Video Anomaly Detection Paper Critique

Zackary Amo

June 24, 2024

Abstract

The purpose of this paper is to analyze and critique the theoretical aspects of a paper called ”Learning Normal Dynamics in Videos with Meta Prototype Network” (LND-VMPN) written by Hui Lv, Chen Chen, Zhen Cui, Chunyan Xu, Yong Li, Jian Yang. The paper introduces a state of the art method for video anomaly detection using auto encoders, dynamic processing units, and meta-learning. I will breakdown the datasets used, preprocessing steps, model architechture, objective functions, meta-learning implementation, and the process of validating results. As I break these aspects down, I will point out some possible areas of improvement and question some theoretical choices. The review provides insight into the paper’s contributions and suggest potential improvements for future video anomaly detection research.

Contents

1 Introduction

1.1 What is VAD?

Video anomaly detection (VAD) is a popular machine learning problem that has been researched for decades. Reliable VAD systems have applications in various fields as surveillance, traffic monitoring, healthcare, and fraud detection. The crux of the problem is to model ”normal” dynamics in the video data, so that ”abnormal” events can be detected. This problem is difficult to solve in situations where what is considered an ”anomaly” is ambiguous [1]. How can one build a model that can learn what constitutes an anomaly in different situations?

1.2 Problem Statement

LNDVMPN introduces method combining auto encoders (AEs), dynamic prototype units (DPUs), and meta-learning to detect anomalies in video data. AEs are models that encode input data into a compressed format and then decodes it to rebuild the original input. AEs are useful for anomaly detection because they can be used to encode normal patterns in the video data and reconstruct the original frames using the learned patterns. Then the difference between the real image and the output image is usually a good indicator if there were any abnormal events. However, auto encoders alone face the issue of overgeneralizing, meaning all events including anomalies can be predicted well due to how good convolutional neural networks have become [1].

Papers before LNDVMPN introduced a memory bank into the AE to store encoded normal data, but this approach is memory-consuming and is not able to adapt well to unseen data. In real world applications, VAD models need the ability to adapt quickly to unseen data. Without this, the model will only be able to detect anomalies that were present in the training data, making this model useless for most real world use cases [1].

1.3 Research Objectives

LNDVMPN proposes integrating a DPU within the AE to extract normal prototypes from the training data in real-time. This allows for much faster processing time since it avoids reads to a global memory bank. Additionally, the paper introduces a meta-learning algorithm for the DPU, enablings the model to adapt to new, unseen data, therefore improving generalization. This approach achieves state-of-the-art (SOTA) performance on popular VAD datasets while significantly increasing the processing speed and ability to adapt to new data [1].

1.4 Structure of the Review/Critique

The review will start with an in-depth overview and critique of each theoretical aspect of the paper, including the datasets, data preprocessing, model architecture, and objective functions. After that, the methods for validating the results will be examined. In the Critique section, the overall strengths and weaknesses of the model architecture and model performance will be discussed, along with the writing style and the depth of explanation. The review will conclude with potential improvements and future research in VAD.

2 Theoretical Aspects

2.1 Dataset Selection

There were four datasets used in this paper: UCSD Ped1 (34 training videos and 36 testing videos), UCSD Ped2 (16 training videos and 12 testing videos), CUHK Avenue (16 training videos and 21 test videos), ShanghaiTech (330 training videos and 107 testing videos), and the UCF Crime dataset with an undisclosed amount of videos. These datasets are very popular in VAD research which gives a good benchmark for comparing the proposed model to other existing SOTA models [1].

Although there is not a lot of training and testing data used here, the amount of data used is sufficient for training the DPU and meta learning algorithm. This mainly due to the architecture of the model enabling it to learn normal patterns in real-time. This aligns with the scarcity of data in most real-world VAD applications.

For the base AE model, more data could potentially enhance the model's performance. Since the AE is "frozen" after training, meaning the parameters are not modified, more data might help the AE encode and decode normal patterns better. AEs can suffer from over generalization so adding more data should be done with care. Excess data might lead to the AE encoding anomalies or noise as normal patterns.

2.2 Data Preprocessing

In the paper, the video data was resized to 256×256 and normalized to the range $[-1, 1]$ [1]. I think this is sufficient for the task at hand, however, one can likely use imagery processing techniques to enhance normal patterns in each frame to increase the accuracy. These techniques include histogram equalization, noise reduction, and edge enhancement which could enhance important normal features and improve model performance. Although, this would likely lower the inference speed of the model. The optimal preprocessing steps vary with each application.

2.3 Model Architecture

The overall model consists of three parts: the AE, the DPU, and the meta-learning algorithm.

