공학석사학위논문

아동 학대 감지를 위한 자가 증류 접근법 기반 다중 인스턴스 학습

Multiple Instance Learning for Child Abuse Detection based on Self-distillation Approach

충 북 대 학 교 대 학 원

전기·전자·정보·컴퓨터학부 컴퓨터과학전공

Chinbold Gansukh

2021 년 8 월

공학석사학위논문

아동 학대 감지를 위한 자가 증류 접근법 기반 다중 인스턴스 학습

Multiple Instance Learning for Child Abuse Detection based on Self-distillation Approach

지도교수 Aziz Nasridinov

전기・전자・정보・컴퓨터학부 컴퓨터과학전공

Chinbold Gansukh

이 논문을 공학석사학위 논문으로 제출함.

2021 년 8 월

본 논문을 Chinbold Gansukh의 공학석사학위 논문으로
인정함.

심 사 위 원 장 류 관 희 (인)

심 사 위 원 최 상 현 (인)

심 사 위 원 Aziz Nasridinov (인)

충 북 대 학 교 대 학 원

2021 년 8 월

Multiple Instance Learning for Child Abuse Detection based on Self-distillation Approach

Chinbold Gansukh

Accepted in partial fulfillment of the requirements for the degree of Master of Computer Science

Jul 8, 2021

Thesis Advisor _____
Prof. Aziz Nasridinov

Committee member _____
Prof. Kwan-Hee Yoo

Committee member _____
Prof. Sang-Hyun Choi

Table of Contents

Multiple Instance Learning for Child Abuse Detection based on Self-distillation Approach*

Chinbold Gansukh

School of Electrical Engineering and Computer Science,
Graduate School of Chungbuk National University,
Cheongju, Korea
Supervised by Professor Aziz, Nasridinov

Abstract

We refer to child abuse as physical abuse or violence at a child. In South Korea, child abuse is increasing year by year. In this paper, we propose to learn child abuse by using normal and abnormal weakly labeled videos based on multiple instance learning (MIL). We propose to learn abnormal through the self-distillation approach by assisting the mirrored transformation of the video dataset. In our approach, a deep neural network learns from distilled knowledge between a video and its mirrored transformation and generalizes the abnormal and normal video segments. Furthermore, we introduce the self-distillation loss function assisted by MIL ranking loss of the previous state-of-the-art method. We also introduce a new dataset for child abuse detection (CAD). It contains videos with physical violence directed at a child and videos without any violence as normal. Our experiments show that our self-distillation approach achieves CAD performance as compared with the previous state-of-the-art methods.

Keywords: Video anomaly detection, Child abuse detection, Multiple instance learning, Self-distillation, Knowledge distillation

---

* A thesis for the degree of Master in August 2021.

List of Figures

List of Tables

List of Algorithms

Table of Abbreviations

MIL	Multiple Instance Learning
KL	Kullback Leibler
VAD	Video anomaly detection
CAD	CAD
AUC	Area Under Curve
ROC	Receiver Operating Characteristic
BCE	Binary Cross Entropy
MSE	Mean Squared Error
CNN	Convolutional Neural Network
C3D	Convolutional 3D
I3D	Inflated 3D ConvNet

Table of Notations

D	Set of videos (dataset)
V_i.	The video with identifier i
$\widehat{V_i}$	The mirror transformation of the video with identifier i
V_a	The abnormal video
V_n	The normal video
F	The extracted features
K	The number of feature segments
s_n	The set of anomaly score of the normal video
s_a	The set of anomaly score of the abnormal video
$\hat{s_i}$	The anomaly score of the mirror transformation of the video with identifier i

Ⅰ. Introduction

1.1 Background

Korean law defines child abuse as a means that an adult, including a guardian, commits physical violence that may harm the health, welfare or impair the normal development of a person under the age of 18. The number of child abuse cases is increasing year by year, especially in South Korea. As shown in Figure 1.1, in 2019, the number of child abuse cases increased six times that of 2013. It may indicate people are becoming more aware of child abuse, and their responsibility for reporting increased. However, this is hazardous that the number of child abuse cases is likely to increase in further years.

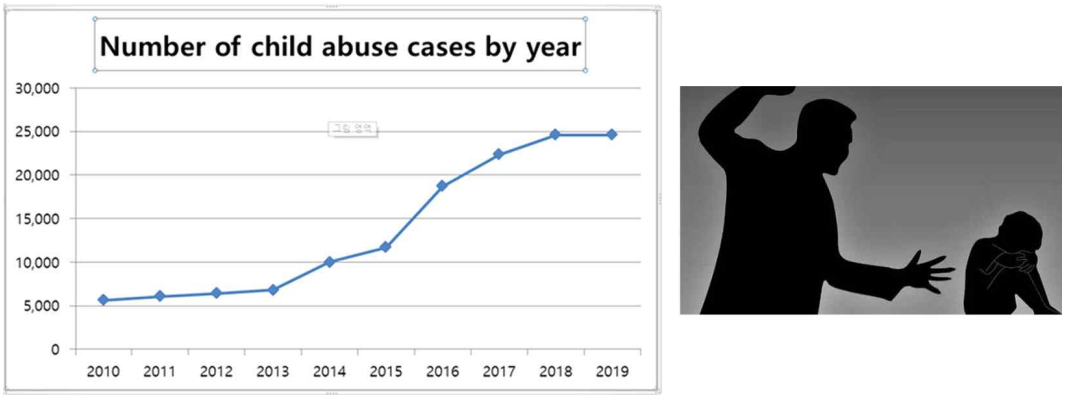


Figure 1.1 Child abuse statistics of South Korea [35]

Developing a robust deep learning network for automatic child abuse detection (CAD) is a pressing need. We consider CAD as a regression problem. Abnormal scenes are indicated by high numbers, but normal scenes are

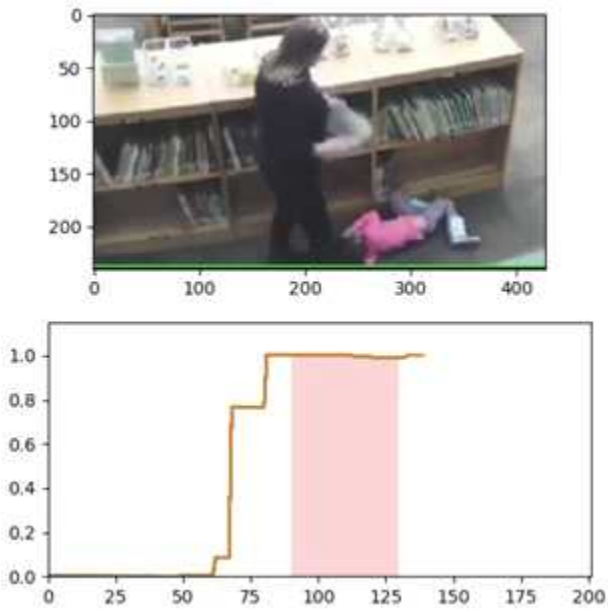


Figure 1.2 Video anomaly detection

indicated by low number. As shown in Figure 1.2, CAD is designed as detecting abnormal scores for a time window of a video. Normal scenes tend to the minimum score in the time window, and abnormal scenes tend to the maximum score. We also consider CAD as an anomaly detection problem. However, anomaly detection focuses on wider than CAD, such as crime, robbery, fighting, car accident, etc. Anomalous actions rarely occur in the real world and, it is the same as child abuse which also occurs rarely.

1.2 Motivation

Video anomaly detection (VAD) has been studied by many researchers. Sultani et al. [1] proposed the first large-scale anomaly detection dataset, namely UCF-Crime, which contains illegal actions, e.g., burglary, robbery, abuse, traffic

accident and others, in videos. The weakly labeled dataset does not have annotations like segment-level, frame-level label, or bounding box. There is only a video level label given for the dataset, which describes if the video of a dataset is anomaly or normal. Basically, anomalous video is likely to contain normal frames in its most segments. The authors adopted multiple instance learning (MIL) for detecting the weakly labeled video by splitting a video into segments, called instances. To extract the features of video segments, the authors utilized a deep-learned feature extractor [33]. It is worth noting that C3D [33] is a neural network model for video recognition that has been driven mainly by expanding 2-d image architectures into spacetime. The middle layers of the network return a useful video representation feature, i.e., the layer named fc7 for C3D and mix_5c for I3D [34]. Then, the extracted features are fed into a deep network to generate scores. The predicted anomaly scores are range between 0 to 1. To train the network model, the authors proposed sparsity and smooth constraint loss functions.

In this thesis, we discover ensemble learning for better performance of CAD in videos. One way to improve the performance of machine learning is the ensemble method. However, the ensemble method is computationally too expensive for a deep neural network to train many models and get an average of prediction. Model compression [2] has been proposed to compress an ensemble of models into a single model. Hinton et al. [28] proposed a model distillation which is an extended version of model compression. Still, its purpose is not only for compressing but also improves the accuracy of the model. Model distillation

transfers knowledge from a complex model to a simple model. The complex model, which is called the teacher model, is large and pretrained. Firstly, the complicated teacher network is trained, and then the smaller student network is trained to mimic the teacher network. Due to optimization methods, the student network can match or even outperform the teacher network [3].

Zang et al. [3] proposed online distillation in which teacher model and student model are from the same network and share knowledge in runtime. In online distillation, training begins with several untrained student models. The performance of each student is increased by following hard label loss. However, each student has different initialization and learns different representations. In training, the soft predictions of each student are matched. This matching procedure helps for better generalization of testing data. Xu et al. [27] proposed a third type of distillation, called self-distillation, a special case of online distillation. Self-distillation optimizes a single network without assistance with a complex network or duplication of the network. It learns only from single network itself. And, it improves generalization performance and improves the accuracy of the network. It maintains two significant training losses: hard label loss and mimicry loss [3], which measures difference the between soft predictions from data distributions. For details, it uses the Kullback Leibler (KL) divergence to estimate the difference between two soft predictions.

