Multimedia Systems (2025) 31:127
https://doi.org/10.1007/s00530-025-01706-8

REGULAR PAPER

Few-shot fine-tuning with auxiliary tasks for video anomaly detection

Jing Lv[1] · Zhi Liu[1,2] · Gongyang Li[1,2,3]

Received: 16 August 2024 / Accepted: 31 January 2025

Abstract
Anomaly detection in surveillance videos aims to identify video frames that exhibit unexpected behavior. Most existing methods follow an unsupervised setup, training with normal videos and testing with videos from the same scene. However, in real-world deployments, the performance of existing models significantly degrades when faced with unseen scenes. To address this issue, we introduce the auxiliary tasks of segmentation and optical flow estimation into the fine-tuning process, proposing a novel Segmentation and Optical Flow Fine-tuning (SOFF) framework. This framework enables the existing models to adapt to new scenes with only a few samples for fine-tuning. To integrate these auxiliary tasks, we design a Segmentation and Flow Output Network (SFO-Net). SFO-Net enhances fine-tuning performance in unseen scenes by extracting rich shape and motion information through the execution of auxiliary tasks during the fine-tuning process. Additionally, SFO-Net can be flexibly cascaded with existing models that output images to form the SOFF framework. Experiments on multiple datasets demonstrate that our framework improves the performance of existing models when faced unseen scenes through few-shot scenes fine-tuning and achieves competitive performance.

Keywords Video anomaly detection · Fine-tuning · Segmentation · Optical flow

1 Introduction

Video anomaly detection is an important task in the field of computer vision, involving the identification of abnormal behaviors or events in video clips. Given its critical applications in public safety, especially in surveillance videos, this area has received considerable attention in recent years. In practical applications, abnormal events occur much less frequently, leading to a scarcity of abnormal samples. This imbalance between the number of normal and abnormal samples poses significant challenges for dataset creation and model training. Therefore, most previous approaches [1–8] addressed this issue through unsupervised learning techniques, where models are trained using only normal videos and then tested on the test set of the same dataset. However, when the trained anomaly detection models are faced with new scenes, their performance tends to decline sharply. To address this issue, in this paper, we focus on enhancing the model's cross-dataset video anomaly detection capabilities.

Early research in video anomaly detection primarily relied on handcrafted features for identifying anomalies. These features included motion trajectories [9], object speed and size [10], histograms of optical flow [11], histograms of oriented gradients (HOG) [12], SIFT [13], sparse feature points [14], and mixtures of dynamic textures (MDTs) [15]. Subsequently, deep learning was applied to video anomaly detection tasks, achieving significant results. Techniques such as the autoencoder [2, 3], generative adversarial network (GAN) [1], and diffusion model [4] were utilized. These methods utilized deep learning to extract features, providing more robust representations compared to handcrafted features and achieved better detection performance. However, they were all trained and tested on the same dataset, without considering that in real-world deployment, the model

Communicated by Teng Li.

✉ Zhi Liu
liuzhisjtu@163.com

✉ Gongyang Li
ligongyang@shu.edu.cn

Jing Lv
lvjing@shu.edu.cn

[1] School of Communication and Information Engineering, Shanghai University, Shanghai, China

[2] Wenzhou Institute of Shanghai University, Wenzhou, China

[3] Yunnan Key Laboratory of Service Computing, Yunnan University of Finance and Economics, Kunming, China

Published online: 24 February 2025

would encounter new scenes, which could lead to a decline in performance. In recent years, some research explored meta-learning methods [16–19], which enabled models to adapt to new scenes with only a few samples. These methods could alleviate the challenge of cross-dataset setting, i.e., differing training and testing environments, which was crucial for real-world deployment. However, these methods often required specific training setups or specially designed models, limiting their broad applicability.

To address the performance degradation of traditional unsupervised methods in new scenes and the limited applicability of meta-learning methods, in this paper, we present a novel framework called Segmentation and Optical Flow Fine-tuning (SOFF) for few-shot and scene-adaptive anomaly detection. Unlike existing methods, SOFF doesn't need special training or model changes and works with existing models that output images. SOFF adapts an existing anomaly detection model to new scenes with just a few training samples. As depicted in Fig. 1, video anomaly detection can be performed through future frame prediction. The model is trained on normal data to predict future frames. During testing, a substantial discrepancy between the predicted and actual frames signals an anomaly, enabling the distinction between normal and anomalous frames. In Fig. 1, the dashed box represents the standard fine-tuning process, while

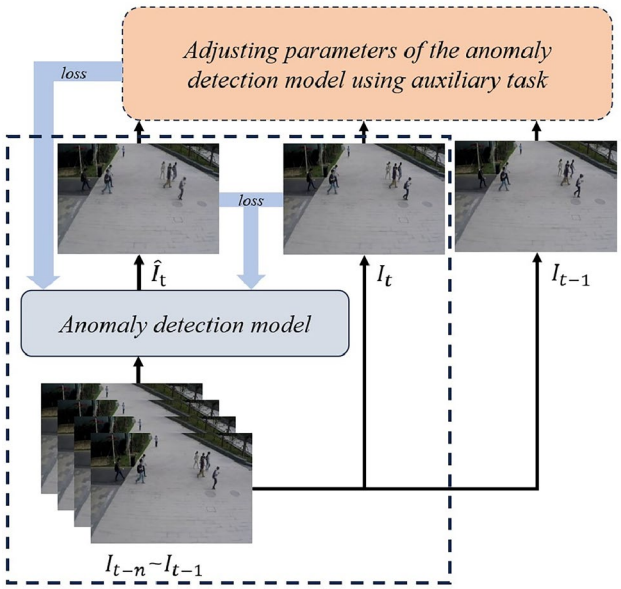

Fig. 1 Overview of the proposed Segmentation and Optical Flow Fine-tuning (SOFF) framework. During the fine-tuning process, the existing anomaly detection model generates a future predicted frame. Using both the predicted future frame and the actual future frame, the model generates pairs of predicted segmentation and optical flow maps as well as corresponding ground truth segmentation and optical flow maps, all based on the current ground truth frame. The model parameters are updated by computing losses from the segmentation maps, optical flow maps, and images

the SOFF framework extends this process. Typically, the standard fine-tuning involves updating an existing model's parameters by minimizing the loss between predicted and actual images. While effective with ample data, this standard approach can overfit when samples are scarce. Specifically, SOFF incorporates two additional auxiliary tasks of segmentation and optical flow estimation. These tasks extract more information from the limited data and impose extra constraints during fine-tuning, alleviating overfitting.

In particular, SOFF enhances fine-tuning by introducing temporal and spatial information through two auxiliary tasks, i.e., optical flow estimation and segmentation. The optical flow estimation task captures the motion of objects, while the segmentation task acquires their shapes, focusing on objects critical for anomaly detection, such as vehicles, pedestrians, and related items. To perform these auxiliary tasks, we develop a Segmentation and Flow Output Network (SFO-Net) with three components, including an optical flow branch, a segmentation branch, and an optical flow and segmentation fusion block. We adopt a three-stage training strategy, decomposing the overall training into several relatively independent stages, each focusing on training a specific part of the network. This modular learning strategy effectively reduces the complexity of the training process and ensures stable convergence of each module. Our SOFF framework is particularly suited for the practical deployment of video anomaly detection models, eliminating the need for special modifications to existing models or specific training setups. As long as the model outputs images, our SOFF framework can be applied for few-shot scenes fine-tuning across various anomaly detection models, showcasing exceptional compatibility. To summarize, our contributions are as follows:

- We propose a novel Segmentation and Optical Flow Fine-tuning (SOFF) framework for video anomaly detection that incorporates segmentation and optical flow estimation tasks into the fine-tuning process. By adding these auxiliary tasks, we provide diverse loss supervisions that enhance the model's performance during few-shot fine-tuning.
- We design the Segmentation and Flow Output Network (SFO-Net) to improve fine-tuning within the SOFF framework. SFO-Net is an important part of the SOFF framework, and specifically implements the auxiliary tasks, enhancing fine-tuning performance. It can generate segmentation and optical flow maps. By combining optical flow with segmentation features, SFO-Net improves accuracy, especially for small targets in surveillance videos.
- We conducted comprehensive experiments on multiple challenging datasets, including Shanghai Tech [20], UCF Crime [21], UCSD Ped1 [15], UCSD Ped2 [15],

and CUHK Avenue [22]. Our SOFF framework significantly improves fine-tuning performance for anomaly detection in new scenes with few samples. Specifically, on the UCSD Ped2 dataset, our method pre-trained using the Shanghai Tech dataset outperforms the best method MPN [17] by 0.61% in AUC under the 10-shot setting. While pre-training with the UCF Crime dataset, our method surpasses the best method VADNet [19] by 5.01% in AUC under the 10-shot setting. On the UCSD Ped1 dataset, our method pre-trained using the UCF Crime dataset exceeds the best method rGAN (Meta) [16] by 0.59% under the 10-shot setting. Moreover, in the 1-shot setting, our SOFF framework achieves competitive results across these datasets.