2.3.1 Auto Encoder

The specifics of the AE will not be covered extensively since it is treated as a pretrained model in the context of the entire model and the use of the DPU and the meta-learning algorithm are the important parts. However, to provide an overview, the AE takes T video frames as input: $\mathbf{x}_k = (I_{k-T+1}, I_{k-T+1}, ..., I_k)$ for the k-th video. The AE encodes these frames giving $\mathbf{X}_t = f_\theta(\mathbf{x}_t) \in \mathbf{R}^{h,w,c}$, which is then passed to the DPU. The output of the DPU is then put though the remaining layers of the AE producing the output $y_k = I_{k+1}$. This output y_k is the prediction of the next frame hence why $y_k = I_{k+1}$ [1].

2.3.2 Dynamic Prototype Unit

The DPU aggregates the normal encoding of the frames input by the AE with a normalcy encoding through three processes called Attention, Encoding, and Retrieving. Before the attention process, the t-th AE encoding map $\mathbf{X}_t$ is broken down into $N = wh$ vectors $\{\mathbf{x}_t^1, \mathbf{x}_t^2, ..., \mathbf{x}_t^N\}$ [1].

Attention generates a pool of dynamic prototypes. This is done using M attention mapping functions $\{\psi_m : \mathbf{R}^c \rightarrow \mathbf{R}\}_{m=1}^{M}$ to get normalcy weights $\mathbf{W}_t^m = \psi_t^m(\mathbf{X}_t)$ (i.e. m-th normalcy map) [1].

Ensemble then creates a prototype pool $\mathbf{P}_t$ of prototypes p_t^m from the N encoding vectors and the normalcy weights using the equation:

$$\mathbf{p}_t^m = \sum_{n=1}^{N} \frac{\mathbf{w}_t^{n,m}}{\sum_{n'=1}^{N} \mathbf{w}_t^{n',m}} \mathbf{x}_t^n \tag{1}$$

Hence, $\mathbf{P}_t = \{\mathbf{p}_t^m\}_{m=1}^{M}$ [1].

Retrieving uses the encoding vectors $\mathbf{x}_t^n$ as queries to get prototypes in the prototype pool and to reconstruct a normalcy encoding as:

$$\tilde{\mathbf{x}}_t^n = \sum_{m=1}^{M} \beta_t^{n,m} \mathbf{p}_t^m \tag{2}$$

where

$$\beta_t^{n,m} = \frac{\mathbf{x}_t^n \mathbf{p}_t^n}{\sum_{m'=1}^{M} \mathbf{x}_t^n \mathbf{p}_t^{m'}} \tag{3}$$

This normalcy map is then combined withe original AE encoding using a channel-wise sum operation [1].

The DPU, as expected, processes and extracts normal features from video frames in real-time. Not only are the normal patterns boosted, the abnormal events are suppressed as desired. When the AE decodes the normalcy-encoded frames, anomalies more likely to be highlighted, making the error between the predicted and actual frames high enough to indicate an anomaly [1].

2.4 Objective Functions

LNDVMPN uses several objective functions to evaluate the loss of normalcy dynamics and feature reconstruction in the DPU and the overall frame prediction of the model. The overall loss function is divided into two parts, the loss of the frame prediction L_{fra} and the loss of the feature reconstruction L_{fea}:

$$L = L_{fra} + \lambda_1 L_{fea} \tag{4}$$

where λ_1 is balancing weight [1].

The frame prediction loss is picked to be the Euclidean distance formula between the actual frame y_t and the predicted frame $\hat{y}_t$ [1]:

$$L_{fra} = ||\hat{y}_t - y_t|| \tag{5}$$

The feature reconstruction loss is further split into two loss functions for feature compactness L_c and feature diversity L_d:

$$L_{fea} = L_c + \lambda_2 L_d \tag{6}$$

where λ_2 is another balancing weight [1].

The compactness term measures the overall ability of the DPU to reconstruct the encoding vectors with a small amount of descriptive prototypes. It is calculated as the mean Euclidean distance between each encoding vector and its most descriptive prototype:

$$L_c = \frac{1}{N} \sum_{n=1}^{N} ||x_t^n - p_t^*|| \tag{7}$$

where

$$* = argmax_{m \in [1,M]} \beta_t^{n,m} \tag{8}$$

* is the indices of the most descriptive prototype [1].

The diversity term makes each prototype sufficiently distinct in terms of its descriptive features:

$$L_d = \frac{2}{M(M-1)} \sum_{m=1}^{M} \sum_{m'=1}^{M} [-\|p_m - p_{m'}\| + \gamma]_+ \tag{9}$$

where γ is the desired difference between prototypes and $[x]_+ = max(0, x)$ [1].