According to training time, offline distillation takes a long time, followed by self-distillation and single model because it must learn teacher model first and

then learn student model. According to a number of parameters, offline distillation has more parameters than self-distillation and single model, where self-distillation and single model have the same number of parameters. According to performance, offline distillation and self-distillation are better than a single model. Offline distillation and self-distillation are almost equal in performance. To utilize the capacity of a single network, we propose self-distillation with mirrored data for CAD. Self-distillation has been proposed for image classification and other areas. It was also proposed for a fully supervised network to increase the potential capacity of a single network. For applying self-distillation to the problem of CAD, we have three main components: feature extraction for a video and its mirrored transformation, soft prediction generator single network, multiple self-distillation losses for mimicking each other with hard loss by assisting MIL ranking. In this thesis, we introduce a new dataset with untrimmed real-world videos for CAD.

1.3 Contribution

The main contributions of this thesis are a self-distillation approach with mirrored data to detect child abuse and a new dataset for CAD. To the best of our knowledge, this is the first work that targeted child abuse. Our results achieved an accomplishment and consistently outperformed the state-of-the-art methods. Our contributions are as follows:

- We propose a self-distillation approach for CAD, which requires a transformation of the dataset, i.e., is mirror transformation of a video. Mirror

transformation allows us to differentiate the same video while keeping whole video actions and without altering the input labels. For managing the structure of the video dataset, MIL is adopted. MIL is a practical approach for training weakly labeled data on time window detection.

- We propose multiple self-distillation loss for training child abuse detector with MIL ranking loss and hard label loss assistance. MIL ranking loss manages abnormal video and normal video relations by pushing negative video and positive video far apart. Multiple self-distillation loss leads to effective training by increasing the consistency of a video and its mirror transformation. And, hard label loss enforces video segments to follow ground truths.
- We propose a new dataset for detecting child abuse which contains long and untrimmed videos with normal and abnormal actions. We have collected this dataset through surfing the internet video platform such as Youtube and DailyMotion with multi-language search. Our dataset videos include scenes of a child as a victim and an adult as an abuser in some scenes. We have also conducted experiments designed to evaluate the self-distillation model by comparing our results with the state-of-the-art methods. Our method outperforms Sultani model [1] and Tian model [5] by 10.4% and 1.6% on CAD dataset.

1.4 Thesis organization

The rest of the thesis is structured as follows. Section 2 analyzes previous work

related to anomaly detection, CAD, and self-distillation techniques. In Section 3, we propose our self-distillation approach with mirrored data transformation based on MIL. Afterward, in Section 4, we introduce our new child physical abuse dataset and its characteristics. We also detail an evaluation and experimental results between our proposed method and the state-of-the-art methods on CAD dataset. Finally, in Section 5, we conclude the thesis and discuss future work.

Ⅱ. Related Work

In this section, we review video VAD techniques and knowledge distillation techniques. In Section 2.1, we demonstrate VAD techniques and how they provide solutions for anomaly detection. Afterward, in Section 2.2, we demonstrate knowledge distillation techniques and how it is developed from model distillation to self-distillation.

2.1 VAD approaches

Sultani et al. [1] firstly proposed MIL based method. They used a deep neural network for learning abnormal and normal segments. They introduced fixed segments under the meaning of MIL. The number of fixed segments is 32, for example. In one segment, if there are more than 16-frames, they find an average of features. The logic for loss function is based on a comparing the largest score of abnormal segments and the largest score of normal segments. The max score in the abnormal video should be higher than the max score in the normal video. For training, they choose an equal number of abnormal and normal videos. The authors proposed temporal smoothness loss because abnormal action occurs gradually. The authors also proposed sparsity loss because abnormal occurs rarely. The authors also proposed a large-scale dataset that contains 1900 untrimmed surveillance videos with 128 hours long and 13 anomalies such as Vandalism, Shoplifting, Stealing, Shooting, Abuse, Robbery, Arrest, Fighting, Arson, Explosion, Assault, Burglary, and Accident. Among the total videos, 950 videos are normal,

and 950 are abnormal videos. The video-level label is available on the training data.

Liu et al. [12] proposed an architecture based on deep residual 3D network and temporal encoding, named T-C3D. Feature extraction is done using the 3D-CNN network. Features from 3D-CNN are aggregated for video-level features. Zhong et al. [25] applied a fully supervised action classifier to extract video features. They utilized a graph convolutional neural network consisting of feature similarity and temporal consistency module for excavating similar characteristic and temporal proximity of anomaly segments. The authors applied fully supervised action classifier for feature extraction and graph convolutional neural network consisting of feature similarity and temporal consistency module for similar characteristic and temporal proximity of anomaly segments. Dubey et al. [16] used a pre-trained 3D Resnet model instead of C3D. Kamoona et al. [10] proposed a temporal encoding network to capture spatio-temporal information of video segments. The authors utilized a temporal convolutional network that encodes and decodes video features based on 1-d convolutional layers. Manuel et al. [8] proposed an iterative learning framework that consists of MIL and a Bayesian classifier. The Bayesian classifier discriminates the video globally as abnormal or normal. The authors also proposed a self-supervised network based on the output of the Bayesian classifier. Lv et al. [11] divides video into segments and extract features using CNN for frame by frame in segments and aggregates them before the network. Zaheer et al. [17] proposed a clustering-based self-reasoning technique that divides segments into two

clusters using K-means. Zaheer et al. [24] proposed dynamic segmenting instead of splitting the video into fixed 32 segments. The proposed clustering loss distance on the video feature by utilizing k-means clustering with two centers. They assumed that the clustering distance of abnormal features should be greater than a normal feature. In MIL network, they multiply output of fully connected layer by its output of softmax activation function. The authors mentioned that their proposed method outperforms the state-of-the-art methods.

Degardin et al. [9] proposed an iterative learning framework by leveraging a Bayesian framework responsible for filtering out the segments. The authors also proposed the Random Forest ensemble method on score level and proposed a fully annotated dataset on frame-level, named UBI-Fight. The authors also proposed a self-supervised learning method that discriminates abnormal and normal segments. Maqsood et al. [23] modified the UCF-Crime dataset and applied spatial augmentation to provide frame-level annotation on the training set. This semi-supervised learning has no bounding box or pixel-level label for a frame. The authors utilized spatial augmented features using 3D ConvNets [33] which is fine tuned and trained on activity-based datasets. The augmentation is done by horizontal and vertical flipping. The videos were prepared from horizontally and vertically augmented frames. Tian et al. [5] proposed a method that enables top-k MIL approaches for weakly supervised VAD. They utilized context aggregation for gathering local information and non-local network for gathering global information of the video feature. The new feature is a result of the concatenation of these

two pieces of information. They assumed that the norm of a new feature in an abnormal video should be greater than the norm of a new feature in a normal video. They select top-k maximum norms. Only corresponding scores to the top-k norms are processed in the loss function. Feng et al. [21] proposed a two-stage self-training procedure that consists of self-guided attention boosted feature encoder and multiple instance pseudo label generator. Lv et al. [6] proposed a dynamic prototype unit to learn normal dynamics, an attention operation on an autoencoder encoding map. The authors proposed a meta-learning approach for VAD, named meta prototype unit. Their meta-learning model effectively learns to learn the normalcy of videos. Like most of these studies, we formulate the CAD problem in a weakly supervised manner. We aim to detect whether a video segment is abnormal or normal and provide an abnormal score for the segments.

2.2 Knowledge distillation techniques

Knowledge distillation techniques have been studied in many areas: NLP, neural machine translation, audio classification, face classification and human segmentation. Model compression is a useful technique for compressing a large network to a small network. Hinton et al. [28] proposed model distillation, namely offline distillation, which is an extended version of model compression, but its purpose is not only for compressing but also for improving the model's accuracy. As shown in the first block of Figure 2.1, model distillation transfers knowledge from a complex model to a simple model. The complex model, which is called

the teacher model, is pretrained. Using model distillation it is able to use the result of a big and complex model on a low-capacity device such as smartphones and other devices. Model distillation has been proposed to compress large networks without accuracy loss. It has been used for training classification networks where the teacher network guides the student network on training.

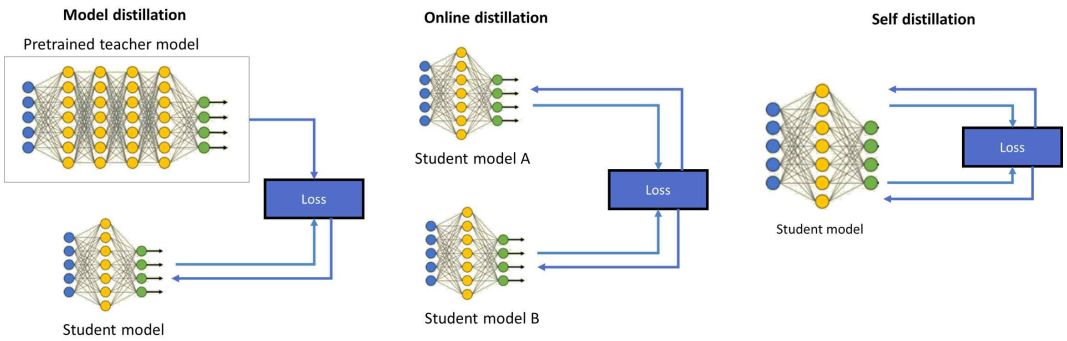


Figure 2.1 Types of knowledge distillation [27]

During training, if only the regression output of the student model is worse than that of the teacher, additional loss is applied [13]. Chen et al. [13] used model distillation to preserve accuracy with a small network for multi-class object detection. Ensemble distillation [20] is a sub variation of model distillation that distills knowledge from the ensemble of models instead of the complex teacher model. Wang et al. [15] used model distillation for model compression. The student model gets hints from the teacher model during feature extraction.

