Weakly Supervised Two-Stage Training Scheme for Deep Video Fight Detection Model

Zhenting Qi*  
ZJU-UIUC Institute  
Zhejiang University  
Haining, China  
Zhenting.19@intl.zju.edu.cn

Ruike Zhu*  
Department of Computer Science  
University of Illinois at Urbana-Champaign  
Urbana, IL, USA  
ruike2@illinois.edu

Zheyu Fu*  
ZJU-UIUC Institute  
Zhejiang University  
Haining, China  
Zheyu.19@intl.zju.edu.cn

Wenhao Chai*  
ZJU-UIUC Institute  
Zhejiang University  
Haining, China  
Wenhaochai.19@intl.zju.edu.cn

Volodymyr Kindratenko  
National Center for Supercomputing Applications  
University of Illinois at Urbana-Champaign  
Urbana, IL, USA  
0000-0002-9336-4756

Abstract—Fight detection in videos is an emerging deep learning application with today’s prevalence of surveillance systems and streaming media. Previous work has largely relied on action recognition techniques to tackle this problem. In this paper, we propose a simple but effective method that solves the task from a new perspective: we design the fight detection model as a composition of an action-aware feature extractor and an anomaly score generator. Also, considering that collecting frame-level labels for videos is too laborious, we design a weakly supervised two-stage training scheme, where we utilize multiple-instance-learning loss calculated on video-level labels to train the score generator, and adopt the self-training technique to further improve its performance. Extensive experiments on a publicly available large-scale dataset, UBI-Fights, demonstrate the effectiveness of our method, and the performance on the dataset exceeds several previous state-of-the-art approaches. Furthermore, we collect a new dataset, VFD-2000, that specializes in video fight detection, with a larger scale and more scenarios than existing datasets. The implementation of our method and the proposed dataset is available at https://github.com/Hepta-Col/VideoFightDetection.

Index Terms—Computer Vision, Weakly Supervised Learning, Self-Training, Video Anomaly Detection, Video Fight Detection

I. INTRODUCTION

Video fight detection (VFD) aims to detect and localize fight events in videos temporally or spatially. With the prevalence of surveillance cameras and streaming media, VFD is getting increasingly important as it serves as an automatic tool to monitor and identify aggressive scenes without human intervention, thus largely reducing the cost. For example, online videos can be automatically checked and tagged with age restrictions if containing violent scenes. Also, surveillance systems equipped with VFD modules can provide a real-time warning to the police and people around.

VFD has been investigated for years due to its practical prospect. Early studies on VFD [1]–[3] largely rely on Action Recognition techniques, since human fights can be intuitively considered as a subset of human actions. These approaches usually use a video encoder to obtain action-aware features and then apply a classifier to the features to decide whether normal or not. Therefore, the encoders are usually trained on very short video clips [4], [5] due to the device memory limitation on one batch of videos. Although these models show good performance in terms of various binary classification metrics, they ignore the temporal dependencies between time periods of an event happening in real life.

Recently, with the development of Video Anomaly Detection (VAD) methods, some researchers tried treating VFD as a subtask of VAD [6], [7]. In this sense, VFD models are trained to learn the temporal context information of videos, therefore they are able to detect anomalies precisely even if given long videos. But the training process shares the same challenge with VAD that it is hard for models to fully learn the inherent discrepancies between normal videos and abnormal videos. Moreover, fight events differ from general video anomalies in that they usually happen in short periods, which poses an even greater challenge for the model to detect them.

Considering these aspects of VFD, our idea to construct a model and design a training scheme is two-fold: first, we take into account action recognition and anomaly detection together; second, we train the model on weak labels, so that collecting training data would not require too much human labor. To this end, we develop the model as a composition of a feature extractor head and an anomaly score generator tail; next, we utilize a two-stage training framework, where multiple-instance-learning loss is applied to train the score generator from scratch and pseudo labels are generated to supervise another training iteration of it. Experiments show that our model outperforms the current video anomaly detection state-of-the-art (SOTA) models on the UBI-Fights dataset by reaching AUROC of 93.1%, which is 1.2 points higher than current SOTA models reported in the literature.

Existing fight detection datasets are limited in size and scenarios. To address this issue, we construct VFD-2000, a...
large-scale, multi-scenario dataset with both video-level and
frame-level labels. Our dataset has the following advantages
over others: i) it contains fight behaviors in multiple scenarios,
thus reducing the effect of background in fight detection;
ii) it contains the corresponding non-fight videos under the
same scene as fight videos, thus eliminating the background
differences between fight and non-fight videos; iii) it includes
clips that are showing ambiguous behaviors between fight and
non-fight; iv) it contains both short and long, vertical- and
horizontal-view videos for video-level weakly supervised and
supervised training.