The paper did not provide reasoning for using Euclidean distance in the terms L_{fra}, L_c, and L_d. This metric is useful for the goal of optimizing the descriptiveness and compactness of normalcy encoding and the prototypes. However, it would be useful to see how other similarity metrics such as mean-squared error and mean-absolute error impact the model's performance.

Additionally, adding an orthogonality term in the diversity loss function might improve model performance. This could lead to a more descriptive and potential for a lower dimensional prototype pool. However, one would need to be careful that the prototype pool does not become too descriptive, picking up on anomalies. The goal of the DPU is reconstruct the AE encoding with normal features only.

2.5 Meta-Learning Implementation

To introduce adaption to the VAD model, LNDVMPN introduces a few-shot meta-learning algorithm. Few-shot learning is a machine learning technique that is used in scenarios where little data is available. This approach adapts the base model to a few frames of a normal scene. In the case of surveillance, these frames would be images of base scene that the camera will be capturing. This will allow the model the desired ability to adapt to new scenes even if the base model has not been explicitly trained on this data [1].

The paper formulates an algorithm f_θ, referred to as the Meta-Prototype Unit (MPU), that is a composite of the AE decoder and the DPU:

$$f_\theta = D_\delta \circ P_\tau \tag{10}$$

where $\theta = \delta \cup \tau$, δ, and τ are the parameters to their respective models [1].

In the context of the entire model, the MPU operates as follows:

$$f_\theta(E_\eta(x)) = D_\delta(P_\tau(E_\eta(x))) \tag{11}$$

The Meta-Learning algorithm is used in the training phase of the model development. The MPU starts with the model parameters θ_0 and T iterations of an update function U is applied. The update functions is defined as:

$$U(\theta, \nabla_\theta L, \alpha) = \theta - \alpha \odot \nabla_\theta L \tag{12}$$

where α defines the step size of the gradient descent and $\odot$ is the element-wise product of two vectors. α is a vector the same size as parameter set θ [1]. This allows for different step sizes for each corresponding parameter in θ.

Each training iteration uses a randomly selected input/output pair $(\mathbf{x}_k, \mathbf{y}_k)$ from the training set. One iteration of the initial model θ_0 is as follows:

$$\theta_0^{i+1} = U(\theta_0^i, \nabla_{\theta_0^i} L(\mathbf{y}_k), f_{\theta_0}(E_\eta(\mathbf{x}_k))) \tag{13}$$

After T iterations, we get the adapted model $\hat{\theta}$ [1].

This model $\hat{\theta}$ is then tested using another random input/output training pair (x_j, y_j). The initialization model θ_0^* and the update step size α^* is then updated by the following equation [1]:

$$\theta_0^*, \alpha^* = argmin_{\theta_0, \alpha}\mathbf{E}[L(y_j, f_{\hat{\theta}}(E_\eta(x_j)))] \tag{14}$$

There are several potential problems with this meta-learning approach. The most glaring issue is the assumption that historical data of the target scene exists. Additional efforts may be needed to collect a few images or a short video of the target scene for model adaption. Another issue could be the quality of the scene adaption frames. If the target scene only has a few low-quality frames to train with, the final model may not pick up on the correct normal patterns. It is also very possible that the target scene is changing frequently. In this case the model may need to be adapted frequently. Perhaps a periodic, automated adaption of the model can take place to keep the model up to date. The frequency of this adaption should be dependent on how often the scene changes. This would also assume that the computational resources are available when the model needs to adapt.

2.6 Training

The training of the entire model is split into two parts, training the base AE model and then meta-training. The AE is trained on all training data to optimize the frame prediction loss function L_{fra} using gradient descent [1]. The meta-training process is as described in the previous subsection.

2.7 Testing

Testing this model consists of getting the first few frames of the testing data to adapt the model to. Then one can use the updated model to evaluate the performance of the model [1]. The model evaluation techniques used in this paper will be analyzed in the next section.

3 Validation of Results

3.1 Testing Scenarios

LNDVMPN evaluates the performance of the model in two problems settings. This first setting only uses normal data to train the model. This provides a good way to compare the baseline performance of the model with other SOTA models. The second setting focuses on testing the meta-learning aspect of the model by using training and testing data from multiple datasets. This gauges the model's ability to adapt to new scenarios [1].

3.2 Evaluation Metrics

The evaluation metric used in this paper is area under the ROC curve (AUC). This metric is widely used for model evaluation, providing a great way to compare performance with other existing models.