Mullapudi et al. [14] proposed an online model distillation method that does not require a pre-training teacher models. The student only gets supervision from the teacher when only its accuracy drops. For details, if their network accuracy is less

than the desired threshold, the model is updated using the teacher prediction. Zhou et al. [22] distill knowledge from a multi-teacher model. The authors assume that distilling from the previous model helps them to preserve old data information. The authors reconstructed previous models using the pruning method. Sarfraz et al. [18] proposed several knowledge distillation methods on different architectures and datasets and demonstrated that knowledge distillation is effective for noisy and imbalanced datasets. Tran et al. [19] proposed a technique in which a light model with multiple heads distills knowledge from each ensemble member. Each head shares the same network and preserves the diversity of each member of the ensemble. Radosavovic et al. [7] proposed data distillation in which multiple transformations are performed on input data and pass through the teacher model. And its output is used as annotated data for the student model.

Teacher model of model distillation is large and complex. Due to the increased complexity of training, model distillation is challenging for industrial usage [4]. Online distillation, i.e., the student-student mechanism, is a more cost-effective method that trains a duplication of the network [4]. It is also worth trying to train the n-copy of a model to reach the quality of a large model [4]. As shown in the second block of Figure 2.1, in online distillation, teacher and student model are updated together, and learn while influencing each other. Distillation methods focus on knowledge of logits, but Chung et al. [26] proposed a technique that also focuses on the feature map. The method transfers the knowledge of feature maps. Their loss functions consist of logit loss and feature map loss. Wu et al.

[29] integrate peer collaboration networks into online ensembles for image detection. The authors created multiple counterparts of an image using random augmentation and then fed these counterparts into multiple branch networks, respectively. The authors construct an ensemble teacher model for these multiple branch networks and improve the performance of the image classifier. Bhat et al. [31] proposed an online distillation model based on self-supervised learning. Each student model is trained under self-supervised learning and distills knowledge from each other by aligning softmax output scores.

Xu et al. [27] proposed a third type of distillation, called self-distillation. As shown in the third block of Figure 2.1, in self-distillation, it is performed in one model. Self-distillation is a special case of online distillation, and teacher is selected within the network. It learns representations only from a single network itself. Therefore, it improves generalization and improves the accuracy of the network. For VAD, matching segment scores is the principle of self-distillation. Zhang et al. [32] divided the image classifier neural network into several sections. An additional bottleneck layer is added to each section. All sections are trained as student models and distill knowledge from the deepest layer. Li et al. [30] proposed online distillation with an ensemble method based on self-distillation for image detection. Self-distillation converts feature maps of ensemble models into the fused feature model. And then student model distills knowledge from the fused feature. This approach boosts the generalization ability of the model.

Ⅲ. Proposed method

In this section, we demonstrate the proposed method, self-distillation with mirrored data based on MIL. Section 3.1 describes the overall view of our proposed method and CAD. In Section 3.2, we describe feature extraction. Afterward, in section 3.3, we propose multiple loss terms, including MIL ranking loss, self-distillation loss and, ground truth loss. Finally, Section 3.4 describes the training process and how multiple loss terms and single network is mixed together in CAD framework.

3.1 Overview

Figure 3.1 describes the overall flow of the proposed method. Dataset D consists of videos V that are labeled as $y \in \{0,1\}$ which are abnormal and normal annotation. Video representation techniques, such as C3D [33] and I3D [34], are feature extractors which take a 16-frames as input and extract features F for a video V_i. Based on MIL [1], features are transformed into K number of feature segments by utilizing mean operation where $K = 32$. In weakly labeled videos, some features represent normal context, and the rest represent anomaly context among the K features of abnormal video V_{ai}. Among the K features of normal video V_n, all of them represent the normal context. After feature extraction, feature instances are fed into three-layer fully connected single neural network for learning video representation features. The architecture of the single network is shown in Figure 3.2. Its first layer consists of 1024 units, followed by the second

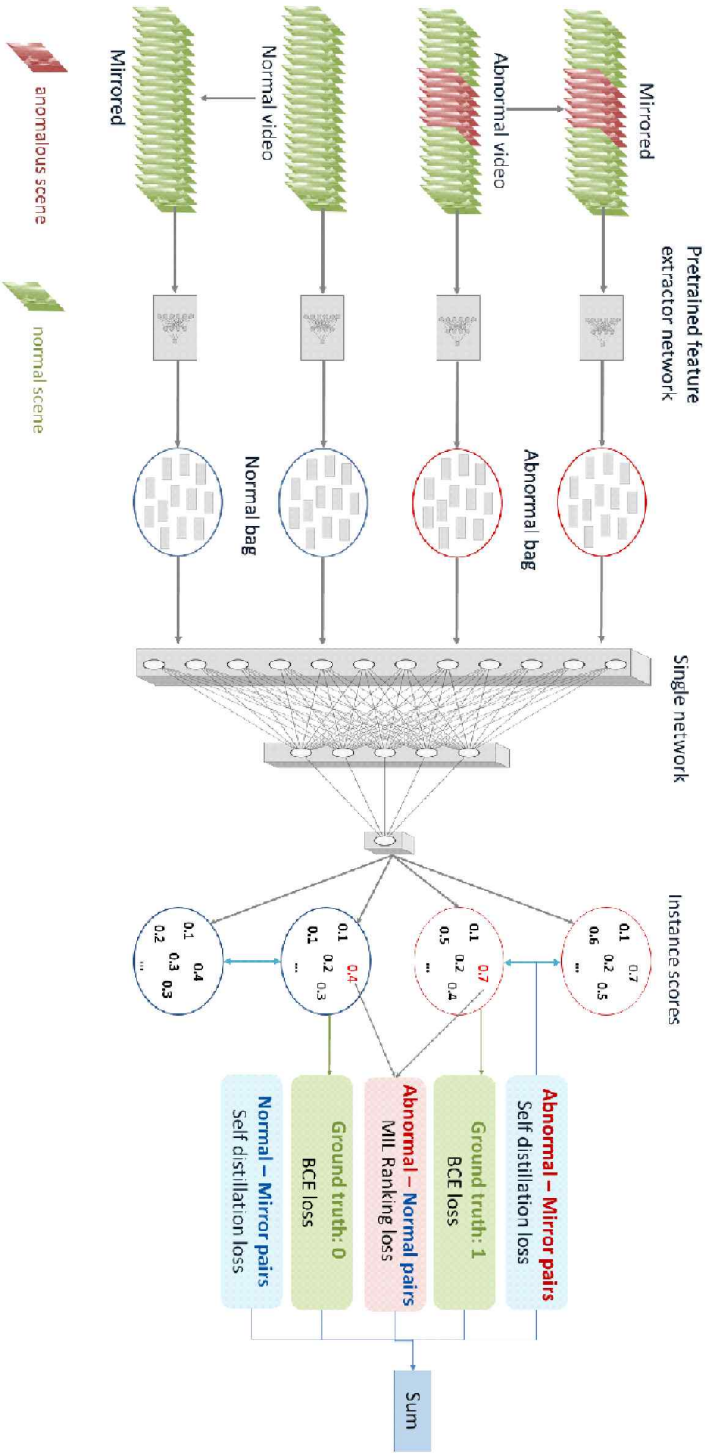


Figure 3.1 Overall flow of proposed model

and third layers with 128 and 1 unit. This network outputs anomaly scores S for K features of video with range [0,1]. For providing a self-distillation technique on our model, we create a mirror transformation video $\widehat{V_i}$ for each of the videos. Video V_i and its mirror transformation $\widehat{V_i}$ are the same due to time window action and labels, but the extracted features differ. It means single network may have many knowledge to learn from them under the task of keeping the consistency of the video-mirror pair. This learning process accelerates the capacity of a single network. Hence, a video and its transformation video are able to distill knowledge from each other.

In training, we process abnormal and normal videos and mirrored variations of them simultaneously. Abnormal and normal pairs help us in training to distinguish them from each other. We have three main learning terms related to abnormal-normal pair, video-mirror pair, and ground truth. MIL ranking loss is responsible for abnormal-normal pair; the self-distillation loss is responsible for video-mirror pair, and binary cross-entropy [37] (BCE) loss is responsible for ground truth. MIL ranking loss is estimated based on a comparison of instance scores of abnormal and normal videos. Self-distillation loss is estimated based on a distillation of video and its mirrored video. Self-distillation consists of two-loss terms, which are Kullback Leibler divergence and Mean Squared Error loss. Video and its mirrored videos distill knowledge from each other. This self-distillation loss optimizes a single network without need of any labels. Over iterations, the self-distillation loss is decreased by distilling knowledge.

