Appearance-Motion Co-memory Network for Video Anomaly Detection

Final Year Project Report

A 4th Year Student Name

Le Duc Anh, Nguyen Ba Duong
HE130082, HE130658

Under the supervision of

Dr. Phan Duy Hung

Bachelor of Computer Science

Hoa Lac Campus – FPT University

Spring 2021

DECLARATION

Project Title	Appearance-Motion Co-memory Network for Video Anomaly Detection
Author	*Le Duc Anh, Nguyen Ba Duong*
Student ID	HE130082, HE130658
Supervisor	Dr. Phan Duy Hung

I declare that this thesis entitled *Appearance-Motion Co-memory Network for Video Anomaly Detection* is the result of my own work except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Le Duc Anh, Nguyen Ba Duong
HE130082, HE130658

Department of Computer Science
Hoa Lac Campus – FPT University

Date: April 27, 2021

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my research supervisor, *Dr. Phan Duy Hung*, for giving me the opportunity to do research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

I am greatly indebted to my honorable teachers of the Department of Computer Science at the Hoa Lac Campus – FPT University who taught me during the course of my study. Without any doubt, their teaching and guidance have completely transformed me to a person that I am today.

I am extremely thankful to my parents for their unconditional love, endless prayers, caring and immense sacrifices for educating and preparing me for my future. I would like to say thanks to my friends and relatives for their kind support and care.

Finally, I would like to thank all the people who have supported me to complete the project work directly or indirectly.

Le Duc Anh, Nguyen Ba Duong
Hoa Lac Campus – FPT University
Date: April 27, 2021

Dedicated to
everyone who loves Video Anomaly Detection area.

– Le Duc Anh, Nguyen Ba Duong

ABSTRACT

Recently, video anomaly detection is currently a challenge and has attracted much attention from many researchers, which apply in the variation field like traffic accident detection, violence detection, intrusion detection systems, surveillance systems. The most common approach adopted the convolutional autoencoder that fused with appearance and motion representation to enhance the model's ability to describe each ordinary object's spatial and temporal behavior and quantifies the predicted error during the testing process. However, the drawback of this approach is the limit number of normal patterns which a model can learn. When training with a considerable amount of normal data, information about the normal pattern recorded in the hidden cells will be compressed, leading to missing or misleading information. This limitation is handled by a completely new improved model that applies memory modules to both the motion-appearance network and shares the same encoder, decoder. The testing on the two public datasets has shown that our model is efficient and indicates significant results improvements.

Keywords: Co-memory network, Video Anomaly Detection, Optical Flow Estimation, Frame Prediction.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem and Motivations

In the age of the information technology explosion, the application of new technologies to surveillance systems or intrusion detection systems such as deep learning plays a crucial role in improving system efficiency or solving new problems. Indeed, the issue of detecting video anomalies plays essential terms and is a consequence of this development. Besides, this problem had garnered more and more attention over the past decade for its essential application in surveillance systems, which can serve as a coarse event filter, when the system developed. However, evaluating the problem of detecting abnormal events in the video remains challenging for two features of anomaly [7]. First, it's difficult to define whether the determined events are abnormal or not, and the definition has a massive impact on the circumstances in which the event is placed (For example, when a car stops on a highway would be abnormal. On other hand, if the vehicle park in the parking, it would be a regular event). Next, anomalous data is challenging to collect because abnormal events are very diverse, and their frequency of occurrence is much less than regular events. For that reason, instead of considering the problem of detecting anomalous events in the video as a supervision approach using anomalous data for training, building a model to describe normal data is a more accurate choice. Following the conventional method, the scientists state that when normal data use the inputs of the model, what describes usual events with normality events as training data [8]. Indeed, with much scientific evidence, this approach has achieved awe-inspiring results in public datasets.

After the revolution of remarkable application of deep learning in computer vision [9, 10, 11], video anomaly detection has been shifted the approach method from using the hand-crafted method [12, 13], which human time consuming and less effective, to the deep learning method [14, 15, 16]. In general, the standard method to solve this problem is to adopt deep encoder-decoder architecture [17, 18], a powerful method to build a model to describe multi-dimensional data types, to build a model to learn regular events. The strategies adopted from the approach above are divided into main categories: the reconstruction-based framework and the prediction-based framework. Still, in general, both above methods have two main parts: the encoder, which transforms input data in a lower-dimensional latent representation, and the decoder, which extracts latent representation into higher dimensional data. In particular solutions, the goal of this approach is minimum the prediction error or reconstruction error of normal data and maximum error of anomaly outliers because lots number of studies assumed that the model will perform worse against unseen data which not contain in training data and is an unusual event in our case [19, 20, 21]. However, the limitation when using the method above is that there are many cases where the anomaly detection model works well with anomaly data. The reason for this situation is that the encoder extracts the common compositional pattern, which appears in both abnormal and normal data, such as edge patterns, instead of a normal feature. Moreover, the decoder works too ”good” when decoding bottleneck lower-dimensional features. To overcome that drawback, some scientists have used some extended methods to transform or augment the latent representation by guiding the normal memory [22, 23, 16]. Using the above method is to augment latent representation that maximizes the distance between normal and abnormal features. Consequently, the model is likely to maximum the reconstruction or prediction error of unusual events. Besides, an extraordinary event is not only based on spatial information but also the motion pattern, which helps detect an abnormal event to solve the problem of lacking motion information the memory network approached which not included.

Inspired by [23, 15, 24], we introduced the model with two streams capable of extracting information from both appearance and motion and added with memory module guidance at each branch. In the training stage, we allow update operation to affect the memory module, which makes memories possible to record normal representation of both motion and appearance. In testing, we disable update operation to control memory size within an acceptable range. Our model is optimized according to the difference between the results obtained and the ground truth of the input data (optical flow - optical flow GT, predicted image - predicted image GT).

1.2 Related Works

1.2.1 Abnormal detection

Abnormal detection is an unsupervised learning problem, which refers to detect abnormal activities that divert observed normal events, and during the training process, the use of abnormal data is hard to modeling normally events due to the unpredictable nature of the data. For a particular solution, training datasets only contain normal activities which are more undemanding to collect data, and more approachable than abnormal activities [7]. Abnormal activities identification model has been built based on cluster method such as flexible genre model (FGM) [25]. There are also reconstructive and discriminative approaches. Reconstructive approaches such as deep structured energy based models (DSEBMs) [26], repulsive forces and sparse reconstruction [13], and generative models [27] encode normal information then learn their representation. Discriminative approaches such as Gaussian process regression [28], one-class support vector machines [29] use a probability allocation of normal data and decision boundary to contain normal data and eliminate abnormal instances. Using these approaches with high-dimensional data like images and videos, however, the results are often not satisfactory [30]

Some architectures have been created from the autoencoder model associated with CNN such as fully convolutional feed-forward autoencoder [19], spatiotemporal autoencoder [21], stacked RNN framework [31], and generative adversarial network (GAN) [14] gave quite promising results. But, they still have drawbacks. During reconstruction to detect anomalies, sometimes they can even reconstruct abnormalities into typical instances due to the ability of CNNs. Another method to detect abnormal detection is to train a network to predict future frames and compare the predicted frames corresponding to the future frame of input frames [15]. This method will increase runtime cost by extracting optical flow features of video frames but its performance is outstanding.