2 Related work

2.1 Intra-dataset video anomaly detection

Video anomaly detection can be categorized into three paradigms based on the learning approach, i.e., unsupervised video anomaly detection [23, 24], weakly supervised video anomaly detection [25, 26], and supervised video anomaly detection [27]. Given the difficulty of obtaining anomalous data in the real world and the high cost of meticulously labeling large-scale datasets, we focus on unsupervised video anomaly detection. Unsupervised video anomaly detection uses proxy tasks, such as frame reconstruction and future frame prediction. During testing, if there is a significant prediction error between the model's output and the corresponding ground truth frame, the frame is classified as an anomaly. Due to the powerful feature representation and generalization capabilities of neural networks, autoencoders commonly used in reconstruction-based methods [2, 3] can sometimes accurately reconstruct anomalous frames as well [28]. This poses a challenge for anomaly detection. An alternative approach is future frame prediction. This method involves predicting the next frame in a sequence given a set of consecutive frames. For anomalous samples, the predicted future frame significantly deviates from the actual future frame, thereby enabling the identification of anomalous frames. Liu et al. [5] employed a UNet [29] architecture as the generator for predicting future frames. They also incorporated a discriminator to form a structure akin to a generative adversarial network (GAN), enhancing the accuracy of the anomaly detection. Lei et al. [6] proposed a UNet model incorporating an attention mechanism with multi-scale feature extraction and a weakly supervised data augmentation network (WSDAN) to improve the accuracy and robustness of video anomaly detection. Ren et al. [7] proposed a two-stream spatio-temporal generative model (TSSTGM) that uses reconstruction error and prediction error to detect anomalous behavior, and also extracts the motion features of objects in the video through optical flow loss. Wang et al. [8] proposed an anomaly detection method that combines a dual-branch autoencoder and a memory module, enhancing the robustness of the autoencoder and improving detection performance through the branches of frame prediction and frame reconstruction.

Our architecture can be applied to methods that are based on reconstruction or prediction and produce image outputs. To demonstrate the effectiveness of our architecture, we applied it to the widely referenced work [5].

2.2 Few-shot inter-dataset video anomaly detection

In the field of few-shot learning, researchers aim to emulate the rapid and flexible learning capabilities exhibited by humans when faced with limited data, enabling quick adaptation and application of acquired knowledge in new contexts [30]. Transfer learning and meta-learning methods are commonly employed in this domain. For instance, Rostami et al. [31] and Tai et al. [32] utilized transfer learning approaches for SAR image classification under few-shot conditions. Yu et al. [33] introduced a novel transfer learning framework for semi-supervised few-shot learning. Ghani et al. [34] applied transfer learning to the task of bioacoustic classification with limited samples. However, while transfer learning is predominantly used for simpler classification tasks, meta-learning methods are more prevalent in tackling the more challenging task of anomaly detection. Meta-learning methods are primarily categorized into three types: optimization-based [16, 17], metric-based [18, 19], and model-based approaches [35]. In relation to video anomaly detection, Lu et al. [16] and Lv et al. [17] adopted an optimization-based meta-learning approach, where meta-training enables the model to learn a universal initialization parameter. This facilitates rapid adaptation to new scenarios by starting from this initialization parameter during practical applications. Meanwhile, Hu et al. [18] and Huang et al. [19] employed a metric-based approach, constructing a prototype network and using various distance functions to compute similarity. This method allows for application in previously unseen scenes without requiring fine-tuning.

In contrast to methods requiring special designs for training processes or models, our framework leverages an auxiliary task network independent of the basic model. The SFO-Net provides optical flow information and detailed object characteristics, facilitating fine-tuning of the model under few-shot conditions and enhancing its applicability across a broader range of scenarios.

3 Method

In this section, we introduce our Segmentation and Optical Flow Fine-tuning (SOFF) framework designed for few-shot scenarios. The goal of SOFF framework is to enhance the performance of existing models on benchmark datasets when tested on new scenes of other datasets.

3.1 SOFF framework

Our SOFF framework extends the traditional fine-tuning approach. Specifically, the standard fine-tuning framework begins with a pre-trained anomaly detection model predicting future frames from past ones. The model then computes loss based on the discrepancy between predicted and actual frames, updating parameters accordingly. Unlike standard methods, our SOFF framework diverges by creating optical flow and segmentation maps to dissect the motion and shape characteristics of objects within the video. It leverages the resulting differences as a loss function to optimize the anomaly detection model's parameters. Our SOFF framework notably improves fine-tuning with a few samples. Figure 2 illustrates the details of the proposed SOFF framework. In our experiments, we applied the SOFF framework to fine-tune two different anomaly detection models. The first is the UNet anomaly detection model from [5], which we refer to as Ours-UNet-3layer. The second is a modified version of Ours-UNet-3layer, created by adding an additional downsampling layer, and is referred to as Ours-UNet-4layer.

We denote a DNN-based anomaly detection model as $f(I_{(t-n)} \sim I_{(t-1)};\theta)$, where θ represents the model's parameters. It takes n consecutive past frames as input to predict the next frame $\hat{I}_t$. This prediction is pivotal, as it, in conjunction with the preceding frame $I_{(t-1)}$, is fed into the SFO-Net to generate optical flow map $f_{(o-\mathrm{g})}$ and segmentation map $f_{(s-\mathrm{g})}$. The SFO-Net, denoted as $g(I_{(t-1)} \sim I_t;\phi)$, has ϕ as the collective representation of its network parameters. To effectively utilize both dynamic and static video information, the SOFF framework compares predicted and actual frames for optical flow and segmentation maps. These differences are used to update the model's parameters. We obtain the real optical flow map $f_{(o-\mathrm{truth})}$ and segmentation map $f_{(s-\mathrm{truth})}$ by applying the same SFO-Net to the real previous frame $I_{(t-1)}$ and future frame I_t. This step replicates the process used for predicted frames but with actual data, providing the real maps $f_{(o-\mathrm{truth})}$ and $f_{(s-\mathrm{truth})}$ as pseudo-labels for loss calculation in the optical flow estimation and segmentation.

To fine-tune our model, we calculate three types of losses by comparing predictions to actual results. First, we compare the predicted future frame $\hat{I}_t$ with the real future frame I_t to calculate the pixel-level prediction loss for the RGB image, denoted as L_p. Then, we compare the predicted optical flow map $f_{(o-\mathrm{g})}$ with the actual optical flow map $f_{(o-\mathrm{truth})}$ to compute the optical flow prediction loss L_o, capturing the motion information. Next, we compare the predicted segmentation map $f_{(s-\mathrm{g})}$ with the real segmentation map $f_{(s-\mathrm{truth})}$ to calculate the segmentation prediction loss L_s, which captures the spatial shape information of the targets. Using back-propagation, we compute gradients for these losses relative to the model's parameters θ. These gradients are crucial for updating the anomaly detection model, especially in unseen scenes with few samples. The combined use of L_o and L_s adds valuable motion and spatial information beyond what L_p provides, enhancing the model's adaptation. Finally, we utilize all three losses L_p, L_o, and L_s simultaneously during fine-tuning. To stabilize and simplify the fine-tuning process, we keep the pre-trained SFO-Net parameters fixed and only update those of the anomaly detection model.