3.2 Feature extraction

As we mentioned in the previous subsection, we extract videos into feature representation using a pretrained C3D model. C3D model was pretrained on an action recognition dataset, Sports-1M. This dataset consists of over a million videos from Youtube with 487 sports-related classes. Thus, we expect that the extracted video features efficiently represent human actions in videos. Figure 3.2 shows the algorithm of feature extraction using a pretrained model. On line 1, the algorithm initializes $features$ with an empty list. On lines 2 - 3, the algorithm

Algorithm 1. Feature extraction

Input:

(1) *videoPath*: path of video to be extracted

(2) *featureExtractor*: pretrained C3D model of sports1m dataset

Output:

(1) *features*: extracted C3D features of input video

Algorithm:

1. Initialize *features = {}*
2. Set *videoClips = getVideoClips(videoPath).* /*every 16-frames of video*/
3. Set *numFrames = getNumberofFrames(videoPath).*
4. **FOR** *i=1* **TO** *enumerate(videoClips)* **DO**
5. **IF** *len(videoClips) < 16* **THEN**
6. **CONTINUE**
7. Set *feature = featureExtractor.predict(videoClips(i))*.
8. Put *feature* into *features*.
9. Return *features*.
10. **END**

Figure 3.2 An algorithm for feature extraction of video

gets every 16-frame clip of a video and the total number of frames in a video. From lines 4 to 8, it iterates through the clip set and finds the prediction of extracted features from the object of the pretrained model. Finally, on line 9, the algorithm returns the extracted $features$.

3.3 Multiple loss terms

Our proposed multiple loss terms consist of MIL ranking loss, self-distillation loss and ground truth loss. MIL ranking loss is estimated for the pairs of abnormal and normal video. Self-distillation loss is estimated for the pairs of a single video and its mirror transformation. Ground truth loss is estimated for only single video compared to its ground truth score.

3.3.1 MIL ranking loss

MIL is the solution for weakly labeled video. It divides a video into multiple segments and estimates annotation for each segment using MIL ranking loss. MIL ranking loss is responsible for comparing output instance scores of abnormal and normal video pairs. MIL ranking loss has been studied by several studies [1, 5, 11] in multiple variations. Sultani et al. [1] proposed a loss which means the segment with the highest anomaly score s_a in the abnormal bag should be higher than the segment with the highest anomaly score s_n in the normal bag.

$$\max_{1 \le i \le k} {s_a}^i > \max_{1 \le i \le k} {s_n}^i \qquad (1)$$

A segment with the highest anomaly score in the abnormal video is likely to be

real anomalous scene. A segment with the highest anomaly score in the normal video looks like an anomalous scene. Drawback of Equation 1 [1] is that it ignores the underlying temporal structures of the abnormal video. A video can contain multiple abnormal actions, and a normal video can look like an anomalous scene. We expect that its drawback is reduced by the self-distillation approach. MIL ranking loss is described as:

$$loss_{ranking} = \max(0, 1 - \max_{1 \le i \le k} s_a^i + \max_{1 \le i \le k} s_n^i) \qquad (2)$$

Equation 2 [1] distinguishes anomaly instances and normal instances according to anomaly score. We utilize MIL ranking loss and its two helper loss terms temporal smoothness constraint and sparsity constraint loss terms. The most segments of abnormal video is likely to be normal. Thus, sparsity constraint loss imposes a prior that few segments have high anomaly scores. Sparsity constraint loss [1] is defined as:

$$loss_{sparsity} = \lambda_1 \sum_{i}^{n} s_a^i \qquad (3)$$

where s_a is the anomaly score of the abnormal video. Temporal smoothness constraint enforces temporal smoothness between the scores of instances. Temporal smoothness constraint loss [1] is described as:

$$loss_{smoothness} = \lambda_1 \sum_{i}^{k-1} (s_a^i - s_a^{i+1})^2 \qquad (4)$$

where k is the number of instances.

3.3.2 Self-distillation loss

The self-distillation is responsible for how anomaly scores of a video and its mirror transformation are consistent. In the ideal case, anomaly scores of a video and its mirror transformation should be equal since they are the same. Our proposed the self-distillation loss consists of two loss terms that are Kullback Leibler (KL) divergence and mean squared error (MSE) loss. KL divergence is responsible for measuring how a video's probability distribution of anomaly scores from another probability of anomaly scores of the mirror transformation. Then we define KL divergence loss [3] as:

$$D_{KL}(S||\hat{S}) = \sum_{i=1}^{k} s_i \log\left(\frac{s_i}{\hat{s}_i}\right) \qquad (5)$$

where s_i is the anomaly score of the video V_i and $\hat{s}_i$ is the anomaly score of the mirrored video $\hat{V}_i$. KL divergence on both directions that are S to $\hat{S}$, and $\hat{S}$ to S is then described as [3]:

$$Loss_{KL} = D_{KL}(S \,||\, \hat{S}) + D_{KL}(\hat{S} \,||\, S) \qquad (6)$$

MSE loss is estimated directly on predicted scores for input video and its mirror transformation to estimate their consistency. Mean squared error is defined as:

$$loss_{MSE} = \frac{1}{k}\sum_{i=1}^{k}(s_i - \hat{s}_i)^2 \qquad (7)$$

3.3.3 Ground truth loss

We maintain BCE [37] loss for the prediction of ground truths. BCE [37] loss function is defined as:

$$loss_{BCE} = \sum_{i=1}^{k} -(y \log(s_i) + (1-y)\log(1-s_i))) \quad (8)$$

where y represents the ground truth of the current video. Estimating ground truth loss has good impact on training normal videos and positive segments of abnormal videos. Its drawback is that it is likely mess up the learning process on negative segments of abnormal videos since the dataset has no segment-level labeling.

3.3.4 Final loss

Finally, our final loss for abnormal video, normal video, and those mirrored videos is estimated as the sum of all loss functions:

$$loss_{total} = loss_{KL} + loss_{MSE} + loss_{BCE} + loss_{ranking} + loss_{smoothness} + loss_{sparsity} \quad (9)$$

We expect that our self distillation single network will efficiently generalize abnormal video and normal video by utilizing these loss functions.

3.4 Training

This subsection describes the structure of the training and evaluation process for CAD. Figure 3.3 shows the algorithm of the training procedure. On line 1 - 2,

the algorithm initializes Adam optimizer with *learningRate = 0.001* and

Algorithm 2. Main block

Input:

(1) *D*: Dataset

Output:

(1) Save the best trained model

Algorithm: Initialize *model = SingleNetwork()*.

1. Initialize optimizer = *Adam(learningRate = 0.001)*.
2. Initialize bestAUC = 0.
3. Set *epoch = 15k*.
4. Set *abnormalBatches = DataLoader(D, type = "Abnor", batchSize=16)*.
5. Set *normalBatches = DataLoader(D, type = "Normal", batchSize=16)*.
6. Set *evaluationBatches = DataLoader(D, type = "Eval", batchSize=1)*.
7. **FOR** *i=1* **TO** *epoch* **DO**
8. Set *abnormalMirrorBatch = correspondingMirror(abnormalBatches(i))*.
9. Set *normalMirrorBatch = correspondingMirror(normalBatches(i))*.
10. *train(abnormalBatches(i), abnormalMirrorBatch, normalBatch(i), normalMirrorBatch, model, optimizer)*.
11. Set *AUC = evaluation(evaluationBatches, model)*.
12. **IF** *AUC > bestAUC* **THEN**
13. Set *bestAUC = AUC*.
14. Save *model*.
15. **END**

Figure 3.3 An algorithm for training CAD

Algorithm 3. Training one iteration

Input:

(1) *abnormalBatch*: abnormal training video features with labels

(2) *abnormalMirrorBatch*: batch of mirrored abnormal training video features

(3) *normalBatch*: batch of abnormal training video features

(4) *normalMirrorBatch*: batch of mirrored normal training video features

(5) *model*: single model object

(6) *optimizer*: object of Adam training optimizer

Output:

(1) update the single network

Algorithm:

1. Set *scoresAbnormal = model(abnormalBatch).*
2. Set *scoresMirrorAbnormal = model(abnormalMirrorBatch).*
3. Set *scoresNormal = model(normalBatch).*
4. Set *scoresMirrorNormal = model(normalMirrorBatch).*
5. Set *scoresOrig = concat(scoresAbnormal, scoresNormal).*
6. Set *labelOrig* according to abnormal and normal
7. Set *cost = MILLoss(scoresAbnormal, scoresNormal) + KLLoss(scoresAbnormal, scoresMirrorAbnormal) + KLLoss(scoresNormal, scoresMirrorNormal) + MSELoss(scoresAbnormal, scoresMirrorAbnormal) + MSELoss(scoresNormal, scoresMirrorNormal) + BCELoss(scoresOrig, labelOrig) + sparse(scoresAbnormal) + smooth(scoresAbnormal)*.
8. *optimizer.zeroGrad()*.
9. *cost.backward()*.
10. *optimizer.step()*.
11. **END**

Figure 3.4 An algorithm for training one iteration

bestAUC = 0, which preserves the best AUC during evaluation. Line 3 sets epoch

size as 15000. On lines 4 - 6, the algorithm fetches a set of an abnormal batch, normal batch, and evaluation batch from the data loader for training. From lines 7 to 14, it iterates in epochs and perform training procedure. On lines 8 - 9, the algorithm gets the batches of mirror transformation features based on corresponding abnormal and normal batches. Then it feeds all batches, model and optimizer to main train procedure on line 10. Afterwards, the algorithm gets evaluation *AUC* result on line 11. If the *AUC* result is higher than *bestAUC*, the algorithm preserves it as best AUC. Finally, on line 14, the algorithm saves *model*. After epochs, the algorithm finds the best AUC and the corresponding model.

