be a part of the same domain as the source classes. It proposes a novel domain-adversarial prototypical network(DAPN) for the DA-FSL problem. Combining existing methods for FSL and DA won't

work because here the target classes are not in the same embedded space as source classes.

# 3 METHODOLOGY

## 3.1 BASELINE ARCHITECTURE

The first step in the project was to look for a baseline architecture and develop on top of that. Recently, prediction learning is attracting more and more researchers' attention in light of its potential applications in unsupervised feature learning for video representation. Since anomaly detection is the identification of events that do not conform the expectation, it is more natural to predict future video frames based on previous video frames, and compare the prediction with its ground truth for anomaly detection. Thus the decided architecture leverages video prediction for anomaly detection. Three kinds of losses, that put constraints on both spacial and temporal information are used to get the best reconstruction[14]. The error between the reconstruction and the ground truth image is used to decide on the abnormality of the frame. Given a video with consecutive t frames $I_1$, $I_2$, . . . , $I_t$, they are sequentially stacked and all these frames are used to predict a future frame $I_{t+1}$.

Generative Adversarial network is used for the frame reconstruction with U-Net[15] acting as the generator. Using U-Net suppresses gradient vanishing and results in information symmetry. The details of the network have been illustrated in Figure 3.2[14]. To reconstruct the frame as close as possible, three kind of losses are used which are described below :-

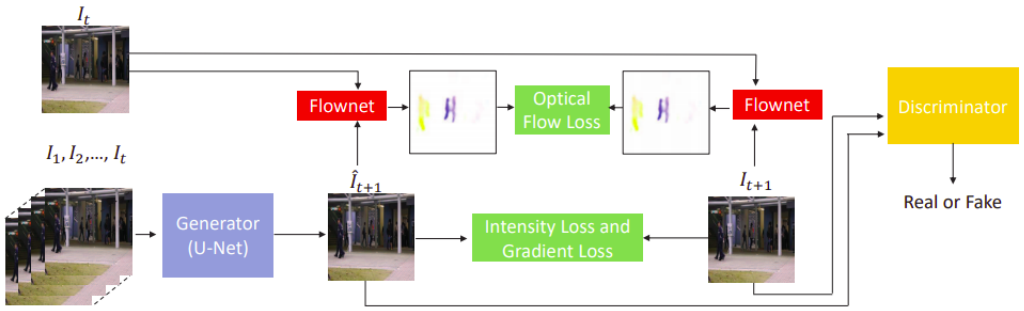



Figure 3.1: Pipeline of the baseline architecture

**Intensity Loss** : The intensity penalty guarantees the similarity of all pixels in RGB space

$$L_{int}(I', I) = ||I' - I||_2^2 \tag{3.1}$$

**Gradient Loss** : The gradient loss tries to sharpen the reconstructed images

$$L_{gd}(I', I) = \sum_{i,j} |||I'_{i,j} - I'_{i-1,j}| - |I_{i,j} - I_{i-1,j}|||_1 + |||I'_{i,j} - I'_{i,j-1}| - |I_{i,j} - I_{i,j-1}|||_1 \tag{3.2}$$

**Flow loss** : A temporal loss is defined as the difference between optical flow of prediction frames and ground truth. Flownet[16] is used for optical flow estimation.

$$L_{op}(I', I) = ||f(I'_{t+1}, I_t) - f(I_{t+1}, I_t)||_1 \tag{3.3}$$

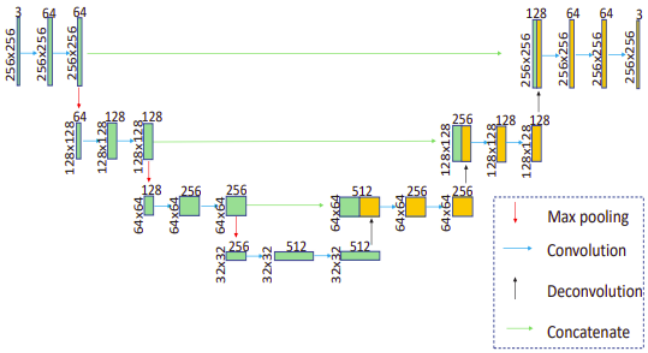



Figure 3.2: U-Net architecture used as the generator

Pretrained $f$ is used in this baseline.