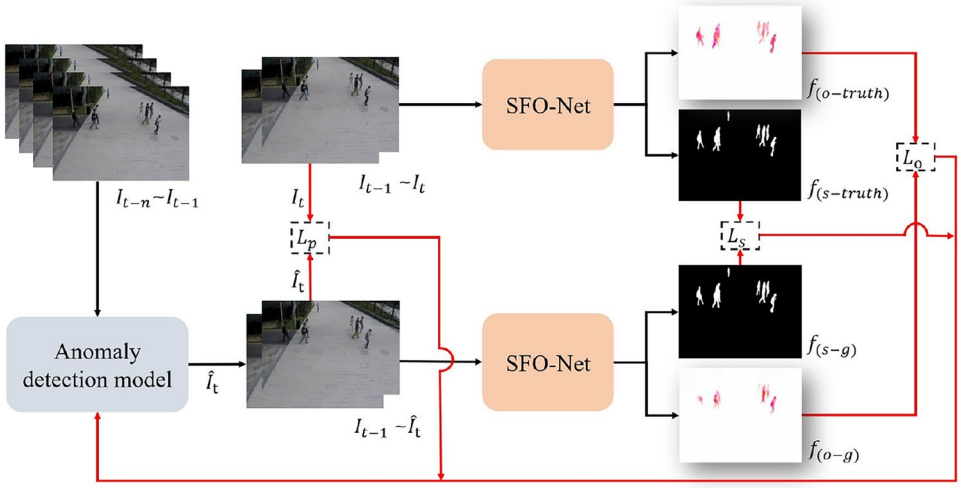

Fig. 2 The details of SOFF framework. To update the parameters of the anomaly detection model, the process follows these steps. First, inputting I_{t-1} and $\hat{I}_t$ into the SFO-Net to generate $f_{(o-\mathrm{g})}$ and $f_{(s-\mathrm{g})}$. Next, inputting I_{t-1} and I_t into the SFO-Net to generate $f_{(o-\mathrm{truth})}$ and $f_{(s-\mathrm{truth})}$, and computing L_p, L_o, and L_s. Finally, updating the model's parameters. Here, black arrows represent the feed-forward inference process, while red arrows indicate the gradient back-propagation

3.2 SFO-net

SFO-Net is an important component of our framework. It generates optical flow and segmentation maps from two consecutive images, providing richer information for fine-tuning the video anomaly detection model. SFO-Net is based on the two-branch network architecture, where one branch is used for generating the optical flow map and another branch is used for generating the segmentation map. Additionally, we incorporate the Optical Flow and Segmentation Fusion (OSF) block, which merges features from the optical flow branch into the segmentation branch to enhance the accuracy of the segmentation map. As shown in Fig. 3, SFO-Net takes two consecutive frames $(f_1,f_2) \in \mathbb{R}^{H\times W\times 6}$ as input, where H and W are the height and width of frames, and 6 represents the combined channels of the two frames. The output includes an optical flow map $f_o \in \mathbb{R}^{H\times W\times 2}$ and a segmentation map $f_s \in \mathbb{R}^{H\times W\times 1}$. The optical flow map f_o has two channels, corresponding to the x-y flow fields that represent the optical flow. The segmentation map f_s has a single channel, indicating the foreground objects and background in the image. The outputs of the SFO-Net, f_o and f_s, depend on the input frames. Depending on the input configuration, f_o represents either $f_{(o-g)}$ or $f_{(o-\text{truth})}$, while f_s represents either $f_{(s-g)}$ or $f_{(s-\text{truth})}$. Specifically, when the input consists of the real previous frame I_{t-1} and the predicted next frame $\hat{I}_t$, the outputs are $f_{(o-g)}$ and $f_{(s-g)}$, generated based on the predicted frame. Conversely, when the input consists of the real previous frame I_{t-1} and the future ground truth frame I_t, the outputs are $f_{(o-\text{truth})}$ and $f_{(s-\text{truth})}$, generated based on the ground truth frame. Through this architecture, SFO-Net effectively extracts and leverages both the motion information and target shape information from the consecutive frames to produce precise optical flow and segmentation maps.

3.2.1 Optical flow branch

The optical flow branch uses convolutional neural networks (i.e., encoder) to process two consecutive frames. Initially, features are extracted through two identical processing streams that share parameters, enhancing efficiency and generalization. These streams then undergo downsampling to capture features at various scales, which are subsequently fused. Following this, the decoder performs stepwise upsampling to restore the fused features to the spatial resolution of the input frames. To maintain effective information flow during training, we implement skip connections between the encoder and decoder. These connections help preserve detailed information that might otherwise be lost during the encoding process. In the decoder, feature fusion operations merge corresponding encoder and decoder features. Feature fusion helps preserve high-resolution details during upsampling, crucial for accurately estimating object edges and texture changes in dynamic scenes.

3.2.2 Segmentation branch

We use an end-to-end training approach to predict segmentation maps, leveraging CNNs for their ability to capture complex patterns. For the segmentation branch, we employ a lightweight variant of the UNet-like network, known for efficient feature extraction and reconstruction.

3.2.3 Optical flow and segmentation fusion block

In surveillance video scenes, target objects are typically small and the video resolution is generally low, which complicates segmentation. To improve segmentation performance, we apply an OSF block after the segmentation branch. This block uses an attention mechanism to

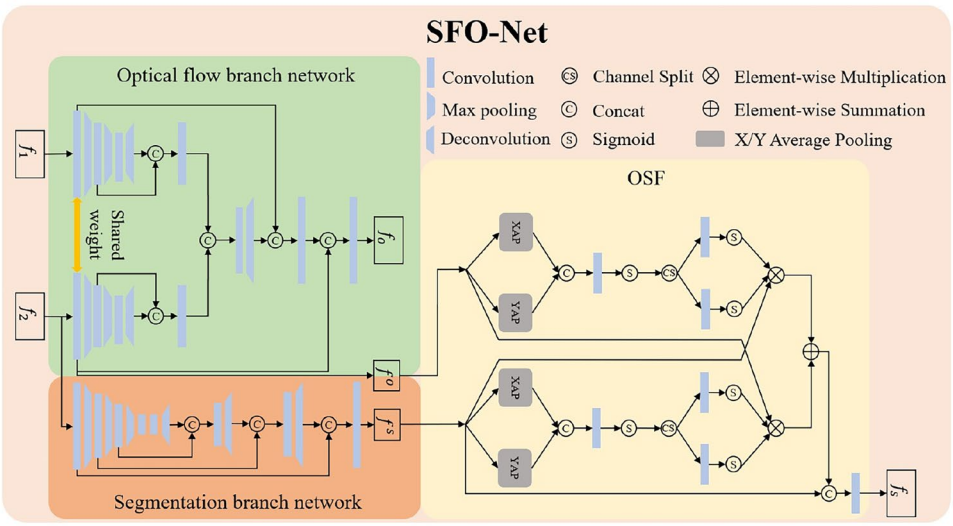

Fig. 3 Illustration of our SFO-Net, which consists of three main components, including the optical flow branch, the segmentation branch, and the OSF block. Optical flow branch is responsible for predicting the optical flow map. Segmentation branch is used for predicting the segmentation map. The OSF block fuses features of optical flow branch and segmentation branch to generate a more accurate segmentation map

deeply fuse shallow features from the optical flow branch (i.e., $f^o \in \mathbb{R}^{H\times W\times 48}$) with high-level features from the segmentation branch (i.e., $f^s \in \mathbb{R}^{H\times W\times 48}$), enhancing the segmentation map. Channel attention [36–38] improves feature representation by modeling interdependencies between channels, but it overlooks spatial positional information. Inspired by Coordinate Attention [39], we embed positional information into channel attention to preserve spatial layout better.

