# Cross-modal Assisted Training for Abnormal Event Recognition in Elevators

Xinmeng Chen*
Xuchen Gong*
Data Science Research Center, Duke Kunshan University
Kunshan, China
xinmeng.chen@dukekunshan.edu.cn
xuchen.gong@dukekunshan.edu.cn

Ming Cheng
Data Science Research Center, Duke Kunshan University
Kunshan, China
ming.cheng@dukekunshan.edu.cn

Qi Deng
Technology Asia and Escalator (CRD), KONE Elevators Co., Ltd.
Kunshan, China
qi.deng@kone.com

Ming Li†
Data Science Research Center, Duke Kunshan University
Kunshan, China
ming.li369@dukekunshan.edu.cn

## ABSTRACT

Given that very few action recognition datasets collected in elevators contain multimodal data, we collect and propose our multimodal dataset investigating passenger safety and inappropriate elevator usage. Moreover, we present a novel framework (RGBP) to utilize multimodal data to enhance unimodal test performance for the task of abnormal event recognition in elevators. Experimental results show that the best network architecture with the RGBP framework effectively improves the unimodal inference performance on the Elevator RGBD dataset by 4.71% (accuracy) and 4.95% (F1 score) with respect to the pure RGB model. In addition, our RGBP framework outperforms two other methods for "multimodal training and unimodal inference": MTUT [1] and the two-stage method based on depth estimation.

## CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**.

## KEYWORDS

Multimodal learning, abnormal event recognition in elevators

**ACM Reference Format:**
Xinmeng Chen, Xuchen Gong, Ming Cheng, Qi Deng, and Ming Li. 2021. Cross-modal Assisted Training for Abnormal Event Recognition in Elevators. In *Proceedings of the 2021 International Conference on Multimodal Interaction (ICMI '21), October 18–22, 2021, Montréal, QC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3462244.3479908

*Both authors contributed equally to this research.
†corresponding author


*ICMI '21, October 18–22, 2021, Montréal, QC, Canada*

ACM ISBN 978-1-4503-8481-0/21/10…$15.00
https://doi.org/10.1145/3462244.3479908

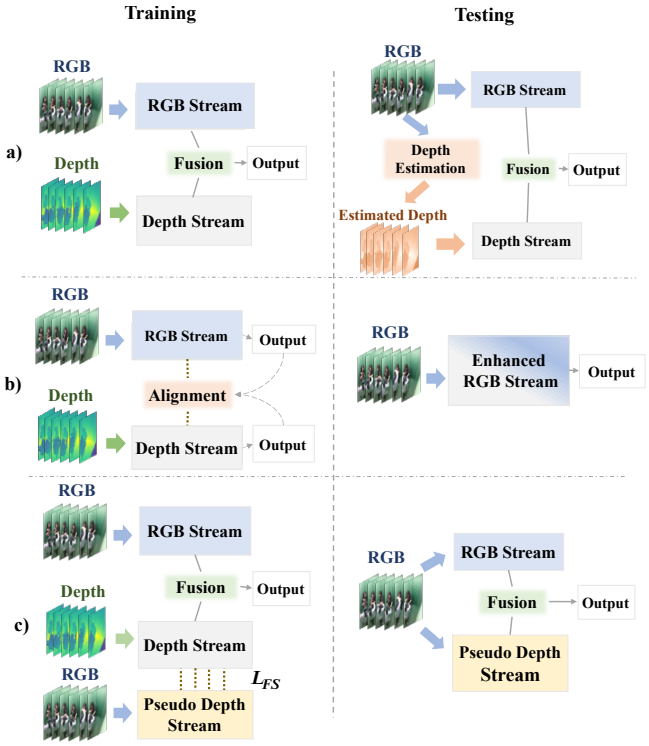



**Figure 1: Comparison of our RGBP framework with two other frameworks that use depth and RGB data in the training stage and only RGB data in the inference stage. a) a two-stage framework that involves depth estimation. b) MTUT c) RGBP (proposed).**