The main contributions of our work are:

- We propose an effective deep learning model specifically
developed for video fight detection.
- We propose a two-stage training framework to train the
fight detection model from coarse to fine in a weakly-
supervised manner.
- Our experimental results on the UBI-Fights dataset out-
perform current SOTA models.
- We collect a large-scale, multi-scenario dataset with both
video-level and frame-level labels made specifically for
video fight detection tasks.
- We conduct several benchmark tests on our proposed
VFD-2000 and report SOTA results.

II. RELATED WORK

A. Fight Detection

Fight detection in videos is usually considered a binary
classification task. A large body of work is based on a video
feature extractor followed by a classifier that outputs fight or
no fight label for each video. The features of interest typically
contain spatial-temporal and/or motion information. Zhang et
al. [8] use the Gaussian Model of Optical Flow to extract
candidate violence region, and then the Orientation Histogram
of Optical Flow is applied to classify the violent behavior.
Ditsanthia et al. [9] propose an approach that utilizes the
convolutional neural network (CNN) as a feature extractor,
and also present a new kind of visual feature descriptors
to describe visual information in a video frame. Ullah et
al. [10] implement a CNN-based model to recognize objects
and remove unwanted frames. Other works also utilize action
recognition techniques to solve fight detection tasks. Accatoli
et al. [1] apply a pre-trained 3D CNN to extract motion
features. The authors use these descriptors as an input for a
linear SVM to classify videos as violent or non-violent. Li et
al. [2] combine both the RGB-based model and the optical
flow Graph Model, and feed these two kinds of images into a
3D-CNN model to get the final results. Kang et al. [3] propose
a pipeline that combines spatial attention with the 2D-CNN
model to extract features and use a Temporal Squeeze-and-
Excitation block to highlight the periods that might have fight
events. Recently, Choquelelque-Roman et al. [11] utilize a Fast-
RCNN-based architecture to realize violence detection in a
weakly supervised manner, with modules designed especially
for human actions and dynamics. However, we notice that few
researchers except Tan et al. [7] and Degardin et al. [6] treat
fight detection as an anomaly detection task.

B. General Video Anomaly Detection

Given a video, a VAD model is supposed to localize the
anomaly time span(s). Here anomaly is defined as a deviation
from what is typically expected based on the majority of the
video frames. However, obtaining frame-level annotations is
usually too laborious. Hence, most studies on VAD focus
on unsupervised or weakly supervised methods. Unsupervised
methods [12]–[15] usually train models using purely normal
data, and the model is expected to recognize the distinctive
data as abnormal data. For weakly supervised methods [16]–
[20], the video-level labels (normal/abnormal) are available. In
this scenario, the task is dubbed as Weakly Supervised Video
Anomaly Detection (WSVAD).

Past work on WSVAD has tried several ways to leverage
these weak labels. Sultani et al. [16] first propose multiple-
instance-learning (MIL), where each video is treated as a
video segment bag and the model is trained to maximize
the separability between the normal bags and abnormal bags.
However, this method suffers from noisy scores and the inability
to consider the temporal relation between clips. Motivated
by these drawbacks, later studies concentrate on improving the
MIL-based training scheme. Zhong et al. [17] treat WSVAD
as a supervised learning task with noisy labels, and use a
GCN-based model to clear the noisy labels; Lv et al. [18]
propose a high-order context encoding model to utilize the
temporal context, and apply a new strategy to facilitate noise
inference; RTFM model proposed by Tian et al. [19] uses
a multi-scale temporal network (MTN) to encode both long-
range and short-range temporal relations between video clips,
greatly improving the robustness of MIL-based model training.
And in MIST [20], a modified MIL loss is used to train a
score generator, which then outputs clip-level pseudo labels
to supervise the training of a self-guided attention module.

C. Self-training

Self-training has been investigated extensively in semi-
supervised learning. Much of the past work on self-training
focused on pseudo-labeling strategies. At each iteration, self-
training selects a portion of the unlabeled training data for
pseudo-labeling. An early approach sets the selecting threshold
to the average of respectively positive and negative predictions
[21]. Lee et al. [22] leverage neural networks as a supervised
classifier and pick up the class with the maximum predicted
probability to infer pseudo-labels. In more recent work, Iscen et
al. [23] implement pseudo-labeling through graph-based
label propagation. They make predictions on the entire dataset
by employing a transductive label propagation method and use
the predictions to generate pseudo-labels.

