B.Tech Project II : Anomaly Detection in Videos

Rohan Bansal, 170070058

June 30, 2021

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 continue our work from BTP1. In BTP1, we decided our reconstruction baseline and touched upon prototypical networks. We also stated few shot learning as a part of future work. Thus, in this project, we combine the two concepts of memory model based prototype networks with few shot learning.

This project focuses on implementing the prototypical networks using the process described by the few shot learning paper[12]. After the implementation, the results are compiled by varying majorly three parameters - number of scenes in training(N), the number of iterations, and the number of frames of new scene used in meta-testing(k). The datasets that we use are UCSD Ped2 [3], CUHK Avenue [4] and the IITB Corridor dataset [5].

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 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 and implement it in reconciliation with few shot learning.

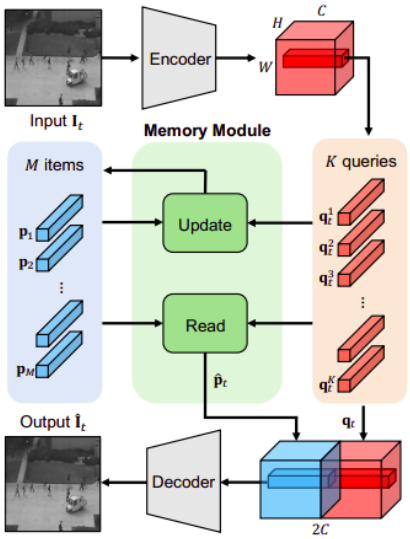

Figure 3.1: Overview of the prototypical network framework

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.

3.2 Few Shot learning

Existing anomaly detection approaches assume that model learned from training videos can be directly used in test videos. This is reasonable only if the training and test videos are from the same scene. If the testing is done on a completely different scene, the performance drops. One way is to train on a set of diverse scenes, which is not an ideal approach as such a model requires a large capacity. In many real-world applications, the anomaly detection system is often deployed on edge devices with limited computing powers. As a result, even if we can train a huge model that generalizes well to different scenes, we may not be able to deploy this model.For practical purposes, cameras are generally deployed to work on a single scene. So camera only needs to work properly on that scene. Few shot learning introduces a method by which a pre-trained model adapts to a particular scene by using just few frames from that scene.

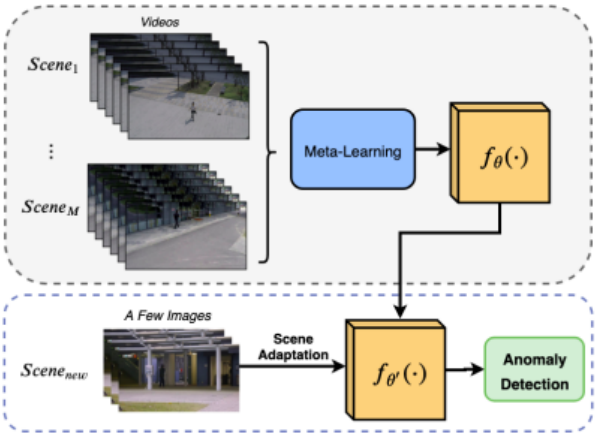

Figure 3.2: Overview of Few shot learning

Few shot learning employs a meta-learning approach, MAML[19], which consists of meta-training and meta-testing. In meta-learning, a model is trained from a large number of few-shot scene-adaptive anomaly detection tasks constructed using the videos available in meta-training, where each task corresponds to a particular scene. Basically, during meta-training, the model learns to adapt on a unknown scene, which is the primary objective. IN meta-testing, few frames from an unknown scene(not encountered during meta-training)

are provided on which the model adapts. This adapted model is expected to achieve high performance on the unknown scene.