## 1 INTRODUCTION

Elevators are widely used public infrastructures that facilitate daily transportation. However, the structure and position of elevators make them vulnerable to several abnormal activities, such as forced opening the door. Moreover, some emergent events in elevators,

such as passenger fainting, may not be noticed in time. Therefore, elevators are important scenarios to implement abnormal event detection, which can help ensure life and property safety, enhance facility preservation, and reduce labor costs with an improved monitoring efficiency. Abnormal event detection is one popular application of the video-based action recognition [2, 5, 8–10, 28, 33, 46] and mainly involves classifying human actions into normal and abnormal activities [24].

The narrow space, crowded scene, and special angle of the monitoring camera make it usually insufficient to use the single RGB modality to detect abnormal events. One common way to compensate for the information, such as position and distance, is to add depth data [18, 22, 27]. In real-life situations, however, depth data is rarely available due to the expensiveness of depth cameras. Hence, if we involve depth data to assist the training process and only use the single RGB modality in the inference stage, the performance of abnormal event detection in elevators would possibly be enhanced with relatively low cost. There are mainly two common ways to deal with this task. One is depth estimation based on the RGB images [6, 13, 29], whose output would be estimated depth data and used as the input of depth feature extractor. The other is to use the information from the depth modality to improve the performance of single RGB modality [1]. This paper proposes a new framework, called the RGBP framework, which generates a pseudo depth model and fuses it with the RGB model in the inference stage. Figure 1 shows the comparison of these three different frameworks.

As mentioned above, our proposed RGBP framework can use both RGB and depth data during the training process and only use RGB data in the inference stage. The realization of our framework contains three stages, as specified in section 3. The first stage is to pre-train an RGB and depth joint model (RGBD). The second stage is to train a pseudo depth model that learns to generate the depth features from RGB data, which is supervised by a proposed *feature similarity loss*. The third stage is to substitute the depth model with the pseudo depth model and perform inference. All the models are trained on our proposed Elevator RGBD Dataset introduced in section 4.1. We compared our RGBP frameworks with the other two frameworks aforementioned in section 4.2, and our framework performs the best on the Elevator RGBD Dataset.

In summary, the contributions of this paper are as follows. First, we propose a new framework that generates a Pseudo Depth stream during the training process to imitate and replace the Depth stream in the inference stage. Second, we propose the *feature similarity loss* to give differentiated supervision to each block of the Pseudo Depth feature extractor. Third, we introduce the Elevator RGBD Dataset, which overcomes some limitations of existing datasets.