Besides leveraging single classifiers in self-training algo-
rithms, some studies have proposed to utilize two classifiers,
where each model learns on the output of the other. This
method is also called co-teaching, in which two classifiers
successively switch roles to assign pseudo-labels for the other

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
to train. Xie et al. [24] show the simplicity and effectiveness of
teaching to leverage unlabeled data. They propose a model
with manually injected noises in the student model to learn
beyond the teacher's knowledge. Chen et al. [25] propose to
use cross pseudo supervision to regularize and enforce the
consistency between the two segmentation networks.

III. METHODS

A. Problem Formulation

We consider Video Fight Detection as an intersection of
Action Recognition and Video Anomaly Detection. Different
from most of the previous works that treat VFD as a classifi-
cation task, we formulate it as a regression task that relies
on scores to define the output. Plus, since the annotations
are video-level binary labels, i.e. normal (does not contain
fight event) and abnormal (contains at least one fight event),
our problem formulation largely follows the existing WSV AD
work [16]–[20]: given a video \( V = \{ v_i \}_{i=1}^{N} \) consisting of \( N \) clips,
each of which has \( T \) frames, along with a video-level label \( Y \in \{ 0, 1 \} \)
indicating whether the video contains any fight event or not, the model is trained and is expected to
calculate anomaly scores \( S = \{ s_i \}_{i=1}^{N} \) for an unseen video,
with each clip \( v_i \) giving a score \( s_i \in [0, 1] \) indicating the
probability of a fight event occurring.

B. Video Fight Detection Model

To combine action recognition and anomaly detection tech-
niques, we propose a fight detection model composed of a
feature extractor head \( f \) and an anomaly score generator tail
\( g \), as shown in Fig. 1. The feature extractor encodes the video
with special attention to human actions: accepting a video
\( V = \{ v_i \}_{i=1}^{N} \) broken up into \( N \) clips, it outputs features
\( X = \{ x_i \}_{i=1}^{N} \) corresponding to all clips, where:

\[
x_i = f(v_i).
\]

Then the score generator is supposed to rate the clips in terms
of their relevance to fight: it takes the extracted features as
input and calculates the anomaly scores \( S = \{ s_i \}_{i=1}^{N} \) for each
clip of the video:

\[
S = g(X),
\]

where the score of every clip is calculated according to the
features \( X \) of the whole group of clips, whereby temporal
relation between clips is injected into each score.

Fig. 1. Workflow and structure of the fight detection model.

C. Two-stage Training Framework

Next, to train such a model on weak labels, we design a two-
stage training framework as shown in Fig. 2. To distinguish
between these two stages, we name the first generator as
network A and the second as network B, while keeping the
feature extractor unchanged. Theoretically, this gives rise to
two scenarios: A equals B, and A differs from B. We compare
these two scenarios in Section IV.

1) Stage One: Score Generator Training via Multiple In-
stance Learning: To train the anomaly score generator A from
scratch, we follow previous WSVAD work [16]–[20] to use an
MIL ranking loss as constraint that maximizes the separability
between normal and abnormal videos. Given a pair of normal
video \( V^n = \{ v^n_i \}_{i=1}^{N} \) and abnormal video \( V^a = \{ v^a_i \}_{i=1}^{N} \)
(“\( n \)” means “abnormal” and “\( a \)” means “normal”), the feature
extractor \( f \) first encodes the clips into \( D \)-dimensional feature
vectors \( X^n = \{ x^n_i \}_{i=1}^{N} \) and \( X^a = \{ x^a_i \}_{i=1}^{N} \):

\[
x^n_i = f(v^n_i), \quad x^a_i = f(v^a_i),
\]

where \( x_i \in \mathbb{R}^D \). Then generator A calculates scores \( S^n \) and
\( S^a \) from the features:

\[
S^n = g_A(X^n), \quad S^a = g_A(X^a),
\]

where \( S = \{ s_i \}_{i=1}^{N} \) are scalar values between 0 and 1. Using
these scores, an MIL loss function can be defined as:

\[
\mathcal{L}_{MIL} = \max(0, \epsilon - d(S^n, S^a)),
\]

where \( \epsilon \) is a fixed margin, and function \( d \) measures the
distance, or separability, between normal and abnormal videos.
The instantiation of \( d \) depends on the choice of the genera-
or. Following existing WSVAD work [16]–[20], we add the
sparsity and smoothness constraints to the loss. Note that the

Fig. 2. Two-stage training framework. At stage 1, each video is cut into \( N \)
clips (\( N \) is a constant), and Network A is trained with MIL loss (note that if
\( A \neq B \) then \( B \) should also be trained from scratch at stage one); after stage 1 is
done, each video is cut into \( M \) clips so that each has exactly \( T = 32 \) frames
(\( M \) is a variable depending on video length), and pseudo labels for each clip
are generated; at stage 2, the pseudo labels are used to supervise Network
B’s training.
feature extractor is not involved in the parameter updating procedure; and if $A \neq B$, $B$ should also be trained from scratch at stage one, following the same manner as training $A$.