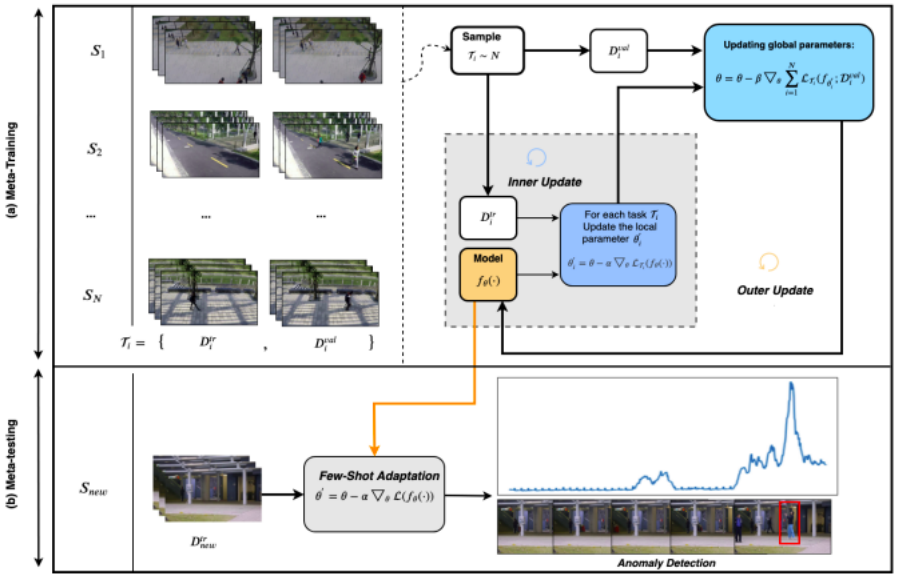

Fig. 2. An overview of our proposed approach. Our approach involves two phases: (a) meta-training and (b) meta-testing. In each iteration of the meta-training (a), we first sample a batch of N scenes $S_1, S_2, ..., S_N$. We then construct a task $\mathcal{T}_i = \{D_i^{tr}, D_i^{val}\}$ for each scene S_i with a training set D_i^{tr} and a validation set D_i^{val}. D_i^{tr} is used for *inner update* through gradient descent to obtain the updated parameters θ_i' for each task. Then D_i^{val} is used to measure the performance of θ_i'. An *outer update* procedure is used to update the model parameters θ by taking into account of all the sampled tasks. In meta-testing (b), given a new scene S_{new}, we use only a few frames to get the adapted parameters θ' for this specific scene. The adapted model is used for anomaly detection in other frames from this scene.

Figure 3.3: Detailed representation of MAML

3.3 Overall mechanism

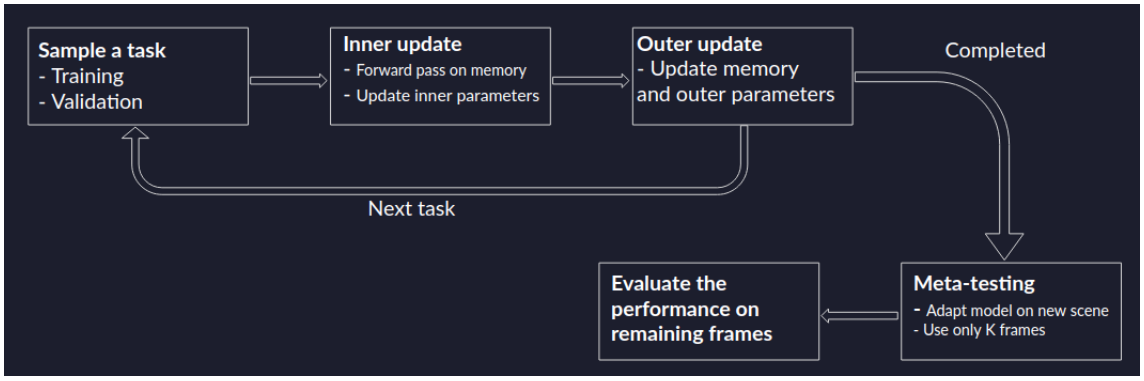

Figure 3.4: Overall approach

The most challenging task was to implement the prototypical networks in accordance with few shot learning. Figure 3.4 presents the overall mechanism of the combined implementation. At every update, the memory module is updated based on the reconstruction loss, feature compactness loss and the feature separation loss. After iterating through all the training scenes, the model adapts to the new scene using just a few frames. The final stage is to test the model on remaining frames of the new scene.