In our OSF block, for f^o, we encode each channel with pooling kernels of spatial ranges $(H, 1)$ and $(1, W)$, respectively. Thus, the output at height h and width w for the c-th channel can be written as the following two equations:

$$z_c^h(h) = \frac{1}{W} \sum_{0\le i<W} f_c^o(h, i), \tag{1}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0\le j<H} f_c^o(j, w). \tag{2}$$

Next, we concatenate z^h and z^w along the spatial dimension and feed them into a 1×1 convolution layer. After applying a sigmoid function, we obtain $X^o \in \mathbb{R}^{8\times(H+W)}$. X^o is then split into two parts along the spatial dimension, i.e., one part is along the height direction, denoted as $X_h^o \in \mathbb{R}^{8\times H}$, and the other part is along the width direction, denoted as $X_w^o \in \mathbb{R}^{8\times W}$. Following this, we apply a 1×1 convolution layer to X_h^o and X_w^o, and then use the sigmoid function to obtain two attention maps, i.e., $Y_h^o \in \mathbb{R}^{48\times H}$ and $Y_w^o \in \mathbb{R}^{48\times W}$. These attention maps represent the attention maps of the optical flow features in the height and width dimensions.

A similar process is applied to f^s of the segmentation branch, resulting in two corresponding attention maps $Y_h^s \in \mathbb{R}^{48\times H}$ and $Y_w^s \in \mathbb{R}^{48\times W}$. After obtaining these attention maps, we apply coordinate attention maps from the optical flow branch to f^s of the segmentation branch. Specifically, for a given position (i, j), its output f^{so} is expressed as:

$$f^{so}(i, j) = f^s(i, j) \times Y_h^o(i) \times Y_w^o(j). \tag{3}$$

Similarly, we apply coordinate attention maps from the segmentation branch to f^o of the optical flow branch, with the output f^{os} expressed as:

$$f^{os}(i, j) = f^o(i, j) \times Y_h^s(i) \times Y_w^s(j). \tag{4}$$

In this way, we obtain two new feature representations, i.e., f^{so} and f^{os}, which incorporate the attention maps from the segmentation and optical flow branches, respectively. Next, we perform an element-wise addition operation on these two features to produce a fused feature $F \in \mathbb{R}^{H\times W\times 48}$. Finally, we concatenate the original segmentation feature f^s and the fused feature F along the channel dimension to form a new feature representation. This new feature representation is then processed by a convolution layer, producing the final segmentation map $f_s \in \mathbb{R}^{H\times W\times 1}$. This segmentation map not only integrates the information from both the optical flow and segmentation branches but also leverages the attention mechanism to enhance the capture of details, ensuring a more precise segmentation map.

3.3 Objective function

3.3.1 Loss function of SFO-Net

To train the SFO-Net suitable for surveillance video data, we pretrain it using the ShanghaiTech dataset [20], originally meant for anomaly detection and lacking direct labels for optical flow estimation and segmentation. To overcome this, we employ a pretrained optical flow estimation model [40] to generate optical flow labels. We do this by feeding consecutive frames from the ShanghaiTech training set into the model, producing predictions that serve as training labels for the optical flow branch, denoted as $f_{(o-\text{label})} \in \mathbb{R}^{H\times W\times 2}$. For segmentation labels, we first utilize YOLOv8 [41] to detect objects and generate bounding boxes as prompts. These prompts are then processed by MobileSAM [42] to produce detailed segmentation maps, which subsequently serve as training labels for the segmentation process, denoted as $f_{(s-\text{label})} \in \mathbb{R}^{H\times W\times 1}$.

It is worth noting that $f_{(o-\text{label})}$ and $f_{(s-\text{label})}$ differ from $f_{(o-\text{truth})}$ and $f_{(s-\text{truth})}$ mentioned in Sect. 3.1. Specifically, $f_{(o-\text{label})}$ and $f_{(s-\text{label})}$ are generated using pre-trained models and are intended for training the SFO-Net. Once trained, the SFO-Net is integrated into the SOFF framework. During fine-tuning with the SOFF framework, the parameters of the SFO-Net remain frozen, and only the anomaly detection model is updated. When consecutive real frames are input into the SFO-Net, the outputs are $f_{(o-\text{truth})}$ and $f_{(s-\text{truth})}$, which are utilized to update the anomaly detection model.

We employ a three-stage training strategy to train the SFO-Net. In the first training stage, we train only the optical flow branch with the OSF block removed, while keeping all other parameters of SFO-Net fixed. We use two consecutive frames, i.e., f_1 and f_2, as input. The loss for this stage $L_o(f_1, f_2, f_{(o-\text{label})}; \theta_f)$, where θ_f are the parameters of the optical flow branch, is computed using the endpoint error (EPE). EPE measures the Euclidean distance between the estimated

optical flow $f_{(o-\mathrm{g})} \in \mathbb{R}^{H\times W\times 2}$ and the ground truth $f_{(o-\mathrm{label})}$. The loss is formulated as follows:

$$L_o(f_1,f_2,f_{(o-\mathrm{label})}) = \frac{1}{N}\sum_{i,j}\sqrt{\left(f^x_{(o-\mathrm{g})}(i)-f^x_{(o-\mathrm{label})}(i)\right)^2+\left(f^y_{(o-\mathrm{g})}(j)-f^y_{(o-\mathrm{label})}(j)\right)^2}, \quad (5)$$

where N is the number of pixels, x and y are the optical flow components in their respective directions, and i, j denote the spatial indices of a video frame.

In the second training stage, we train only the segmentation branch without using the OSF block. We use a single frame, f_2, as input. The segmentation branch first produces $f^s \in \mathbb{R}^{H\times W\times 48}$, which is then passed through convolutional layers to generate $f_{(s-\mathrm{g})} \in \mathbb{R}^{H\times W\times 1}$. The loss function for this stage $L_s(f_2,f_{(s-\mathrm{label})};\theta_s)$, where θ_s are the segmentation branch's parameters, combines intensity loss $L_{(s-\mathrm{I})}$ and gradient loss $L_{(s-\mathrm{G})}$. The loss is formulated as follows:

$$L_s(f_2,f_{(s-\mathrm{label})}) = L_{(s-\mathrm{I})}(f_2,f_{(s-\mathrm{label})}) + L_{(s-\mathrm{G})}(f_2,f_{(s-\mathrm{label})}), \quad (6)$$

$$L_{(s-\mathrm{I})}(f_2,f_{(s-\mathrm{label})}) = \frac{1}{N}\sum_{i,j}\left(f_{(s-\mathrm{g})}(i,j)-f_{(s-\mathrm{label})}(i,j)\right)^2, \quad (7)$$

$$L_{(s-\mathrm{G})}(f_2,f_{(s-\mathrm{label})}) = \frac{1}{N}\sum_{i,j}\Big(\left|\nabla_x f_{(s-\mathrm{g})}(i,j)-\nabla_x f_{(s-\mathrm{label})}(i,j)\right| + \left|\nabla_y f_{(s-\mathrm{g})}(i,j)-\nabla_y f_{(s-\mathrm{label})}(i,j)\right|\Big). \quad (8)$$

In the third training stage, the OSF block is integrated into SFO-Net, and only its parameters (θ_{fs}) are trained, while the rest of the SFO-Net's parameters remain fixed. We input two consecutive frames, f_1 and f_2, into the SFO-Net. The optical flow and segmentation features from the two branches serve as inputs to the OSF block. Using the segmentation map $f_{(s-\mathrm{label})}$ as the label, the output is $f_{(s-\mathrm{g})} \in \mathbb{R}^{H\times W\times 1}$. The loss function for this stage is the same as in the second training stage.

Algorithm 1 Three-Stage Training Strategy for SFO-Net

Input: Training data $D = \{f_1, f_2, f_{o\text{-}label}, f_{s\text{-}label}\}$; Hyperparameters: α, epochs1, epochs2, epochs3.
Output: Trained SFO-Net model.