1.2.2 Memory networks

Long short-term memory (LSTM) [32] is a classic and popular memory, it is used to store information in hidden states during training models and reuse this information as needed. However, it also has limitations on the size of the cell or the information in the hidden state is compressed, causing the information to be transformed from the original information. Therefore, new memory network structures are born to solve problems better than classical memory. Kim et al.[33] solved most of the two notorious issues of unsupervised GAN training using a learnable memory network. Memory can significantly improve the performance of DGMs as well as density estimation, image generation, and missing value imputation, Li et al.[34].

Moreover, memory networks [35] also use the query, key, and value (QKV) concept. The QKV concept is often used when the target information of the current input exists at the other inputs. In this case, memory networks set the current input and the other inputs as the query and memory, respectively. The key and value are extracted from memory, and the correlation map of the query and memory is generated through a non-local matching operation of the query and key feature. Then, the weighted average value based on the correlation map is retrieved. The QKV concept is widely used in a variety of tasks, including natural language processing [36, 37, 38], image processing [31,54], and video recognition [39, 40, 41]. In VOS, STM [42] has achieved significant success by repurposing the concept of the QKV.

1.3 Contribution

In brief, our approach considers both appearance and motion features based on the perception that compared with normal behaviors, an abnormal behavior differs in their appearance and motion patterns by using memory module. In summary, this thesis makes the following contributions:

- We proposed a model that includes a motion branch with a motion representation extraction task combined with an appearance branch to increase prediction efficiency. Also integrates the memory module into the motion branch to store information about the normal movement of objects.

- We proposed a loss function synthesized from memory, prediction, optical flow losses to take advantage of these loss functions to the training model.
- Our model was trained on 2 different datasets and resulted in 2.7% (UCSD Ped2 dataset [5]), 0.9% (CUHK Avenue dataset [6]) increase compared to the original MNAD model [16]. Hence, we have achieved state-of-the-art with both datasets Ped2 and Avenue for abnormal detection in video frames.
- We also provide a breakdown of memory size - an important factor in memory modules.

Chapter 2

Background

2.1 Auto encoder and its variant

Auto-encoder. Auto-encoders are an unsupervised learning technique. Specifically, a bottleneck is imposed in the neural network architecture, which plays a significant role in compressing feature representations of the input. If all of the input's features are independent of each other, this will cause difficulties in compression and reconstruction tasks. Otherwise, if there are some interactive, mutually supportive features in the data, the auto-encoder model will learn this relational structure through bottleneck layers. However, conventional auto-encoder models are only suitable for reconstruction tasks because they only know how to compress the data based on features discovered from data in training these models. When taking an unlabeled dataset and use it into a supervised learning problem with input x then reconstruction an outputting $\hat{x}$. This model can be trained by computing the differences between the input and the output reconstruction then minimize it.

U-net. U-Net is a convolutional neural network architecture (CNN) that expanded with few changes in architecture. Olag Ronneberger et al. [43] introduced the U-Net architecture for biomedical image segmentation, and the role is not only to classify whether there is an infection or not but also to detect wherein the area of infection. The encoder and decoder are two main parts of U-net architecture. The encoder is used to extract the features in the image by convolutional layers followed by pooling operations. The second part decoder consists of up convolution, concatenation, regular convolution operations.

Skip-connection. In many convolutional architectures, skip connection is a widely-used technique because it affects convergence, fixes vanishing gradients, and lever-

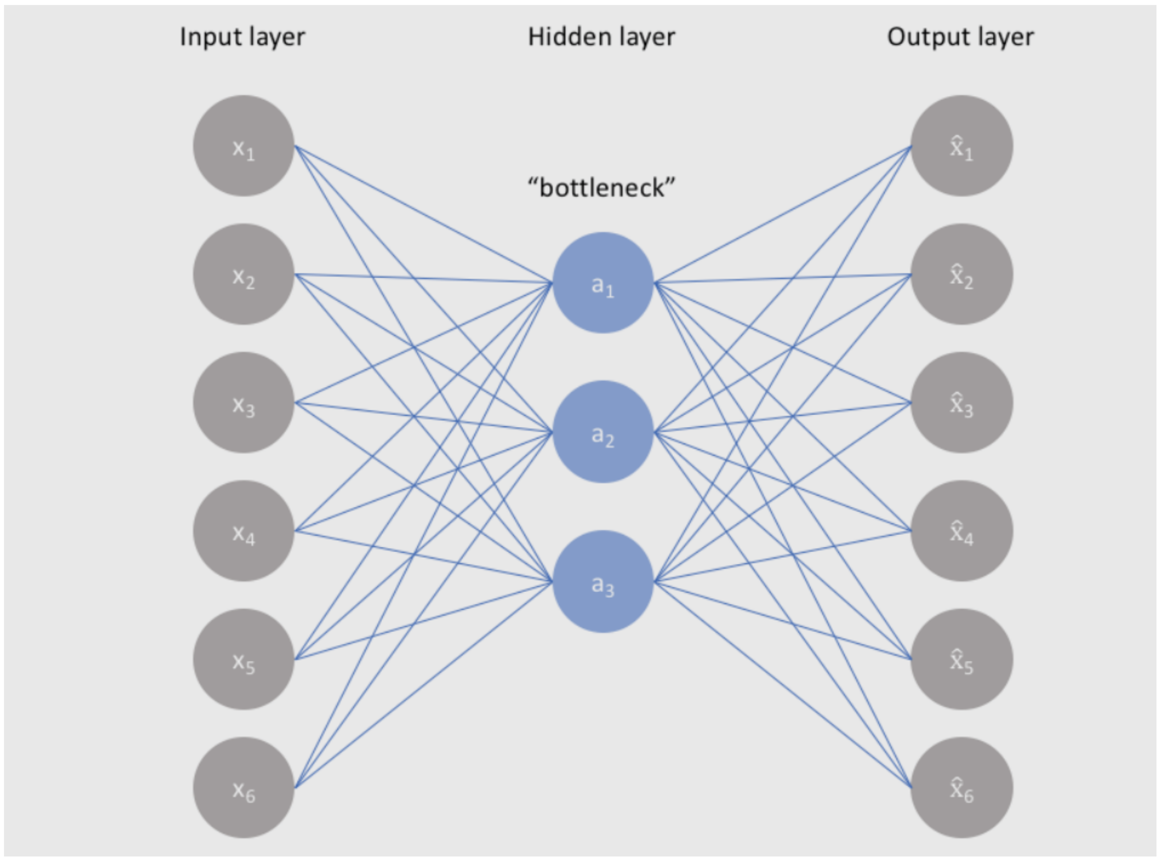

Figure 2.1: Auto-encoder architecture.

ages data from previous layers. The skip connection works by creating paths between layer after layer; the previous layer's output would become the input of the following layers. Note that the skip connection does not create a path of two adjacent layers. For a long structured model, in the backward stage, multiplying too many results less than one together often results in a gradient of approximately zero. Therefore, the update will be suspended in the early layers. In this state, a skip connection path can yield significant improvements. There are two basic ways to use skip connections:

- Addition skip connection as in ResNet [2].
- Concatenation skip connection as in DenseNet [3].

