ENPM809K Final Project
Weakly Supervised Multi-Task Variational Auto-Encoder for Video Anomaly Detection

Pin-Hao Huang
University of Maryland, College Park
pinhao17@umd.edu

Po-Lun Chen
University of Maryland, College Park
pchen115@umd.edu

Jeffin Johny Kachappilly
University of Maryland, College Park
jeffinjk@umd.edu

Abstract

The goal for Video anomaly detection (VAD) aims to identify abnormal activities in a video sequence, more specifically, to make frame-level predictions indicating whether the frame is normal or not. Typically, existing works addressing VAD can be formulated into two categories: one-class formulation which only normal data is available during training and the weakly-supervised setting, which only video-level annotations are provided. In this project, we focused on the weakly-supervised setting and designed a Multi-Task Variational Auto-Encoder to generate pseudo normal and abnormal video features. We evaluated our models on two commonly used video anomaly detection dataset: the small scale ShanghaiTech and the large scale UCF-Crime dataset.

1. Introduction

Nowadays, the prevalence of surveillance cameras play a critical role in maintaining public order and safety. However, manually identifying abnormal events such as accidents, crime events in video sequences is exasperating and time consuming as these events are usually rare in nature. Therefore, it can be beneficial to the society if we develop technologies that can automatically detect abnormal events in videos for us. With the advancement of computer vision and deep learning algorithms recently, anomaly detection in images have long been addressed. In contrast, video anomaly detection is still an active and challenging problem yet to be solved. In this project, we aim to design deep learning approaches for the video anomaly detection task. First, we designed a naive method which serves as the baseline. Next, we planned to investigate few related works and come up with a novel approach that hopefully out performs our baseline.

2. Problem Statement

Given a video input, video anomaly detection aims to output per-frame predictions indicating whether the anomaly event falls in the underlying frame. Typically, the learning process is often categorized into two types of formulations. The first setting is the one-class setting, which during training, the learner only has access to normal data. The other setting is the so-called weakly supervised formulation, which the learner has access to both normal and abnormal data. However, only video level labels are given instead of per-frame labels for the abnormal data. In this project, we are going to focus on the second setting.

2.1. Problem Formulation

This project aims to solve the *weakly-supervised video anomaly detection* task. Given a video $\mathbf{x} = \{x_0, x_1, \cdots, x_t, \cdots, x_T\}$, the model outputs $\mathbf{y} = \{y_0, y_1, \cdots, y_t, \cdots, y_T\}$ where x_t represents each individual frames in the video and $y_t \in [0, 1]$ is the predicted score for each frame indicating whether it contains abnormal event.

2.2. Dataset

The formulated model has been tested on two popular multi-scene benchmark datasets for weak supervised anomaly detection: ShangaiTech [7] and UCF-Crime [8]. Most of the existing datasets only contain videos captured with one fixed-angle camera, and it lacks a diversity of scenes and view angles, while these datasets address this issue.

ShanghaiTech [7] is a medium-scale dataset. It contains 13 scenes with complex light conditions and camera angles. A total of 437 videos, which contain 107 anomalous videos comprising of 130 anomalous events. The anomalies in this dataset are caused by sudden motion, such as chasing and brawling. This characterization is more suitable for real-life scenarios, and the dataset has over 270,000 training frames with pixel level ground truth of abnormal events annotations.

Even though normal training data is available, in reality, this is not the actual case. Our approach mainly focuses on the usage of a weakly labeled dataset(i.e. noisy, limited, or imprecise sources are used to provide supervision signal for labeling large amounts of training data in a supervised learning setting), inspired by Zhong et al. [11] work. For that, a subset of test videos was included in the training videos to create a weakly supervised training set, such that both training and test contains all 13 scenes. The split is such that the training set is made up of 238 videos, and the testing set contains 199 videos. Table 1 shows the statistics of the dataset split.

Table 1: Statistics of the ShanghaiTech dataset

	Training Set	**Test Set**	**Total**
Normal Videos	175	155	330
Anomaly Videos	63	44	107
Total	238	199	437

UCF-Crime [8] is a large-scale dataset with over 1900 untrimmed videos totaling up to 128 hours of real-world surveillance footage. The background in UCF-Crime is not static, unlike what is found in Shanghai dataset. The anomalies comprise 13 different classes which are Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. The training set contains 1,610 videos with video-level labels and the remaining test set with 290 videos have frame-level labels. The training set consists of 800 normal and 810 anomalous videos and the testing set includes the remaining 150 normal and 140 anomalous videos. Both training and test have all 13 anomalies and some videos even have multiple anomalies as well. Table 2 shows the statistics of the dataset split.