2) Stage Two: Score Generator Refining with Pseudo Labels: Inspired by previous self-training methods in the field of semi-supervised learning [20], [24], [25], we apply a self-training scheme at stage two. After the first stage is done, we break every video in the training set into $M$ equal-length clips $V = \{v_i\}_{i=1}^M$ (here $M$ is a variable depending on every video length) and pass them through the model again:

$$\hat{x}_i^n = f(\hat{v}_i^n), \quad \hat{x}_i^a = f(\hat{v}_i^a),$$

(6)

$$\hat{S}_i^n = g_A(\hat{X}_i^n), \quad \hat{S}_i^a = g_A(\hat{X}_i^a),$$

(7)

$$\hat{Y} = h(\hat{S}),$$

(8)

where $\hat{X} = \{\hat{x}_i\}_{i=1}^M$, $h$ is a transformation of raw scores $\hat{S} = \{\hat{s}_i\}_{i=1}^M$, and $\hat{Y} = \{\hat{y}_i\}_{i=1}^M (\hat{y}_i \in [0, 1])$ are the generated pseudo labels. Note that we use soft pseudo labels here to prevent over-confidence, and pseudo labels are generated only for abnormal videos since we can safely assign 0 to all clips of any normal video. Then following the same workflow as $A$, the generator $B$ generates another set of scores $S'$ by:

$$S'^n = g_B(\hat{X}_i^n), \quad S'^a = g_B(\hat{X}_i^a),$$

(9)

and a cross-entropy loss is applied to the pseudo labels $\hat{Y}^*$ and scores $S'$. After two stages are done, only generator $B$ is used for inference.

The motivation for applying a two-stage training scheme here is two-fold: First, the transition from video-level learning to clip-level learning is a coarse-to-fine learning procedure, which helps the model focus on increasingly shorter time spans. At stage one any video would be broken up into a fixed number of clips regardless of its original length ($N$ clips for each video), therefore the scores assigned are coarse-grained, i.e. a long span of frames shares a single score; at stage two, however, the videos are cut up in a way that each clip has identical length $T$ ($T$ is far less than each video length), so more clips will be generated while each clip has a smaller number of frames, which means the scores generated are fine-grained, i.e. scores are assigned to shorter spans of frames. Second, the model gets stronger by learning from faithful predictions generated by itself. If the model is trained powerful enough on the video-level supervision, we can predict that through learning on the high-quality clip-level pseudo labels generated by itself, the model could perform even better in generating anomaly scores.

D. Dataset Collection

1) Existing Fight Detection Datasets: So far, several datasets have been used in fight detection. The videos in those datasets are given video-level or frame-level labels (‘Fight’ and ‘NonFight’). Those datasets have several drawbacks such as fixed video length, small scale, limited scenes, etc. For example, the small-scale problem appears in the movie dataset [26] and surveillance camera fight dataset [27]. They contain 200 videos and 300 videos, respectively, and each video in these two datasets lasts for about 1 to 2 seconds. The hockey dataset [26] is limited in the variety of scenes. It’s sampled based on ice hockey games videos and does not include fight behaviors in other real-life situations. The CCTV-Fight dataset [28] only includes fight videos and does not provide non-fight samples. The UCF-Crime dataset [16] is large-scale with 1900 long videos where each video is longer than 60 seconds, but it’s not specifically focused on fight detection. The RLV [5] and RWF [4] dataset both contain 2000 videos which are large in number, but they only contain short video clips where the length of videos is in the range of 3 to 7 seconds. The UBI-Fights dataset [29] contains 1000 videos with a total of 80 hours and is fully annotated at the frame level. It includes videos from different recording angles such as moving, rotating, and stationary cameras. We will use it as a comparison dataset in our experiments.