Algorithm 4. Single neural network

Input:

(1) *inputBatch*: input batch of video features from dataset

Output:

(1) *scores*: anomaly scores

Algorithm:

1. Set *out* = *ReLU(FC1(inputBatch))*. /*layer with 1024 units*/
2. Set *out* = *Dropout(out)*
3. Set *out* = *ReLU(FC2(out))*. /*layer with 128 units*/
4. Set *out* = *Dropout(out)*
5. Set scores = *Sigmoid(FC3(out))*. /*layer with 1 unit as final*/
6. Return *scores*.
7. **END**

Figure 3.5 An algorithm for single network

Figure 3.4 shows the algorithm of the training function, which is responsible for training batch in one iteration of epochs. On lines 1 - 4, the algorithm runs a single network for all batches and gets corresponding anomaly scores. On line 5, the algorithm concatenates scores of an abnormal and normal batch for further usage. On line 6, the algorithm sets default label values for abnormal and normal batch segments, respectively. On line 7, the algorithm estimates MIL ranking loss between abnormal and normal batch, KL and MSE loss for plain and mirror batch pair, BCE loss for scores of plain and the labels, smoothness and sparsity constraints for abnormal batches. And all the loss terms are summed as a final loss. Finally, from lines 8 to 10, the algorithm backpropagates the loss function.

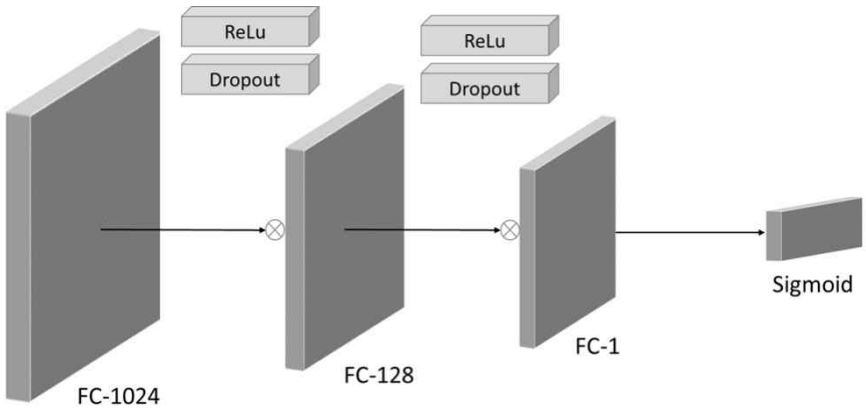


Figure 3.6 Architecture of single network

Figure 3.5 shows the algorithm for the single network, which is a three-layer fully connected neural network. As shown in Figure 3.6, the first layer is 1024, units followed by the second and third layers with 128 and 1 unit, respectively. As we mentioned in the previous section, it generates scores as a regressor, e.g.,

high scores for anomaly segments and low scores for normal segments. On lines 1 - 4, the algorithm passes the input batch through the network layer, ReLU activation, function and dropout regularization. On line 5, the algorithm calls the output layer followed by the sigmoid function to get anomaly scores. And, on line 6, it returns anomaly scores.

Algorithm 5. Evaluation

Input:

(1) *evaluationBatches*: evaluation video features from dataset

Output:

(1) *resAUC*: evaluation result in area under curve metric

Algorithm:

1. Initialize *totalLogits = {}*
2. Set *groundTruth = load(GROUNDTRUTHPATH)*.
3. **FOR** *i=1* **TO** *enumerate(evaluationBatches)* **DO**
4. Set *logits = model(evaluationBatches(i))*.
5. Put *logits* into *totalLogits*.
6. Set *predictions = expand(totalLogits , 16)*. */*transform instance scores into frame scores*/*
7. Set *fpr, tpr = RocCurve(groundTruth, predictions)*.
8. Set *resAUC = AUC(fpr, tpr)*.
9. Return *resAUC*.
10. **END**

Figure 3.7 An algorithm for evaluation

Figure 3.7 shows the algorithm of training evaluation. On line 1, it initializes the anomaly scores *totalLogits* with an empty list, and on line 2, it loads *groundTruth* from the path. From line 3 to 5, the algorithm iterates through

evaluation batches to get the anomaly scores of every batch as *logits* and accumulates into *totalLogits* list. Since the single network returns the anomaly scores for segments, the algorithm extrapolates the scores into frame level by repeating 16 times on line 6. On line 7, the algorithm calculates false positive rate (FPR) and true positive rate (TPR) as *fpr* and *tpr* based on ground truth and predictions. Finally, on lines 8 - 9, it calculates *AUC* using *fpr* and *tpr*, and returns the result.

Ⅳ. Performance Evaluation

This section shows the results of the experiments. The main goal of these experiments is to detect child abuse. Subsection 4.1 proposes a CAD dataset that we used in the experimental evaluation. Afterward, subsection 4.2 describes implementation detail to perform CAD. Subsection 4.3 describes experimental evaluation on these datasets using our proposed method.

4.1 Dataset

This subsection provides detailed information on our proposed dataset. Our new dataset is created for CAD. It contains abnormal and normal videos. We collected 229 real videos consisting of 119 abnormal and 110 normal videos with an average length of 1 minute. Each video's frame dimension size is 320x240, and the frame rate is 30 fps. Each video is labeled as "abnormal" and "normal". Abnormal video contains abnormal actions, and it also can contain normal actions, but the normal video only contains normal actions. Figure 4.1 depicts sample frames of the dataset and some challenging videos are included in normal videos, such as preventing a child from falling or injuries. These actions look similar to abnormal activities in some way. For training, video-level labels are utilized to learn multiple instances. For training, video-level labels are utilized to learn multiple instances. Although to evaluate our model on testing, we create a frame label for abnormal testing video. Then, we use ROC curves and corresponding AUC for the result. Table 4.1 shows the video splits of the dataset and we divide

our dataset as a training set consisting of 99 normal and abnormal videos and the

Table 4.1 Video splits of CAD dataset

Events	For training	For testing
Abnormal	99	20
Normal	99	11
Total	198	31

(a) Normal (b) Abnormal

Figure 4.1 Sample frames of CAD dataset

Table 4.2 CAD dataset frames

Total	Training	Testing	Normal	Abnormal
451,898	441,632	10,266	197,488	254,400

testing set consisting of 11 normal and 20 abnormal videos. Table 4.2 shows frame numbers by several group such as training, testing, normal and abnormal. The frames of abnormal category contains both normal and abnormal frames.

4.2 Implementation detail

We extract video features from the FC6 layer of the pre-trained C3D [33] network. C3D model outputs 4096 x 1 array for every 16 frames of a video. For the score mapping single neural network, we divide a video into non-overlapping 32 segments. We obtain a C3D feature for the segment by averaging of all 16 frame clips in a segment. Hence, for each video, we have a 32 x 4096 feature for C3D. For the single neural network, we employ Adam optimizer with an initial learning rate of 0.001. Dropout regularization and ReLu activation function are used between these layers, and Sigmoid activity function is used in the last layer. As we mentioned in previous sections, the single network architecture is three-layer fully connected network with 1024 units, 128 units and 1 unit as the final layer. We have tested variations of units in the layers, and there was no impact on changing the units of layers. We use the Pytorch framework for all modules. For training, 16 abnormal videos and 16 normal videos are trained as a mini-batch with an iteration of 15k. Thus, the original 32 videos and their mirrored videos and 64 videos as mini-batch are processed in one iteration. We compute loss as shown in Equation 9 and back-propagate the loss.

Our machine architecture is x86_64, and it runs 40 CPUs whose specification is Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz and GPU which is NVIDIA TITAN RTX with 64G memory. We utilized the Docker environment for OS management. We have installed the NVIDIA-PyTorch image, which was downloaded from the Nvidia GPU Cloud website with the following docker command:

docker run --gpus all -it ---ipc=host nvcr.io/ea-clara-agx/agx-pytorch:xx.xx-py3

Add ipc=host parameter, which increases the size of shared memory. It is important for Torch multi-threaded dataloaders. Docker image version is 20.03, and PyTorch version is 1.5.0.

4.2.1 Evaluation metric

To evaluate our proposed model, we create the frame-level annotations for the test videos. For evaluation metrics, following previous studies [1, 5, 11], we use the frame-based receiver operating characteristics (ROC) [36] curve and its area under the curve (AUC) [36]. Figure 4.2 shows the performance example of the result of CAD model in the plot. The plot shows the predicted abnormal scores for the video frames in the histogram. The X-axis represents abnormal scores, and the Y-axis represents the number of frames. The blue area represents abnormal, and its ground truth score is 1. And the red area represents normal, and its ground truth score is 0. Some of the normal frames have the predicted scores greater than zero, and some abnormal frames have the predicted scores

lower than one. Firstly we transfer this plot to ROC curve using classification thresholds to estimate the accuracy of the plot. Our classification thresholds are iteratively set through possible abnormal scores with a range between 0 to 1. For example, threshold = 0.5 means we classify all the frames as normal if their scores are below 0.5, and we classify all the frames as abnormal if their scores are above 0.5. ROC curve draws a point of TPR and FPR of the thresholds. TPR is described as:

$$True\ Positive\ Rate = \frac{true\ positives}{all\ positives} \qquad (10)$$

FPR is described as:

$$False\ Positive\ Rate = \frac{false\ positives}{all\ negatives} \qquad (11)$$

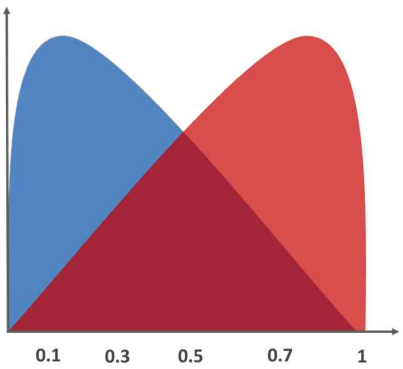


Figure 4.2 A histogram plot example of CAD model performance

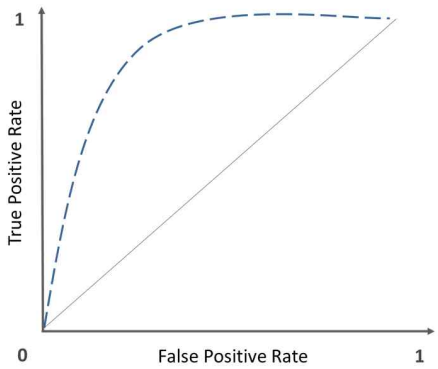


Figure 4.3 ROC curve

TPR is expressed in the y axis and FPR is expressed in the x axis of the ROC curve as shown in Figure 4.3. AUC is the area of the entire ROC curve as

shown Figure 4.4 and it measures the performance of all classification thresholds.