Table 2: Statistics of the UCF-Crime dataset

	Training Set	**Test Set**	**Total**
Normal Videos	800	150	950
Anomaly Videos	810	140	950
Total	1610	290	1900

2.3. Evaluation Metric

The metric to evaluate the performance of video anomaly detection typically uses Area Under the Curve (AUC) of the frame-level Receiver Operator Characteristic (ROC). We follow this convention and use AUC score as our evaluation metric.

2.4. Related Works

Video action recognition, a task that aims to classify the action categories in a video, has a close relationship with the video anomaly detection task. [1] suggested an inflated 3D convolution architecture (I3D) that benefits from the ImageNet pretrained 2D models by inflating the 2D convolution kernels to 3D. With the advantage of the robust features learned by I3D, recent literature that addressed the video anomaly detection task employed I3D as the backbone feature extractor [2], [6], [9], [10]. More specifically, [2] proposed pseudo label generation and finetuned on the generated pseudo labels. [6] designed a transformer-based multi-sequence network and use self-training technique to address the VAD task. [9] proposed feature magnitude learning to separate anomaly and normal snippet features. [10] designed a dictionary learning technique to reconstruct normal features from anomaly features and unified a framework for both one-class VAD and weakly-supervised VAD.

3. Method

Given the input video $\mathbf{x} = \{x_0, x_1, \cdots, x_t, \cdots, x_T\}$, we group the video frames into snippet-level video sequence $\{\mathbf{v}_s\}_{s=1}^{S}$ with S snippets, where each snippet contains 16 consecutive frames. Then, following recent literature [2][6][9][10], we adopt a pre-trained I3D network [1] to extract snippet-level features $\{\mathbf{f}_s\}_{s=1}^{S}$, where $\mathbf{f}_s$ is a $n \times d$ matrix. n is the "n-crop" augmentation during inference which refers to cropping images into the center, four corners, and their mirrored counterparts during feature extraction, we suggest the readers refer to [1] for more details. d is the feature dimension. Following recent work, $n = 10$ and $d = 2048$ in our work. Next, we describe our approach to output per-frame anomaly predictions given the extracted snippet features as input.

3.1. Baseline Approach

A baseline approach is designed in the first place. First, We use a multi-layer perceptron (MLP) followed by a sigmoid function as the frame classifier: $\hat{\mathbf{y}}_s = \sigma(\phi(\mathbf{f}_s))$, where $\phi(\cdot)$ and $\sigma(\cdot)$ denote the MLP and the sigmoid function, respectively, and $\hat{\mathbf{y}}_s \in [0, 1]$ is the prediction score indicating the "anomalous" of the s-th snippet. Since no frame-wise label is available, one naive approach is to average the snippet scores as the video score $\hat{\mathbf{Y}} = \frac{1}{S}\sum_{s=1}^{S} \hat{\mathbf{y}}_s$ in order to supervise it directly with the corresponding video-level label. We use binary cross-entropy as the loss function:

$$\mathcal{L}(\mathbf{Y},\hat{\mathbf{Y}}) = -\frac{1}{N}\sum_{i=1}^{N}\mathbf{Y}_i\cdot log(\hat{\mathbf{Y}}_i)+(1-\mathbf{Y}_i)\cdot log(1-\hat{\mathbf{Y}}_i) \quad (1)$$

During inference, the snippet level predictions $\hat{\mathbf{y}}_s$ are used directly. To obtain the frame level predictions, for simplicity, we directly assign the snippet score to the frames within it.

3.2. Pseudo Feature Generation

Recently, few existing works were able to boost their model performances by utilizing generated pseudo labels [2][6]. On the other hand, [10] employed dictionary learning and used the attention mechanism to filter out normal-event features in order to generate pseudo abnormal features. We follow [10] to generate pseudo features. However, we came up with a different approach to design a Variational Auto-Encoder (VAE)[5] to achieve this.