Stage 1: Initialize θ_f, freeze θ_s and θ_{fs} ▷ removing the OSF block
for *epoch* $\leftarrow 1$ **to** epochs1 **do**
  for each $(f_1, f_2, f_{o\text{-}label})$ in D **do**
    Compute $f_{(o-\mathrm{g})}$ ▷ using the optical flow branch
    $\theta_f \leftarrow \theta_f - \alpha\nabla_{\theta_f}\mathcal{L}_o(f_1, f_2, f_{o\text{-}label})$ ▷ using Eq. (5)
  end for
end for
Stage 2: Initialize θ_s, freeze θ_f and θ_{fs} ▷ removing the OSF block
for *epoch* $\leftarrow 1$ **to** epochs2 **do**
  for each $(f_2, f_{s\text{-}label})$ in D **do**
    Compute $f_{(s-\mathrm{g})}$ ▷ using the segmentation branch
    $\theta_s \leftarrow \theta_s - \alpha\nabla_{\theta_s}\mathcal{L}_s(f_2, f_{s\text{-}label})$ ▷ using Eq. (6)
  end for
end for
Stage 3: Initialize θ_{fs}, freeze θ_f and θ_s ▷ adding the OSF block
for *epoch* $\leftarrow 1$ **to** epochs3 **do**
  for each $(f_1, f_2, f_{o\text{-}label}, f_{s\text{-}label})$ in D **do**
    Compute $f_{(s-\mathrm{g})}$ ▷ using SFO-Net
    $\theta_{fs} \leftarrow \theta_s - \alpha\nabla_{\theta_{fs}}\mathcal{L}_s(f_2, f_{s\text{-}label})$ ▷ using Eq. (6)
  end for
end for

3.3.2 Loss function for few-shot fine-tuning

In our proposed SOFF framework, the calculation formula for L_s during few-shot fine-tuning of SFO-Net is the same as the one used during SFO-Net training, while the optical flow loss L_o during fine-tuning is calculated as the intensity loss of the optical flow map combined with the L_o from the SFO-Net training phase. The image prediction loss L_p is defined as follows:

$$L_p(I_t, \hat{I}_t) = L_{(p-\mathrm{I})}(I_t, \hat{I}_t) + L_{(p-\mathrm{G})}(I_t, \hat{I}_t), \tag{9}$$

where $L_{(p-\mathrm{I})}(I_t, \hat{I}_t)$ represents the intensity loss for the image, and $L_{(p-\mathrm{G})}(I_t, \hat{I}_t)$ represents the gradient loss for the image. These losses are the same as the intensity loss $L_{(s-\mathrm{I})}$ and the gradient loss $L_{(s-\mathrm{G})}$ used during the training of the SFO-Net.

Finally, the total loss function L_{total} during few-shot fine-tuning is defined as follows:

$$L_{\text{total}} = L_p + \alpha L_s + \beta L_o, \tag{10}$$

Table 1 Comparison of K-shot (K = 0; 1; 10) scene-adaptive anomaly detection under the cross-dataset testing setting

Dataset	Methods	0-shot	1-shot	10-shot
UCSD Ped1	rGAN [16] (Finetune)	73.1	76.99	78.23
	rGAN [16] (Meta)	73.1	80.6	82.38
	MPN [17] (Meta)	74.45	78.54	80.20
	ADNet [43] (Meta)	–	**82.23**	**85.42**
	Ours-UNet-3layer	73.20	78.49	79.44
	Ours-UNet-4layer	**78.35**	80.83	82.65
UCSD Ped2	rGAN [16] (Finetune)	81.95	85.64	91.11
	rGAN [16] (Meta)	81.95	91.19	92.8
	MPN [17] (Meta)	90.17	94.46	95.75
	AADNet [18] (Meta)	–	92.76	94.42
	VADNet [19] (Meta)	–	93.96	95.12
	ADNet [43] (Meta)	–	94.52	94.82
	Ours-UNet-3layer	**91.10**	93.97	**96.36**
	Ours-UNet-4layer	90.23	**95.52**	95.98
CUHK Avenue	rGAN [16] (Finetune)	71.43	75.43	77.77
	rGAN [16] (Meta)	71.43	76.58	78.79
	MPN [17] (Meta)	74.06	78.92	81.69
	AADNet [18] (Meta)	–	78.88	80.59
	VADNet [19] (Meta)	–	80.83	82.62
	ADNet [43] (Meta)	–	**83.71**	85.33
	Ours-UNet-3layer	76.89	81.22	84.63
	Ours-UNet-4layer	**80.67**	81.85	**85.72**

We use the Shanghai Tech dataset for pre-training, reporting results in the form of AUC (%). Note that K = 0 represents the models are only pre-trained without any fine-tuning

The best performing results are marked in bold, while the second-best results are marked with an underline

where α and β are weighting hyper-parameters. We set α to 0.5 and β to 1 in our experiments. We use the total loss L_{total} to update the parameters θ of the anomaly detection model.

4 Experiments

4.1 Dataset

The aim of this paper is to adapt the model to new scenarios with only a small number of fine-tuning samples, enabling it to detect anomalous events in previously unseen scenes. This setup requires that the training and testing videos come from different scenes. To validate the effectiveness of the SOFF framework, we use several popular and challenging datasets.

Shanghai Tech [20] contains 330 training videos with only normal events and 107 testing videos featuring anomalies like chasing or cycling. It spans 13 different scenes.

UCF Crime [21] is a large-scale dataset with 1,900 videos. The training set includes both 800 normal and 810 abnormal videos, while the testing set has 150 normal and 140 abnormal videos. It features 13 types of anomalies, such as theft and explosions, across diverse and complex scenes.

UCSD Ped1 [15] includes 34 training videos and 36 testing videos, all from the same scene, showing normal pedestrian activities and anomalies like walking on grass.

UCSD Ped2 [15] has 16 training and 12 testing videos, featuring lateral pedestrian movements with similar anomalies to UCSD Ped1.

CUHK Avenue [22] comprises 16 training and 21 testing videos from the same scene at different times. Normal activities involve pedestrian movement, with test set anomalies including running and littering. Some anomalies are simulated, and camera shake adds complexity.

In our experiments, Shanghai Tech and UCF Crime datasets are used for training, respectively. UCSD Ped1, UCSD Ped2, and CUHK Avenue datasets are used only for testing, providing a few samples for fine-tuning the pre-trained model.

4.2 Evaluation metrics

We use a frame prediction model trained only on normal samples, which can accurately predict future frames for normal scenarios. During testing, anomalies result in significant differences between predicted and actual frames. By comparing these differences, we identify anomalies.

To compute the anomaly score for each frame, we calculate the Peak Signal-to-Noise Ratio (PSNR) between the predicted future frame $\hat{f}$ and the actual future frame f, denoted as $P(\hat{f}, f)$. The PSNR is given by the following formula:

$$P(\hat{f},f) = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right), \tag{11}$$

$$\mathrm{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\|\hat{f}(i,j) - f(i,j)\right\|^2, \tag{12}$$

where MAX is the maximum pixel value of the image.

Next, for a test video segment consisting of t frames, we compute t PSNR values $P(\hat{f}^t,f^t)$ for each frame. These PSNR values are then normalized to obtain $\bar{P}(\hat{f}^t,f^t)$, ensuring that the results are constrained within the range [0, 1]. The normalization is performed using the min–max normalization method, as described in the following formula:

$$\bar{P}(\hat{f}^t,f^t) = \frac{P(\hat{f}^t,f^t) - \min_t P(\hat{f}^t,f^t)}{\max_t P(\hat{f}^t,f^t) - \min_t P(\hat{f}^t,f^t)}. \tag{13}$$

The value of $\bar{P}(\hat{f}^t,f^t)$, used as the anomaly score, reflects the degree of anomaly in a frame. By comparing it with a threshold, we can determine whether a frame is anomalous. To evaluate the model's anomaly detection performance on the dataset, we use the AUC (Area Under the Curve) based on $\bar{P}(\hat{f}^t,f^t)$ as the evaluation metric, following previous works [16–19]. The AUC is calculated as the area under the Receiver Operating Characteristic (ROC) curve, which is obtained by varying the threshold of the anomaly scores. A higher AUC indicates better performance in distinguishing between normal and anomalous samples. In the experiments, we report results using the AUC (%).