**Adversarial training** : As mentioned before, GAN module is used to generate realistic images. U-net is used as the generator. For Discriminator, patches of the image are used. Generator learns to generate frames that are hard to be classified by the Discriminator, while Discriminator aims to discriminate the frames generated by the Generator.

**Training** : Four consecutive frames are taken at a time to predict the fifth frame. The frames dimensions are set at 256x256.

**Testing** : The difference between predicted frame I' and its ground truth I can be used for anomaly prediction. MSE is one popular way to measure the quality of predicted images by computing a Euclidean distance between the prediction and its ground truth of all pixels in RGB color space. Here, Peak Signal to Noise Ratio (PSNR) is used as it is a better way for image quality assessment.

$$PSNR(I, I') = 10log_{10}\frac{[max_{I'}]^2}{\frac{1}{N}\sum_{i=0}^{N}(I_i - I'_i)^2} \tag{3.4}$$

High PSNR of the t-th frame indicates that it is more likely to be normal. A threshold can be set to distinguish between the normal and abnormal frames.

## 3.2 RUNNING THE CODE

The code was run on the UCSD Ped1 dataset. The dataset consists of 36 Training and 36 testing videos, both having normal and abnormal frames. The loss functions were run separately and together to capture what each constraint is doing.

**Training** : The number of iterations is set at 1000. Four consecutive frames are taken at a time to predict the fifth frame. The frames dimensions are set at 256x256. First, all the loss functions are used. Below are the reconstructions and SNR's at different iterations.

```
GeneratorModel : Step 300, lr = 0.000100
                 Global      Loss :  0.07542455
                 intensity   Loss : (0.0050 * 1.0000 = 0.0050)
                 gradient    Loss : (0.0636 * 1.0000 = 0.0636)
                 adversarial Loss : (0.1192 * 0.0500 = 0.0060)
                 flownet     Loss : (0.0774 * 0.0100 = 0.0008)
                 PSNR  Error      :  28.991758
```

Figure 3.3: Reconstructed frame after 300 iterations and the respective SNR

```
DiscriminatorModel: Step 800 | Global Loss: 0.234964, lr = 0.000010
GeneratorModel : Step 800, lr = 0.000100
                 Global      Loss :  0.037823737
                 intensity   Loss : (0.0013 * 1.0000 = 0.0013)
                 gradient    Loss : (0.0294 * 1.0000 = 0.0294)
                 adversarial Loss : (0.1363 * 0.0500 = 0.0068)
                 flownet     Loss : (0.0351 * 0.0100 = 0.0004)
                 PSNR  Error      :  34.920628
```

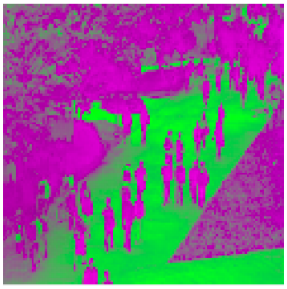
Figure 3.4: Reconstructed frame after 800 iterations and the respective SNR

```
DiscriminatorModel: Step 1000 | Global Loss: 0.237883, lr = 0.000010
GeneratorModel : Step 1000, lr = 0.000100
                 Global      Loss :  0.034664903
                 intensity   Loss : (0.0009 * 1.0000 = 0.0009)
                 gradient    Loss : (0.0267 * 1.0000 = 0.0267)
                 adversarial Loss : (0.1361 * 0.0500 = 0.0068)
                 flownet     Loss : (0.0290 * 0.0100 = 0.0003)
                 PSNR  Error      :  36.67826
```

Figure 3.5: Reconstructed frame after 1000 iterations and the respective SNR

In Figure 3.3, Figure 3.4 and Figure 3.5 we can see that the reconstruction improves as the GAN is trained. The increase in the SNR for each iteration also signifies the improvement in reconstruction.