3.2.1 Variational Auto-Encoder

The architecture of a standard VAE is shown in figure 1. A VAE consists of an encoder and a decoder. Given input x, (x could be a feature vector or an image for CNN based VAE), the encoder is a neural network $\mathbf{e}(\cdot)$ that maps the input to a Gaussian distribution in the latent space by predicting the means μ_x and standard deviations σ_x: $(\mu_x, \sigma_x) = \mathbf{e}(x)$. Sampled from μ_x, σ_x, one can obtain $z \sim \mathcal{N}(\mu_x, \sigma_x)$. z is then fed to a decoder $\mathbf{d}(\cdot)$ which is also a neural network that maps z back to the original feature space $\hat{x} = \mathbf{d}(z)$. The objective is to minimize the reconstruction error:

$$e_r = ||x - \hat{x}||^2 \quad (2)$$

Additionally, a regularisation term on the latent space is introduced to regularise the distribution output by the encoder to be close to the standard normal distribution. This regularisation term is expressed as the Kulback-Leibler divergence between the predicted distribution and the standard Gaussian distribution:

$$L_r = KL(\mathcal{N}(\mu_x, \sigma_x), \mathcal{N}(0,1)) \quad (3)$$

Once finished training, one can randomly sample $z \sim \mathcal{N}(0,1)$ and generate a new sample using the decoder $\tilde{x} = \mathbf{d}(z)$.

3.2.2 VAE for Pseudo Feature Generation

As shown in the previous section, one can train a VAE to learn to generate a certain type of features from the data points sampled from the standard Gaussian distribution. In this section, we explain how we can apply VAE to the video anomaly detection task.

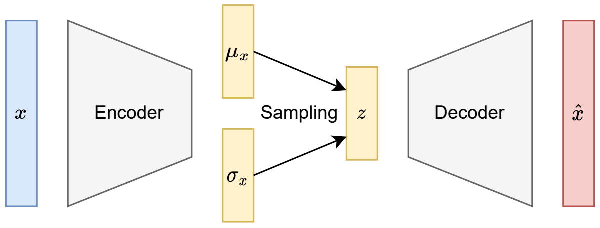

Figure 1: Architecture of a Variational Auto-Encoder

One naive approach is to learn different VAEs for each task, i.e. a normal VAE that learns to generate normal features and an abnormal VAE that learns to generate abnormal features. Each generated pseudo normal/abnormal features can then be represented as the feature of a single pseudo snippet, and can be trained jointly with the original snippet features to boost the performance of the model. Although this seems like a feasible approach, it has a few limitations. One of the main issue of this approach is that we cannot model the relationship between two generated features. In other words, this approach only allows the model to sample a set of individual features rather than a sequence of snippet features that contains temporal information. This may not be a problem if the model we used for anomaly detection simply treat each snippet features independently to predict the anomalous score. However, if we are using a more complex model that aims to aggregate nearby snippet features to allow temporal reasoning, the model may not be able to benefit from the generated features since they may not follow the temporal order. We will discuss how we address this issue in the next section.

3.2.3 Multi-Task VAE

As discussed in the previous section, the goal is to generate pseudo normal/abnormal features that maintain temporal consistency. To solve this, we designed a Multi-Task VAE that consists of an encoder and two decoders. The training of the Multi-Task VAE is no different from a conventional VAE. We briefly show the training procedure of our Multi-Task VAE in figure 2. First, a normal snippet feature x_n or an abnormal snippet feature x_a is fed into the encoder. The encoder maps the input feature into a Gaussian distribution in the latent space: $(\mu_{\mathbf{f}}^-, \sigma_{\mathbf{f}}^-) = \mathbf{e}(\mathbf{f}^-)$ or $(\mu_{\mathbf{f}}^+, \sigma_{\mathbf{f}}^+) = \mathbf{e}(\mathbf{f}^+)$. Note that the normal and abnormal snippet features are denoted as $\mathbf{f}^-$ and $\mathbf{f}^+$, respectively. Then we sample $z^- \sim \mathcal{N}(\mu_{\mathbf{f}}^-, \sigma_{\mathbf{f}}^-)$ and $z^+ \sim \mathcal{N}(\mu_{\mathbf{f}}^+, \sigma_{\mathbf{f}}^+)$, then fed into two separate decoders for feature reconstruction: $\hat{\mathbf{f}^-} = \mathbf{d}^-(z^-)$ and $\hat{\mathbf{f}^+} = \mathbf{d}^+(z^+)$. The objective is to minimize the reconstruction loss:

$$\mathcal{L}_r = ||\mathbf{f}^+ - \hat{\mathbf{f}^+}||^2 + ||\mathbf{f}^- - \hat{\mathbf{f}^-}||^2 \quad (4)$$

