Anomaly Detection in Surveillance Videos using Deep Learning

by

Yiwei Lu

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

Master of Science

Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Thesis advisor

Yang Wang

Author

Yiwei Lu

Anomaly Detection in Surveillance Videos using Deep Learning

Abstract

We address the problem of anomaly detection in videos. The goal is to identify unusual behaviors automatically by learning exclusively from normal videos. Most existing approaches are usually 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. In this thesis, we propose a novel few-shot scene-adaptive anomaly detection problem to address the limitations of previous approaches. Our goal is to learn to detect anomalies in a previously unseen scene with only a few frames. A reliable solution for this new problem will have huge potential in real-world applications since it is expensive to collect a massive amount of data for each target scene. We propose a meta-learning based approach for solving this new problem; extensive experimental results demonstrate the effectiveness of our proposed method.

Contents

Abstract ii
Table of Contents iv
List of Figures v
List of Tables vii
Publications ix
Acknowledgments x
Dedication xi

1 Introduction 1
1.1 General Introduction 1
1.2 Thesis Outline 6

2 Related Work 7
2.1 Anomaly Detection with Hand-crafted Features 7
2.2 Anomaly Detection with Deep Learning 8
2.3 Few-Shot and Meta Learning 9

3 Background and Problem Setup 10
3.1 Variational Autoencoder 10
3.2 Variational Recurrent Neural Network 11
3.3 Conditional Generative Adversarial Network 11
3.4 Problem Setup 12
3.4.1 Future Frame Prediction for Anomaly Detection 12
3.4.2 Few-Shot Scene-Adaptive Anomaly Detection 12

4 Our Approach 14
4.1 MAML for Scene-Adaptive Anomaly Detection 16
4.1.1 Tasks in Meta-learning 16
4.1.2 Meta-Training 17
4.1.3 Meta-Testing 18
4.2 Backbone Architecture 19

4.2.1 r-GAN 20
Generator 20
Discriminator 21
Anomaly Detection 21
4.2.2 Conv-VRNN 22
Prior Distribution in VAE 22
Encoder 24
Recurrence 24
Decoder 25
Model Learning 25

5 Datasets and Experimental Setup 26
5.1 Datasets 26
5.2 Evaluation Metrics 29
5.3 Implementation Details 29
5.4 Baselines 30

6 Experimental Results 31
6.1 Backbone Architectures 31
6.1.1 Conv-VRNN vs r-GAN 31
6.1.2 Analysis on Conv-VRNN 33
6.2 Few-shot Scene-adaptive Anomaly Detection 38
6.2.1 Results on Shanghai Tech 39
6.2.2 Cross-dataset Testing 40
6.2.3 Ablation Study 41
6.2.4 Qualitative Results 41

7 Conclusion 42

Bibliography 48

List of Figures

1.1 An example of our proposed video anomaly detection method. Our method uses a future frame prediction framework. Given several observed frames in a video, our model predicts the future frame. If the future frame is an anomaly, the predicted future frame is likely to be very different from the actual future frame. This prediction error allows us to detect the anomaly in a video. 2

1.2 An overview of our proposed problem setting. During training (1st row), we have access to videos collected from M different camera scenes. From such training data, we use a meta-learning method to obtain a model f_θ with parameters θ. Given a target scene (2nd row), we have access to a small number of frames from this target scene. Our goal is to produce a new model $f_{\theta'}$ where the model parameters θ' are specifically adapted to this scene. Then we can use $f_{\theta'}(\cdot)$ to perform anomaly detection on the remaining videos from this target scene. . . 3

4.1 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. 15

4.2 An overview of our backbone architecture. Our anomaly detection model consists of a Sequential Image Generator $\mathcal{G}(\cdot)$ and a Discriminator $\mathcal{D}(\cdot)$. Given an image sequence $I_1, I_2, ..., I_t$ as the input, $\mathcal{G}(\cdot)$ outputs a prediction $\hat{I}_{t+1}$ of the next frame. A prediction loss is computed between $\hat{I}_{t+1}$ and the actual frame I_{t+1} for parameter updating. $\mathcal{D}(\cdot)$ takes both $\hat{I}_{t+1}$ and I_{t+1} as its input to classify which one is real and which one is fake. These two networks are trained adversarially to obtain a good $\mathcal{G}(\cdot)$ that is able to fool $\mathcal{D}(\cdot)$. 19

4.3 An overview of our proposed Conv-VRNN model at one time-step of a sequence. Our model requires 4 steps to process the input: (a) calculating the prior distribution in VAE; (b) encoder for posterior distribution and latent variable; (c) recurrence module for sequence modelling; (d) decoder for prediction. 23

5.1 Example frames from the datasets used for meta-training. The first row shows examples of different scenes from the Shanghai Tech dataset. The second row shows examples of different scenes from the UCF crime dataset. 27

5.2 Example frames from datasets used in meta-testing. The first row shows examples of normal frames for four datasets, and the second row shows the abnormal frames. Note that training videos only contain normal frames. Videos with abnormal frames are only used for testing. 27

6.1 Examples of anomaly detection on three datasets using Conv-VRNN. We plot the anomaly score of our model and the ground-truth anomaly score. Again, the bounding boxes are for visualization purpose. 34

6.2 ROC curves of our Conv-VRNN method, Conv-VAE (w/o optical flow) and Conv-VAE (with optical flow) on three datasets. 35

6.3 Qualitative results on three benchmark datasets using a pre-trained model on the Shanghai Tech dataset. Different columns represent results on different datasets. Each row shows few-shot scene-adaptive anomaly detection results with different numbers of training samples K. The red bounding boxes showing the abnormal event localization are for visualization purposes. They are not the outputs of our model which only predicts an anomaly score at the frame level. 40

List of Tables

6.1 Comparison of anomaly detection performance among our backbone architecture (r-GAN and Conv-VRNN), and existing state-of-the-art in the standard setup (i.e. without scene adaptation). We report AUC (%) of different methods on UCSD Ped1 (Ped1), UCSD Ped2 (Ped2), CUHK Avenue (CUHK) and Shanghai Tech (ST) datasets. We use the same train/test split as prior work on each dataset. Our proposed backbone architecture *r-GAN* outperforms the existing state-of-the-art on almost all datasets. 32

6.2 Comparison of anomaly detection in terms of AUC (%) of different methods on the UR fall detection dataset. This dataset contains depth images. We simply treat those as RGB images. Our proposed backbone architecture r-GAN is state-of-the-art among all the methods. 33

6.3 Comparision of Conv-VAEs versus Conv-VRNN in terms of AUC on three datasets. 33

6.4 Evaluation of different combinations of various loss terms in the objective functions in our Conv-VRNN network on the Ped1 dataset. The results show that the combination of all loss terms gives the best performance. 35

6.5 Comparison between our Conv-VRNN model with different VAE-based models (with or without optical flow features). Our proposed Conv-VRNN outperforms Conv-VAE (with optical flow) even if our model does not use optical flow features. 36

6.6 Comparison of K-shot scene-adaptive anomaly detection on the Shanghai Tech dataset. We use 6 scenes for training and the remaining 7 scenes for testing. We report results in terms of AUC (%) for $K = 1, 5, 10$. The proposed approach outperforms two baselines. . . . 36

6.7 Comparison of K-shot ($K = 1, 5, 10$) scene-adaptive anomaly detection under the cross-dataset testing setting. We report results in terms of AUC (%) using the Shanghai Tech dataset for meta-training. We compare our results with two baseline methods. Our results demonstrate the effectiveness of our method on few-shot scene-adaptive anomaly detection. 37

6.8 Comparison of K-shot ($K = 1, 5, 10$) scene-adaptive anomaly detection under the cross-dataset testing setting. We report results in terms of AUC (%) using the UCF crime dataset for meta-training. We compare our results with two baseline methods. Our results demonstrate the effectiveness of our method on few-shot scene-adaptive anomaly detection. 38

6.9 Ablation study for using different number of sampled tasks ($N = 1$ or $N = 5$) during each epoch of meta-training. The results show that even the performance of training with one task is better than fine-tuning. However, a larger number of tasks is able to train an improved model. 39

Publications

There are two published/submitted papers related to this thesis:

(1) Future Frame Prediction Using Convolutional VRNN for Anomaly Detection, Yiwei Lu, Mahesh Kumar Krishna Reddy, Seyed shahabeddin Nabavi and Yang Wang, accepted to the 2019 IEEE International Conference on Advanced Video and Signal-based Surveillance.

(2) Few-shot Scene-adaptive Anomaly Detection, Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy and Yang Wang, submitted to the 2020 European Conference on Computer Vision.

Acknowledgments

I would like to begin by thanking my advisor, my committee, my parents, my significant other, and all the people who have supported me along the way.

This thesis is dedicated to somebody special. You know who you are.

Chapter 1

Introduction

1.1 General Introduction

Recently, anomaly detection is becoming an essential problem in video surveillance. Given a video, the goal is to identify frames where abnormal events happen. This is a very challenging problem since the definition of "anomaly" is ambiguous – any event that does not conform to "normal" behaviours can be considered an anomaly. As a result, we cannot solve this problem via a standard classification framework since it is impossible to collect training data that covers all possible abnormal events.