2) Data Collection: We propose VFD-2000, a video fight detection dataset containing more than 2000 videos. We use YouTube as the data source. We search for specific scenarios and use “fight” as a search keyword, for example, “street fight”, “beach fight”, and “violence in the restaurant”. We collect 200 videos under 20 different scenes. Some of those videos are collections of very short fight videos, and most of the videos are recorded by smartphone or moving camera device, therefore the scenes in those videos vary significantly. In order to keep the scene in the video to be more consistent, we use the PySceneDetect package [30] to cut the video into the scene clips. We set the threshold of the scene detector to be 30 and min clip length to 2 seconds. After the clipping, we get about 3500 video clips with a time length from 2 seconds to 3 minutes. But there are still many unnecessary clips containing video introductions, transition animations, or black screen, having shaky video angles or blurred picture quality, or some special filters added by the video producer. To eliminate those clips we manually examine the videos and add the video-level labels to each clip. After the filtering, there are 2490 video clips left: 1296 of them are fight videos, and 1194 are non-fight videos. We then divide our dataset into four categories: long and short video in vertical and horizontal views. Those longer than 96 frames will be counted as long videos, while others will be viewed as short videos. The horizontal view refers to those having their width longer than the height. The ratio of vertical and horizontal videos in our dataset is about 1:2, and we also keep this ratio when splitting the training and testing data. Fig. 3 shows a few example frames for different scenes. Table I shows the four categories and corresponding number of videos. We also provide the distribution of length of videos for our dataset in Fig. 4, and compare our dataset with other datasets in Table II.

3) Data annotation: For the annotation, we manually add video-level labels for each video. For long videos, we manually add frame-level labels to those fight videos in test data to test our model. For short videos, we use the video-level label in training and testing. We save the predicted results into a JSON file in a dictionary format with each video containing the predicted results by our model, scene labels as well as the
portion of new features of each video are passed to the VFD-2000 short video.

Our dataset includes 20 different scenes, Fig. 4 gives detailed description. We split our dataset into long and short videos by number of video frames.

1.6~1.96 360x288 video-level Hockey
1000 1~600 640x360 both real life
1000 5~720 variable frame-level CCTV
2000 3~7 variable video-level real life
200 1.6~2 720x480 video-level movies

#Videos Length(sec) Resolution Annotation Scenario

| Name          | Videos | Length/sec | Resolution | Annotation | Scenario |
|---------------|--------|------------|------------|------------|----------|
| CCTV-Fights [25] | 1000   | 3.2/2      | variable   | Hockey     | CCTV     |
| Hockey Fight [26] | 1000   | 1.6/1.96   | 360x288    | video-level| Hockey   |
| Movies Fight [26] | 200    | 1.6/2      | 720x480    | video-level| movies   |
| UCF-Crime [16]   | 1900   | 60/600     | variable   | surveillance| surveillance |
| UBI-Fights [29]  | 1000   | 1/600      | 640x360    | both       | real life |
| RLV [5]          | 2000   | 3.7        | variable   | video-level| real life |
| RWF [4]          | 2000   | 5          | variable   | video-level| surveillance |
| Surveillance [27]| 300    | 2          | variable   | video-level| surveillance |

Ours\(^a\) 938(both) 3.2 variable video-level multiple\(^b\)
Ours\(^b\) 1552(long) 3.2 208 variable both multiple\(^b\)

\(^a\) We split our dataset into long and short videos by number of video frames.
\(^b\) Our dataset includes 20 different scenes, Fig. 4 gives detailed description.

### IV. Results and Discussion

#### A. Implementation Details

We choose I3D model [31] pretrained on Kinetic400 [32] as the action-aware feature extractor. Before detection model training, we first pre-process the videos by converting raw mp4 format to npy format and resizing them to 224×224 pixels. Then we cut the videos into clips with \( T = 32 \) frames per clip, which are then passed through the I3D model one by one. The clip-level features are extracted from “mix_5c” layer of I3D (\( D = 20/48 \)), grouped by video name, and stored as npy files.

In the first stage, we follow Sultani et al. [16] to cut each video into \( N = 32 \) clips, regardless of its original length. To realize this, for each video we average every \( \frac{T}{N} \) portion of clip features of the whole span to get new \( N \) clip features. Then these \( N \) new features of each video are passed to the detection model as a whole. In the second stage, clip features of each video are directly passed through the detection model without ensuring that the number of clip features is constant (since videos have different lengths).

As mentioned in Section III, we have tried different implementations of generator A and B (details are shown in Part D). We experiment with Multi-scale Temporal Network (MTN), a dilated convolution-based module proposed by RTFM [19] (current SOTA model on WSVD task), and MLP, a simple regressor consisting of 3 layers with the number of units being 512, 32, and 1. During first-stage training, both MTN and MLP are trained for 10000 epochs with Adam optimizer (learning rate = 0.001). Then for self-training, MTN is trained for 50 epochs with Adam optimizer (learning rate = 0.00001) while MLP is trained for 200 epochs with Adam optimizer (learning rate = 0.001). The score transformation, \( h \), is instantiated by identical mapping and min-max normalization.