, and the Kulback-Leibler divergence between the predicted distribution and the standard deviation:

$$\mathcal{L}_{kl}^+ = KL(\mathcal{N}(\mu_\mathbf{f}^+, \sigma_\mathbf{f}^+), \mathcal{N}(0,1)) + KL(\mathcal{N}(\mu_\mathbf{f}^-, \sigma_\mathbf{f}^-), \mathcal{N}(0,1)) \quad (5)$$

The Multi-Task VAE is trained using the joint loss:

$$\mathcal{L} = \mathcal{L}_r + \alpha \cdot (\mathcal{L}_{kl}^+ + \mathcal{L}_{kl}^-) \quad (6)$$

where α is a hyperparameter to weight the kl divergence loss.
We are done describing the training and objectives of our Multi-Task VAE, and we are left to justify that it is a solution to generate temporal consistency pseudo features. The pseudo generating process is briefly illustrated in figure 3. To generate pseudo features, instead of randomly sampling from the standard Gaussian distribution, we take advantage of the original snippet features and used them as priors. To be more specific, when generating pseudo abnormal videos, given a sequence of snippet features from a normal video, we randomly sample $T/2$ snippet features from it and fed them to the encoder. Then we sample the latent feature vectors from the predicted distributions in the latent space and passed to the *abnormal* decoder to generate pseudo *abnormal* features. As a result we can obtain a pseudo abnormal video represented by $T/2$ original normal features and $T/2$ pseudo abnormal features. Similar process is applied to the abnormal videos to generate pseudo normal videos, but rather than sampling snippets randomly, instead we fed the entire sequence of snippet features to the encoder this time since we don't know which snippets are normal or abnormal in the first place. This is fine since during training, the normal and abnormal features shared the same encoder, therefore the normality of the original normal snippet features will still be preserved while the original abnormal snippet features will be mapped to generate pseudo normal features. Thus, we can obtain the pseudo normal videos. Note that although during training, the Multi-task VAE model didn't learn any temporal information between snippets, the above process can preserve the temporal consistency of the original video to some extent since the pseudo features are generated from the distribution in the latent space that came from the original features.

3.2.4 Feature Selection for Multi-Task VAE Training

In this section, we detail the feature selection process before training the Multi-Task VAE. It is rather simple for selecting normal features since they can be directly sampled from

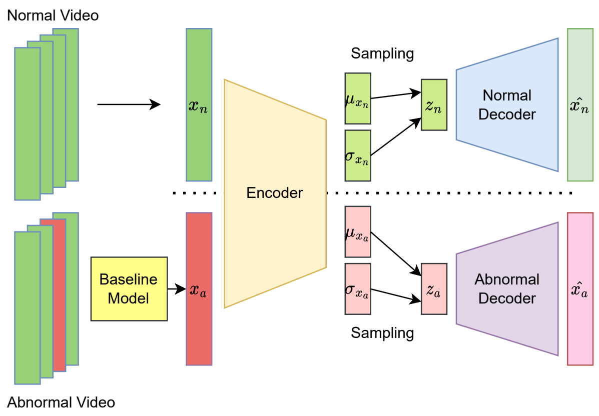

Figure 2: Architecture of our Multi-Task VAE with a shared encoder during training. First, a snippet feature is classified as normal or abnormal. (For normal videos we can ignore this step). Then, reconstruct normal features using normal decoder and abnormal features using abnormal decoder.

the snippet features from any normal videos. For abnormal features, we took advantage of the trained baseline model described in section 3.1 and select the snippets that have anomalous score higher than a defined threshold. We set this threshold to 0.5. Although the baseline isn't perfect, it still has a rather fair capability in identifying most of the abnormal snippets.