A lot of prior work (e.g. Hasan et al. [2016]; Masci et al. [2011]; Sabokrou et al. [2016]; Chalapathy et al. [2017]; Sabokrou et al. [2018]; Abati et al. [2019]) in anomaly detection uses feature reconstruction. These approaches learn a model to reconstruct the normal training data and use the reconstruction error to identify anomalies. However, it has been observed that the reconstruction errors of normal and abnormal

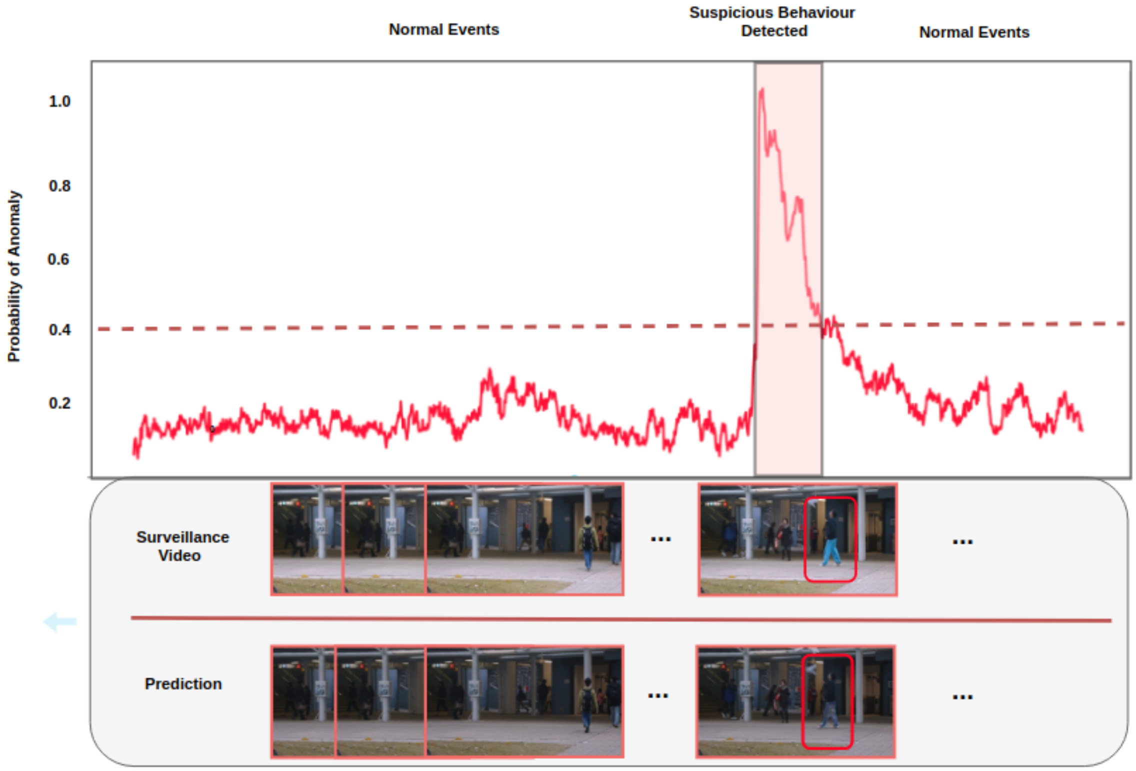


Figure 1.1: An example of our proposed video anomaly detection method. Our method uses a future frame prediction framework. Given several observed frames in a video, our model predicts the future frame. If the future frame is an anomaly, the predicted future frame is likely to be very different from the actual future frame. This prediction error allows us to detect the anomaly in a video.

events are often similar and are not very discriminative for the anomaly detection task. To address the limitation of feature reconstruction approaches, the work in Liu et al. [2018] proposes a future frame prediction framework for anomaly detection. This method learns a model that takes a sequence of consecutive frames as the input and predicts the next frame. The difference between the predicted frame and the actual frame at the next time step is used to indicate the probability of an anomaly. In this work, we follow this future frame prediction framework and propose two novel architectures. We propose to combine sequential modelling with generative models to

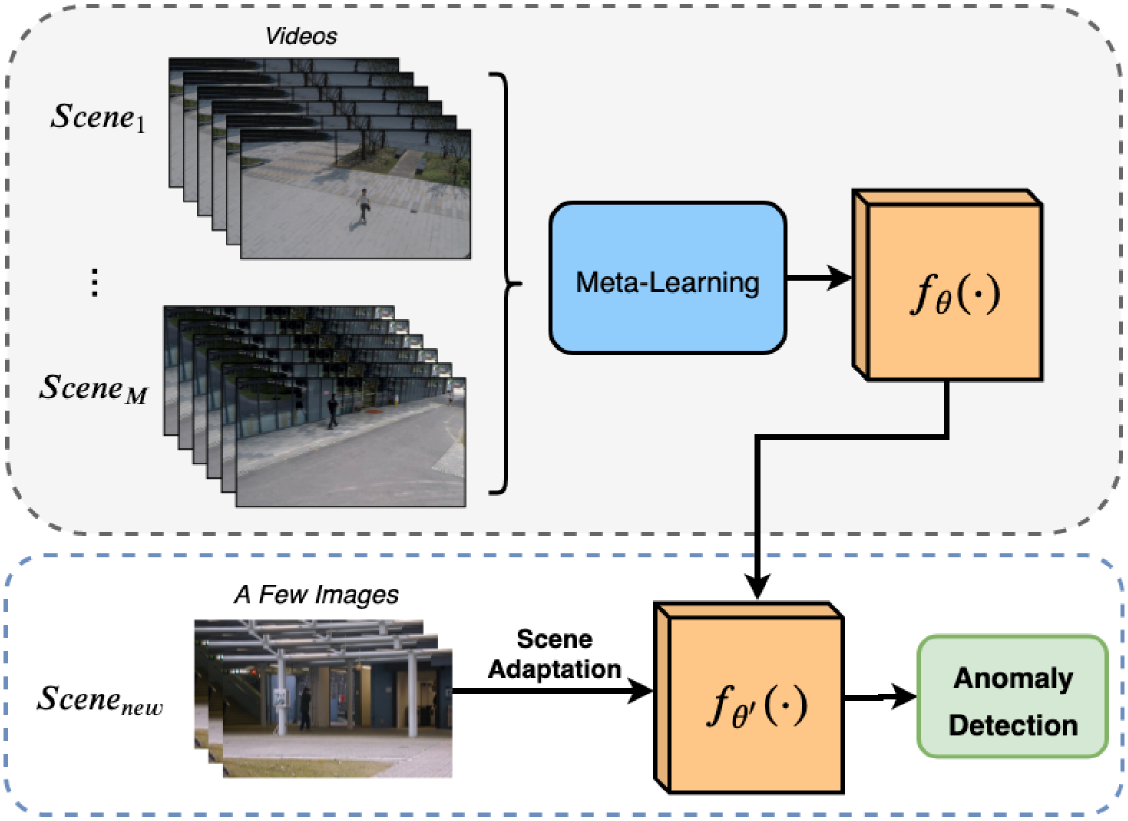


Figure 1.2: An overview of our proposed problem setting. During training (1st row), we have access to videos collected from M different camera scenes. From such training data, we use a meta-learning method to obtain a model f_θ with parameters θ. Given a target scene (2nd row), we have access to a small number of frames from this target scene. Our goal is to produce a new model $f_{\theta'}$ where the model parameters θ' are specifically adapted to this scene. Then we can use $f_{\theta'}(\cdot)$ to perform anomaly detection on the remaining videos from this target scene.

build models that can be trained end-to-end. Although sequential generative models have been previously proposed for speech recognition and music generation (Mogren [2016]; Chung et al. [2015]) they have not been applied in anomaly detection. An example of our proposed video anomaly detection method is showed in Fig 1.1. Given several consecutive frames, our model learns to predict the next future frame. For normal frames, our method is able to predict the next frame reasonably well. When there is anomaly in the future frame, the prediction is often distorted and blurry. By comparing the predicted future frame with the actual future frame, our system can detect suspicious behaviours or events (in this case, the man is throwing his bag up and down) are detected in a video frame. In the rest of this thesis, we will use the two terms *future frame prediction model* and *anomaly detection model* interchangeably since the latter can be directly derived from the former.

However, future frame prediction models also have some limitations. They implicitly assumes that the future prediction model learned from the training videos can be directly used in unseen test videos. This is a reasonable assumption if training and testing videos are from the same scene (e.g. captured by the same camera). In the experiment section, we will demonstrate that if we learn a future frame prediction model from videos captured from one scene and directly use the model in a completely different scene, the performance will drop. Of course, one possible way of alleviating this problem is to train the future frame prediction model using videos collected from diverse scenes. Then the learned model will likely generalize to videos from new scenes. However, this approach is also not ideal. In order to learn a model that can generalize well to diverse scenes, the model requires a large capacity. In many real-world applications, the anomaly detection system is often deployed on edge devices with limited computing powers. Thus, even if we can train a huge model that generalizes well to different scenes, we may not be able to deploy this model.

Our work is motivated by the following key observation. In real-world anomaly detection applications, we usually consider one particular scene for testing since the surveillance cameras are normally installed at a fixed location. As long as a model works well in this particular scene, it does not matter at all whether the same model works on images from other scenes. In other words, we would like to have a model specifically adapted to the scene where the model is deployed. In this thesis, we propose a novel problem called the *few-shot scene-adaptive anomaly detection* illustrated in Fig. 1.2. During training, we assume that we have access to videos collected from multiple scenes. During testing, the model is given a few frames in a video from a new target scene. Note that the learning algorithm does not see any images from the target scene during training. Our goal is to produce a future frame prediction model specifically adapted to this target scene using these few frames. We believe this new problem setting is closer to real-world applications. If we have a reliable solution to this problem, we only need a few frames from a target camera to produce an anomaly detection model that is specifically adapted to this camera. In this thesis, we propose a meta-learning based approach to this problem. During training, we learn a model 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.

The contributions of this thesis are manifold:

- We introduce a new problem called few-shot scene-adaptive anomaly detection, which is closer to the real-world deployment of anomaly detection systems.
- We propose a novel meta-learning based algorithm for solving this problem.
- We propose two novel anomaly detection backbone architectures using the idea of sequential generative modelling which showed superior performance over state-of-the-art methods.

- We demonstrate that our proposed approaches significantly outperforms alternative methods on several datasets.

1.2 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 discuss relevant literature in anomaly detection in surveillance videos. Chapter 3 lays out the required background and our problem setting for *few-shot scene-adaptive anomaly detection* for the thesis. Chapter 4 describes our algorithm to solve our proposed problem and two novel backbone architectures using future frame prediction. Dataset and evaluation methods are addressed in Chapter 5. Experimental results are discussed in Chapter 6. Conclusions of the thesis are presented in Chapter 7.

Chapter 2

Related Work

2.1 Anomaly Detection with Hand-crafted Features

Early work in video anomaly detection uses hand-crafted features. Tung et al. [2011]; Wu et al. [2010] use trajectory features to represent normal behaviours. However, these methods can not be applied to crowded scenes. To address this limitation, low-level features such as histogram of oriented gradient and histogram of oriented flows are also applied (Dalal and Triggs [2005]; Dalal et al. [2006]) for human detection. Zhao et al. [2011]; Lu et al. [2013]; Cong et al. [2011] represent each scene by a dictionary of temporal and spatial information. These approaches have low performance due to the fact that the dictionary does not ensure the capacity of normal events and cannot classify anomaly correspondingly. Statistical-based models have also been proposed. For example, Kim and Grauman [2009] proposes an approach based on a mixture of probabilistic PCA (MPPCA) with optical flow pattern. Gaussian mixture model (Mahadevan et al. [2010]) has also been applied for anomaly detection.