4.3 Implementation details

We conducted our experiments using the PyTorch framework. During the pre-training phase of the frame prediction model, all input frames were resized to 256×256, with a batch size of 16 and a base learning rate of 0.0002. Other configurations followed the default settings described in the anomaly detection model [5], except for the optical flow and discriminator settings, which were not used during the pre-training of our anomaly detection model. In the testing phase, for the UCSD Ped2 and CUHK Avenue datasets, we resized the input frames to 256×172. For the UCSD Ped1 dataset, which has smaller original dimensions of 238×158, we resized the input frames to 236×156. In this experiment, the number of consecutive frames n fed into the model was set to 4. It is worth noting that, in the experiments and visualization examples presented in this paper, unless otherwise specified, we use the Ours-UNet-3layer model pre-trained on the Shanghai Tech dataset for ablation studies and visualization generation.

Table 2 Comparison of K-shot (K = 0; 1; 10) scene-adaptive anomaly detection under the cross-dataset testing setting

Target	Methods	0-shot	1-shot	10-shot
UCSD Ped1	rGAN [16] (Finetune)	66.87	71.70	74.68
	rGAN [16] (Meta)	66.87	78.44	81.62
	MPN [17] (Meta)	75.52	77.19	79.53
	Ours-UNet-3layer	74.35	77.39	79.77
	Ours-UNet-4layer	**76.01**	**78.51**	**82.21**
UCSD Ped2	rGAN [16] (Finetune)	62.53	65.58	78.32
	rGAN [16] (Meta)	62.53	83.08	90.21
	MPN [17] (Meta)	86.04	88.43	89.89
	AADNet [18] (Meta)	–	87.62	90.28
	VADNet [19] (Meta)	–	88.20	90.40
	Ours-UNet-3layer	90.85	**92.65**	**95.41**
	Ours-UNet-4layer	**90.98**	91.88	94.69
CUHK Avenue	rGAN [16] (Finetune)	64.32	66.70	70.61
	rGAN [16] (Meta)	64.32	72.62	79.02
	MPN [17] (Meta)	**82.26**	**85.62**	**85.91**
	AADNet [18] (Meta)	–	79.84	81.30
	VADNet [19] (Meta)	–	80.20	81.90
	Ours-UNet-3layer	73.84	80.41	85.73
	Ours-UNet-4layer	79.05	80.84	85.03

We use the UCF Crime dataset for pre-training, reporting results in the form of AUC (%). Note that K = 0 represents the models are only pre-trained without any fine-tuning

The best performing results are marked in bold, while the second-best results are marked with an underline

4.4 Performance comparison

We conducted experiments to evaluate the fine-tuning performance of the SOFF framework when adapting to new scenes. For comparison, we selected related works on few-shot scene-adaptive anomaly detection tasks [16–19]. The anomaly detection model was pre-trained separately on the Shanghai Tech and UCF Crime datasets. We then tested the SOFF framework's fine-tuning performance on three previously unseen datasets: UCSD Ped1, UCSD Ped2, and CUHK Avenue. Only a small number of samples from these datasets were used for fine-tuning the model.

The comparison results are presented in Tables 1 and 2. When using the model pre-trained on the Shanghai Tech dataset, as shown in Table 1, our method achieved outstanding performance in the 10-shot setting on the UCSD Ped2 and CUHK Avenue datasets. When using the model pre-trained on the UCF Crime dataset, as shown in Table 2, our method outperformed previous methods on UCSD Ped1 and UCSD Ped2 and achieved near-optimal performance on CUHK Avenue. Additionally, across the aforementioned cross-scene settings, our SOFF framework demonstrated a significant improvement when transitioning from 0-shot to 10-shot scenarios. Furthermore, both versions of our

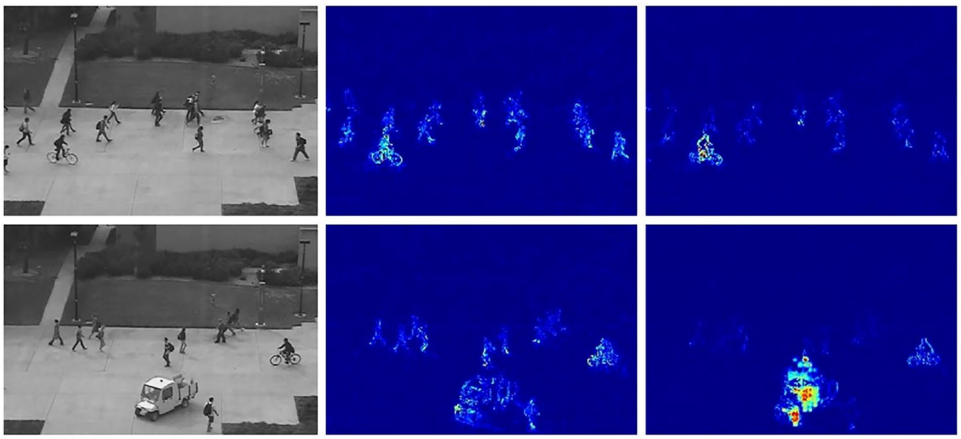

Fig. 4 The visualization of prediction result. The first column shows the real frames. The second column displays the prediction errors from the pre-trained model without fine-tuning, where highlighted areas indicate the differences between the predicted frames and the real frames. The third column presents the prediction errors after fine-tuning with the SOFF framework

Table 3 Ablation studies of three parts of the proposed SFO-Net when evaluated on the UCSD Ped2

Optical flow	Segmentation	OSF	UCSD Ped2 (1-shot)	UCSD Ped2 (10-shot)
×	×	×	92.15	93.46
✓	×	×	92.93	95.82
×	✓	×	92.61	94.37
✓	✓	×	<u>93.62</u>	<u>96.04</u>
✓	✓	✓	**93.97**	**96.36**

The best performing results are marked in bold, while the second-best results are marked with an underline

anomaly detection model, Ours-UNet-3layer and Ours-UNet-4layer, exhibited strong performance in Tables 1 and 2, further validating the versatility and effectiveness of the proposed SOFF framework.

To further demonstrate the effectiveness of the SOFF framework, we visualized some predicted images in Fig. 4. As illustrated, our method significantly reduces prediction errors compared to models without fine-tuning. Additionally, in non-anomalous areas, our method exhibits lower prediction errors than the model without fine-tuning. This improvement allows the model to better predict normal regions, concentrating prediction errors in anomalous areas. Consequently, this enhances the model's ability to distinguish between normal and abnormal conditions.

4.5 Ablation study

4.5.1 Effectiveness of each component

To evaluate the effectiveness of each component of the SFO-Net, we tested its performance by systematically removing each part. When the optical flow branch was removed, we also removed the OSF block since it links to

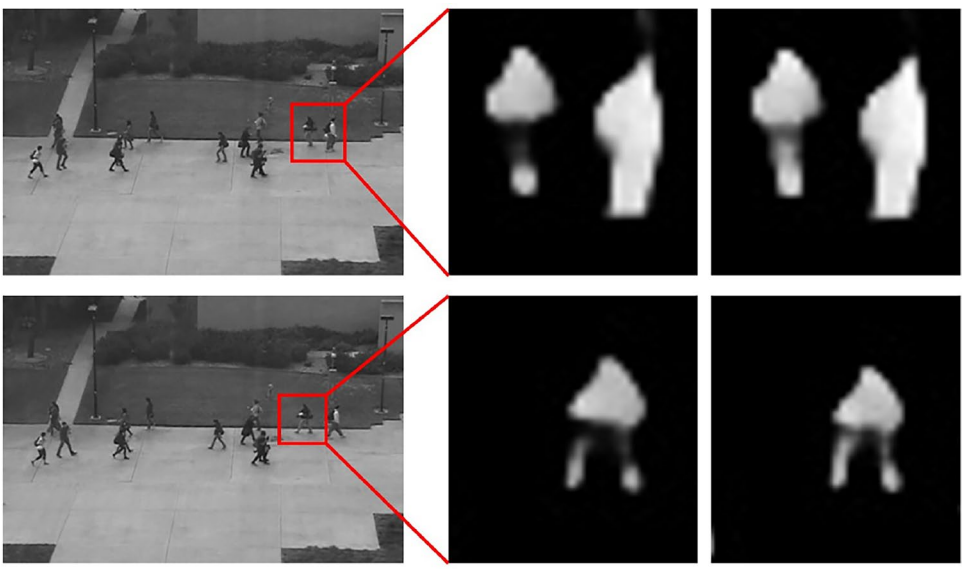

Fig. 5 The segmentation maps predicted by the SFO-Net. The first column shows the RGB images. The second column presents the segmentation maps output without processing by the OSF block, and the third column shows the segmentation maps output with processing by the OSF block