4. Experiments

4.1. Implementation Details

Follow [2][6][9][10], we employed the I3D network [1] pre-trained on the Kinetics-400 dataset [3] as the snippet feature extractor. To find the best configuration to train our model, we randomly sample 1/5 of the training dataset as our validation set. To ensure our validation set contains abnormal videos, we sample separately for normal and abnormal videos. We trained our baseline model in total 100 epochs through the Adam optimizer [4], with the learning rate of 0.001 and batch size of 32. Since the unification of the video size is required for batch training, each video is sampled to the same number of snippets using linear interpolation. Here we set the number of snippets for each video to $S = 32$.
Our designed Multi-Task VAE consists of a total 18 fully-connected layers, where the encoder has 6 layers: $[1024, 512, 256]$ with each number representing the output dimensions of each hidden layer. The input and output dimension of the encoder are 2048 and 256, respectively, where the output dimension is the dimension of the latent space. The decoder is simply the reverse layout of the encoder: $[512, 1024, 2048]$. Both the normal and abnormal

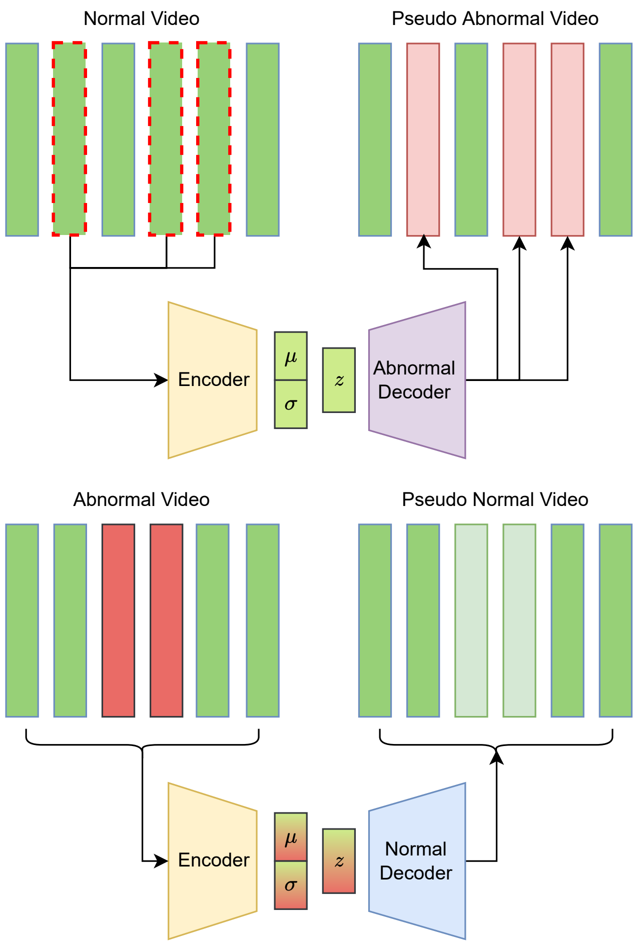

Figure 3: The process of generating pseudo normal/abnormal videos representations. (Top) To generate pseudo abnormal videos, we sample $T/2$ snippet features from a normal video and fed to the Multi-task VAE with the abnormal decoder to generate abnormal pseudo features. (Bottom) To generate pseudo normal videos, the entire abnormal video features are fed to the Multi-task VAE with the normal decoder to generate normal pseudo features.

decoder share the same structure. During training, for each video, if the number of total snippets exceeds 1024, we randomly sample 1024 snippets. Otherwise we kept the entire features as the batch input for the network. The network was trained in total 100 epochs using the Adam optimizer with the learning rate of 0.0001. The learning rate was decayed every epoch exponentially with a factor of $\gamma = 0.95$. Once the Multi-Task VAE was trained, we generated two pseudo videos for each video. More specifically, we generate 350 pseudo abnormal video from the original 175 normal videos and 126 pseudo normal videos from the original 63 abnormal videos.

4.2. ShanghaiTech Results

We evaluate our models on the ShanghaiTech testing dataset and compared with other existing frameworks. As mentioned in section 2.3, we use the AUC of the frame-level ROC as the evaluation metric.

4.2.1 Baseline with Multi-Task VAE

To justify the effectiveness of our proposed method, we designed a simple baseline model mentioned in section 3.1 and compared the performance between using only original data and using original data + generated pseudo features. As shown in table 3, the AUC score is **92.62**% for the baseline and **94.21**% for the baseline model trained with the additional generated pseudo features. The +1.59% improvement show that apparently the proposed method for pseudo feature generation can boost the performance of the baseline model even if the model is as simple as it contains only MLPs.