Now we try to analyse the importance of each constraint by carrying out the reconstruction using two losses at a time.



```
DiscriminatorModel: Step 1000 | Global Loss: 0.000051, lr = 0.000010
GeneratorModel : Step 1000, lr = 0.000100
                 Global      Loss :  0.056682616
                 intensity   Loss : (0.0000 * 1.0000 = 0.0000)
                 gradient    Loss : (0.0306 * 1.0000 = 0.0306)
                 adversarial Loss : (0.5000 * 0.0500 = 0.0250)
                 flownet     Loss : (0.1094 * 0.0100 = 0.0011)
                 PSNR  Error      :  9.324005
```

Figure 3.6: Reconstructed frame using only gradient and flow loss and the respective SNR

```
DiscriminatorModel: Step 1000 | Global Loss: 0.249845, lr = 0.000010
GeneratorModel : Step 1000, lr = 0.000100
                 Global      Loss :  0.0078487005
                 intensity   Loss : (0.0013 * 1.0000 = 0.0013)
                 gradient    Loss : (0.0000 * 1.0000 = 0.0000)
                 adversarial Loss : (0.1262 * 0.0500 = 0.0063)
                 flownet     Loss : (0.0206 * 0.0100 = 0.0002)
                 PSNR  Error      :  34.778656
```

Figure 3.7: Reconstructed frame using only intensity and flow loss and respective SNR

```
DiscriminatorModel: Step 1000 | Global Loss: 0.237598, lr = 0.000010
GeneratorModel : Step 1000, lr = 0.000100
                 Global      Loss :  0.03640741
                 intensity   Loss : (0.0010 * 1.0000 = 0.0010)
                 gradient    Loss : (0.0288 * 1.0000 = 0.0288)
                 adversarial Loss : (0.1317 * 0.0500 = 0.0066)
                 flownet     Loss : (0.0000 * 0.0100 = 0.0000)
                 PSNR  Error      :  36.029972
```

Figure 3.8: Reconstructed frame using only intensity and gradient loss and respective SNR

The above three figures(Figure 3.6, Figure 3.7 and Figure 3.8) signifies the importance of each loss function. Without the intensity loss, the color reconstruction does not happen which results in a very low SNR. Without the gradient loss, the reconstruction is blurred out a bit. Thus, gradient loss is responsible for the sharpness of the image. Without the flow loss, we still get a pretty good reconstruction. The importance of flow loss arises when there is a sudden change in the scene.

**Testing**: The trained model with all the losses is tested on the test set consisting of 36 different videos. Below is shown a couple of test frames with their SNR's :

Figure 3.9 depicts a normal frame, we see no abnormal activity happening in that image. Figure 3.10 depicts an abnormal frame, as we can see a rider riding a bicycle which is

```
video = 02 / 36, i = 4 / 11, psnr = 33.015442
video = 02 / 36, i = 5 / 11, psnr = 33.044666
video = 02 / 36, i = 6 / 11, psnr = 32.923862
video = 02 / 36, i = 7 / 11, psnr = 32.695908
video = 02 / 36, i = 8 / 11, psnr = 33.262806
video = 02 / 36, i = 9 / 11, psnr = 33.369194
```

Figure 3.9: An example of Normal image with SNR

```
video = 06 / 36, i = 5 / 11, psnr = 26.158175
video = 06 / 36, i = 6 / 11, psnr = 26.016529
video = 06 / 36, i = 7 / 11, psnr = 26.110794
video = 06 / 36, i = 8 / 11, psnr = 26.426878
video = 06 / 36, i = 9 / 11, psnr = 26.679722
```

Figure 3.10: An example of abnormal image with SNR

deviation from normal behaviour. Also, observe the difference in the SNR's, the SNR of normal frame being higher than that of abnormal frame.

## 3.3 Prototypical Networks

After deciding the baseline, the next step is to extend it to using prototypical networks. [2] proposes an unsupervised learning technique for anomaly detection using prototypical networks. They assume that a single prototypical feature is not enough to represent various patterns of normal data. They propose a memory module for anomaly detection, where individual items in the memory correspond to prototypical features of normal patterns. The overall framework is shown in Figure 3.11[2]. We leverage their work on top of our baseline for anomaly detection.