As mentioned above, the skip connection can help the model to reuse the data of earlier layers. This feature is of concatenate skip connection, so concatenation is

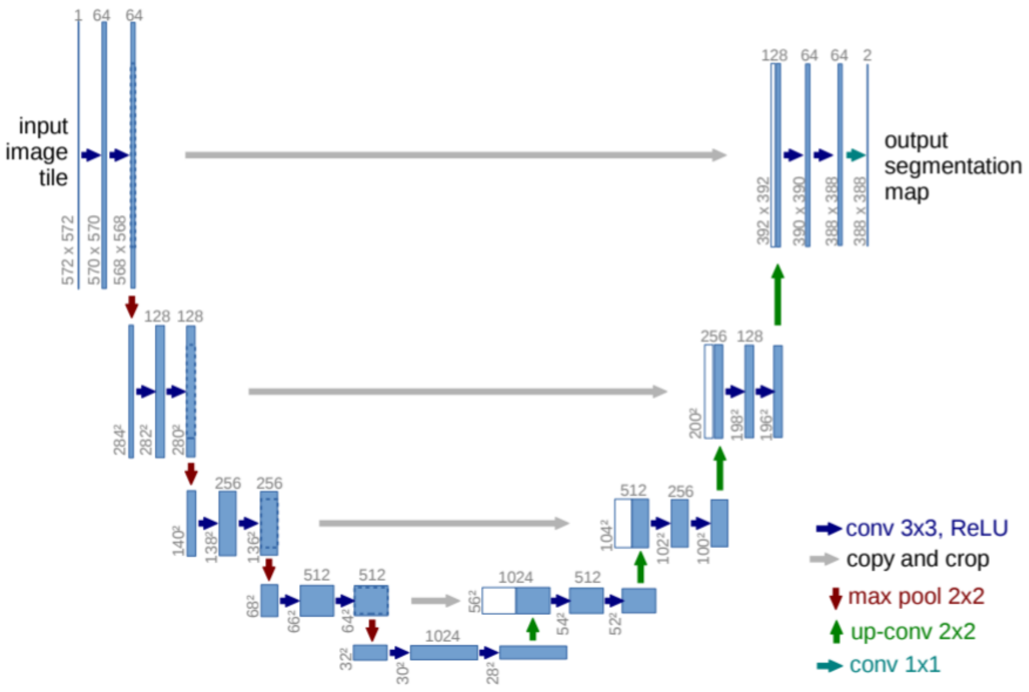

Figure 2.2: U-net architecture. Image from [1]

used quite commonly. Note that the addition skip connection has to have the same dimension and concatenation also. With concatenation, there can be a difference in the number of channels. In addition, the skip connection has two types of setups:

- Short skip connections.
- Long skip connections.

The short skip connections are used for consecutive layers and have whole input dimensions (ResNet figure 2.3). Its main effect is to stabilize gradient training and convergence. Long skip connection is mainly used in symmetrical architecture; a good example is U-net (U-net figure 2.2). It is used to transport the features extracted from the encoder to the decoder to recover spatial information lost during downsampling. Long skip connection is utilized for tasks that have the same spatial dimension as the input, such as image segmentation, optical flow estimation, video prediction, etc.

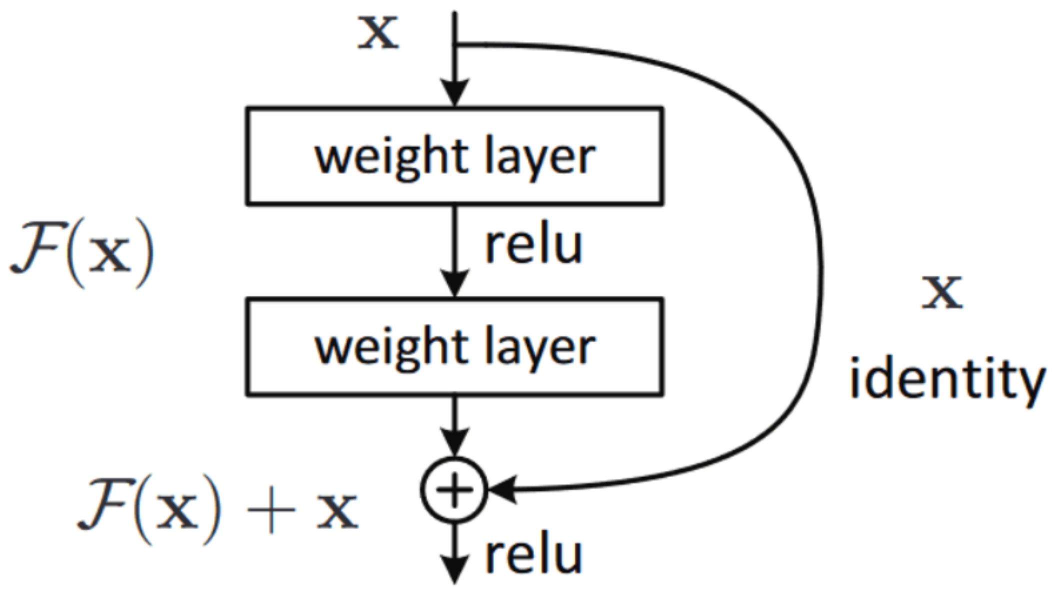
x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.3: This is example for addition and short connection. Image from [2]

2.2 Deep feature learning for anomaly detection

Image reconstruction. Image reconstruction or image restoration is essential for problems applied in practice that use images as an input. The corrupted images such as noise, motion blur, and low resolution will recover the clean images. Image reconstruction makes images obtained from the actual camera-less affected by the surrounding environment, weather, etc., improving the model's accuracy.
Image prediction. Image prediction is a method of predicting the future based on deep learning. With the input being a series of consecutive frames, feature representations are extracted then reconstruct a future frame from the features. To predict a frame with high quality, appearance features combined with motion features. Motion constraints obtained by performing the optical flow between future predicted frames and ground truth frames to be consistent [15].
Optical flow. Optical flow can be seen as the motion attribute of objects in a series of consecutive frames. It computes the motion vector of each pixel in the video frame. Most optical flow methods consider each pixel in a video frame to be an object, and the color/intensity of each pixel is the feature to identifying an object in a series of consecutive frames. Assuming that the color/intensity of each pixel is