3.3 Hyperparameters

LNDVMPN implements the model with a learning rate of 0.0001 and a batch size of 4. The DPU is placed into the AE after the third convolutional layer counting from the bottom of the model. The training epochs are 60 for Ped1, 60 for Ped2, 60 for CUHK Avenue, and 10 for ShanghaiTech. The MPU is trained after the AE is trained and frozen. The learning rate for the update function U is 0.00001 and there are 1000 training epochs. The constants defined in the section 2.4 are set to the following: $\lambda_1 = 1$, $\lambda_2 = 0.01$, and $\gamma = 1$ [1].

The rationale for choosing these hyper parameters is not provided in the paper. One would need to train and test this model on their own to find the optimal hyperparameters. Nonetheless, having 60 training epochs for the smaller datasets and only 10 for the larger dataset might overfit to the small datasets relative to the larger one. My initial instinct would be to lower the training epochs for the smaller datasets and/or increase the training epochs for the larger dataset.

3.4 Results

The paper compares the base model's performance with other SOTA VAD models. The base model with the DPU performs on par with the other SOTA methods and scores the best on the CUHK Avenue dataset [1].

When conducting cross-dataset training using the meta-training algorithm, the MPU outperformed the then-best VAD model rGAN [1]. This shows the effectiveness of the meta-learning approach to enhancing the DPU model.

In addition to increasing the SOTA accuracy scores, this model has 80 times faster inference speed than rGAN. MPU also has much faster update iterations than rGAN [1]. This is great for real-world applications that need anomalies to be detected quickly.

4 Critique

4.1 Strengths of the Model

The introduction of the DPU and meta-learning to VAD is a significant advancement in model accuracy in diverse scenarios and inference speed. This model also significantly reduced the amount of memory needed as compared to other AE approaches using a memory bank. All of these advancements are great step towards more reliable real-world VAD systems.

In addition to notable achievements, this model was trained and tested on multiple different datasets. This demonstrates the model's ability to generalize to different scenarios as desired.

4.2 Weaknesses of the Model

The increase in model performance was mostly due to the meta-learning algorithm used to adapt the model to new scenarios. This assumes that data from the desired scene already exists or is possible to collect.

In scenes where the environment is changing frequently, the model may need to be adapted periodically to maintain the performance gain from the meta-learning. To automate this periodic adaptation, additional work would need to be done and computational resources would be needed.

The hyperparameter selection was not explained in this paper. Most of the hyperparameters seemed reasonable, but 10 training epochs for the ShanghaiTech dataset may be insufficient.

My intuition is that the model will overfit to the smaller datasets since they have 60 training epochs and underfit to the ShanghaiTech dataset since its much larger. It is possible that the training epochs for the ShanghaiTech datasets were lowered to decrease the overall training time.

4.3 Validation And Testing

The paper used AUC as the evaluation technique which is very popular in VAD research. This provides a reliable way to compare different models. However, it may be useful to add more metrics for evaluating the models performance.

It also would be useful to test the model with larger datasets or for varying fields of application. Testing with larger datasets would test the models ability to scale. The performance decrease with the ShanghaiTech dataset may indicate scalability issues, so that may be worth testing further. Testing for different use cases would be useful as well. This would show if the model is able to adapt to different kinds of VAD tasks.

4.4 Writing Style and Explanations

The writing style of the paper is generally good. Most explanations were informative and straight to the point. However, there are some instances where the notation is unclear and where the reasoning behind certain choices was absent. For example, I believe that all the vectors and matrices should be made bold or else the reader can get confused what is a scalar and what is a vector or a matrix. Also, the lacks explanations for the chosen objective functions and hyperparameters. This makes it more difficult for academics to implement improvements to this paper and to similar VAD models.

5 Conclusion

Overall, this paper introduces significant advancements for VAD in real-world applications. The DPU enableds encoding of normal patterns in real-time and significantly reduced the amount of memory needed to perform VAD with AEs. The MPU allows the model to adapt to different scenarios with little data. It also led to increases in model accuracy and it's capability to generalize. This model also has much faster inference speed than previous models.

Despite the significant improvements, there are flaws in some of the theoretical aspects of the paper. For example, the increase in model accuracy requires there to be data from the scene of interest. Also, although the model is meant work well with little data, there may be issues scaling this model to larger datasets. Some aspects were completely void of explanation, making it difficult for readers to understand the rationale behind certain choices.

There is more interesting work to be done regarding VAD using DPUs and meta-learning. Improvements can be achieved by trying different objective functions, tweaking hyperparameters, and modifying the DPU and meta-learning algorithms. It would be interesting to evaluate this model performance in other VAD tasks other than surveillance, such as medical imaging and autonomous vehicles.

References

[1] Hui Lv, Chen Chen, Cui Zhen, Chunyan Xu, and Jian Yang. Learning normal dynamics in videos with meta prototype network. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2021.