The entire pipeline is implemented in PyTorch, and the calculations are carried out on HAL system [33].

#### B. Datasets and Metrics

We conduct experiments mainly on UBI-Fights and our VFD-2000. UBI-Fights is a large-scale video anomaly detection dataset especially focusing on fights, which contains 80 hours of video fully annotated at the frame levels. Of its total 1000 videos, 784 of them are shot from normal daily life while 216 of them contain a fight event. All videos are resized to be 640x360 pixels, with the frame rate set to 30 fps. Importantly, unnecessary video segments, such as video introductions and text overlays that could disturb the learning process are removed.

Following previous work, we test our model in terms of Area Under the Receiver Operating Characteristic (AUROC) and Equal Error Rate (EER). AUROC is a widely used performance metric in classification tasks, which tells about the models ability to discriminate between positive examples and negative examples. It is calculated as the area under the ROC curve where the false positive rate and false positive rate across different decision thresholds. EER is the location on a ROC curve where the false positive rate and false negative rate are equal. Generally speaking, the lower the EER is, the higher accuracy the model can achieve.

#### C. Comparison with SOTA Models

We compare our model’s performance with previous SOTA video anomaly detection and human action recognition models on UBI-Fights and our VFD-2000. For fairness, we present experimental results for both AUROC and EER.

1) On UBI-Fights Dataset: Results of UBI-Fights are shown in Table III, showing that our method surpasses all the previous approaches. For AUROC, our method reaches 93.1%, which is at least 1.2 points higher than current SOTA models. It is worth mentioning that the previous best performance (AUROC=91.9%) produced by Tan et al. [7] uses 10-crop video preprocessing for model training, which is very time- and space-consuming, while our model is trained with just...
Fig. 3. VFD-2000 consists of outdoor and indoor videos recorded by both stationary and moving cameras.

Fig. 4. The left bar chart shows the distribution of our dataset by number of frames. The right bar chart shows the number of videos for each scene, among all these scenario street fight account for the majority of videos.

1-crop. Meanwhile, the EER of our model predictions falls below 10% for the first time.

2) On the Collected VFD-2000 Dataset: We also conduct experiments on the proposed VFD-2000, as shown in Table IV. We compare our model with previous video anomaly detection, human action recognition, and violence detection models trained by us on our dataset. On VFD-2000-short, we train C3D [34] and TimeSformer [35] with video-level labels based on MMAction2 [36] framework. We also trained the ESTMM [3] pipeline. Among those models, the TimeSformer model reaches the highest accuracy at 79.79%. Note that since WSVAD methods are designed for detecting anomalies in long videos, we did not run our model or any WSVAD-based model on VFD-2000-short. On VFD-2000-long, where we compare WSVAD methods, our model exceeds RTFM, the current SOTA model on WSVAD, by 1.49 points in terms of AUROC, and decreases EER to 40.55%.

D. Ablation Study

Here we perform ablation studies to reveal the influences on the result brought by different model implementations. All the experiments are conducted on UBI-Fights.

1) Two-stage Framework: To verify the effectiveness of the two-stage framework, we implement both one-stage and two-stage schemes to see whether two-stage training improves performance. As shown in Table V, compared with the one-stage framework, both MLP and MTN achieve higher AUROC scores under two-stage training. Specifically, MLP’s performance increases by 1.30 points, and that of MTN increases by 2.71 points.

2) Score Generator Choice: Here we compare different combinations of score generators, as Table VI shows. Note

| Method | Supervised | AUROC (%) | EER (%) | Acc (%) |
|--------|------------|-----------|---------|---------|
| Hasan et al. [12] | Un. | 52.8 | 46.6 | |
| Adv. Generator [37] | Un. | 53.3 | 48.4 | |
| LSTM-AE [38] | Weak | 54.1 | 48.0 | |
| s2-VAE [39] | Un. | 61.0 | 42.7 | |
| Degardin et al. [6] | Weak | 84.6 | 21.6 | |
| Sultani et al. [16] | Weak | 89.2 | 18.6 | |
| RTFM* [19] | Weak | 90.4 | 12.6 | |
| GMM [40] | Weak | 90.6 | 16.0 | |
| Tan et al. † [7] | Weak | 91.9 | - | |
| Our model** | Weak | 93.1 | 8.5 | |

*Using 1-crop. †Using 10-crop.