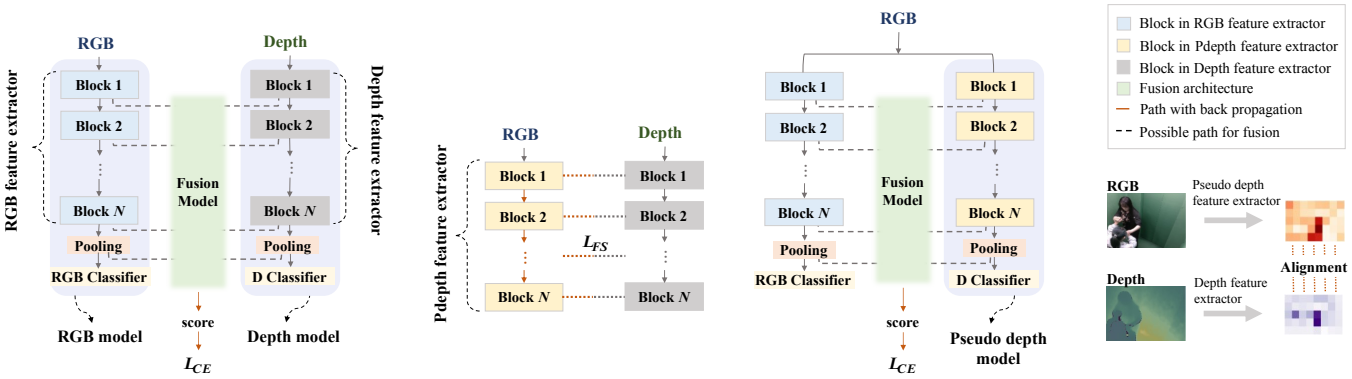
## 2 RELATED WORK

**Action Recognition and Abnormal Event Detection.** In action recognition, a label of an action is predicted based on the RGB videos, depth data, or the skeleton sequences, etc. It is fundamental to behavioral analysis and is widely used in video surveillance [5]. Abnormal event detection, as an important application of action recognition in videos, is aimed to detect the abnormal events in certain scenarios to prevent potential dangers [24]. When representing the behavior of the target in the videos, some early work [12, 21, 26, 32, 39] analyze the histograms, variation, or acceleration of the motions using optical flow vectors. However, such computationally expensive methods only extract global features. In handling this issue, local features such as interest points are extracted to represent the significant motion variations of the abnormal actions [23, 28], and texture features are extracted for each moving target [25, 31, 37]. The object tracking method is also employed [2, 7] to obtain the trajectory of each object described by a sequence of coordinates corresponding to different frames. Then, to adapt to the challenges resulting from the crowded scenes and non-static objects, Rao et al. and other authors [41, 42, 45] propose the spatiotemporal volume features by using the temporal information obtained from consecutive frames.

Moreover, multiple studies have explored abnormal event detection in the particular scenario of elevators. Shu et al. [33] learn the movement characteristics of targets from corner kinetic energy and use SVM to identify the violent behaviors in real-time. Zhu et al. [46] recognize the people who fall down and then use image entropy of Motion History Image [4] to detect violent behavior. However, both work focus more on passenger safety and lack the detection of inappropriate usage of the elevators. Moreover, many datasets collected in elevators make some assumptions about passenger behaviors. For example, the dataset used by Jia et al. [19] assumes that people get in or out only when the elevator stops at a certain floor, and it ignores the dangerous behaviors of forcing open the door, etc. The dataset used by Xiao et al. [40] assumes an ideal setup where the cameras can capture the faces of the occupants. While in reality, the cameras are usually at the corner of the ceiling, unable to capture the passengers' faces. In addition, unlike the dataset we propose in section 4.1, none of the above datasets contains depth data, so they can not be used to train the models that employ fusion methods. Furthermore, our collected database can also be used to study the abnormal activity detection from depth images only, protecting user privacy.

**Multimodal Fusion.** Since RGB data is a 2D representation of the 3D world with one dimension of information lost, and the depth data compensates for the lost information in the RGB videos. Therefore, combining the information extracted from RGB data and the depth data is common to integrate the complementary information. Fusion can be carried out at different stages by concatenation, weighted average, etc. Early fusion integrates data of different modalities into one feature volume as the input to a machine-learning algorithm [30]. Late fusion (i.e., decision-level fusion) involves the aggregation of prediction results made by multiple models trained on data of multiple modalities. As a favored fusion method because of its easy implementation and good performance, late fusion is widely used when the errors of two networks are relatively uncorrelated [35, 38]. For intermediate fusion, Karpathy et al. [20] show that a model slowly fuses temporal information throughout the network outperforms their models that employ early fusion or late fusion. Hu et al. [17] propose a dense multimodal intermediate fusion network for effective joint representation of the features.