Figure 4.4 AUC

4.3 Experiment results

This subsection presents the experiments designed to evaluate the self-distillation model. We compare our results with the state-of-the-art methods. We also analyze the qualitative performance of our CAD model.

4.3.1 Comparison with the state-of-the-art methods

Table 4.3 shows the main result of CAD dataset. We compare our method with the state of the art methods. Thus, our model outperforms Sultani model [1] by 10.4% and Tian model [5] by 1.6%. This result demonstrates that our proposed model is able to detect child abuse videos effectively. We have tested our method without smoothness and sparsity constraints. The smoothness and sparsity losses had good impact on training. Figure 4.5 shows the comparison of ROC.

Table 4.3 AUC comparison on CAD dataset

Model	AUC(%)
Sultani et al. 2018	73.41%
Tian et al. 2021	82.29%
Ours	**83.88%**

We have also tested the impact of a parameter on multiple losses. We added parameters on multiple losses, and these losses are multiplied by that parameters. However, we did not see any difference or improvement for the evaluation.

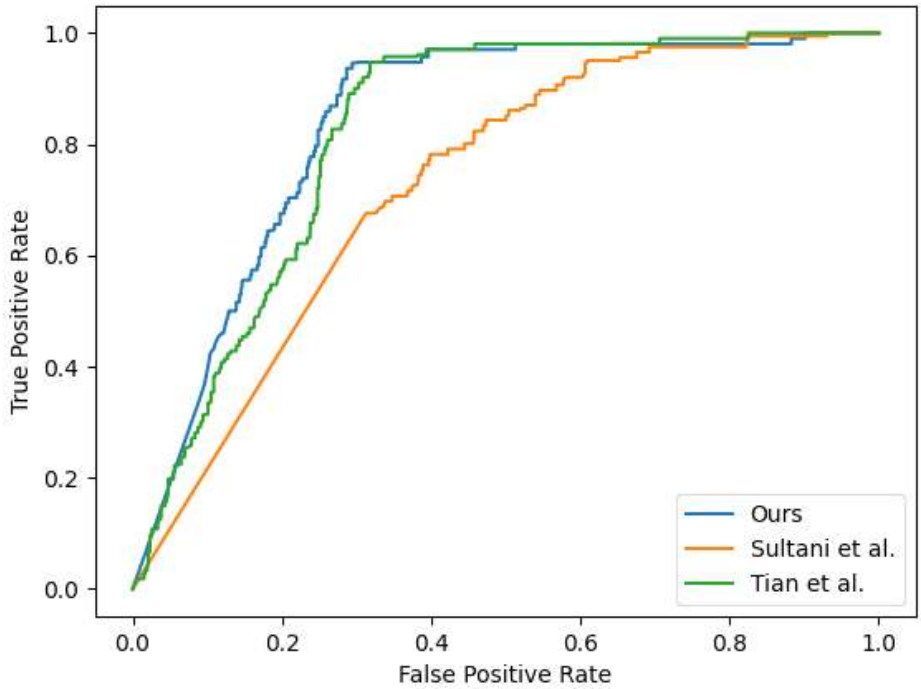
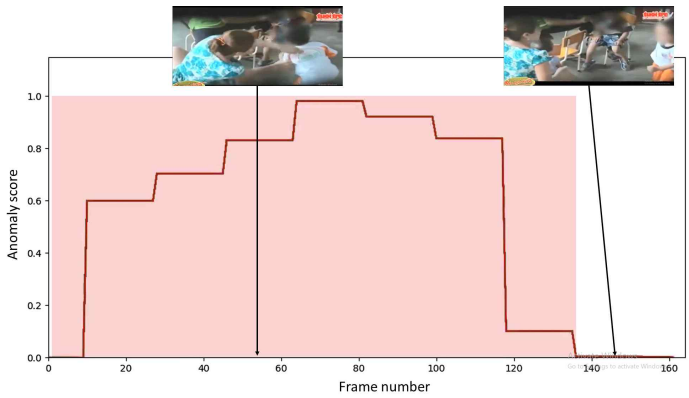


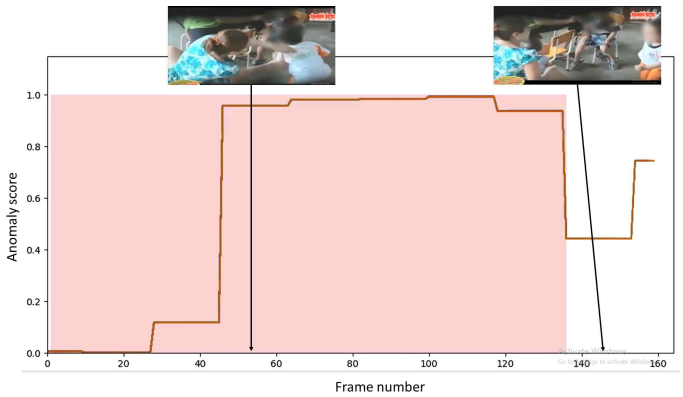
Figure 4.5 ROC comparison of Sultani et al. [1] (orange) Tian et al. [5] (green) and our CAD method (blue)

4.3.2 Qualitative analysis

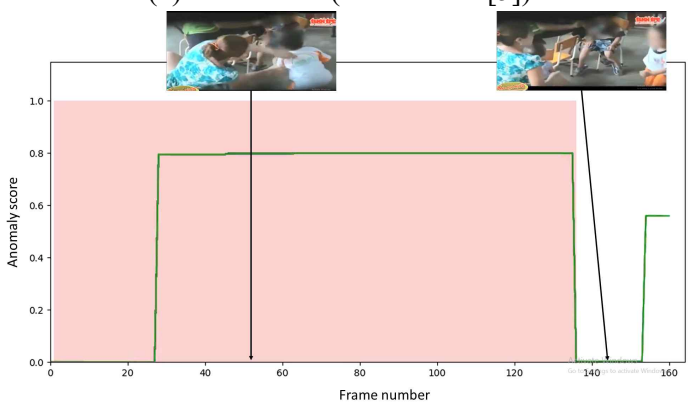
Figure 4.6 shows qualitative results of our method on testing videos from our proposed CAD dataset. A red rectangle area indicates the ground truth of abnormal, and a line indicates prediction scores in the figures. The X-axis represents a frame number of a video, and the Y-axis represents an abnormal score with a range from 0 to 1. Figures 4.6(a) - (f) show the result of abnormal videos, and Figures 4.6(g) - (l) show that of normal videos. The first abnormal sample is depicted from Figure 4.6(a) to (c), and our model outputs high scores close to the ground truth for abnormal sections. However, Sultani [1] and Tian [5] models both give false high scores for the short time of normal section. The second abnormal sample is shown from Figure 4.6(d) to (f), and our model generates high scores in abnormal sections, and it matches to ground truth. Sultani [1] and Tian [5] models generate high abnormal scores in most normal parts. The first normal sample is presented from Figure 4.6(g) to (i); our model outputs a low score for the normal video that matches to the ground truth. However, the Sultani [1] model gives high scores for whole parts and is not acceptable. Tian [5] model also generates false scores in several parts. The second normal sample is depicted from Figure 4.6(j) to (l); our model generates low scores and does not throw false alerts for the normal video. Sultani's [1] model generates false high anomaly scores in the end of the video. Tian [5] model also generates false scores in small parts. These examples demonstrate that our model performs robust detection on abnormal and normal actions than other approaches.



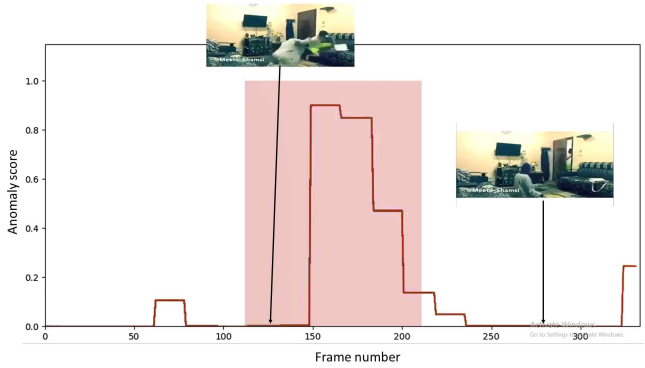
(a) Abnormal (ours)



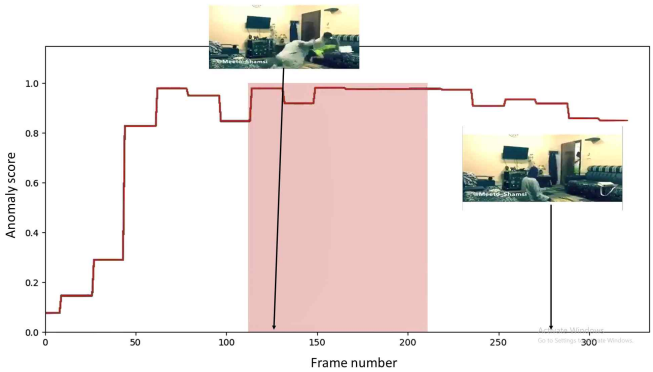
(b) Abnormal (Tian et al. [5])