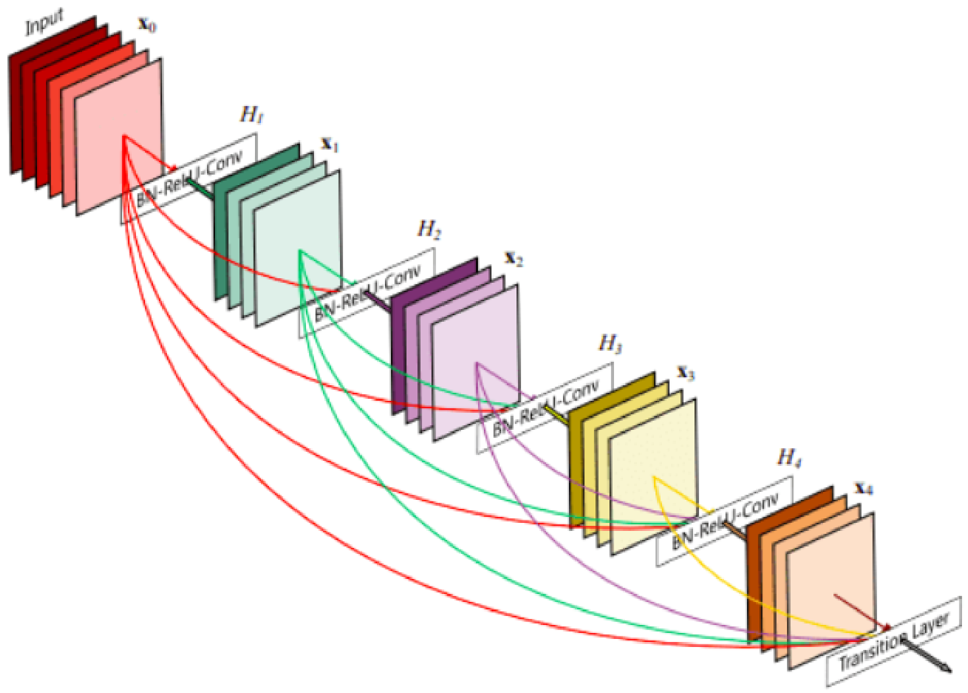

Figure 2.4: Example for long skip connection and feature reusability by concatenation of 5 layers. Image from [3]

constant across video frame s, optical flow methods estimate the motion of a pixel by monitoring the movement of its color/intensity over time in a video. The flow fields are then analyzed to create segments into regions, which can be linked to moving objects. Optical flow has two main variants:

- Sparse optical flow computes the vectors of several pixels represent the edges or corners of an object in the video frame.
- Dense optical flow computes the vectors of all pixels in the video frame.

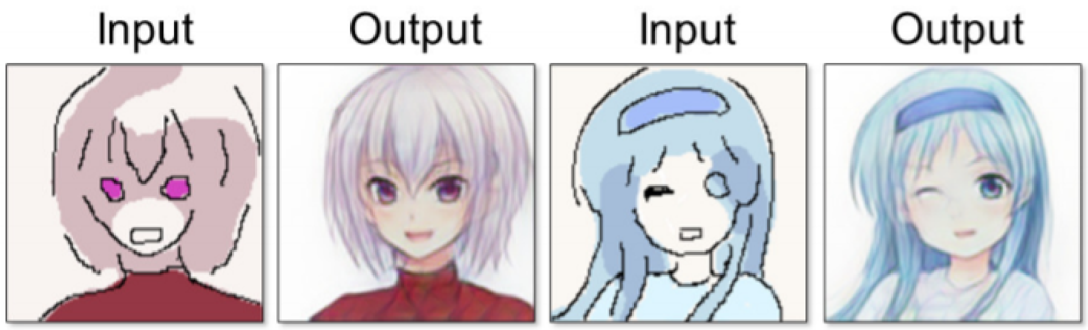

Figure 2.5: Example for reconstruction. Image from [4]

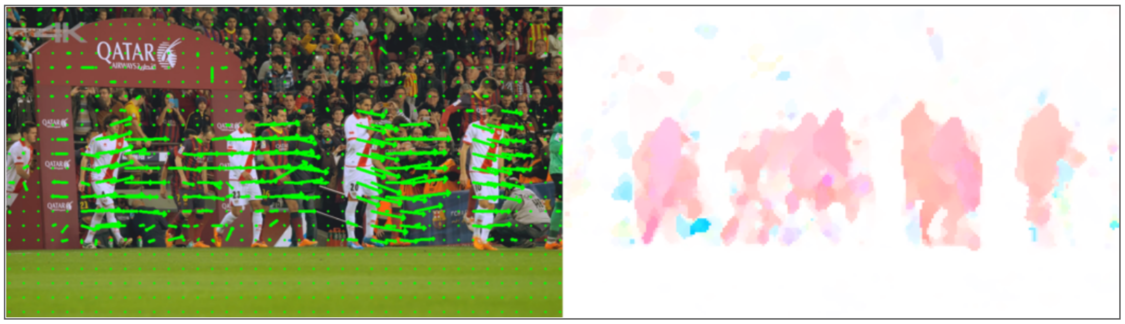

Figure 2.6: Example for sparse (left) and dense (right) optical flows.

Chapter 3

Proposed Method

3.1 Overview

Appearance (spatial) and motion (temporal) constraints are both important factors in the which discussed in Section 3.2. In related studies about video frame generation [44, 45], it has been shown that using individual appearance constraints for predicting or reconstructing cannot guarantee extract motion features. Therefore, we propose adding an optical flow extractor branch to ensure prediction efficiency besides the frame predicted branch. Indeed, several articles have applied this approach in various fields like future frame prediction [15] and action recognition [46, 47], combining both temporal and spatial stream significant accuracy improvements [22, 24, 46, 47].

The memory module is a type of architecture that attracted much attention by the researcher and be applied to video abnormally detection in recent years. It contains the two most common functionality, which is read and updates operations. Moreover, that memory module in the model treats actual physical memory, which keeps normal data representation integrity during training and testing stages. That is why the memory modules and their properties to store both normal motion and appearance representations are included in our model. This section is carefully discussed in Section 3.3.

Our model design is based on memory architecture [16] and appearance (spatial) constraints combined with motion (temporal) constraints [15, 24]. Furthermore, according to experiments about enhanced anomaly events detection, they demonstrated that motion features integrate with frames prediction and learn to describe normal events better, so by inheriting that property, we introduce a method that

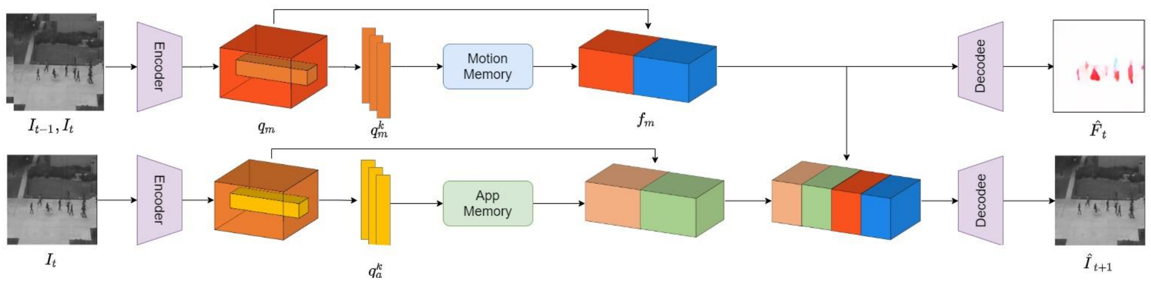

Figure 3.1: The proposal motion-appearance co-memory network for predict future video frame and extract optical flow. Our model consists of two branches motion (upper branch) and appearance (lower branch) with each branch is composed of 3 main components: encoder, memory, and decoder modules.

uses both frame reconstruction and optical flow extraction to take advantage of property from both appearance features and motion features.