**Multimodal Training and Unimodal Inference.** Different from multimodal fusion, which aims to use the representation from multiple modalities together to enhance the performance, the method




(a) Training Stage 1 (b) Training Stage 2 (c) Inference Stage (d) Representation Learning

**Figure 2: The RGBP framework works in three stages, which involve two stages of training and one stage of testing. In training stage 1, the pre-trained RGB model and the pre-trained depth model are fused. Supervised by $L_{CE}$, only the fusion model of RGBD is trainable. In training stage 2, supervised by $L_{FS}$, the feature extractor of the pseudo depth model is trained to learn the representation output by the depth model, with the depth model frozen. Then, to assemble the model for testing, for the RGBP model trained in (a), its depth model is replaced by the pseudo depth model trained in (b). Then, the resulting model is ready for inference.**

of "Multimodal Training and Unimodal Inference" achieves that goal when the inference inputs are only from one modality. In our case, depth data enhances the performance of an RGB model, while during inference, such data is usually unavailable since many elevators are only equipped with RGB cameras.

Commonly, the most intuitive method to make inferences based on both RGB and depth information when only the RGB data is available is to employ "depth estimation." Though the depth data is unavailable, we can first estimate the depth data via the RGB data and then use the RGB data and the estimated depth data to make a joint inference. Many work [6, 13, 34] achieve depth estimation based on the RGB images, and the estimated results can be used in the joint modeling. Such a method allows for fusion in the absence of depth input, while it has two limitations. First, it requires two models during the inference: RGB-to-Depth and RGBD joint classification models. Therefore, both the model size and the inference time get increased. Second, the error of the RGB-to-Depth model will lead to poor depth inputs given to the classification model, resulting in a bigger accumulative error.

In an attempt to handle this problem, Abavisani et al. propose a training method called MTUT [1], which encourages a model of one modality to utilize the learned information from multiple modalities. By aligning the spatiotemporal semantic information of the RGB model, the optical flow model, and the depth model during training, their method enhances the performance of each model. Regarding knowledge transfer, an inspiring idea is proposed by Hinton et al. [16], where a well-trained teacher network provides extra supervision for the training of the student network. Therefore, in the work of Aydogdu et al.[3], the radar model is trained on both the raw radar data and the knowledge distillation from the camera data, making it able to use the radar modality alone with enhanced performance during inference.

In all, for our task of action recognition and event detection, the actions of interest and the special setting in elevators, characterized by crowded space and limited views, are largely unavailable in the current public datasets. Therefore, we propose a dataset specifically designed for our task. Moreover, we propose our own "multimodal training, unimodal testing" framework. As discussed in section 3, unlike MTUT, which encourages one modality to learn multiple modalities, we build one more model and guide it to generate the features of another modality. This way, we can better maintain the features extracted from different modalities without having one of them interfering implicitly with the other.

## 3 PROPOSED METHOD

Our proposed method is a framework called "RGBP," which adopts data from multiple modalities for training while uses only one modality data for inference. Specifically, it takes RGB data alone as input, while it can make a prediction based on both the features of the RGB data and the predicted features of the depth data. It is achieved by having the RGBP composed of three 3D convolutional networks, the RGB model, the depth model, and the pseudo depth model. All three models have similar, or in our case, the same 3D CNN architecture. The RGB model and the pseudo depth model take RGB data as input, and the depth model takes depth data as input. Since we will only use the RGB model and the pseudo depth model during inference, our RGBP framework can have enhanced performance even when only RGB inputs are available. The pseudo depth model aims to output the depth features, even though its input is the RGB data. Such learning is called "representation learning" [44], where a model aims to directly return the representation of the data of modality $A$ from the data of modality $B$. This way, the pseudo depth model can supersede the depth model in many fusion architectures without the need for depth input data while maintaining the benefits of multimodal fusion.

To achieve this, as illustrated in Figure 2 (d), we train the intermediate outputs of the pseudo depth model to be consistent with those of the depth model. If some inconsistency is found, the inconsistency will result in the *feature similarity loss*, which is defined in section 3.2. The gradients of such loss will flow through the feature extractor of the pseudo depth model and modify its parameters.

Therefore, our method involves multimodal training since the depth model directly learns from the depth input, and the pseudo depth model is trained to learn the depth model's output representations. Our method also realizes unimodal inference since it only uses the RGB model and the pseudo depth model during testing.