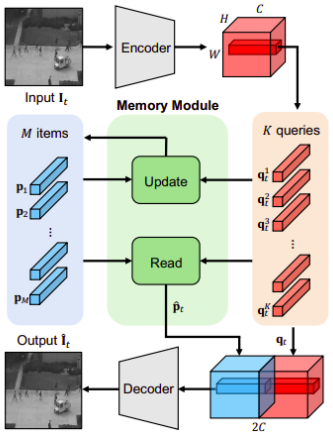



Figure 3.11: Overview of the prototypical network framework

The model mainly consists of three components: an encoder, a memory module, and a decoder. The encoder and decoder are same as those used in the baseline[14]. The proposed memory module contains M items recording various prototypical patterns of normal data. The encoder output is denoted by query $q_t$. These queries are input to the memory module and the the prototypes or the memory items are updated. Cosine similarity is computed between each query and memory item. Then these memory items are updated as weighted product of probabilities based on these cosine similarities. Concatenation of memory items and the queries is given as input to the decoder. This enables the decoder to reconstruct the input frame using normal patterns in the items, lessening the representation capacity of CNNs, while understanding the normality. To train the model, these losses are used:

**Reconstruction Loss** : In [2], only the intensity loss has been used. We plan to use all the three losses(intensity loss, gradient loss and flow loss) from our baseline to achieve best reconstruction.

**Feature Compactness Loss** : The feature compactness loss encourages the queries to be close to the nearest item in the memory, reducing intra-class variations.

**Feature Separateness Loss** : The Feature Compactness Loss makes similar queries and memory items to be close to each other. We also want these memory items to be far away from one another to separate different patterns in the normal data. For this purpose, Feature Separateness Loss is used.

# 4 CONCLUSION AND FUTURE WORK

In this project, we study and develop an architecture to tackle the problem of Anomaly detection in video frames. First, we work out a baseline that uses reconstruction of frames to recognise anomalies. Three kinds of losses are used in the baseline to achieve best reconstruction. We run some experiments to study about the performance of the baseline and to gain some better understanding of the constraints and metrics used. On top of our baseline architecture, we plan to implement prototypical networks. This helps in lessening the representation capacity of CNN's. Next, we need to study more about the prototypes and what kind of prototypes work best for our model.

Our model requires large amount of videos to achieve good results. This shortcoming has been worked out in a previous work[12] where they use few shot learning for anomaly detection. As a future work, we plan to explore how few shot learning can be leveraged in our architecture to make it more useful for practical purposes. Another work[17] proposes few shot learning using prototypical networks. As we are employing prototypical networks in this project, we can think of it as a future work too.

# 5 REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3):15, 2009

[2] Hyunjong Park and Jongyoun Noh and Bumsub Ham. Learning Memory-guided Normality for Anomaly Detection, arXiv:2003.13228

[3] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. IEEE TPAMI, 2013. 2

[4] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 FPS in MATLAB. In ICCV, 2013

[5] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In ICCV, pages 2720–2727, 2013.

[6] N. Navneet and B. Triggs. Histograms of oriented gradients for human detection. In CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005

[7] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In ECCV, pages 428–441. Springer, 2006.

[8] Singh, Dinesh and C. K. Mohan. “Graph formulation of video activities for abnormal activity recognition.” Pattern Recognit. 65 (2017): 265-272.

[9] K. Vignesh, G. Yadav and A. Sethi, "Abnormal Event Detection on BMTT-PETS 2017 Surveillance Challenge," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, 2017, pp. 2161-2168, doi: 10.1109/CVPRW.2017.268.

[10] Mohammad Sabokrou and Mohsen Fayyaz and Mahmood Fathy and Zahra Moayedd and Reinhard klette. Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes, arXiv:1609.00866, 2017

[11] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, Advances Neural Information Processing Systems (2012) 1097–1105.

[12] Yiwei Lu and Frank Yu and Mahesh Kumar Krishna Reddy and Yang Wang. Few-shot Scene-adaptive Anomaly Detection, arXiv:2007.07843, 2020

[13] Domain-Adaptive Few-Shot Learning. An Zhao and Mingyu Ding and Zhiwu Lu and Tao Xiang and Yulei Niu and Jiechao Guan and Ji-Rong Wen and Ping Luo, arXiv:2003.08626, 2020

[14] Future Frame Prediction for Anomaly Detection – A New Baseline. Wen Liu and Weixin Luo and Dongze Lian and Shenghua Gao, arXiv:1712.09867, 2018

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 234–241. Springer, 2015.

[16] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In ICCV, pages 2758–2766, 2015

[17] Jake Snell and Kevin Swersky and Richard S. Zemel. Prototypical Networks for Few-shot Learning, arXiv:1703.05175, 2017