2.2 Anomaly Detection with Deep Learning

In order to address the limitation of hand-crafted features in anomaly detection, there has been recent work that explores the use of deep learning approaches. They can be roughly categorized as either reconstruction-based or prediction-based methods. Reconstruction-based methods train a deep learning model to reconstruct the frames in a video and use the reconstruction error to differentiate the normal and abnormal events. Examples of reconstruction models include convolutional autoencoders (Masci et al. [2011]; Hasan et al. [2016]; Sabokrou et al. [2016]; Chalapathy et al. [2017]; Gong et al. [2019]), latent autoregressive models(Abati et al. [2019]), deep adversarial training (Sabokrou et al. [2018]), etc. Prediction-based detection methods define anomalies as anything that does not conform to the prediction of a deep learning model. Sequential models like Convolutional LSTM (ConvLSTM) (Xingjian et al. [2015]) have been widely used for future frame prediction and utilized to the task of anomaly detection (Luo et al. [2017a]; Medel and Savakis [2016]). Popular generative networks like generative adversarial networks (GANs) (Goodfellow et al. [2014]) and variational autoencoders (VAEs) (Kingma and Welling [2013]) are also applied in prediction-based anomaly detection. Liu et al. (Liu et al. [2018]) propose a conditional GAN based model with a low level optical flow (Dosovitskiy et al. [2015]) feature. Moreover, Gong et al. [2019] apply optical flow prediction constraint on a reconstruction based model.

2.3 Few-Shot and Meta Learning

To mimic the fast and flexible learning ability of humans, few-shot learning aims at adapting quickly to a new task with only a few training samples (Lake et al. [2015]). In particular, meta learning (also known as *learning to learn*) has been shown to be an effective solution to the few-shot learning problem. The research in meta-learning can be categorized into three common approaches: metric-based (Koch et al. [2015]; Vinyals et al. [2016]; Sung et al. [2018]), model-based (Santoro et al. [2016]; Munkhdalai and Yu [2017]) and optimization-based approaches (Ravi and Larochelle [2016]; Finn et al. [2017]). Metric-based approaches typically apply Siamese (Koch et al. [2015]), matching (Vinyals et al. [2016]), relation (Sung et al. [2018]) or prototypical networks (Snell et al. [2017]) for learning a metric or distance function over data points. Model-based approaches are devised for fast learning from the model architecture perspective (Santoro et al. [2016]; Munkhdalai and Yu [2017]), where rapid parameter updating during training steps is usually achieved by the architecture itself. Lastly, optimization-based approaches modify the optimization algorithm for quick adaptation (Ravi and Larochelle [2016]; Finn et al. [2017]). These methods can quickly adapt to a new task through the meta-update scheme among multiple tasks during parameter optimization. However, most of the approaches above are designed for simple tasks like image classification. In our proposed work, we follow a similar optimization-based meta-learning approach proposed in Finn et al. [2017] and apply it to the much more challenging task of anomaly detection. To the best of our knowledge, we are the first to cast anomaly detection as meta-learning from multiple scenes.

Chapter 3

Background and Problem Setup

3.1 Variational Autoencoder

Variational autoencoder (VAE) [Kingma and Welling, 2013] has been shown to be effective in reconstructing complex distributions for non-sequential data. Given an input x, VAE applies an encoder (also known as inference model) $q_\theta(z|x)$ to generate the latent variable z that captures the variation in x. It uses a decoder $p_\phi(x|z)$ to approximate the observation given the latent variable. The inference model represents the approximate posterior using the mean μ and variance σ^2 calculated by a neural network $q_\theta(z|x) \sim \mathcal{N}(\mu_x, \sigma_x^2)$, where μ_x and σ_x^2 are outputs of some neural networks that take x as the input. A prior $p(z)$ is chosen to be a simple Gaussian distribution. With the constraints of distribution on latent variables, the complete objective function can be described as below:

$$L(x|\theta, \phi) = -KL(q_\theta(z|x)||p(z)) + \mathbb{E}_{q_\theta(z|x)}[logp_\phi(x|z)] \tag{3.1}$$

where $KL(q_\theta(z|x)||p(z))$ is the Kullback-Leibler divergence [Hershey and Olsen, 2007] between the prior and the posterior.

3.2 Variational Recurrent Neural Network

VAE is a generative model. It cannot directly be used to model sequential data. For the problem of anomaly detection, our data are inherently sequential since we need to consider the information in several consecutive frames in order to predict the next frame. Variational Recurrent Neural Network (VRNN) [Chung et al., 2015] is an extension of vanilla VAE. It combines VAE with a recurrent neural network in order to model sequential data. Since this approach shares the same inspiration with our Conv-VRNN approach, we will explain the technical details in the next section.

3.3 Conditional Generative Adversarial Network

Conditional GAN [Isola et al., 2017] was first proposed for image translation. Here we use it for anomaly detection. It consists of a generator $\mathcal{G}$ which tries to reconstruct the input and a discriminator $\mathcal{D}$ which aims at discriminating the output of the generator and the target image. Both $\mathcal{G}$ and $\mathcal{D}$ are trained jointly following a two-player min-max game [Mirza and Osindero, 2014]. Through this training scheme, the network can learn the distribution of regular data and output a poor reconstruction on anomalies in testing data. The detection of abnormal frames can be implemented using the reconstruction error or the output score of the discriminator.

3.4 Problem Setup

Following Liu et al. [2018], we consider anomaly detection using the framework of future frame prediction. For completeness, we first briefly summarize the future frame prediction based framework for anomaly detection as defined in Liu et al. [2018]. Then we describe our problem setup of *few-shot scene-adaptive anomaly detection*.

3.4.1 Future Frame Prediction for Anomaly Detection

We follow the work of Liu et al. [2018] and define anomaly detection in videos as a future frame prediction problem. Given t consecutive frames $I_1, I_2, ..., I_t$ in a video, the goal is to learn a model $f_\theta(x_{1:t})$ with parameters θ that takes these t frames as its input and predicts the next frame at time $t+1$. We use $\hat{I}_{t+1}$ to denote the predicted frame at time $t+1$. The anomaly detection at time $t+1$ is determined by the difference between the prediction frame $\hat{I}_{t+1}$ and the actual frame I_{t+1}. If this difference is larger than a threshold, the frame I_{t+1} is considered an anomaly.

During training, the goal is to learn the future frame prediction model $f_\theta(\cdot)$ from a collection of normal videos. Note that the training data only contains normal videos since it is usually difficult to collect training data with abnormal events for real-world applications.

3.4.2 Few-Shot Scene-Adaptive Anomaly Detection

The standard future frame prediction setting for anomaly detection described above has some limitations that make it difficult to apply in real-world scenarios. It implicitly assumes that the prediction model $f_\theta(\cdot)$ learned from the training videos can generalize well on test videos. In practical applications, it is unrealistic to collect training videos from the target scene where the system will be deployed. In most cases, training and test videos will come from different scenes. The prediction model $f_\theta(\cdot)$ can easily overfit to the particular training scene and will not generalize to a different scene during testing. We will empirically demonstrate this in the experiment section.

In this thesis, we introduce a new problem setup that is closer to real-world applications. This setup is motivated by two crucial observations. First of all, in most anomaly detection applications, the test images come from a particular scene captured by the same camera. In this case, we only need the learned model to perform well on this particular scene. Second, although it is unrealistic to collect a large number of videos from the target scene, it is reasonable to assume that we will have access to a small number of images from the target scene. For example, when a surveillance camera is installed, there is often a calibration process. We can easily collect a few images from the target environment during this calibration process.

Motivated by these observations, we propose a problem setup called *few-shot scene-adaptive anomaly detection*. During training, we have access to videos collected from different scenes. During testing, the videos will come from a target scene that never appears during training. Our model will learn to adapt to this target from only a few initial frames and the adapted model is expected to work well in the target scene.

Chapter 4

Our Approach

We propose to learn few-shot scene-adaptive anomaly detection models using a meta-learning framework, in particular, the MAML algorithm Finn et al. [2017] for meta-learning. Figure 4.1 shows an overview of the proposed approach. The meta-learning framework consists of a meta-training phase and a meta-testing phase. During meta-training, we have access to videos collected from multiple scenes. The goal of meta-training is learning to quickly adapt to a new scene based on a few frames from it. During this phase, the 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. In each task, our method learns to adapt a pre-trained future frame prediction model using a few frames from the corresponding scene. The learning procedure (meta-learner) is designed in a way such that the adapted model will work well on other frames from the same scene. Through this meta-training process, the model will learn to effectively perform few-shot adaptation for a new scene. During meta-testing, given a few frames