## 3.1 RGBP Framework

Since the pseudo depth model is trained to utilize the RGB input to produce the representations of the depth data, it can supersede the depth model during inference. Such an idea can be realized in our proposed three-stage framework, which can be summarized as "train an RGBD model," "train a pseudo depth model as a substitution," and "substitute in the RGBD," respectively. In training stage 1, we first train an RGB model and a depth model, with each of them consisting of a feature extractor and a classifier. Then, we fuse the outputs of their feature extractors. The fusion can be at the decision level or elsewhere, and as illustrated in Figure 2 (a), the dotted lines denote all the possible paths for fusion. Then, the fusion model (colored in green in Figure 2) is supervised by the cross-entropy loss denoted as $L_{CE}$, and the red lines represent the paths the gradients of the loss backward will flow through. In this stage, only the fusion model is trainable, and both the RGB model and the depth model are frozen.

In training stage 2, the pseudo depth model utilizes the RGB data to learn similar representations of the depth data. As illustrated in Figure 2 (b), we use the *feature similarity loss*, denoted as $L_{FS}$, to make the representations output by the pseudo depth feature extractor conform with those output by the depth feature extractor. Since the depth feature extractor is used as a paragon, it is frozen to avoid being affected by the poor representations from the pseudo depth feature extractor. The classifier of the pseudo depth model is a copy of the depth model's classifier, and it is not affected in this stage. Therefore, the gradients of the loss backward will only flow to and modify the feature extraction layers in the pseudo depth model. The performance of the pseudo depth model can be evaluated by $L_{CE}$, while we do not let the classification loss produce any backward gradients.

In stage 3, the model for inference is assembled by the models trained in the two training stages. After the pseudo depth feature extractor has learned to play the role of the depth feature extractor well, we use the pseudo depth model to substitute the depth model in the fusion model from stage 1. Then, the resulting RGBP model can be either directly used for inference or slightly finetuned for better performance before usage.

The differences between our method and the MTUT [1] lie in two aspects. First, two modalities of the MTUT exchange information only by the loss function, while our RGBP framework generates a pseudo depth model to realize various modality fusions. Second, when both RGB and depth data are available, predictions from two modalities of MTUT can only be fused at the score level. In contrast, our method allows fusion to happen at many stages in an end-to-end manner.

## 3.2 Feature Similarity Loss

Ideally, though given the RGB data as the input, the pseudo depth model could extract features and make predictions as if the given input were the depth data. Therefore, apart from the class label, which can enforce supervision on the class scores, the intermediate outputs of the depth model provide timely supervision as well. This supervision is imposed by the *feature similarity loss*, which measures the similarity between the in-depth feature maps of the depth model and those of the pseudo depth model. Then, during loss backward, we apply its gradients to the pseudo depth model alone to avoid degrading the depth model's capability of producing representations.

Our measurement of similarity is generic and adaptive to the specific architecture of the applied model. In general, it has two components, which measure the difference in the correlations of the feature maps and the absolute consistency of the feature maps, respectively.

Since the outputs of the in-depth layers contain high-level features, or in other words, the semantic information [11], one level of similarity between the feature maps returned by different models can be evaluated by their semantic closeness. Here, we employ the method of measuring their semantic closeness in the way proposed by MTUT [1]: Let $F^{depth}$, $F^{pseudo} \in R^{W\times H\times T\times C}$ represent two feature maps from the depth model and the pseudo depth model, each of which has width W, height H, temporal dimension size $T$, and channel number C. Then, by reshaping $R^{W\times H\times T\times C}$ into $R^{D\times C}$ where $D = W \times H \times T$, we express the spatiotemporal information in one dimension of size $D$. By encouraging the elements in the reshaped $F^{depth}$ and $F^{pseudo}$ to have similar correlation patterns, we are expecting two feature maps to have analogous semantic representations. The correlation of a feature map of modality $m$ is calculated by