Table 3: Comparison of the baseline and baseline with Multi-Task VAE augment learning on the testing set of ShanghaiTech

Method	**Feature**	**AUC)** (%)
Baseline	I3D	92.62
Baseline + Multi-Task VAE	I3D	**94.21** (+1.59)

4.2.2 RTFM with Multi-Task VAE

Our proposed method can be built on top of any existing frameworks. To show this, we tried to reproduce the RTFM[9] framework by using their publicly available source code[*]. Note that although they reported a 97.21% AUC in the paper, we weren't able to reach that performance and only got 95.86%. We then modified some of their source codes to enable the joint training with our generated pseudo features and was able to reach 96.85%. We want to emphasize that although it seems like a boost in performance when using our generated pseudo features on our end, the conclusion is still not clear until we can completely reproduce their results. We leave this for our future work. Additionally, we show the comparison of our results and recent works in table 4. We also show the ROC curves of our different approaches in figure 5.

Table 4: Comparison of frame-level AUC on the testing set of the ShanghaiTech dataset. (RTFM* indicates our reproduced result of RTFM)

[*]For the reproduction of RTFM, we modified the code from https://github.com/tianyu0207/RTFM

Method	Feature	AUC (%)
MIST [2]	I3D	94.83
RTFM* [9]	I3D	95.86
RTFM* + Multi-Task VAE	I3D	96.85 (+0.99)
RTFM [9]	I3D	97.21
MSL [6]	I3D	96.08
S3R [10]	I3D	97.48

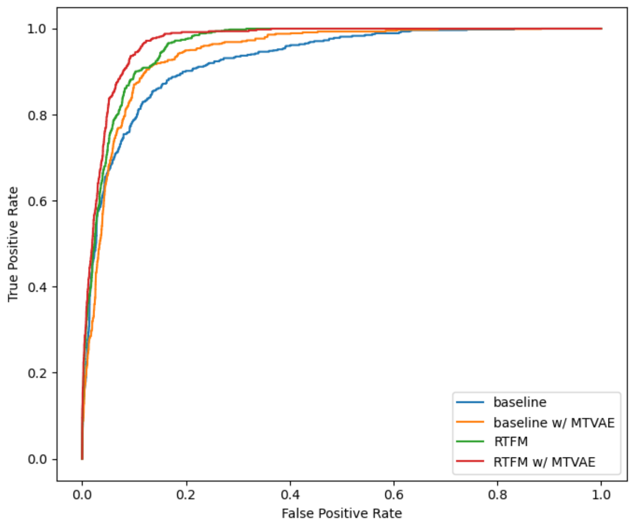

Figure 4: The ROC curves of the baseline model, baseline model with Multi-Task VAE generated pseudo features, our reproduced RTFM, and RTFM with Multi-Task VAE generated pseudo features

4.2.3 Ablation Study on ShanghaiTech

We conducted several ablation study to find the best architecture to generate pseudo features. As mentioned in section 4.1, we randomly sampled 1/5 videos from the training normal videos and 1/5 videos from the training abnormal videos to form the validation set. We then trained our Multi-Task VAE and report the AUC of the validation set. We used our baseline model as our video anomaly detector.

Architecture We tested three architectures, each with different layers and different dimensions in the latent space. We denote the architectures as Arch1, Arch2 and Arch3, where Arch1: [1024, 512, 256], Arch2: [1024, 512, 256, 128], Arch3: [1024, 512, 512, 256, 256]. The numbers represent the output dimension of each layer of the encoder. We show the ablation study in Table 5.

Table 5: Ablation Study on different architecture for the Multi-Task VAE on the validation set of ShanghaiTech)

Architecture	AUC (%)
Arch1	93.89
Arch2	92.91
Arch3	92.92

Sample ratio for psuedo videos generation We also validated the sampled ratio for generating pseudo videos using the Arch1 architecture. We chose the 1/2, 1, 2 settings, where each number represent the sampled ratio from the original videos (1/2 means randomly sample 1/2 of the total original videos).

Table 6: Ablation Study on different sample ratio for the Multi-Task VAE on the validation set of ShanghaiTech)

Sample Ratio	AUC (%)
1/2	93.12
1	93.89
2	93.31

4.3. UCF-Crime

We also evaluated our models on the testing set of the large scale UCF-Crime dataset. Since training on this dataset required a larger amount of time compared to the ShanghaiTech and due to the limitation of computation resources available, we only reported the reproduced results of RTFM [9] and the results of the joint training with the generated pseudo features from our Multi-Task VAE. The results are shown in table 7 along with the results of recent works. Similar to the ShanghaiTech dataset, we weren't able to reproduce the reported 84.30% result and instead got 83.30. Also, after adding the generated pseudo features, it didn't improve the performance but instead got a similar result to the original one. We made an assumption regarding this result: The UCF-Crime is a much larger dataset and contains longer videos, a larger and more complicated network might be required to learn a better VAE. Since we have limited time and computational resources, we leave this issue to be addressed in our future works.