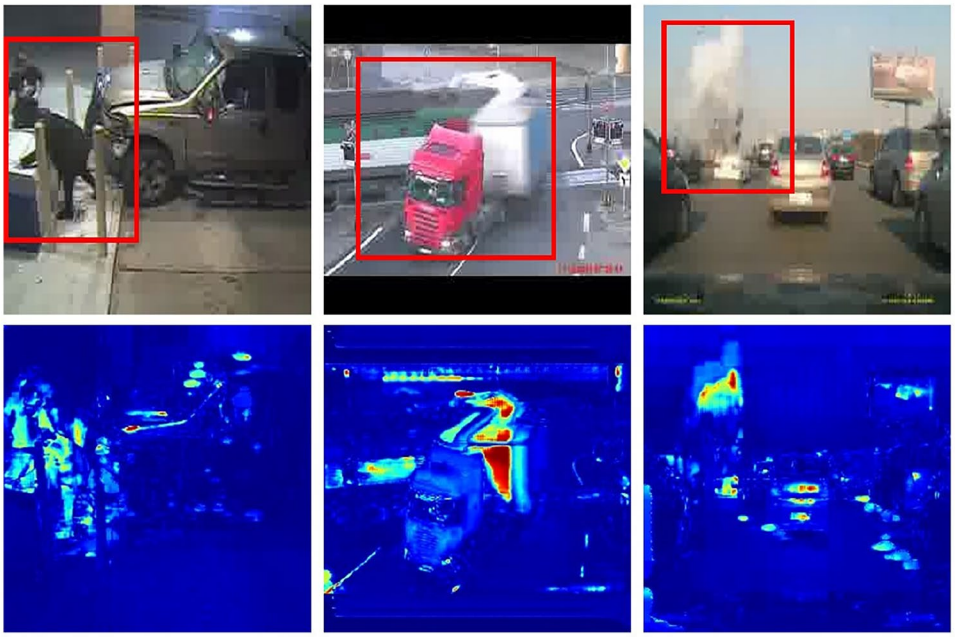

Fig. 6 Visualization of prediction results for different types of anomalies. From left to right, the anomaly types are theft, road accidents, and explosion. The first row shows the ground truth frames, where the red boxes highlight the anomalous regions. The second row presents the corresponding prediction errors generated by the model

both the optical flow and segmentation branches, leaving only the segmentation branch for predicting segmentation maps. Similarly, removing the segmentation branch also removed the OSF block, leaving just the optical flow branch to predict optical flow maps. Without the OSF block, both branches remained to predict their respective maps independently. Using a model pre-trained on the Shanghai Tech dataset, we conducted few-shot fine-tuning tests on the UCSD Ped2 dataset. The results are presented in Table 3. Notably, the first row, which does not use the SOFF framework, serves as the baseline and represents the standard fine-tuning method. Both the segmentation and optical flow branches improved performance over the baseline, indicating that spatial and temporal information enhances the fine-tuning process. Combining both branches further boosted performance, demonstrating their complementarity. The inclusion of the OSF block yielded the best results, enhancing feature fusion and improving segmentation accuracy during fine-tuning.

To further illustrate the effectiveness of the OSF block, we visualized several segmentation maps predicted by the SFO-Net in Fig. 5. As shown, with the inclusion of the OSF block, the SFO-Net can accurately segment even smaller objects. This demonstrates that the OSF block can effectively integrate features from both the optical flow branch and the segmentation branch, enhancing the final segmentation maps.

We utilized a model pretrained on the UCF Crime dataset and fine-tuned on the UCSD Ped2 dataset to detect various types of anomalies. As shown in Fig. 6, the model is able to accurately detect diverse anomalies, such as theft, road accidents, and explosions. This demonstrates the fine-tuned model's strong adaptability and applicability across a range of real-world scenarios.

Table 4 Comparison of different training strategies

Training strategy	UCSD Ped2 (1-shot)	UCSD Ped2 (10-shot)
One-stage	<u>93.76</u>	95.13
Two-stage	93.45	<u>95.69</u>
Three-stage	**93.97**	**96.36**

The best performing results are marked in bold, while the second-best results are marked with an underline

4.5.2 Effectiveness of training strategy

To assess the impact of different training strategies on the SFO-Net's fine-tuning performance, we tested one-stage, two-stage, and three-stage training methods. In the one-stage training approach, the entire SFO-Net is trained simultaneously. For the two-stage training, the process begins by training only the optical flow branch while keeping the other parameters frozen. In the second stage, the optical flow branch is frozen, and the training focuses on the segmentation branch and the OSF block. The three-stage training method involves a sequential approach: initially, the optical flow branch is trained with the rest of the network

frozen; next, the segmentation branch is trained while freezing the remaining parts; finally, the OSF block is trained independently.

We used the Shanghai Tech dataset to pre-train the anomaly detection model and evaluated the performance of different training strategies for the SFO-Net on the UCSD Ped2 dataset during few-shot fine-tuning. As listed in Table 4, the three-stage training strategy achieved the best performance. This indicates that progressively training each part of the network enhances training stability and ensures that each component learns effectively for its specific task, thereby improving the overall fine-tuning performance.

5 Conclusion

In this paper, we tackle the challenge of few-shot scene-adaptive anomaly detection by introducing a framework called Segmentation and Optical Flow Fine-tuning (SOFF). This framework allows existing anomaly detection models to be fine-tuned to adapt to new scenes with just a few samples from those new scenes. It is worth noting that the SOFF framework can be widely applied to existing anomaly detection models with image-based outputs. To realize this, we integrate a network called SFO-Net into the SOFF framework. This network is then cascaded with existing anomaly detection models to perform two auxiliary tasks: segmentation and optical flow estimation. SFO-Net operates in a self-supervised manner, generating segmentation and optical flow maps based on predicted and actual frames to calculate loss, thereby leveraging rich temporal and spatial information. Extensive experiments demonstrate that, after fine-tuning with the SOFF framework, existing models can adapt well to new scenes. Further ablation studies reveal that the tasks of segmentation and optical flow estimation are complementary, jointly enhancing the few-shot generalization ability of the SOFF framework. After training SFO-Net, the SOFF framework can be applied to existing anomaly detection models with image-based outputs, demonstrating significant practical value.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 62471285 and 62401350, in part by the Shanghai Sailing Program under Grant 24YF2713000, and in part by the Foundation of Yunnan Key Laboratory of Service Computing (No. YNSC24109).

Author Contributions J. L. wrote the main manuscript text and prepared all the tables as well as Figs. 3, 4, 5 and 6. G. L. prepared Figs. 1 and 2, and reviewed and revised the manuscript. Z. L. reviewed and revised the manuscript.