Our model structure consists of 2 branches motion and appearance branch and each branch has three main components including encoder, memory, decoder modules. The motion branch extracts optical features from two frames (I_{t-1}, I_{t-2}), which are then used for reading, updating normal patterns in memory. We synthesized optical features with normal patterns read from memory and inserted them into the decoder module to achieve optical flow ($\hat{F}_t$).

Our appearance branch is similar to our motion branch about encoder, decoder, memory architecture, forward and backward propagation, training loss. The differences are that the input of the app-encoder is a frame (I_t); the features synthesized from the motion-branch and the app-branch will be concatenated to increase information to the decoder for prediction of the input video frame ($\hat{I}_{t+1}$).

3.2 Appearance and Motion branch

According to reference [16], our encoder and decoder architecture is exploited from U-Net architecture [48], which is widely used to extract feature representations from input frames and reconstruct the frames from feature representations. Since ReLU layers [10] do not accept negative values, which limit the diversity of feature representations, we removed them together with the last batch normalization[49] in both encoder modules. Instead, we use an L2 normalization layer, so that feature representations have a common scale. Our model mainly focuses on prediction instead of reconstruction, so we keep skip-connections to utilize their capacity.

3.2.1 Motion branch

Optical flow estimates the pixel-wise motion velocity and direction between two consecutive frames and is a popular low-level motion representation in videos. The utilization of optical flow for motion modeling has appeared extensively in most two-stream-based convolutional networks [46, 47]. Although optical flow estimation is an expensive computational cost and makes the model have a larger size, motion representations are very useful. Inspired by [15, 46, 47], we build a motion branch using optical flow and inheritance memory.

The motion-encoder converts the consecutive frames I_{t-1} ,I_t to motion representations. Motion representations are denoted as feature query map q_m of size $Height \times Weight \times Channels$. The query map q_m^k contains queries of size $1 \times 1 \times C$ where $q_m^k \in R^C (k = 1, ...K)$ with $K = Height$ x $Weight$. Queries are then input into motion memory modules $M_{motion}(.)$ to update normal patterns during the training stage or read normal patterns in memories.

$$p_m^k = M_{motion}(q_m^k) \tag{3.1}$$

Applying the advantage of the skip connection in using the concatenate operation to fuse the feature query map q_m with obtained normal motion feature $\hat{p}_m$ to enhance the knowledge of featuring both compositional patterns and normally a pattern which the encoder obtained by extracting from pair of frames:

$$f_m = Concat(q_m, \hat{p}_m) \tag{3.2}$$

The motion-decoder D_{motion} inputs the feature f_m and gives an optical flow image F_t.

$$\hat{F}_t = D_{motion}(f_m) \tag{3.3}$$

3.2.2 Appearance branch

As we mentioned in section 3.2.1, appearance encoder and decoder inherit architecture from the motion branch. The process of extracting feature representations and reading normal patterns is similar to the branch it inherits.

$$f_a = Concat(q_a, \hat{p}_a) \tag{3.4}$$

where q_a is appearance feature query map and $\hat{p}_a$ is normal appearance features.

The differences between the two branches encoder are that the app-encoder converts video frames and gives appearance representations. Appearance representations are denoted as query map q_a of size $H \times W \times C$. To complete the information required for prediction, motion features are exploited from motion memory f_m is combined with f_a:

$$f = Concat(f_a, f_m) \tag{3.5}$$

For predicting the future frame $\hat{I}_{t+1}$ our app-decoder using the feature f as the input, and besides the predicted optical flow the predicted frames both use to compute the anomaly score (The detailed descriptions of the anomaly score are presented in the section 3.5)

3.3 Co-memory module

The reading and updating stage is presented in Figure 3.2, then we will explain some math formulas. We denote normal pattern in the memory by $p_n \in R^C (m = 1, ...N)$. Where N is the number of normal patterns recorded in the memory module.

3.3.1 Read

To read the normal patterns, there are two main steps. Step 1, we calculate the matching probabilities w_n^k. First, we compute the cosine similarity between each feature query q^k and all normal patterns p_n in the memory. Cosine similarity yield a two-dimensional correlation map with N x K size for all feature query (N x 1 for each feature query). The matching probabilities w_n^k obtained through the softmax function.

$$w_n^k = \frac{\exp((p_n)^T q^k)}{\sum_{\acute{n}=1}^{N} \exp((p_{\acute{n}})^T p^k)} \tag{3.6}$$

Step 2, we compute the feature $\hat{p}^k \in R^C$ for each feature query q^k through a weighted average of the matching probabilities w_n^k and the corresponding normal

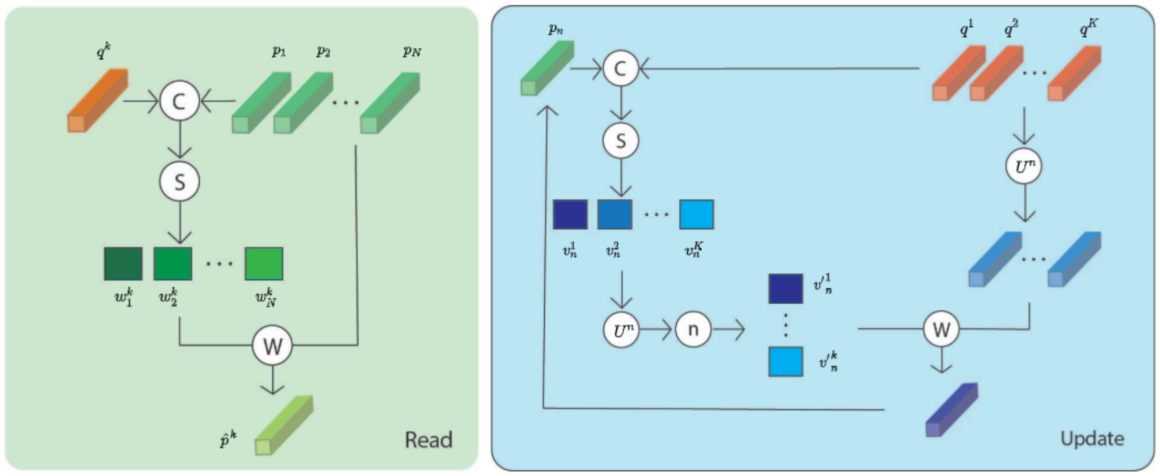

Figure 3.2: The architecture of the memory module. Due to both motion and appearance memories share the same architecture, so the subscripted symbols of appearance a and motion m will not be displayed. Reading stage is executed by calculating matching probabilities w_n^k between feature query q_k and normal patterns in memory $(p1, p2, ...pN)$. We then compute a weighted average of the probabilities with the normal patterns to achieve $\hat{p}^k$. In the updating state, we are obtained matching probabilities v_n^k be computed the similarity between feature queries $(q1, ...qK)$ and normal pattern in memory p_n. The average weighted will be calculated based on the feature queries in the set U^n along with the matching probabilities, the end of the process will be a new normal pattern which is initialized and added to p_n. **C**: cosine similarities; **S**: softmax function; **W**: weighted average; **n**: max normalization; $\mathbf{U^n}$: a set of indices for the n-th memory item.

patterns items in memory p_n.

$$\hat{p}^k = \sum_{\acute{n}=1}^{N} w_n^k P_{\acute{n}} \tag{3.7}$$

After using reading operation for feature query map qm or qa, we obtain corresponding $\hat{P} \in R^{HxWxC}$.Then we concatenate $\hat{p}$ to its feature query map by C dimension and output (Equations 3.2, 3.3).