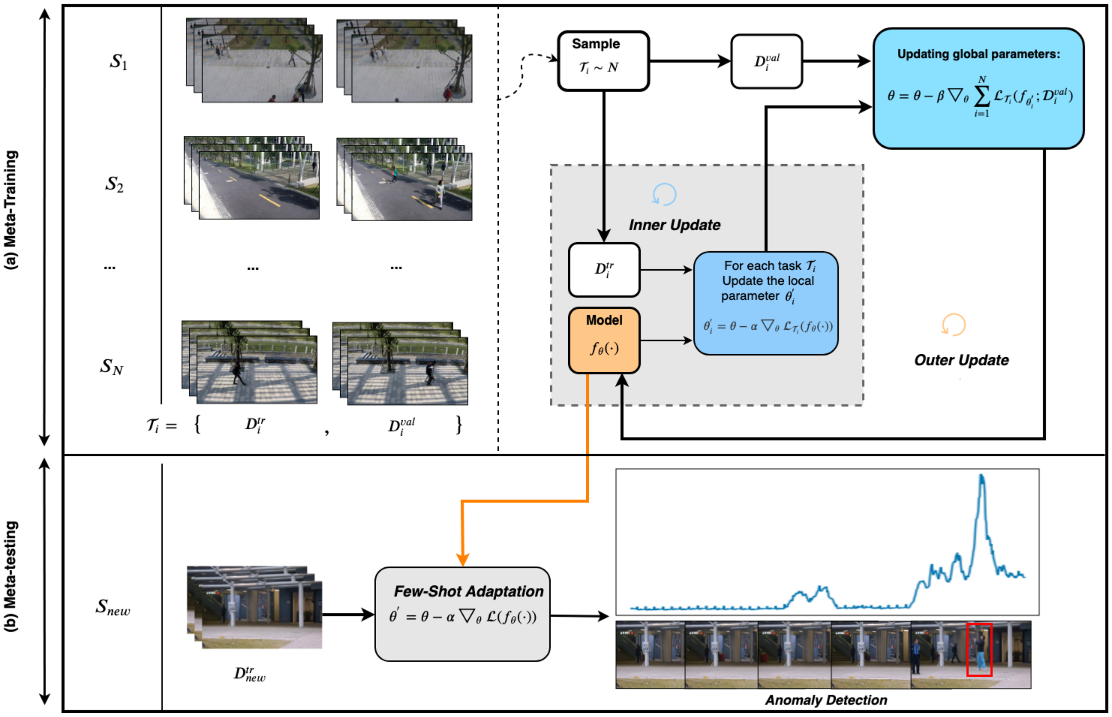

(a) Meta-Training
S_1
S_2
...
S_N
$\mathcal{T}_i = \{ D_i^{tr} , D_i^{val} \}$
Sample
$\mathcal{T}_i \sim N$
D_i^{val}
Updating global parameters:
$\theta = \theta - \beta \nabla_\theta \sum_{i=1}^{N} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}; D_i^{val})$
Inner Update
D_i^{tr}
For each task $\mathcal{T}_i$ Update the local parameter θ_i'
$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta(\cdot))$
Model
$f_\theta(\cdot)$
Outer Update
(b) Meta-testing
S_{new}
D_{new}^{tr}
Few-Shot Adaptation
$\theta' = \theta - \alpha \nabla_\theta \mathcal{L}(f_\theta(\cdot))$
Anomaly Detection

Figure 4.1: 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.

from a new target scene, the meta-learner is used to adapt a pre-trained model to this scene. Afterwards, the adapted model is expected to work well on other frames from this target scene.

Our proposed meta-learning framework can be used in conjunction with any future frame prediction model as the backbone architecture. In Sec. 4.1, we first introduce the meta-learning approach for scene-adaptive anomaly detection in a general way that is independent of the particular choice of the backbone architecture. In Sec. 4.2, we then describe the details of the proposed backbone architectures used in this thesis.

4.1 MAML for Scene-Adaptive Anomaly Detection

Our goal of few-shot scene-adaptive anomaly detection is to learn a model that can quickly adapt to a new scene using only a few examples from this scene. To accomplish this, the model is trained during a meta-training phase using a set of tasks where it learns to quickly adapt to a new task using only a few samples from the task. The key to applying meta-learning for our application is how to construct these tasks for the meta-training. Intuitively, we should construct these tasks so that they mimic the situation during testing.

4.1.1 Tasks in Meta-learning

We construct the tasks for meta-training as follows. Let us consider a future frame prediction model $f_\theta(I_{1:t}) \rightarrow \hat{I}_{t+1}$ that maps t observed frames $I_1, I_2, ..., I_t$ to the predicted frame $\hat{I}_{t+1}$ at $t+1$. We have access to M scenes during meta-training, denoted as $S_1, S_2, ..., S_M$. For a given scene S_i, we can construct a corresponding task $\mathcal{T}_i = (\mathcal{D}_i^{tr}, \mathcal{D}_i^{val})$, where $\mathcal{D}_i^{tr}$ and $\mathcal{D}_i^{val}$ are the training and the validation sets in the task $\mathcal{T}_i$. We first split videos from S_i into many overlapping consecutive segments of length $t+1$. Let us consider a segment $(I_1, I_2, ..., I_t, I_{t+1})$. We then consider the first t frames as the input x and the last frame as the output y, i.e. $x = (I_1, I_2, ..., I_t)$ and

$y = I_{t+1}$. This will form an input/output pair (x, y). The future frame prediction model can be equivalently written as $f_\theta : x \rightarrow y$. In the training set $\mathcal{D}_i^{tr}$, we randomly sample K input/output pairs from $\mathcal{T}_i$ to learn future frame prediction model, i.e. $\mathcal{D}^{tr} = \{(x_1, y_1), (x_2, y_2), ..., (x_K, y_K)\}$. Note that to match the testing scheme, we make sure that all the samples in $\mathcal{D}^{tr}$ come from the same video. We also randomly sample K input/output pairs (excluding those in $\mathcal{D}_i^{tr}$) to form the test data $\mathcal{D}_i^{val}$.

4.1.2 Meta-Training

Let us consider a pre-trained future frame prediction model $f_\theta : x \rightarrow y$ with parameters θ. Following MAML Finn et al. [2017], we adapt to a task $\mathcal{T}_i$ by defining a loss function on the training set $\mathcal{D}_i^{tr}$ of this task and use one gradient update to change the parameters from θ to θ'_i:

$$\theta'_i = \theta - \alpha \bigtriangledown_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta; \mathcal{D}_i^{tr}), \text{ where} \tag{4.1a}$$

$$\mathcal{L}_{\mathcal{T}_i}(f_\theta; \mathcal{D}_i^{tr}) = \sum_{(x_j, y_j) \in \mathcal{D}_i^{tr}} L(f_\theta(x_j), y_j) \tag{4.1b}$$

where α is the step size. Here $L(f_\theta(x_j), y_j)$ measures the difference between the predicted frame $f_\theta(x_j)$ and the actual future frame y_j. We will describe the details of $L(\cdot)$ in Sec. 4.2. The updated parameters θ' are specifically adapted to the task $\mathcal{T}_i$. Intuitively we would like θ' to perform on the validation set $\mathcal{D}_i^{val}$ of this task. We measure the performance of θ' on $\mathcal{D}_i^{val}$ as:

$$\mathcal{L}_{\mathcal{T}_i}(f_{\theta'}; \mathcal{D}_i^{val}) = \sum_{(x_j, y_j) \in \mathcal{D}_i^{val}} L(f_{\theta'}(x_j), y_j) \tag{4.2}$$

The goal of meta-training is to learn the initial model parameters θ, so that the scene-adapted parameters θ' obtained via Eq. 4.1 will minimize the loss in Eq. 4.2

across all tasks. Formally, the objective of meta-learning is defined as:

$$\min_{\theta} \sum_{i=1}^{M} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'}; \mathcal{D}_i^{val}) \tag{4.3}$$

The loss in Eq. 4.3 involves summing over all tasks during meta-training. In practice, we sample a mini-batch of tasks in each iteration. Algorithm 1 summarizes the entire learning algorithm.

Algorithm 1: Meta-training for few-shot scene-adaptive anomaly detection

```
Input: Hyper-parameters $\alpha, \beta$
Initialize $\theta$ with a pre-trained model $f_\theta(\cdot)$;
while not done do
    Sample a batch of scenes $\{S_i\}_{i=1}^N$;
    for each $S_i$ do
        Construct $\mathcal{T}_i = (\mathcal{D}_i^{tr}, \mathcal{D}_i^{val})$ from $S_i$;
        Evaluate $\bigtriangledown_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta; \mathcal{D}_i^{tr})$ in Eq. 4.1;
        Compute scene-adaptative parameters $\theta_i' = \theta - \alpha \bigtriangledown_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta; \mathcal{D}_i^{tr})$;
    end
    Update $\theta \leftarrow \theta - \beta \sum_{i=1}^N \bigtriangledown_\theta \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}; \mathcal{D}_i^{val})$ using each $\mathcal{D}_i^{val}$ and $\mathcal{L}_{\mathcal{T}_i}$ in
     Eq. 4.2;
end
```

4.1.3 Meta-Testing

After meta-training, we obtain the learned model parameters θ. During meta-testing, we are given a new target scene S_{new}. We simply use Eq. 4.1 to obtained

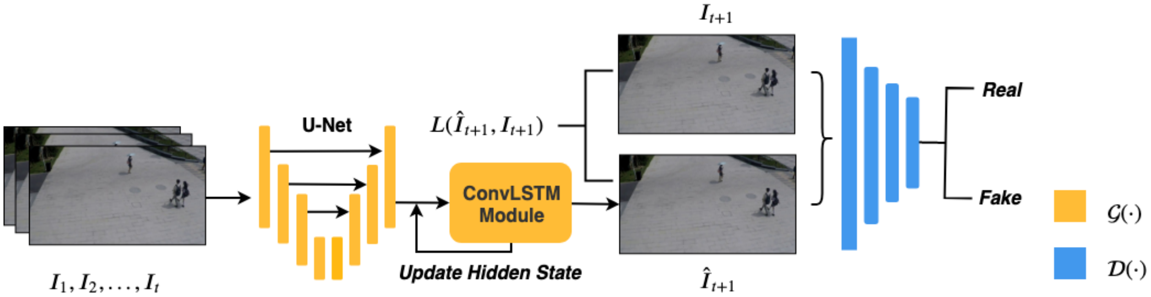


Figure 4.2: An overview of our backbone architecture. Our anomaly detection model consists of a Sequential Image Generator $\mathcal{G}(\cdot)$ and a Discriminator $\mathcal{D}(\cdot)$. Given an image sequence $I_1, I_2, ..., I_t$ as the input, $\mathcal{G}(\cdot)$ outputs a prediction $\hat{I}_{t+1}$ of the next frame. A prediction loss is computed between $\hat{I}_{t+1}$ and the actual frame I_{t+1} for parameter updating. $\mathcal{D}(\cdot)$ takes both $\hat{I}_{t+1}$ and I_{t+1} as its input to classify which one is real and which one is fake. These two networks are trained adversarially to obtain a good $\mathcal{G}(\cdot)$ that is able to fool $\mathcal{D}(\cdot)$.

the adapted parameters θ' based on K examples in S_{new}. Then we apply θ' on the remaining frames in the S_{new} to measure the performance. We use the first several frames of one video in S_{new} for adaptation and use the remaining frames for testing. This is similar to real-world settings where it is only possible to obtain the first several frames for a new camera.

4.2 Backbone Architecture

Our few-shot scene-adaptive anomaly detection method can be used with any future frame prediction model as its backbone architecture. In this thesis, we propose two novel future frame prediction models using the idea of sequential generative models for anomaly detection. These two variants uses either GAN or VRNN as backbone architecture. We call them *r-GAN* and *Conv-VRNN*. We discuss the details of the two models below.