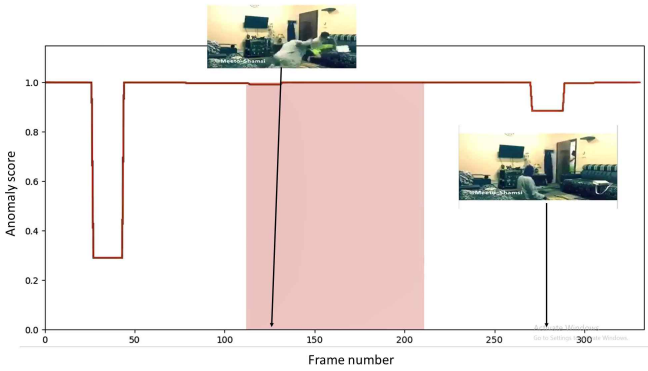
(c) Abnormal (Sultani et al. [1])



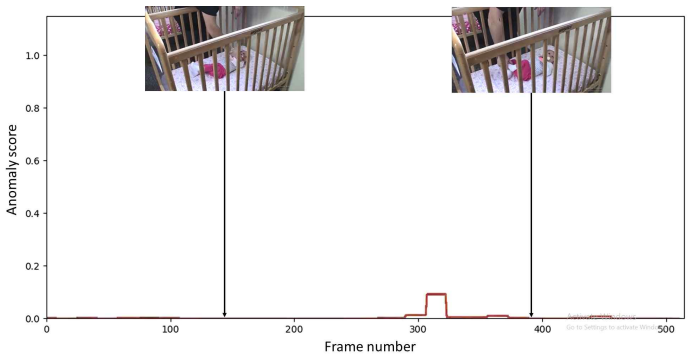
(d) Abnormal (Ours)



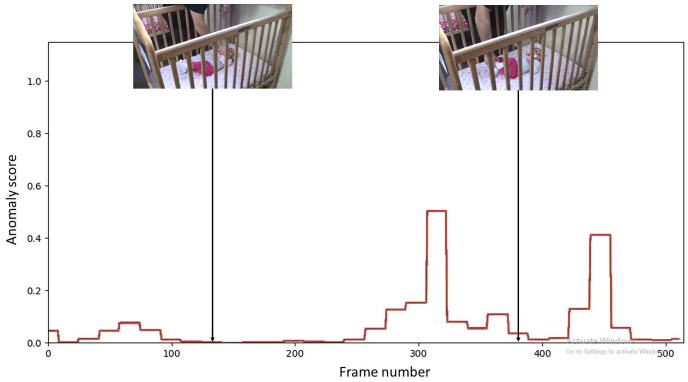
(e) Abnormal (Tian et al. [5])



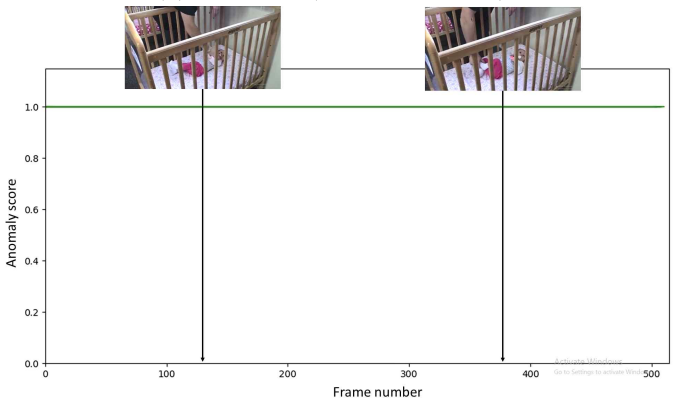
(f) Abnormal (Sultani et al. [1])



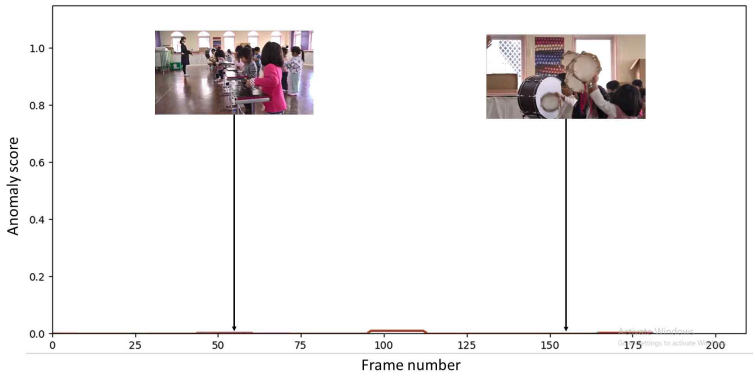
(g) Normal (ours)



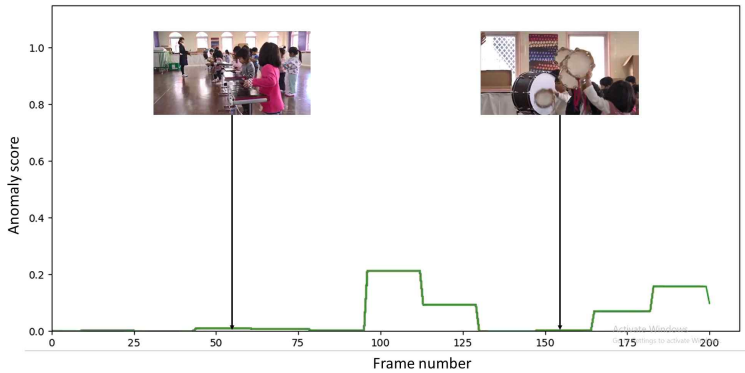
(h) Normal (Tian et al. [5])



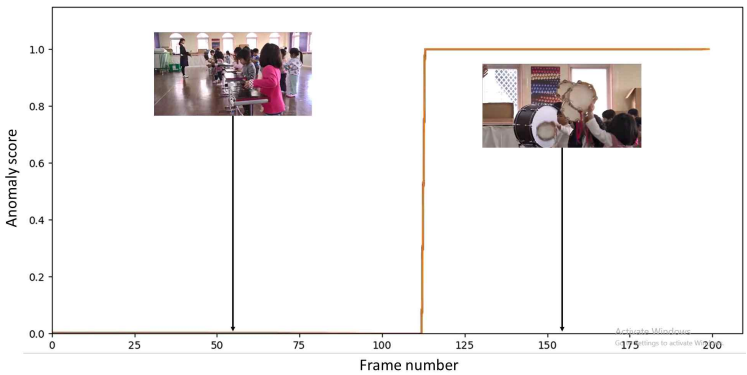
(i) Normal (Sultani et al. [1])



(j) Normal (ours)



(k) Normal (Tian et al. [5])



(l) Normal (Sultani et al. [1])

Figure 4.6 Qualitative results of our method on child abuse testing videos

4.4 Summary of experiments

Experiments show that our proposed method is more robust than the plain MIL model of Sultani [1] and Tian [5] on CAD. Our model learns rich characteristics of abnormal and normal actions based on self-distillation. Our multiple loss terms allow learning the rich characteristics of the video. The main difference between the plain MIL model and CAD model is that the MIL model ignores ground truth-based loss such as BCE, and it only prefers abnormal-normal comparison. Our CAD estimates BCE loss which forces the instances to follow ground truth labels. Since we shrink a video into small fixed 32 instances, forcing these instances to follow ground truth is acceptable. Estimating this loss helps to increase the accuracy like other loss functions.

Ⅴ. Conclusion and Future work

In this thesis, we proposed an effective self-distillation approach with mirrored data for CAD to enhance the capacity of the single network without the need for an assistive model that reduces training costs with high accuracy. We also proposed a new dataset for CAD that consists of two classes of videos in general which are abnormal and normal. We utilized multiple losses with a combination of self-distillation loss and MIL ranking loss for training. We conducted an experimental evaluation by comparing our CAD method with the state-of-the-art methods on our proposed CAD dataset. Our proposed method outperforms the state-of-the-art approaches on our proposed dataset. We improved accuracy by 10.4% and 1.6% by comparing performance with the state-of-the-art methods.

In future research, our proposed method can be improved. For example, multi-scale segments can improve the accuracy of our model. Multi-scale provides a solution for video frame size problem, e.g., the scene of action and actor is shown in small size are with a too big background in the frame. For the improvement of the dataset, we can increase the volume of our CAD dataset. For improving the self-distillation model, we can distill the knowledge inside the single network. By considering every fully connected layer as a classifier, we can distill the knowledge between these layers, e.g., fc3 layer with fc2 layer and fc2 layer with fc1 layer. Distilling knowledge inside the network allows us not to use mirror transformation of the input videos and reduces the time complexity.

References

[1] Waqas Sultani, Chen Chen and Mubarak Shah. (2018). Real-world Anomaly Detection in Surveillance Videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 6479-6488).

[2] Cristian Buciluǎ, Rich Caruana and Alexandru Niculescu-Mizil. (2006). Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 535−541).

[3] Ying Zhang, Tao Xiang, Timothy M. Hospedales and Huchuan Lu. (2018). Deep Mutual Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4320-4328).

[4] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E. Dahl and Geoffrey E. Hinton. (2018). Large scale distributed neural network training through online distillation. arXiv preprint:arXiv:1804.03235.

[5] Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W. Verjans and Gustavo Carneiro. (2021). Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning. arXiv preprint:arXiv:2101.10030.