3.3.2 Update

For updating normal patterns in memory, we first also use cosine similarities and softmax functions to achieve matching probabilities v_n^k as Equation 3.1.

$$v_n^k = \frac{\exp((p_n)^T q^k)}{\sum_{\acute{k}=1}^{K} \exp((p_n)^T q^{\acute{k})}} \tag{3.8}$$

We have a two-dimensional map v_N^K of size N x K when feature queries compute with all normal patterns in the memory. We choose K's highest matching probability in v_N^K, which means each query will be assigned to a normal pattern in the memory. The indexes of these K's highest matching probability will be recorded in U_n. U_n is then used to filter out queries and their corresponding matching probabilities v_n^k. These $\acute{v}_n^k$ are renormalized through a class of max normalization.

$$\acute{v}_n^k = \frac{v_n^k}{\max_{\acute{k} \in U^n} v_n^{\acute{k}}} \tag{3.9}$$

To the end, memory update normal patterns are based on matching index queries in U^n through $g(.)$ is L2 norm.

$$P_n \longleftarrow g(p_n + \sum_{k \in U^n} \acute{v}_n^k q^k) \tag{3.10}$$

In terms of the structure and algorithm of the co-memory module, we mentioned the above formulas. Pack et al. [16] used this memory structure for abnormal detection, but their model had only one appearance branch for both reconstruction and prediction. With the idea of using appearance and motion in the detect abnormal activities model, we have combined memory modules for two branches. The results in table 2 indicate that combining motion branches with memory has brought about significant improvements.

3.4 Loss functions

We use compactness and separation losses for training memory modules. Compactness and separation define the quality of clustering results. A cluster has good compactness when elements are close to each other and good separation when clusters do not overlap [50]. Optical flow and prediction losses are used to train motion and appearance branches, respectively.

3.4.1 Optical flow loss

The optical flow loss calculates the difference between the predicted optical flow and ground truth by minimizing the $L2$ distance between them as flowing:

$$L_{flow}(\hat{F}_t, I_{t-1}, I_t) = ||\hat{F}_t - f(I_{t-1}, I_t)||_2 \tag{3.11}$$

where $\hat{F}_t$ denote optical flow image of decoder output; $f(.)$ stands for the function to extract optical flow ground truth, and we use the FlowNet2 model for this function, due to practical experiments.

3.4.2 Prediction loss

The prediction loss makes the video frame predicted from the decoder similar to its ground truth. Specifically, we minimize the L2 distance between the decoder output $It + 1$ and the ground truth $It + 1$:

$$L_{pred}(\hat{I}_{t+1}, I_{t+1}) = ||\hat{I}_{t+1} - I_{t+1}||_2 \tag{3.12}$$

where we denote I_{t+1} by the predicted frame in appearance branch.

3.4.3 Compactness loss

When training memory modules with compactness loss, the feature queries will be close to the other normal pattern, which is nearest in the memory.

$$L_{compact} = \sum_{k}^{K} ||q^k - p_i||_2 \tag{3.13}$$

where i is an index of the normal pattern, with the distance is closest to the query qk.

$$i = \arg\max_{n \in N} w_n^k \tag{3.14}$$

3.4.4 Separateness loss

For cluster problems, the spacing between clusters is very important. However, when training memory with compactness loss in (13), normal patterns tend to form a single cluster. To solve this problem and enhance the discriminative competence, we apply separateness loss intending to create a margin α between the query and its second nearest.

$$L_{separate} = \sum_{k}^{K} [||q^k - p_i||_2 - ||q^k - p_j||_2 + \alpha] \tag{3.15}$$

where i, j are index of the normal patterns, with the distance are closest and second closest to the query qk, respectively.

$$j = \arg\max_{n \in N, n \neq i} w_n^k \tag{3.16}$$

3.4.5 Training objective

To train our model, we use optical flow, prediction, compactness, and separateness losses ($Lflow$, $Lpre$, $L_{compact}$ and $L_{separate}$, respectively), where $\lambda_f, \lambda_p, \lambda_{ac}, \lambda_{as}, \lambda_{mc}$ and λ_{ms} act as balance parameters.

$$Loss = \lambda_f L_{flow} + \lambda_p L_{pred} + \lambda_{ac} L_{a-compact} + \lambda_{as} L_{a-separate} + \lambda_{mc} L_{m-compact} + \lambda_{ms} L_{m-separate} \tag{3.17}$$

3.5 Abnormality Score

For enhancement the efficiency of using both predicted optical flow and predicted frames when calculating an abnormality score of each frame, we use the score introduced in [38] instead of the common score employed in the deep learning approach like Lp or Peak Signal to Noise Ratio (PSRN). Indeed, the drawback of the common abnormality score, which calculated the similarity between the ground truth frame and the predicted frame is they attend at entire video frames by summation or average, which causes missed information in small regions. In our case, the ratio of the movement brightness pattern in the optical flow is small due to non-moving objects in the background.

The estimated weighted combination score of motion and appearance branch defines as:

$$S = \log[w_F S_F(\tilde{\mathrm{P}})] + \lambda_S[w_I S_I(\tilde{\mathrm{P}})] \tag{3.18}$$

where the the spatial score $S_F(P), S_I(P)$ of each branch at same position P define as follows:

$$\begin{cases} S_I(P) = \frac{1}{|P|}\sum_{i,j\in P}(I_{i,j} - \hat{I}_{i,j})^2 \\ S_F(P) = \frac{1}{|P|}\sum_{i,j\in P}(F_{i,j} - \hat{F}_{i,j})^2 \end{cases} \tag{3.19}$$

where P is the index of spatial image patch, and $|P|$ is the total number of pixels in an image patch. In our experiment the size of image patch P is 16x16, and for particular solutions we adopt convolutional operation with filter size 16x16 for computed partial score of two branches. The inverted score of appearance and motion branch which obtained from n images in training set can computed as:

$$\begin{cases} w_F = [\frac{1}{n}\sum_{i=1}^{n} S_{F_i}(\tilde{\mathrm{P}}_{\mathrm{i}})]^{-1} \\ w_I = [\frac{1}{n}\sum_{i=1}^{n} S_{I_i}(\tilde{\mathrm{P}}_{\mathrm{i}})]^{-1} \end{cases} \tag{3.20}$$

Moreover, the index P̃ can obtained by search the maximum value partial score of examine frames $\tilde{P} \longleftarrow \underset{P\ slide\ on\ frame}{\arg\max} S_F(P)$. Finally, the abnormal score of each frame t in a video with m frames can be computed as:

$$\hat{S}_t = \frac{S_t}{\max(S1..m)} \tag{3.21}$$

Chapter 4

Experimental Results

4.1 Implementation details

4.1.1 Dataset

For evaluation the effectiveness of our model, we use the two most common benchmark dataset: (1) UCSD Ped2 dataset [5] contains 4560 frames which separated into 2550 use for training and 2010 using for testing, and the rare event is riding a bike and driving the vehicle (Table 4.1); (2) CUHK Avenue dataset [6] consists of 30652 frames that split into 16 clips for training and 21 abnormally event clips for testing. The irregular action contains 47 abnormal events such as anomaly actions (e.g., running, throwing), wrong moving direction, anomaly object (e.g., bicycle).