4.2.1 r-GAN

This backbone architecture is based on the model in Liu et al. [2018]. The model in Liu et al. [2018] is built on a conditional GAN architecture with a modified U-Net Ronneberger et al. [2015]. Additionally, Liu et al. [2018] uses a Flownet Dosovitskiy et al. [2015] to capture temporal information of an image sequence. To build an end-to-end model, we remove the Flownet and instead learn the spatial-temporal feature of an image sequence using a ConvLSTM module. We call our model *r-GAN*. Our proposed model consists of two major parts: a sequential image generator and a discriminator. Fig 4.2 shows an overview of *r-GAN*.

Generator

We apply the same modified U-Net with Liu et al. [2018] as the backbone of our generator $\mathcal{G}(\cdot)$. Given an image sequence $I_1, ..., I_t$ (note that we choose $t = 3$ in our case), we pass each image $I_T(T = 1, 2, ..., t)$ to the U-net to generate a prediction $\hat{I}_{T+1}$. A ConvLSTM module then takes $\hat{I}_{T+1}$ and the last hidden state h_T as input and generate the current hidden state h_{T+1}:

$$h_{T+1} = f_{ConvLSTM}(h_T, \hat{I}_{T+1}) \tag{4.4}$$

The hidden state in the ConvLSTM module is used to remember the previous information of an image sequence.

To learn parameters in this module, we combine the least absolute deviation (L_1 loss) Pollard [1991], multi-scale structural similarity measurement (L_{ssm} loss) Wang et al. [2003] and gradient difference (L_{gdl} loss) Mathieu et al. [2016] to define a loss that measures the quality of the predicted frame.These three loss functions can be defined as follows:

(1) L1 loss between ground-truth and prediction is the summation of the absolute value between every pixel of the two images.

(2)We use multi-scale SSIM to represent the structural difference. MSSSIM is a multi-scale version of SSIM, which performs better on video sequences.

(3) Gradient difference is widely used for measuring the performance of a prediction. Gradient difference loss considers the intensities difference between neighbour pixels. Overall, given the predicted frame $\hat{I}_{t+1}$ and the ground-truth I_{t+1}, the complete loss function is defined as:

$$L(\hat{I}_{t+1}, I_{t+1}) = L_1(\hat{I}_{t+1}, I_{t+1}) + L_{msssim}(\hat{I}_{t+1}, I_{t+1}) + L_{gdl}(\hat{I}_{t+1}, I_{t+1}) \tag{4.5}$$

Discriminator

The goal of the discriminator is to differentiate the output of the generator and the ground-truth. Our discriminator in this network targets at classifying I_{T+1} as 1 and $\hat{I}_{T+1}$ as 0. More specifically, we optimize our discriminator $\mathcal{D}(\cdot)$ according to the objective function below:

$$\begin{aligned} L^D_{adv}(\hat{I}_{t+1}, I_{t+1}) = \quad & \frac{1}{2}L_{MSE}(\mathcal{D}(\hat{I}_{t+1}), 0) + \\ & \frac{1}{2}L_{MSE}(\mathcal{D}(I_{t+1}), 1) \end{aligned} \tag{4.6}$$

where L_{MSE} is the Mean Square Error loss function.

Anomaly Detection

Given an input sequence of frames $I_1, ..., I_t$ during testing, we use our model to predict the next frame $\hat{I}_{t+1}$ in the future. This predicted future frame $\hat{I}_{t+1}$ is compared with the ground-truth future frame I_{t+1} by calculating $L(\hat{I}_{t+1}, I_{t+1})$ (see Eq. 4.5). Same as Liu et al. [2018], after calculating the overall spatial loss of each testing video, we normalize the losses to get a score $S(t)$ in the range of $[0, 1]$ for each frame in the video by:

$$S(t) = \frac{L(\hat{I}_{t+1}, I_{t+1}) - \min L(\hat{I}_{t+1}, I_{t+1})}{\max L(\hat{I}_{t+1}, I_{t+1}) - \min L(\hat{I}_{t+1}, I_{t+1})} \tag{4.7}$$

We then use $S(t)$ as the score indicating how likely a particular frame is an anomaly. Note that all of our variants share the same evaluation metrics.

4.2.2 Conv-VRNN

This model extend VAE to model image sequences for anomaly detection and use the idea of Variational Recurrent Neural Network (VRNN) Chung et al. [2015] and build a Conv-VRNN model for future frame prediction. An overview of our proposed model is shown in Figure 4.3. Let $I_T \in \mathbb{R}^{H \times W \times 3}$ be the input image at time T, where $H \times W$ is the spatial dimension of the image. We define $h(T) \in \mathbb{R}^{H \times W \times 3}$ to be the hidden state of a ConvLSTM at time step T. Note that we choose the spatial dimension of h_T to match the image size. Our method consists of four components at each time step T:

Prior Distribution in VAE

This module takes the hidden state h_{T-1} from the previous time step as the input. It then generates a distribution on the latent variable in VAE. We first extract a feature vector from h_{T-1}. Since $h_{T-1} \in \mathbb{R}^{H \times W \times 3}$ is a 3D tensor and can be treated as a image, we can use a standard convolutional neural network (CNN) to extract

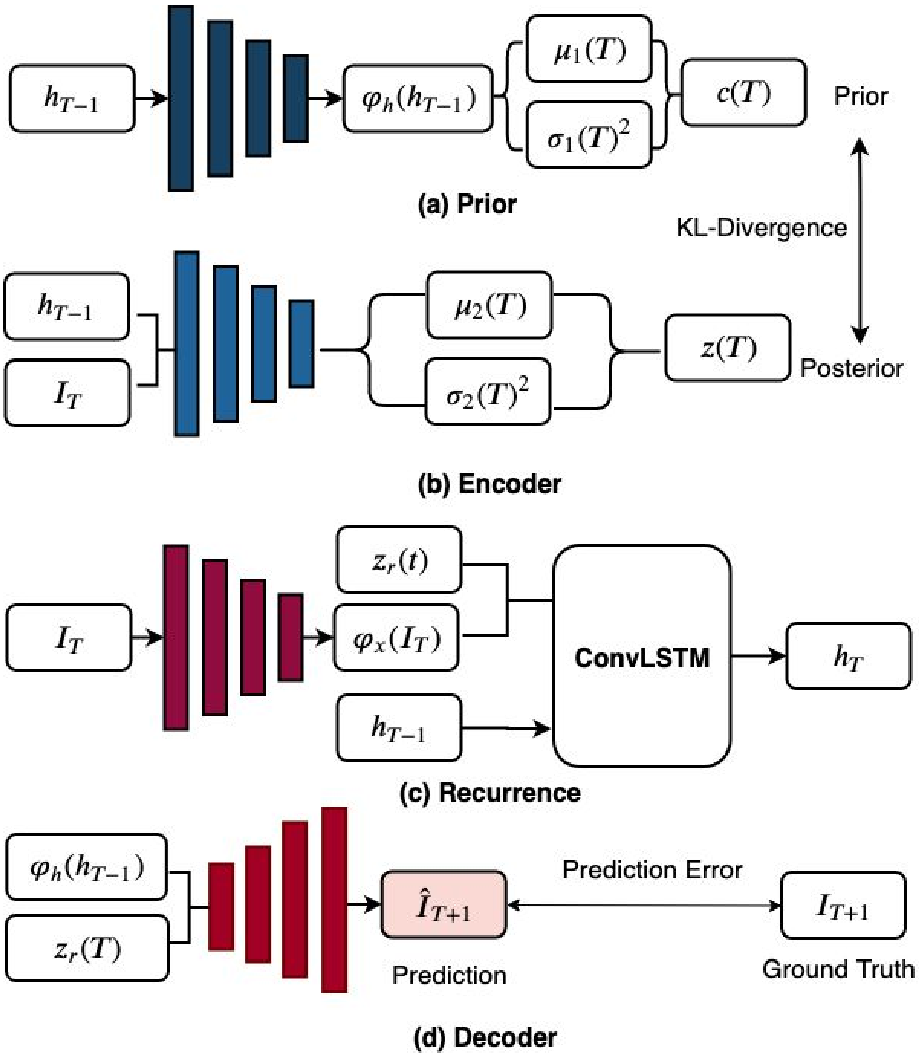


Figure 4.3: An overview of our proposed Conv-VRNN model at one time-step of a sequence. Our model requires 4 steps to process the input: (a) calculating the prior distribution in VAE; (b) encoder for posterior distribution and latent variable; (c) recurrence module for sequence modelling; (d) decoder for prediction.

the feature from h_{T-1}. We denote this feature as $\varphi_h(h_{T-1}) \in \mathbb{R}^{H' \times W' \times F}$, where $H' \times W'$ and F correspond to the spatial dimension and the channel dimension of the CNN feature map. Here we set $H' \times W' \times F = 16 \times 16 \times 32$. We then apply two different fully connected layers on $\varphi_h(h_{T-1})$ to produce two vectors corresponding to the mean and the variance of a Gaussian distribution in VAE, denoted by $\mu_1(T)$ and

$\sigma_1(T)$. In our implementation, the dimension of $\mu_1(T)$ and $\sigma_1(T)$ is set to be 20, i.e. $\mu_1(T), \sigma_1(T) \in \mathbb{R}^{20}$. We then use $\mu_1(T)$ and $\sigma_l(T)$ to define a Gaussian distribution for the prior distribution on the latent variable in VAE as follows:

$$c(T) \sim \mathcal{N}\left(\mu_1(T),\, diag\left(\sigma_1(T)^2\right)\right) \tag{4.8}$$

where $diag(\cdot)$ creates a diagonal matrix from a vector and $c(t)$ represent the prior distribution on the latent variable.

Encoder

The module takes the hidden state h_{T-1} of previous time step $T-1$ and the frame I_T at current time T as the input. It then produces a vector of the latent variable in VAE. We first concatenate I_T and h_{T-1} along their channel dimensions, then apply a CNN to extract a feature map. Again, we apply two different fully connected layers on this feature map to produce $\mu_2(T)$ and $\sigma_2(T)$. Similarly, the dimension of $\mu_2(T)$ and $\sigma_2(T)$ to be 20. We then define the posterior of the latent variable $z(T)$ in VAE as:

$$\begin{gathered} q_\theta\left(z(T) | concat\left(I_T, h_{T-1}\right)\right) \\ \sim \mathcal{N}\left(\mu_2(T),\, diag\left(\sigma_2(T)^2\right)\right) \end{gathered} \tag{4.9}$$

where $z(T) \in \mathbb{R}^{20}$. To measure the distribution loss between Eq. 4.8 and Eq. 4.9 at time step t, we can use the KL-divergence metric $KL\left(q_\theta\left(z\left(T\right)|x_T, h_{T-1}\right)||c\left(T\right)\right)$.