---

| Method | Supervised | AUROC (%) | EER (%) |
|--------|------------|-----------|---------|
| C3D [34] | Weak | 83.88 | 26.56 | 73.40 |
| TimeSformer [35] | Weak | 82.19 | 25.60 | 79.79 |
| ESTMM [3] | Weak | 78.30 | 33.01 | 73.40 |

---

| Split | Method | AUROC (%) | EER (%) | Acc (%) |
|-------|--------|-----------|---------|---------|
| short | C3D [34] | 83.88 | 26.56 | 73.40 |
| | TimeSformer [35] | 82.19 | 25.60 | 79.79 |
| | ESTMM [3] | 78.30 | 33.01 | 73.40 |
| long  | RTFM [19] | 61.20 | 41.44 | 69.01 |
| | Ours (MLP+MLP) | 53.00 | 46.39 | 22.57 |
| | Ours (MTN+MTN) | 62.69 | 40.55 | 72.75 |

*Models predict video-level labels for short videos
†Models predict frame-level labels for long videos

---

| # Stages | Models | AUROC (%) | ∆(%) |
|----------|--------|-----------|------|
| One-stage (baseline) | MLP | 80.51 | - |
| Two-stage | MLP+MLP | 81.81 | +1.30 |
| One-stage (baseline) | MTN | 90.36 | - |
| Two-stage | MTN+MTN | 93.07 | +2.71 |

---

Table III: Experimental results on UBI-Fights

Table IV: Experimental results on VFD-2000

Table V: Comparisons between one-stage and two-stage frameworks on UBI-Fights

---
We have also tried a

We select four representative examples each for fight videos (upper row) and non-fight videos (lower row) in Fig. 5. The red curves stand for model predictions (anomaly scores), and the green curves are ground truth. The rising and dropping trends of the prediction curve fit well with the fluctuation of the ground truth curve. Also, it can be seen that the scores for fight videos are hovering at a high level, while that for non-fight videos are repressed at a low level. It might seem unsatisfactory to some extent that the absolute values of scores in fight videos lack discrimination between normal and abnormal clips, but this is a reasonable result since we use the smoothness loss on abnormal training samples, which exactly imposes restrictions on the fluctuation degree of the prediction curves, without which the prediction would become too noisy.

3) Generalization of Proposed Model: Theoretically, if trained on different datasets, our model could be applied to any specific anomaly action detection task, such as detecting burglary, robbery, shooting, shoplifting, and arson. Plus, the feature extractor and the score generator could be replaced by other advanced models. For example, C3D [34] could be a substitution for I3D.

V. CONCLUSION

In this work, we propose an effective model for video fight detection. To train such a model on weak labels and help it learn to focus on increasingly shorter time spans, we use a two-stage training framework to train the model from coarse to fine. Experimental results on the UBI-Fights dataset reveal that our model outperforms current SOTA models. Additionally, to better facilitate research on fight detection in videos, we collect a large-scale, multi-scenario dataset with both video-level and frame-level labels specifically for video fight detection tasks. Our experimental results, code, and dataset is available at https://github.com/Hepta-Col/VideoFightDetection.

ACKNOWLEDGMENT

This work utilized resources supported by the National Science Foundations Major Research Instrumentation program,
grant #1725729, as well as the University of Illinois at Urbana-Champaign.

REFERENCES

[1] S. Accattoli, P. Semani, N. Falcionelli, D. Mekuria, and A. F. Dragoni, “Violence detection in videos by combining 3d convolutional neural networks and support vector machines,” Applied Artificial Intelligence, pp. 1–16, 02 2020.

[2] C. Li, L. Zhu, D. Zhu, J. Chen, Z. Pan, X. Li, and B. Wang, “End-to-end multiplayer violence detection based on deep 3d cnn,” 12 2018, pp. 227–230.

[3] M.-S. Kang, R.-H. Park, and H.-M. Park, “Efficient spatio-temporal modeling methods for real-time violence recognition,” IEEE Access, vol. 9, pp. 76 270–76 285, 2021.

[4] M. Cheng, K. Cai, and M. Li, “Rwf-2000: An open large scale video database for violence detection,” in 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 4183–4190.

[5] M. Soliman, M. Kamal, M. Nashed, Y. Mostafa, B. Chawky, and D. Khattab, “Violence recognition from videos using deep learning techniques,” 12 2019, pp. 80–85.