Data availibility No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Zaigham Zaheer, M., Lee, J.-H., Astrid, M., Lee, S.-I.: Old is gold: redefining the adversarially learned one-class classifier training paradigm. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14171–14181 (2020). https://doi.org/10.1109/CVPR42600.2020.01419
2. Liu, Y., Liu, J., Lin, J., Zhao, M., Song, L.: Appearance-motion united auto-encoder framework for video anomaly detection. IEEE Trans. Circuits Syst. II Express Briefs **69**(5), 2498–2502 (2022). https://doi.org/10.1109/TCSII.2022.3161049
3. Ribeiro, M., Lazzaretti, A.E., Lopes, H.S.: A study of deep convolutional auto-encoders for anomaly detection in videos. Pattern Recogn. Lett. **105**, 13–22 (2018) https://doi.org/10.1016/j.patrec.2017.07.016
4. Tur, A.O., Dall'Asen, N., Beyan, C., Ricci, E.: Unsupervised video anomaly detection with diffusion models conditioned on compact motion representations. In: Foresti, G.L., Fusiello, A., Hancock, E. (eds.) Image Analysis and Processing—ICIAP 2023, pp. 49–62. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-43153-1_5
5. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection—a new baseline. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6536–6545 (2018).https://doi.org/10.1109/CVPR.2018.00684
6. Lei, S., Song, J., Wang, T., Wang, F., Yan, Z.: Attention u-net based on multi-scale feature extraction and WSDAN data augmentation for video anomaly detection. Multimedia Syst. **30**(3), 118 (2024). https://doi.org/10.1007/s00530-024-01320-0
7. Liu, W., Cao, J., Zhu, Y., Liu, B., Zhu, X.: Real-time anomaly detection on surveillance video with two-stream spatio-temporal generative model. Multimedia Syst. **29**(1), 59–71 (2023). https://doi.org/10.1007/s00530-022-00979-7
8. Wang, D., Hu, Q., Wu, K.: Dual-branch network with memory for video anomaly detection. Multimedia Syst. **29**(1), 247–259 (2023). https://doi.org/10.1007/s00530-022-00991-x
9. Zhang, T., Lu, H., Li, S.Z.: Learning semantic scene models by object classification and trajectory clustering. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1940–1947 (2009). https://doi.org/10.1109/CVPR.2009.5206809. IEEE
10. Basharat, A., Gritai, A., Shah, M.: Learning object motion patterns for anomaly detection and improved object detection. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008). https://doi.org/10.1109/CVPR.2008.4587510. IEEE
11. Saligrama, V., Chen, Z.: Video anomaly detection based on local statistical aggregates. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2112–2119 (2012). https://doi.org/10.1109/CVPR.2012.6247917. IEEE
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893 (2005). https://doi.org/10.1109/CVPR.2005.177. IEEE
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94

14. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 65–72 (2005). https://doi.org/10.1109/VSPETS.2005.1570899. IEEE
15. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1975–1981 (2010). https://doi.org/10.1109/CVPR.2010.5539872
16. Lu, Y., Yu, F., Reddy, M.K.K., Wang, Y.: Few-shot scene-adaptive anomaly detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16, pp. 125–141 (2020). https://doi.org/10.1007/978-3-030-58558-7_8. Springer
17. Lv, H., Chen, C., Cui, Z., Xu, C., Li, Y., Yang, J.: Learning normal dynamics in videos with meta prototype network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15425–15434 (2021).https://doi.org/10.1109/CVPR46437.2021.01517
18. Hu, Y., Huang, X., Luo, X.: Adaptive anomaly detection network for unseen scene without fine-tuning. In: Pattern Recognition and Computer Vision: 4th Chinese Conference, PRCV 2021, Beijing, China, October 29–November 1, 2021, Proceedings, Part II 4, pp. 311–323 (2021). https://doi.org/10.1007/978-3-030-88007-1_26. Springer
19. Huang, X., Hu, Y., Luo, X., Han, J., Zhang, B., Cao, X.: Boosting variational inference with margin learning for few-shot scene-adaptive anomaly detection. IEEE Trans. Circuits Syst. Video Technol. **33**(6), 2813–2825 (2022). https://doi.org/10.1109/TCSVT.2022.3227716
20. Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked rnn framework. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 341–349 (2017). https://doi.org/10.1109/ICCV.2017.45
21. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6479–6488 (2018). https://doi.org/10.1109/CVPR.2018.00678
22. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2720–2727 (2013). https://doi.org/10.1109/ICCV.2013.338
23. Liu, Y., Liu, J., Zhao, M., Yang, D., Zhu, X., Song, L.: Learning appearance-motion normality for video anomaly detection. In: 2022 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2022). https://doi.org/10.1109/ICME52920.2022.9859727. IEEE
24. Pang, G., Yan, C., Shen, C., Hengel, A.v.d., Bai, X.: Self-trained deep ordinal regression for end-to-end video anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12173–12182 (2020). https://doi.org/10.1109/CVPR42600.2020.01219
25. Liu, Y., Liu, J., Ni, W., Song, L.: Abnormal event detection with self-guiding multi-instance ranking framework. In: 2022 International Joint Conference on Neural Networks (IJCNN), pp. 01–07 (2022). https://doi.org/10.1109/IJCNN55064.2022.9892231. IEEE
26. Li, C., Chen, M.: Dy-mil: dynamic multiple-instance learning framework for video anomaly detection. Multimedia Syst. **30**(1), 11 (2024). https://doi.org/10.1007/s00530-023-01237-0
27. Acsintoae, A., Florescu, A., Georgescu, M.-I., Mare, T., Sumedrea, P., Ionescu, R.T., Khan, F.S., Shah, M.: Ubnormal: New benchmark for supervised open-set video anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20143–20153 (2022). https://doi.org/10.1109/CVPR52688.2022.01951
28. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1705–1714 (2019). https://doi.org/10.1109/ICCV.2019.00179
29. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-assisted intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pp. 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28 . Springer
30. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350**(6266), 1332–1338 (2015). https://doi.org/10.1126/science.aab3050
31. Rostami, M., Kolouri, S., Eaton, E., Kim, K.: Sar image classification using few-shot cross-domain transfer learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2019). https://doi.org/10.1109/CVPRW.2019.00120
32. Tai, Y., Tan, Y., Xiong, S., Sun, Z., Tian, J.: Few-shot transfer learning for sar image classification without extra sar samples. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. **15**, 2240–2253 (2022). https://doi.org/10.1109/JSTARS.2022.3155406
33. Yu, Z., Chen, L., Cheng, Z., Luo, J.: Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12856–12864 (2020). https://doi.org/10.1109/CVPR42600.2020.01287
34. Ghani, B., Denton, T., Kahl, S., Klinck, H.: Global birdsong embeddings enable superior transfer learning for bioacoustic classification. Sci. Rep. **13**(1), 22876 (2023). https://doi.org/10.1038/s41598-023-49989-z
35. Munkhdalai, T., Yu, H.: Meta networks. In: International Conference on Machine Learning, vol. 70, pp. 2554–2563 (2017). https://doi.org/10.5555/3305890.3305945. PMLR
36. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018). https://doi.org/10.1109/CVPR.2018.00745
37. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018). https://doi.org/10.1007/978-3-030-01234-2_1
38. Hu, J., Shen, L., Albanie, S., Sun, G., Vedaldi, A.: Gather-excite: exploiting feature context in convolutional neural networks. Adv. Neural Inform. Process. Syst. **31** (2018)https://doi.org/10.5555/3327546.3327612
39. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13713–13722 (2021). https://doi.org/10.1109/CVPR46437.2021.01350
40. Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R.: Learning to estimate hidden motions with global motion aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9772–9781 (2021). https://doi.org/10.1109/ICCV48922.2021.00963
41. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO. https://github.com/ultralytics/ultralytics
42. Zhang, C., Han, D., Qiao, Y., Kim, J.U., Bae, S.-H., Lee, S., Hong, C.S.: Faster segment anything: Towards lightweight sam for mobile applications. arXiv preprint arXiv:2306.14289 (2023)
43. Zahid, Y., Zarges, C., Tiddeman, B., Han, J.: Adversarial diffusion for few-shot scene adaptive video anomaly detection.

Neurocomputing **614**, 128796 (2025). https://doi.org/10.1016/j.neucom.2024.128796