Recurrence

To capture the temporal information among frames in a video, we use a ConvLSTM to represent the recurrent relationship among frames. From the current input image I_T, we apply a CNN to extract a feature map which we denote as $\varphi_x(I_T) \in \mathbb{R}^{H' \times W' \times F}$. To match the dimension of this feature, we also resize the latent variable $z(T)$ (recall $z(T) \in \mathbb{R}^{20}$) as follows. We first use fully connected layers to map $z(t)$ to a high-dimensional space $\mathbb{R}^{1024}$, then reshape to a 3D tensor of dimension $H' \times W' \times F = 16 \times 16 \times 32$. We use $z_r(T) \in \mathbb{R}^{H' \times W' \times F}$ to denote this reshaped tensor. We concatenate the input feature $\varphi_x(I_T)$ with the $z_r(T)$ along the channel dimension and use it as the input to ConvLSTM at time T:

$$h\left(t\right) = f_{ConvLSTM}\left(concat\left(\varphi_x(I_T), z_r\left(T\right)\right), h_{T-1}\right) \tag{4.10}$$

Decoder

This module takes the resized hidden state $z_r(T)$ as its input and produces a predicted frame $\hat{I}_{T+1}$ for the next time-step. Note that the dimensions of $z_r(T)$ match those of the extracted feature of previous hidden state $\varphi_h(h_{T-1})$. We concatenate $z_r(T)$ and $\varphi_h(h_{T-1})$ along the channel dimension. The result is used as the input of this decoder module. The decoder is implemented as a deconvolutional nerual network that generates the predicted frame $\hat{I}_{T+1} \in \mathbb{R}^{H \times W \times 3}$. We take the output of the last time-step of the decoder $\hat{I}_{t+1}$ as our prediction.

Model Learning

Overall, given the last time-step output of the decoder $\hat{I}_{t+1}$ and the groundtruth I_{t+1}, we use the prediction loss defined in 4.5. In conclusion, we define the complete objective function of Conv-VRNN as:

$$L = \sum_{T=1}^{t}(-KL(q_\theta(z(T)|I_T, h_{T-1})||c(T))) + L(\hat{I}_{t+1}, I_{t+1}). \tag{4.11}$$

Chapter 5

Datasets and Experimental Setup

5.1 Datasets

Datasets for Standard Anomaly Detection: This thesis propose two novel sequential generative models for anomaly detection. To test the performance of the two models, we apply experiments on four standard anomaly detection datasets: UCSD Pedestrian 1 (Mahadevan et al. [2010]), UCSD Pedestrian 2 (Ped 2)(Mahadevan et al. [2010]), CUHK Avenue (Lu et al. [2013]) and ShanghaiTech(Luo et al. [2017b]). Additionally, we apply our model to another challenging task of fall detection.UR fall (Kwolek and Kepski [2014]) dataset is used for this experiment. We describe these datasets in detail in the next module.

Datasets for Few-shot Scene-adaptive Anomaly Detection: This thesis addresses a new problem. In particular, the problem setup requires training videos from multiple scenes and test videos from different scenes. There are no existing datasets that we can directly use for this problem setup. Instead, we repurpose several avail-

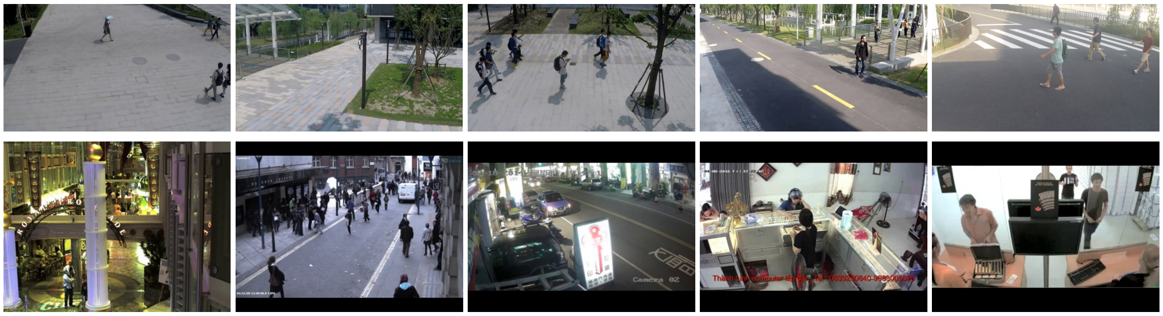


Figure 5.1: Example frames from the datasets used for meta-training. The first row shows examples of different scenes from the Shanghai Tech dataset. The second row shows examples of different scenes from the UCF crime dataset.

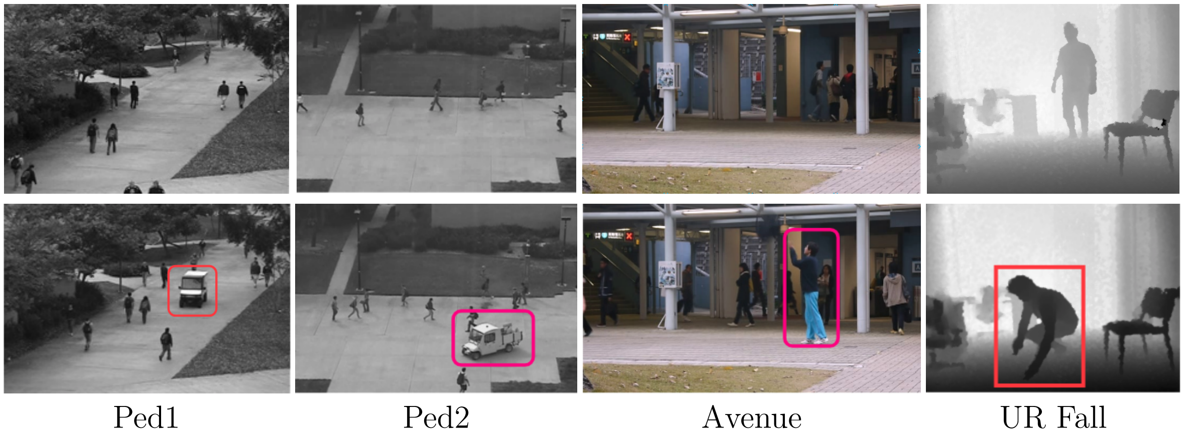


Figure 5.2: Example frames from datasets used in meta-testing. The first row shows examples of normal frames for four datasets, and the second row shows the abnormal frames. Note that training videos only contain normal frames. Videos with abnormal frames are only used for testing.

able datasets.

- Shanghai Tech (Luo et al. [2017b]): This dataset contains 437 videos collected from 13 scenes. The training videos only contain normal events, while the test videos may contain anomalies. In the standard split in Luo et al. [2017b], both training and test sets contain videos from these 13 scenes. This split does not fit our problem setup where test scenes should be distinct from those in training. In our experiment, we propose a new train/test split more suitable for our problem.

We also perform cross-dataset testing where we use the original Shanghai Tech dataset during meta-training and other datasets for meta-testing.

- UCF crime (Sultani et al. [2018]): This dataset contains normal and crime videos collected from a large number of real-world surveillance cameras where each video comes from a different scene. Since this dataset does not come with ground-truth frame-level annotations, we cannot use it for testing since we do not have the ground-truth to calculate the evaluation metrics. Therefore, we only use the 950 normal videos from this dataset for meta-training, then test the model on other datasets. This dataset is much more challenging than Shanghai Tech when being used for meta-training, since the scenes are diverse and very dissimilar to our test sets. Our insight is that if our model can adapt to a target dataset by meta-training on UCF crime, our model can be trained with similar surveillance videos.

- UCSD Pedestrian 1 (Mahadevan et al. [2010]), UCSD Pedestrian 2 (Ped 2)(Mahadevan et al. [2010]), and CUHK Avenue (Lu et al. [2013]): Each of these datasets contains videos from only one scene but different times. They contain 36, 12 and 21 test videos, respectively, including a total number of 99 abnormal events such as moving bicycles, vehicles, people throwing things, wandering and running. We use the model trained from Shanghai Tech or UCF crime datasets and test on these datasets.

- UR fall (Kwolek and Kepski [2014]): This dataset contains 70 depth videos collected with a Microsoft Kinect camera in a nursing home. Each frame is represented as a 1-channel grayscale image capturing the depth information. In our case, we convert each frame to an RGB image by duplicating the grayscale value among 3 color channels for every pixel. This dataset is originally collected for research in fall detection. We follow previous work in Nogas et al. [2018] which considers a person falling as the anomaly. Again, we use this dataset for testing. Since this dataset is drastically different from other anomaly detection datasets, good performance on this dataset will be very strong evidence of the generalization power of our approach.

Figure 5.1 and Figure 5.2 show some example frames from the datasets we used in meta-training and meta-testing.

5.2 Evaluation Metrics

Following prior work Liu et al. [2018]; Luo et al. [2017a]; Mahadevan et al. [2010], we evaluate the performance using the area under the ROC curve (AUC). The ROC curve is obtained by varying the threshold for the anomaly score for each frame-wise prediction.

5.3 Implementation Details

We implement our model in PyTorch. We use a fixed learning rate of 0.0001 for pre-training. We fix the hyperparameters α and β in meta-learning at 0.0001. During meta-training, we select the batch size of task/scenes in each epoch to be 5 on ShanghaiTech, and 10 on UCF crime.

5.4 Baselines

To the best of our knowledge, this is the first work on the scene-adaptive anomaly detection problem. Therefore, there is no prior work that we can directly compare with. Nevertheless, we define the following baselines for comparison.

Pre-trained: This baseline learns the model from videos available during training, then directly applies the model in testing without any adaptation.

Fine-tuned: This baseline first learns a pre-trained model. Then it adapts to the target scene using the standard fine-tuning technique on the few frames from the target scene.

Chapter 6

Experimental Results

In this section, we show our experimental results on our backbone architecture and our experiment on few-shot scene-adaptive anomaly detection.