[6] B. Degardin and H. Proença, “Iterative weak/self-supervised classification framework for abnormal events detection,” Pattern Recognition Letters, vol. 145, pp. 50–57, 2021.

[7] W. Tan and J. Liu, “Detection of fights in videos: A comparison study of anomaly detection and action recognition,” arXiv preprint arXiv:2205.11394, 2022.

[8] T. Zhang, Z. Yang, W. Jia, B. Yang, J. Yang, and X. He, “A new method for violence detection in surveillance scenes,” Multimedia Tools and Applications, vol. 75, 05 2015.

[9] E. Ditsanthia, L. Pipanmaekaporn, and S. Kamonsantiroj, “Video representation learning for cctv-based violence detection,” 12 2018, pp. 1–5.

[10] F. U. M. Ullah, A. Ullah, M. Muhammad, I. Haq, and S. Baik, “Violence detection using spatiotemporal features with 3d convolutional neural network,” Sensors, vol. 19, p. 2472, 05 2019.

[11] D. Choquelouque-Roman and G. Camara-Chavez, “Weakly supervised violence detection in surveillance video,” Sensors, vol. 22, no. 12, p. 4502, 2022.

[12] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 733–742.

[13] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. S. M. Goh, “Anomalynet: Anomaly detection network for video surveillance,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 10, pp. 2537–2550, 2019.

[14] R. Rodrigues, N. Bhargava, R. Velmurugan, and S. Chaudhuri, “Multitimescale trajectory prediction for abnormal human activity detection,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2626–2634.

[15] R. Cai, H. Zhang, W. Liu, S. Gao, and Z. Hao, “Appearance-motion memory consistency network for video anomaly detection,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 2, 2021, pp. 938–946.

[16] W. Sultanii, C. Chen, and M. Shah, “Real-world anomaly detection in surveillance videos,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6479–6488.

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

[18] H. Lv, C. Zhou, Z. Cui, C. Xu, Y. Li, and J. Yang, “Localizing anomalies from weakly-labeled videos,” IEEE transactions on image processing, vol. 30, pp. 4505–4515, 2021.

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

[20] J.-C. Feng, F.-T. Hong, and W.-S. Zheng, “Mist: Multiple instance self-training framework for video anomaly detection,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 14 009–14 018.

[21] G. Tur, D. Hakkani-Tür, and R. E. Schapire, “Combining active and semi-supervised learning for spoken language understanding,” Speech Communication, vol. 45, no. 2, pp. 171–186, 2005.

[22] D.-H. Lee et al., “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in Workshop on challenges in representation learning, ICMIL, vol. 3, no. 2, 2013, p. 896.

[23] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, “Label propagation for deep semi-supervised learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5070–5079.

[24] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10 687–10 698.
[25] X. Chen, Y. Yuan, G. Zeng, and J. Wang, “Semi-supervised semantic segmentation with cross pseudo supervision,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2613–2622.

[26] E. Bermejo, O. Deniz, G. Bueno, and R. Sukthankar, “Violence detection in video using computer vision techniques,” 01 2011, pp. 332–339.

[27] S. Akh, G. A. Tatarolu, and H. K. Ekenel, “Vision-based fight detection from surveillance cameras,” in *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2019, pp. 1–6.

[28] M. Perez, A. C. Kot, and A. Rocha, “Detection of real-world fights in surveillance videos,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2662–2666.

[29] B. Degardin and H. Proença, “Human activity analysis: Iterative weak/self-supervised learning frameworks for detecting abnormal events,” in *2020 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, pp. 1–7.

[30] Brandon Castellano, “Video Scene Cut Detection and Analysis Tool,” 2014-2022. [Online]. Available: https://github.com/Breakthrough/PySceneDetect/

[31] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

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

[33] V. Kindratenko, D. Mu, Y. Zhan, J. Maloney, S. Hashemi, B. Rabe, K. Xu, R. Campbell, J. Peng, and W. Gropp, “Hal: Computer system for scalable deep learning,” 07 2020, pp. 41–48.

[34] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, “Abnormal event detection in videos using generative adversarial nets,” in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 1577–1581.

[35] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?” *CoRR*, vol. abs/2102.05095, 2021. [Online]. Available: https://arxiv.org/abs/2102.05095

[36] MMAction2 Contributors, “OpenMMLab’s Next Generation Video Understanding Toolbox and Benchmark,” 7 2020. [Online]. Available: https://github.com/open-mmlab/mmaction2

[37] B. Degardin, “Weakly and partially supervised learning frameworks for anomaly detection,” Ph.D. dissertation, Universidade da Beira Interior (Portugal), 2020.