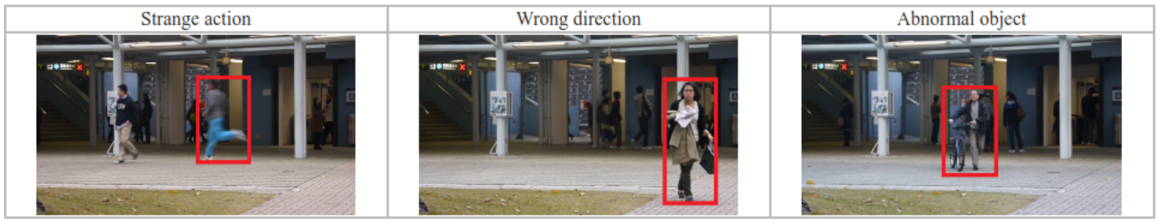

Figure 4.1: Example of anomaly events in Avenue dataset.

Table 4.1: Composition of UCSD Ped2 Anomaly Dataset. *a* number of clips / number of anomaly instances. *b* some clips contain more than one type of anomaly.

Scene	**Nor**	**Abnormal** $_a$			
		Bike	Skater	Vehicle	Total $_b$
Ped2	16	11/19	3/3	1/1	12/23

4.1.2 Training

During the training process, input frames are resized into a size of 256x256, which normalize into the range [-1, 1], and both dimensions of appearance and motion feature dimensions equal 512. The size of memory size for UCSD Ped 2, CUHK Avenue are 10 and 15 for both appearance and motion branch, respectively. We train using Adam optimizer[51] where learning rate, momentum were set to 10^{-4}, 0.9, and we set weight decay to 10^{-4}, 0.1 for CUHK Avenue and UCSD Ped 2. For the optical flow extractor functions f(.), we use the FlowNet2 [52] model, which pre-trained in FlyingThing3D [53] and ChairsSDHom [52] dataset for estimated ground truth. The parameter of λ_f, λ_p, λ_{ac}, λ_{as}, λ_{mc} and λ_{ms} set to 1, 1, 0.1, 0.1, 0.1, 0.1 for both datasets .Our model is implemented in Pytorch [54] and trained in Google Colab.

4.1.3 Evaluation methodology

The frame-level criterion is an algorithm used to evaluate anomaly detection accuracy by predicting frames containing abnormal events. It is based on true positive rates (TPRs) and false-positive rates (FPRs), denoting "an anomalous event" as "positive" and "the absence of anomalous events" as "negative." A-frame containing anomalies is denoted a positive, otherwise a negative. The true and false positive under the frame-level criterion can be determined by comparing the clip's frame-level ground-truth anomaly annotations. Moreover, the True Positive Rate (TPR) and the False Positive Rate (FPR) are two parameters used to draw the Receiver Operating Characteristic (ROC) curve, which is the curve of True Positive Rate (TPR) versus False Positive Rate (FPR), generated by varying an acceptance threshold. TPR and FPR where True/False Positive/Negative are counts for the corresponding class are defined as:

$$TPR = \frac{\# of\ truepositive\ frame}{\# of\ positive\ frame}$$

$$FPR = \frac{\# of\ falsepositive\ frame}{\# of\ negative\ frame}$$

4.2 Results

In this section, we compare our method with different methods for anomaly detection on the UCSD Ped2 dataset [5] and CUHK Avenue dataset [6]. From table 1, we can see that our method results in AUC outperforming all of the other methods and even when compared to models using memory methods like MemAE [23], MNAD [16], AMCM [22]. When comparing our model with motion memory to base model - MNAD [16], the performance of our model increased by 2.7% for the Ped2 dataset and 0.9% for the Avenue dataset. Also, compared to the latest model of appearance-motion memory [22], the AUC result of our model is still higher at 3.1% on Ped2 and 2.8% on Avenue. In the model without (w/o) motion memory, the model also shows an improvement by 0.5% compared to the base model [16] on the Avenue dataset.

Table 4.2: The measurement of AUC results with different methods on the UCSD Ped2 [5] and CUHK Avenue [6] datasets.

Method	Ped2 dataset	Avenue dataset
AMCorrespondence [24]	0.962	0.869
Frame Prediction [15]	0.954	0.851
TSC [31]	0.910	0.806
Stacked RNN [31]	0.922	0.817
ConvLSTM-AE [55]	0.881	0.770
Abnormal GAN [14]	0.935	-
Any-Shot [56]	0.978	0.864
CDAE [57]	0.965	0.860
AMCM [22]	0.966	0.866
MemAE [23]	0.941	0.833
CAC [58]	-	0.870
MNAD [16]	0.970	0.885
Our Method w/o motion memory	0.970	0.890
Our Method w motion memory	**0.997**	**0.894**

We have performed our model experiments with different memory sizes to examine the effect of memory size on the model. From the results obtained from table 2, testing on the UCSD Ped2 dataset [5] shows that a larger memory size does not mean better results are obtained. On the contrary, with a large memory size, the AUC is significantly reduced (1.6% with the Ped2 dataset) compared to app-motion

Table 4.3: The AUC results of our methods on Ped2 and Avenue datasets with different sizes of memory module.

Dataset	M-memory size	A-memory size	AUC
Ped2	0	10	0.970
	10	10	**0.997**
	15	15	0.981
Avenue	0	10	0.890
	10	10	0.891
	15	15	**0.894**

memory size 10. However, in the trial with the CUHK Avenue dataset [6], it gives the best result (0.894) at the app-motion memory size 15. The above comparisons indicate that memory size depends on the number of normal objects in each different dataset. Finally, the results obtained from table 2 show that our method of applying memory to the motion branch brought about positive improvements.

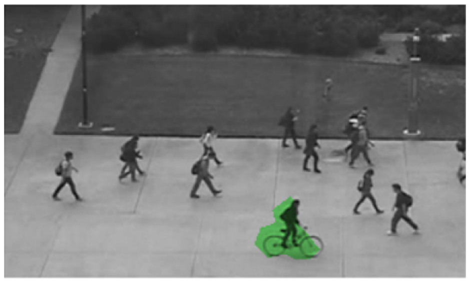

Figure 4.2: Biker

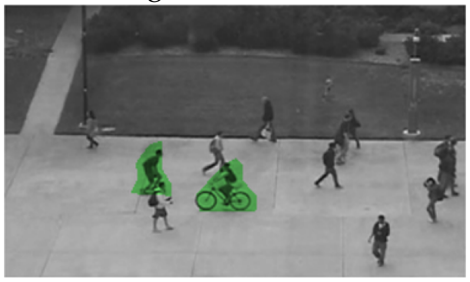

Figure 4.3: Skater and Biker

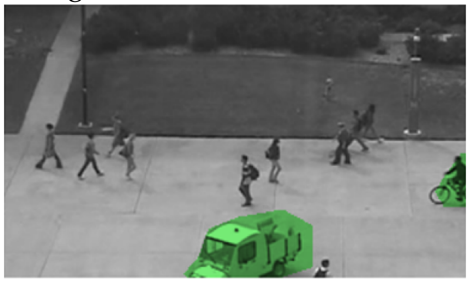

Figure 4.4: Vehicle

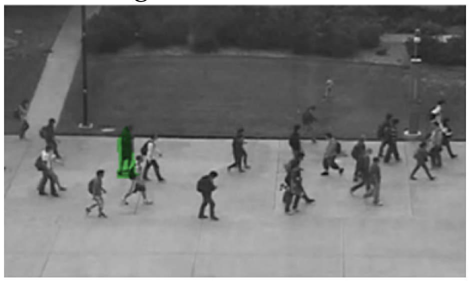

Figure 4.5: Skater

Figure 4.6: Example of anomaly events in Ped2 dataset.