6.1 Backbone Architectures

6.1.1 Conv-VRNN vs r-GAN

We first perform an experiment as a sanity check to show that our proposed backbone architecture is comparable to the state-of-the-art. Note that this sanity check uses the standard training/test setup (training set and testing set are provided by the original datasets), and our model can be directly compared with other existing methods. Table 6.1 shows the comparisons among our proposed architecture (r-GAN and Conv-VRNN),and other methods when using the standard anomaly detection training/test setup on several anomaly detection datasets, in which r-GAN shows superior performance over all its component. Table 6.2 shows the comparison on the

Category	Method	Ped1	Ped2	CUHK	ST
Feature	Kim and Grauman [2009]	59.0	69.3	-	-
	Del Giorno et al. [2016]	-	-	78.3	-
Reconstruction	Hasan et al. [2016]	75.0	85.0	80.0	60.9
	Tudor Ionescu et al. [2017]	68.4	82.2	80.6	-
	Abati et al. [2019]	-	95.4	-	72.5
	Luo et al. [2017a]	75.5	88.1	77.0	-
	Gong et al. [2019]	-	94.1	83.3	71.2
Prediction	Luo et al. [2017b]	-	92.2	81.7	68.0
	Liu et al. [2018]	83.1	95.4	84.9	72.8
	Morais et al. [2019]	-	-	-	73.4
Ours	Conv-VRNN	**86.3**	96.1	**85.8**	77.6
	r-GAN	**86.3**	**96.2**	**85.8**	**77.9**

Table 6.1: Comparison of anomaly detection performance among our backbone architecture (r-GAN and Conv-VRNN), and existing state-of-the-art in the standard setup (i.e. without scene adaptation). We report AUC (%) of different methods on UCSD Ped1 (Ped1), UCSD Ped2 (Ped2), CUHK Avenue (CUHK) and Shanghai Tech (ST) datasets. We use the same train/test split as prior work on each dataset. Our proposed backbone architecture *r-GAN* outperforms the existing state-of-the-art on almost all datasets.

fall detection dataset. We can see that our backbone architecture r-GAN outperforms Conv-VRNN and the existing state-of-the-art methods on almost all the datasets. As a result, we use r-GAN as our backbone architecture to test our few-shot scene-adaptive anomaly detection algorithm in this thesis.

Method	AUC (%)
DAE(Masci et al. [2011])	75.0
CAE(Masci et al. [2011])	76.0
CLSTMAE (Nogas et al. [2019])	82.0
DSTCAE (Nogas et al. [2018])	89.0
Conv-VRNN(ours)	89.7
r-GAN (ours)	**90.6**

Table 6.2: Comparison of anomaly detection in terms of AUC (%) of different methods on the UR fall detection dataset. This dataset contains depth images. We simply treat those as RGB images. Our proposed backbone architecture r-GAN is state-of-the-art among all the methods.

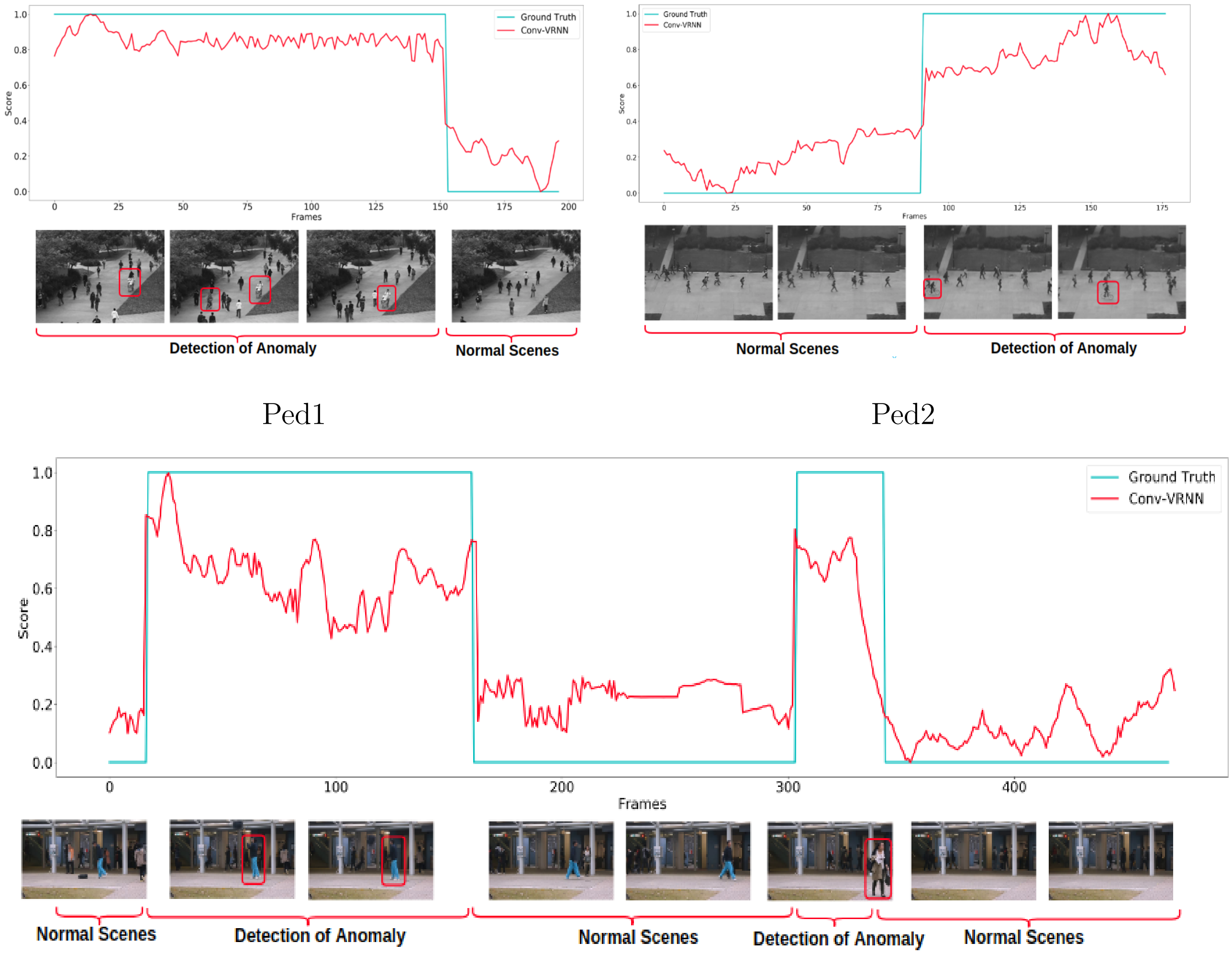
	Ped 1	Ped 2	Avenue
Conv-VAE	82.42%	89.18%	81.82%
Conv-VRNN	**86.26%**	**96.06%**	**85.48%**

Table 6.3: Comparision of Conv-VAEs versus Conv-VRNN in terms of AUC on three datasets.

6.1.2 Analysis on Conv-VRNN

Although Conv-VRNN can not outperform r-GAN on bekchmark datasets, we found out that this end-to-end model still outperforms other state-of-the-art methods. To gain further insight of Conv-VRNN, we perform several ablation studies below. Fig 6.1 also shows some qualitative examples of anomaly detection in videos using Conv-VRNN.

(1) Conv-VAE vs Conv-VRNN: In order to analyze the effect of incorporating temporal information, we implement a variant of our model without RNN. We call this



Ped1

Ped2

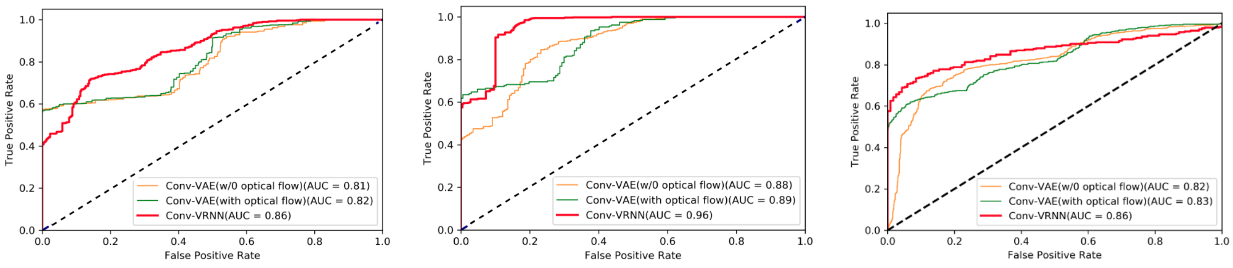
Avenue

Figure 6.1: Examples of anomaly detection on three datasets using Conv-VRNN. We plot the anomaly score of our model and the ground-truth anomaly score. Again, the bounding boxes are for visualization purpose.

variant Conv-VAE. Conv-VAE uses the encoder module to encode a latent variable and uses the decoder module for prediction. We have experimented with Conv-VAE that takes either one input frame or four frames to predict the next frame. The results are shown in Table 6.3. We can see that Conv-VRNN outperforms Conv-VAE. This demonstrates the importance of capturing the temporal information using RNN for

L_1	✓	✓	✓
L_{msssim}	✗	✓	✓
L_{gdl}	✗	✗	✓
AUC	80.29%	83.34%	**86.26%**

Table 6.4: Evaluation of different combinations of various loss terms in the objective functions in our Conv-VRNN network on the Ped1 dataset. The results show that the combination of all loss terms gives the best performance.



Ped1 Ped2 Avenue

Figure 6.2: ROC curves of our Conv-VRNN method, Conv-VAE (w/o optical flow) and Conv-VAE (with optical flow) on three datasets.

anomaly detection.

(2) Analysis on Losses: As we mentioned in Sec 4, we apply three different losses for prediction. The analysis of the impact of the losses can be visualized in Table 6.4. We choose three combinations of objective functions for evaluation: constraint only on intensity (L_1), constraint on intensity and structure ($L_1 + L_{msssim}$), constraint on intensity, structure and gradient ($L_1 + L_{msssim} + L_{gdl}$). The results demonstrate that the appearance information is better captured by the model with more constraints.

	Ped1	Ped2	Avenue
Conv-VAE(w/o optical flow)	80.15%	88.13%	80.92%
Conv-VAE(with optical flow)	81.36%	89.52%	82.23%
Conv-VRNN	**86.26%**	**96.06%**	**85.78%**

Table 6.5: Comparison between our Conv-VRNN model with different VAE-based models (with or without optical flow features). Our proposed Conv-VRNN outperforms Conv-VAE (with optical flow) even if our model does not use optical flow features.

Methods	$K = 1$	$K = 5$	$K = 10$
Pre-trained	70.11	70.11	70.11
Fine-tuned	71.61	70.47	71.59
Ours	**74.51**	**75.28**	**77.36**

Table 6.6: Comparison of K-shot scene-adaptive anomaly detection on the Shanghai Tech dataset. We use 6 scenes for training and the remaining 7 scenes for testing. We report results in terms of AUC (%) for $K = 1, 5, 10$. The proposed approach outperforms two baselines.