4 Results

DATASETS
Training :

- Shanghai Tech(shortened version) - 13 scenes, 5 videos per scene

Testing :

- UCSD Pedestrian 2 - 12 videos from one scene
- CUHK avenue : 21 videos from one scene
- IITB Corridor dataset [18]

PARAMETERS

- N = number of scenes
- it = number of iterations in each epoch
- k = k-shots, i.e., number of frames of new testing scene

N	No. of iterations	AUC score		
		UCSD Ped2	CUHK Avenue	IITB Dataset
4	10	91.74	79.05	67.89
	100	72.01	82.07	70.09
	500	68.215	81.55	65.43
5	10	87.308	79.45	70.08
	100	68.95	82.86	69.65
	500	64.011	75.53	68.03
6	10	85.32	80.04	70.54
	100	88.487	82.9	66.62
	500	51.219	74.69	66.62

Figure 4.1: different N and it

For the first set of results, we change the number of training scenes(N) and iterations for a particular N. The results are shown in Figure 4.1. For a fixed N, the AUC score decreases as we increase the number of iterations. This could be due to overfit occuring due to more training on the training scenes. This could have led to a reduction in the adaptability power of the model. Increasing N doesn't cause any discernible change, which means that there's no need of using a lot of scenes for training.

Dataset	1-shot (K=1)	5-shot (K=5)	7-shot (K=7)
UCSD Ped2	87.481	89.804	91.544
CUHK Avenue	81.92	82.95	80.636

Shanghai Tech

Target	Methods	1-shot (K=1)	5-shot (K=5)	10-shot (K=10)
UCSD Ped 1	Pre-trained	73.1	73.1	73.1
	Fine-tuned	76.99	77.85	78.23
	Ours	**80.6**	**81.42**	**82.38**
UCSD Ped 2	Pre-trained	81.95	81.95	81.95
	Fine-tuned	85.64	89.66	91.11
	Ours	**91.19**	**91.8**	**92.8**
CUHK Avenue	Pre-trained	71.43	71.43	71.43
	Fine-tuned	75.43	76.52	77.77
	Ours	**76.58**	**77.1**	**78.79**

Figure 4.2: varying k; comparison with the few shot paper

IN fig 4.2, we present the best AUC achieved for different values of k. Understandably, the AUC increases on increasing k, because there are more frames to adapt leading to better adaptation. But it is expected that the performance saturates after a certain value of k. In comparison with the paper, we are able to surpass the performance for CUHK Avenue. We were not able test the model for k=10 due to computational restrictions.

Target	Methods	K=1	K=5	k=7
UCSD Ped2	N=1	80.702	88.508	85.563
	N=5	87.481	89.43	87.49
CUHK Avenue	N=1	81.026	81.53	80.636
	N=5	81.92	82.95	82.89

Target	Methods	K=1	K=5	K=10
Ped1	Fine-tuned	76.99	77.85	78.23
	Ours ($N=1$)	79.94	80.44	78.88
	Ours ($N=5$)	**80.6**	**81.42**	**82.38**
Ped2	Fine-tuned	85.64	89.66	91.11
	Ours ($N=1$)	90.73	91.5	91.11
	Ours ($N=5$)	**91.19**	**91.8**	**92.8**
CUHK	Fine-tuned	75.43	76.52	77.77
	Ours ($N=1$)	76.05	76.53	77.31
	Ours ($N=5$)	**76.58**	**77.1**	**78.79**

Figure 4.3: varying N and k; comparison with the few shot paper

IN fig 4.3, we view the performance for two values of N. For N=5, the performance is better than for N=1, as there are more training scenes to help generalise. The variation of k is same as that in the fig 4.2. Here aslo, we surpass the performance for CUHK avenue.

5 Conclusion and Future work

In this project, we study and develop an architecture to tackle the problem of Anomaly detection in video frames. We carry forward our work from BTP-1 and extend it to few shot learning. We implement a novel approach of combining memory module based prototypical networks with few shot learning. Few shot learning only needs a few frames to work well a scene, therefore is quite advantageous from a practical point of view. Whereas prototypical networks help in tackling the drawback of representation capacity of CNN's.We evaluate our model on 3 datasets - UCSD Ped2, CUHK Avenue and IITB Corridor dataset by varying N and k. We also compare these results with the few shot paper[12]. We are able to surpass the paper for one of the datasets (CUHK Avenue).

As a future work, we hope to make some changes to the baseline to improve the performance even further. We would also like to experiment on more combinations of values or some another dataset.

6 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

[18] https://rodrigues-royston.github.io/Multi-timescale_Trajectory_Prediction/

[19] Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017)