[6] Hui Lv, Chen Chen, Zhen Cui, Chunyan Xu, Yong Li and Jian Yang. (2021). Learning Normal Dynamics in Videos with Meta Prototype Network. arXiv preprint:arXiv:2104.06689.

[7] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari and Kaiming He. (2018). Data distillation: Towards omni-supervised learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 4119-4128).

[8] Bruno Degardin and Hugo Proença. (2020). Weakly and Partially Supervised Learning Frameworks for Anomaly Detection. Accessed 13 May 2021. <http://hdl.handle.net/10400.6/10821>

[9] Bruno Degardin and Hugo Proença.(2021). Iterative weak/self-supervised classification framework for abnormal events detection. Pattern Recognition Letters, 50-57.

[10] Ammar Mansoor Kamoona, Amirali Khodadadian Gosta, Alireza Bab-Hadiashar and Reza Hoseinnezhad. (2020). Multiple Instance-Based Video Anomaly Detection using Deep Temporal Encoding-Decoding. arXiv preprint: arXiv:2007.01548.

[11] Hui Lv, Chuanwei Zhou, Zhen Cui, Chunyan Xu, Yong Li and Jian Yang. Localizing Anomalies from Weakly-Labeled Videos. IEEE Transactions on Image Processing (Vol. 30, 4505 - 4515).

[12] Kun Liu, Wu Liu, Huadong Ma, Mingkui Tan and Chuang Gan. (2021). A Real-time Action Representation with Temporal Encoding and Deep Compression. IEEE Transactions on Circuits and Systems for Video Technology, 31(2), 647-660.

[13] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han and Manmohan Chandraker. (2017). Learning efficient object detection models with knowledge distillation. In NIPS, 742−751.

[14] Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan and Kayvon Fatahalian. (2019). Online Model Distillation for Efficient Video Inference. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). (pp. 3573-3582).

[15] Ji Wang, Weidong Bao, Lichao Sun, Xiaomin Zhu, Bokai Cao and Philip S. Yu. (2019). Private Model Compression via Knowledge Distillation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33(01), pp. 1190-1197).

[16] Shikha Dubey, Abhijeet Boragule and Moongu Jeon. (2019). 3D ResNet with Ranking Loss Function for Abnormal Activity Detection in Videos. In Proceedings of 2019 International Conference on Control, Automation and Information Sciences (ICCAIS) (pp. 1-6).

[17] Muhammad Zaigham Zaheer, Arif Mahmood, Hochul Shin and Seung-Ik Lee. (2020). A Self-Reasoning Framework for Anomaly Detection Using Video-Level Labels. IEEE Signal Processing Letters, 27, 1705-1709.

[18] F. Sarfraz, E. Arani and B. Zonooz. (2021). Knowledge Distillation Beyond Model Compression. In Proceedings of 25th International Conference on Pattern Recognition (ICPR) (6136-6143).

[19] Linh Tran, Bastiaan S. Veeling, Kevin Roth, Jakub Swiatkowski, Joshua

V. Dillon, Jasper Snoek, Stephan Mandt, Tim Salimans, Sebastian Nowozin and Rodolphe Jenatton. (2021). Hydra: Preserving Ensemble Diversity for Model Distillation. In Proceedings of ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning (pp. 1-10).

[20] Markus Freitag, Yaser Al-Onaizan and Baskaran Sankaran. (2017). Ensemble Distillation for Neural Machine Translation. arXiv preprint: arXiv:1702.01802.

[21] Jia-Chang Feng, Fa-Ting Hong and Wei-Shi Zheng. (2021). MIST: Multiple Instance Self-Training Framework for Video Anomaly Detection. In Proceedings of CVPR 2021 (pp. 1-14).

[22] Peng Zhou, Long Mai, Jianming Zhang, Ning Xu, Zuxuan Wu, Larry S. Davis. (2020). M2KD: Incremental Learning via Multi-model and Multi-level Knowledge Distillation. In Proceedings of BMVC (pp. 1-13).

[23] "R. Maqsood, UI. Bajwa, G. Saleem, Rana H. Raza, MW. Anwar. (2021). Anomaly Recognition from surveillance videos using 3D Convolutional Neural Networks. arXiv preprint: arXiv:2101.01073."

[24] Muhammad Zaigham Zaheer, Arif Mahmood, Marcella Astrid and Seung-Ik Lee. (2020). CLAWS: Clustering Assisted Weakly Supervised Learning with Normalcy Suppression for Anomalous Event Detection. In Proceedings of Computer Vision – ECCV 2020 European Conference on Computer Vision (pp 358-376).

[25] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H. Li and

Ge Li. (2019). 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 (CVPR), (pp. 1237-1246).

[26] Inseop Chung, Seonguk Park, Jangho Kim and Nojun Kwak. (2020). Feature-map-level Online Adversarial Knowledge Distillation. In Proceedings of roceedings of the 37th International Conference on Machine Learning (PMLR 119:2006-2015).

[27] Ting-Bing Xu and Cheng-Lin Liu. (2019). Data-Distortion Guided Self-Distillation for Deep Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 5565-5572.

[28] Geoffrey Hinton, Oriol Vinyals and Jeff Dean. (2015). Distilling the Knowledge in a Neural Network. NIPS 2014 Deep Learning Workshop, 1-9.

[29] Guile Wu, Shaogang Gong. (2020). Peer Collaborative Learning for Online Knowledge Distillation. arXiv preprint: arXiv:2006.04147

[30] Shaojie Li, Mingbao Lin, Yan Wang, Feiyue Huang, Yongjian Wu, Yonghong Tian, Ling Shao and Rongrong Ji. (2021). Distilling a Powerful Student Model via Online Knowledge Distillation. arXiv preprint: arXiv:2103.14473

[31] Prashant Bhat, Elahe Arani and Bahram Zonooz. (2021). Distill on the Go: Online knowledge distillation in self-supervised learning. Learning from Limited or Imperfect Data (L2ID) Workshop @ CVPR 2021, 1-10.

[32] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao and Kaisheng Ma. (2019). Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 3713-3722).

[33] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani and Manohar Paluri. (2015). Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the International Conference on Computer Vision (ICCV) (pp. 1-9).

[34] Joao Carreira and Andrew Zisserman Quo Vadis. Action Recognition? A New Model and the Kinetics Dataset. (2017). In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-10).

[35] Lee Han-soo (2021). Fatal child abuse stirs outrage in Korea. Accessed 20 May 2021. <https://www.koreabiomed.com/news/articleView.html?idxno=10105>.

[36] Charles E. Metz. (1978). Basic principles of ROC analysis, In Seminars in Nuclear Medicine, 8(4), 283-298.

[37] Antonia Creswell, Kai Arulkumaran and Anil A. Bharath. (2017). On denoising autoencoders trained to minimise binary cross-entropy. arXiv preprint: arXiv:1708.08487

Abstract (Korean)

아동 학대 감지를 위한 자가 증류 접근법 기반 다중 인스턴스 학습

간수크 친볼드

충북대학교 대학원
전기・전자・정보・컴퓨터학부 컴퓨터과학전공
지도교수 나스리디노프 아지즈

요 약

아동 학대는 아동을 대상으로 하는 모든 신체적 학대 또는 폭력 행위이다. 이러한 아동 학대는 국내에서 매년 증가하는 추세로 심각한 사회문제로 부상하고 있다. 본 논문에서는 영상 속에서 발생하는 아동 학대를 자동으로 감지하는 방법을 제안한다. 우리는 영상 데이터로부터 아동 학대를 감지할 수 있는 대표적인 특징들을 추출하기 위해 미러링 변환을 통한 자체 증류 접근법을 기반으로 아동 학대에 해당하는 비정상을 탐지한다. 이 목표를 달성하기 위해, 우리는 심층 신경망을 사용하여 원본 비디오와 미러링된 영상 데이터 사이에서 정제된 지식을 학습한 후 정상과 비정상 영상 세그먼트를 일반화한다. 특히, 영상 수준의 약한 레이블을 효과적으로 학습하기 위해 기존의 최첨단 방식의 MIL 랭킹 손실을 지원하는 자체 증류 손실 기능을 소개한다. 추가적으로, 우리는 아동 학대 감지를 위한 새로운 영상 데이터 집합을 제공한다. 이러한 아동 학대 영상 데이터 집합을 사용한 실험을 통해 우리의 제안방법은 기존의 최첨단 방법들보다 뛰어난 아동 학대 감지 성능을 보인다.

키워드: 영상 이상 감지, 아동 학대 감지, 다중 인스턴스 학습, 자가 증류, 지식 증류

Acknowledgements

First, I want to acknowledge and thank my advisor, Prof. Aziz Nasridinov, for the continuous support of my Master's study and research, for his patience and experience. His valuable comments made this work possible. I am grateful to the rest of my thesis committee: Prof. Kwan-Hee Yoo and Prof. Sang-Hyun Choi, for their thoughtful comments and suggestions. I want to express my gratitude to the Chungbuk National University and the Korean government that funded me through the teaching assistant program, the government projects, and the BK21 program.

I thank my fellow labmates of Data Analytics Laboratory, Jong-Hyeok Choi, Jeong-Hun Kim, U-Ju Kim and Jae-Jun Lee, Tserenpurev Chuluunsaikhan, and Yifan Zhu, for helping me during my study and encouraging me to prepare the critical milestones of the thesis. I want to thank my girlfriend, Uuriintuya Bayarsaikhan, who supports me all the way. Finally, I want to thank my family: my parents, for giving birth to me in the first place and supporting me spiritually.

Chinbold Gansukh

Chungbuk National University

June 2021