(3) Sequential Model vs Optical Flow: Our Conv-VRNN uses a RNN module to capture the temporal information in a video. An alternative way of capturing temporal information is to use optical flow features. We have implemented a Conv-VAE model with such constraint. Following Liu et al. [2018], we apply the pretrained Flownet (Dosovitskiy et al. [2015]) to estimate the optical flow, and use the returned loss of the Flownet as a motion constraint of the network only in training time. Table 6.5, Figure 6.2 show that although adding optical flow in our implementation of Conv-VAE improves the performance compared with Conv-VAE applied on only

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**
UR Fall	Pre-trained	64.08	64.08	64.08
	Fine-tuned	64.48	64.75	62.89
	Ours	**75.51**	**78.7**	**83.24**

Table 6.7: Comparison of K-shot ($K = 1, 5, 10$) scene-adaptive anomaly detection under the cross-dataset testing setting. We report results in terms of AUC (%) using the Shanghai Tech dataset for meta-training. We compare our results with two baseline methods. Our results demonstrate the effectiveness of our method on few-shot scene-adaptive anomaly detection.

raw frames, our proposed Conv-VRNN approach still performs better even if we do not use optical flow features. This demonstrates that it is more effective to design the generative model to directly capture the temporal information instead of relying on low-level optical flow features.

Target	Methods	1-shot (K=1)	5-shot (K=5)	10-shot (K=10)
UCSD Ped 1	Pre-trained	66.87	66.87	66.87
	Fine-tuned	71.7	74.52	74.68
	Ours	**78.44**	**81.43**	**81.62**
UCSD Ped 2	Pre-trained	62.53	62.53	62.53
	Fine-tuned	65.58	72.63	78.32
	Ours	**83.08**	**86.41**	**90.21**
CUHK Avenue	Pre-trained	64.32	64.32	64.32
	Fine-tuned	66.7	67.12	70.61
	Ours	**72.62**	**74.68**	**79.02**
UR Fall	Pre-trained	50.87	50.87	50.87
	Fine-tuned	57.02	58.08	62.82
	Ours	**74.59**	**79.08**	**81.85**

Table 6.8: Comparison of K-shot ($K = 1, 5, 10$) scene-adaptive anomaly detection under the cross-dataset testing setting. We report results in terms of AUC (%) using the UCF crime dataset for meta-training. We compare our results with two baseline methods. Our results demonstrate the effectiveness of our method on few-shot scene-adaptive anomaly detection.

6.2 Few-shot Scene-adaptive Anomaly Detection

We use *r-GAN* as backbone network and demonstrate the effectiveness of our algorithm in this section.

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**

Table 6.9: Ablation study for using different number of sampled tasks ($N = 1$ or $N = 5$) during each epoch of meta-training. The results show that even the performance of training with one task is better than fine-tuning. However, a larger number of tasks is able to train an improved model.

6.2.1 Results on Shanghai Tech

In this experiment, we use Shanghai Tech for both training and testing. In the train/test split used in Liu et al. [2018], both training and test sets contain videos from the same set of 13 scenes. This split does not fit our problem. Instead, we propose a split where the training set contains videos of 6 scenes from the original training set, and the test set contains videos of the remaining 7 scenes from the original test set. This will allow us to demonstrate the generalization ability of the proposed meta-learning approach. Table 6.6 shows the average AUC score over our test split of this dataset (7 scenes). Our model outperforms the two baselines.

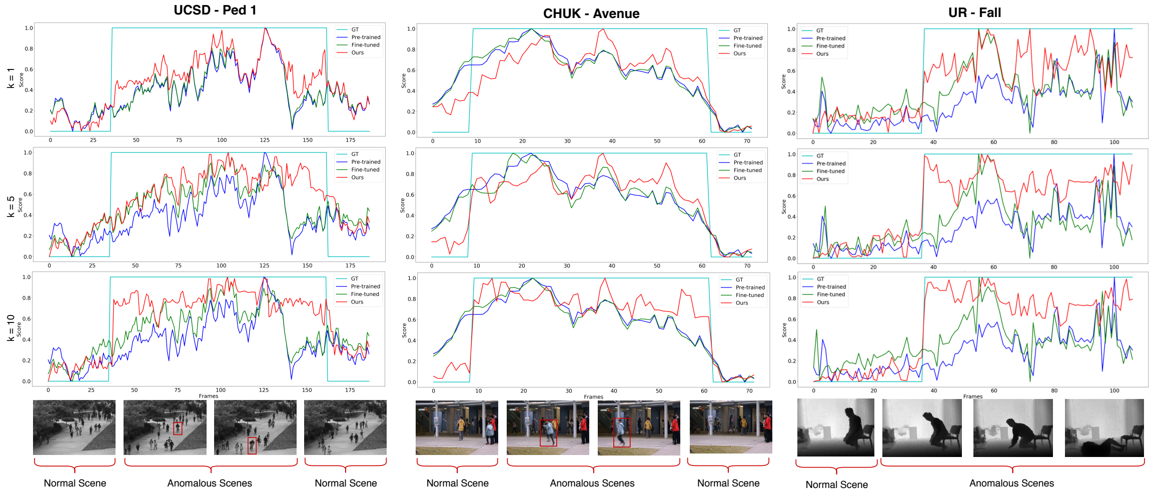


Figure 6.3: Qualitative results on three benchmark datasets using a pre-trained model on the Shanghai Tech dataset. Different columns represent results on different datasets. Each row shows few-shot scene-adaptive anomaly detection results with different numbers of training samples K. The red bounding boxes showing the abnormal event localization are for visualization purposes. They are not the outputs of our model which only predicts an anomaly score at the frame level.

6.2.2 Cross-dataset Testing

To demonstrate the generalization power of our approach, we also perform cross-dataset testing. In this experiment, we use either Shanghai Tech (the original training set) or UCF crime for meta-training, then use the other datasets (UCSD Ped1, UCSD Ped2, CUHK Avenue and UR Fall) for meta-testing. We present our cross-dataset testing results in Table 6.7 for Shanghai Tech and 6.8 for UCF crime. Compared with Table 6.6, the improvement of our approach over the baselines in Table 6.7 and Table 6.8 is even more significant (e.g. more than 20% in some cases). It is particularly exciting that our model can successfully adapt to the UR Fall dataset, considering this dataset contains depth images and scenes that are drastically different from those used during meta-training.

6.2.3 Ablation Study

In this study, we show the effect of the batch size (i.e. the number of sampled scenes) during the meta-training process. For this study, we train r-GAN on the Shanghai Tech dataset and test on Ped 1, Ped 2 and CUHK. We experiment with sampling either one ($N = 1$) or five ($N = 5$) tasks in each epoch during meta-training. Table 6.9 shows the comparison. Overall, using our approach with $N = 1$ performs better than simple fine-tuning, but not as good as $N = 5$. One explanation is that by having access to multiple scenes in one epoch, the model is less likely to overfit to any specific scene.

6.2.4 Qualitative Results

Figure 6.3 shows qualitative examples of detected anomalies. We visualize the anomaly scores on the frames in a video. We compare our method with the baselines in one graph for different values of K and different datasets.

Chapter 7

Conclusion

We investigate the problem of anomaly detection using surveillance videos by introducing a new problem called *few-shot scene-adaptive anomaly detection* based on a future frame prediction framework. Given a few frames captured from a new scene, our goal is to produce an anomaly detection model specifically adapted to this scene. We believe this new problem setup is closer to the real-world deployment of anomaly detection systems. Also, we purpose a meta-learning based approach to this problem. During training, we have access to videos from multiple scenes. We use these videos to construct a collection of tasks, where each task is a few-shot scene-adaptive anomaly detection task. Our model learns to effectively adapt to a new task with only a few frames from the corresponding scene. To demonstrate the effectiveness of our algorithm, we propose two novel state-of-the-art backbone architectures and use the one with better performance for examination. Experimental results show that our proposed approach significantly outperforms other alternative methods.

Bibliography

D. Abati, A. Porrello, S. Calderara, and R. Cucchiara. Latent space autoregression for novelty detection. In *CVPR*, 2019.

R. Chalapathy, A. K. Menon, and S. Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017.

J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In *NeurIPS*, 2015.

Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR*, 2011.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

A. Del Giorno, J. A. Bagnell, and M. Hebert. A discriminative framework for anomaly detection in large videos. In *ECCV*, 2016.

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*, 2015.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1705–1714, 2019.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *CVPR*, 2016.

J. R. Hershey and P. A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *ICASSP*, 2007.

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

J. Kim and K. Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, 2009.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, 2015.

B. Kwolek and M. Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine*, 2014.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015.

W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection–a new baseline. In *CVPR*, 2018.

C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, 2013.

W. Luo, W. Liu, and S. Gao. Remembering history with convolutional lstm for anomaly detection. In *ICME*, 2017a.

W. Luo, W. Liu, and S. Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *ICCV*, 2017b.

V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010.

J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*, 2011.

M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

J. R. Medel and A. Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

O. Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *Constructive Machine Learning Workshop at NeurIPS*, 2016.

R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11996–12004, 2019.

T. Munkhdalai and H. Yu. Meta networks. In *ICML*, 2017.

J. Nogas, S. S. Khan, and A. Mihailidis. Deepfall–non-invasive fall detection with deep spatio-temporal convolutional autoencoders. *arXiv preprint arXiv:1809.00977*, 2018.

J. Nogas, S. S. Khan, and A. Mihailidis. Fall detection from thermal camera using convolutional lstm autoencoder. Technical report, 2019.

D. Pollard. Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, 1991.

S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. 2016.

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*, 2015.

M. Sabokrou, M. Fathy, and M. Hoseini. Video anomaly detection and localization based on the sparsity and reconstruction error of auto-encoder. *Electronics Letters*, 2016.

M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. In *CVPR*, 2018.

A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016.

J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. In *CVPR*, 2018.

F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

R. Tudor Ionescu, S. Smeureanu, B. Alexe, and M. Popescu. Unmasking the abnormal events in video. In *ICCV*, 2017.

F. Tung, J. S. Zelek, and D. A. Clausi. Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance. *Image and Vision Computing*, 2011.

O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003.

S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *CVPR*, 2010.

S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, 2015.

B. Zhao, L. Fei-Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In *CVPR*, 2011.