Table 7: Comparison of frame-level AUC on the testing set of the UCF-Crime dataset. (RTFM* indicates our reproduced result of RTFM)

Method	Feature	AUC (%)
MIST [2]	I3D	82.30
RTFM* [9]	I3D	83.30
RTFM* + Multi-Task VAE	I3D	83.28 (-0.02)
RTFM [9]	I3D	84.30
MSL [6]	I3D	85.30
S3R [10]	I3D	85.99

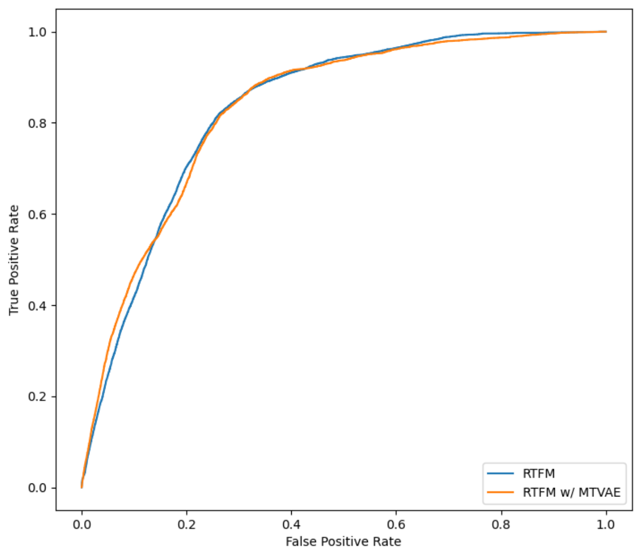

Figure 5: The ROC curves of the our reproduced RTFM, and RTFM with Multi-Task VAE generated pseudo features

5. Conclusion

In this project, we proposed a generative approach to generate pseudo video representations using a Multi-Task Variational Auto-Encoder. To our knowledge, this is the first work that applied a generative-based approach to tackle the video anomaly detection task. Our approach leveraged the shared encoder architecture that enables the network to generate pseudo video features that came from different distribution. This allows us to create pseudo video representations that preserves the temporal consistency in the original videos without the need of a more sophisticated network that is capable of generating an entire sequence of features. We also showed that our generated pseudo video features can improve the performance of a model even if it's a simple network with only MLPs.

Limitations and Future Works. As we looked in the results and visualized the original data, we found out that the model fails to identify the abnormal event if the event itself is barely visible in the frame, i.e., the objects that caused the abnormal events are too small. This is reasonable as our model classified the frames on a global scale (the global scale here means that the feature are extracted as the global feature of each snippet), and an object detecting or segmentation mechanism might be required to capture local events. This requires a more sophisticated architecture and definitely needs more computational resources, we leave this to our future works. Also, due to limited computational resources, we weren't able to investigate much about the UCF-Crime dataset (i.e. using larger network, generating more pseudo features, etc.). Further improvement targeting this dataset should be made in the future. In addition, our current approach was based on the weakly-supervised setting but may not applied to the one class setting where only normal videos are available during training. Finding a way to accommodate the one class setting is an interesting and challenging topic and is definitely worth exploration in our future works.

References

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[2] J.-C. Feng, F.-T. Hong, and W.-S. Zheng. Mist: Multiple instance self-training framework for video anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14009–14018, 2021.

[3] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[6] S. Li, F. Liu, and L. Jiao. Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. *Proceedings of the AAAI, Virtual*, 24, 2022.

[7] W. Liu, D. L. W. Luo, and S. Gao. Future frame prediction for anomaly detection – a new baseline. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018.

[9] Y. Tian, G. Pang, Y. Chen, R. Singh, J. W. Verjans, and G. Carneiro. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4975–4986, 2021.

[10] J.-C. Wu, H.-Y. Hsieh, D.-J. Chen, C.-S. Fuh, and T.-L. Liu. Self-supervised sparse representation for video anomaly detection.

[11] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1237–1246, 2019.