Chapter 5

Conclusion and Future Work

5.1 Future Works

Based on the proposal method given in Chapter 3, some limited is indicated:

- Hyper-parameters for deep models such as kernel size, number of filters, number of layers in our encoder-decoder architectures were chosen based on the expected use in the state of the art architectures classification tasks and anomaly detection tasks without deep turning for our study.
- As we discussed in Chapter 3, the update operations of the memory module only available during the training process, while we freeze it during the testing process because of the calculated W_I, W_F of training datasets when computing the abnormality score. In practical applications, the lack of reading operation during inference time may ham the systems due to a lack of developed ability when the scenario scales up.
- Since our proposal method using two continuous frames as the inputs for the prediction task, our model cannot capture a longer-time dependence of an anomaly, especially in the motion. As a consequence, our model has limitations which cause false alarm when predicts the complex moving case.
- Our parameter of memory size setups by randomly value without analysis carefully.

Many improvements may be applied to the method proposed by us to solve the limitations stated above. Moreover, we introduce some directions for our future works. First, limiting the number of input frames can make the model design easier, making our model more challenging to deal with long-term dependencies. The solution is to adopt LSTM [32] or Conv3D [59] models to represent long-term motion factors instead of convolution in the encoder of the motion branch. Secondly, integration of the update operation during the testing process is affected by the calculation of W_I, W_F parameters during the update, which can change with each update normal pattern in the memory module. Therefore, it is necessary to design a new abnormality score to reduce the effect of update operation on the coefficients while ensuring the properties of the consideration score. Third, for the improvement of inferences times purpose, the replacement of the optical flow extractor to RGB difference maybe apply in our model to reduce the complexation of our model, which inspired by [57].

5.2 Close Remark

This thesis has introduced frameworks where deep convolutional networks learn Spatio-temporal dynamics on optical flow fields and predict a future frame with the additional memory module. Our approach has stepped toward archiving the exploiting motion and appearance feature based on quantitative and qualitative results while extract normal-activation features more efficiently. Moreover, our proposal approach archives state-of-the-art on two benchmark datasets. In conclusion, we hope our proposal method would substantially improve where a normal manifold is described effectively to detect anomaly events advantageously.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[3] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2016.

[4] Sheng You, Ning You, and Minxue Pan. Pi-rec: Progressive image reconstruction network with edge and color domain, 2019.

[5] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):18–32, 2013.

[6] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[8] Muchao Ye, Xiaojiang Peng, Weihao Gan, Wei Wu, and Yu Qiao. Anopcn: Video anomaly detection via deep predictive coding network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1805–1813, 2019.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *2009 IEEE conference on computer vision and pattern recognition*, pages 2921–2928. IEEE, 2009.

[13] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR 2011*, pages 3449–3456. IEEE, 2011.

[14] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1577–1581. IEEE, 2017.

[15] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6536–6545, 2018.

[16] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14372–14381, 2020.

[17] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016.

[20] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

[21] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941, 2017.

[22] Ruichu Cai, Hao Zhang, Wen Liu, Shenghua Gao, and Zhifeng Hao. Appearance-motion memory consistency network for video anomaly detection. 2021.

[23] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019.

[24] Trong-Nguyen Nguyen and Jean Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1273–1283, 2019.

[25] Liang Xiong, Barnabás Póczos, and Jeff Schneider. Group anomaly detection using flexible genre models. 2011.

[26] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, pages 1100–1109. PMLR, 2016.

[27] Namrata Vaswani, Amit K Roy-Chowdhury, and Rama Chellappa. " shape activity": a continuous-state hmm for moving/deforming shapes with application to abnormal activity detection. *IEEE Transactions on Image Processing*, 14(10):1603–1616, 2005.

[28] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. Video anomaly detection and localization using hierarchical feature representation and gaussian

process regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2909–2917, 2015.

[29] Junshui Ma and Simon Perkins. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1741–1745. IEEE, 2003.

[30] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[31] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 341–349, 2017.

[32] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[33] Youngjin Kim, Minjung Kim, and Gunhee Kim. Memorization precedes generation: Learning unsupervised gans with memory networks. *arXiv preprint arXiv:1803.01500*, 2018.

[34] Chongxuan Li, Jun Zhu, and Bo Zhang. Learning to generate with memory. In *International Conference on Machine Learning*, pages 1177–1186. PMLR, 2016.

[35] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*, 2015.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[37] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016.

[38] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR, 2016.

[39] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

[40] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.

[41] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Video multitask transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[42] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019.

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. volume 9351, pages 234–241, 10 2015.

[44] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[45] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[46] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

[47] Zhigang Tu, Wei Xie, Qianqing Qin, Ronald Poppe, Remco C Veltkamp, Baoxin Li, and Junsong Yuan. Multi-stream cnn: Learning representations based on human-related regions for action recognition. *Pattern Recognition*, 79:32–43, 2018.

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[49] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[50] 3 - building energy demand modeling: from individual buildings to urban scale. In Ursula Eicker, editor, *Urban Energy Systems for Low-Carbon Cities*, pages 79–136. Academic Press, 2019.

[51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[52] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[53] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[55] Weixin Luo, Wen Liu, and Shenghua Gao. Remembering history with convolutional lstm for anomaly detection. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 439–444. IEEE, 2017.

[56] Keval Doshi and Yasin Yilmaz. Any-shot sequential anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 934–935, 2020.

[57] Yunpeng Chang, Zhigang Tu, Wei Xie, and Junsong Yuan. Clustering driven deep autoencoder for video anomaly detection. In *European Conference on Computer Vision*, pages 329–345. Springer, 2020.

[58] Ziming Wang, Yuexian Zou, and Zeming Zhang. Cluster attention contrast for video anomaly detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 2463–2471, New York, NY, USA, 2020. Association for Computing Machinery.

[59] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

Appearance-Motion Co-memory Network for Video Anomaly Detection

Final Year Project Report

A 4th Year Student Name

Le Duc Anh, Nguyen Ba Duong
HE130082, HE130658

Under the supervision of

Dr. Phan Duy Hung

Bachelor of Computer Science

Hoa Lac Campus – FPT University

Spring 2021