$$corr(F^m) = \hat{F}^m \, \hat{F}^{mT} \in R^{D\times D} \tag{1}$$

where $\hat{F}^m$ is the feature map obtained after the reshaping, standardization, and normalization of the original feature map $F^m$. Specifically, for the element $f^m_{d,\ c}$ at the position $(d, c)$ of $F^m \in R^{D\times C}$, it is first calculated by $f^{m\,\prime}_{d,c} = \frac{f^m_{d,c}-\mu_d}{\sigma_d}$, where $\mu_d$ and $\sigma_d$ are the mean and standard deviation of the data in the row the element locates at. Then, the standardized element is divided by the $L2$ norm of the row it locates at, resulting in $\hat{f}^m_{d,c} = \frac{f^{m\,\prime}_{d,c}}{\left\|f^{m\,\prime}_d\right\|_2}$.

Then, through minimizing the difference between the correlation of two feature maps calculated by $\left\|corr\left(F^{depth}\right) - corr\left(F^{pseudo}\right)\right\|^2_F$, we encourage the pseudo depth model to have the same understanding of the input as the depth model does. Semantic closeness, as discussed above, measures the level of similarity on a loose level. Meanwhile, a tighter level of similarity is achieved by forcing the feature maps output by two models to be the same. Through minimizing $\left\|F^{depth} - F^{pseudo}\right\|^2_F$, two corresponding feature maps will

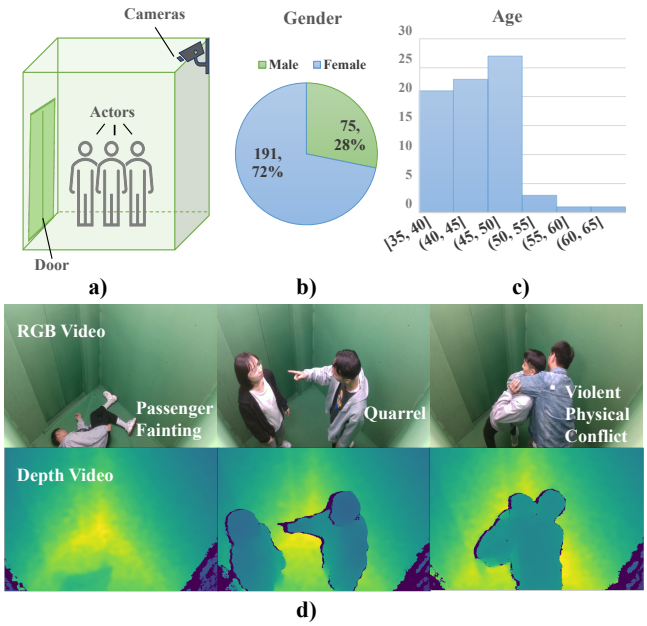



**Figure 3: Data collection information of the Elevator RGBD Dataset. (a) The elevator cabin for data collection, composed of a green screen as the background, an RGBD camera, and a door. (b) The gender distribution of the actors. (c) The age distribution of the actors. (d) Some examples of RGB and depth frames in the Elevator RGBD Dataset.**

develop element-wise analogousness. Combing the above two measurements of similarity together, we define the *feature similarity loss as*

$$L_{FS} = \sum_{p}^{P} \frac{1}{D \times D} \lambda_p \left\| corr\left(F_p^{depth}\right) - corr\left(F_p^{pseudo}\right) \right\|_F^2 + \sum_{q}^{Q} \frac{1}{W \times H \times T \times C} \lambda_q \left\| F_q^{depth} - F_q^{pseudo} \right\|_F^2 \quad (2)$$

where B denotes the batch size, and $F_p^{depth}$ and $F_q^{pseudo}$ refer to the feature maps at layer p of the depth model and that at layer q of the pseudo depth model. Since a CNN can have many layers and not all of them produce feature maps of the same importance, we use $\lambda_p$ and $\lambda_q$ to denote how much we emphasize the correctness of the feature maps of layer $p$ and $q$, respectively. $\lambda_p$ and $\lambda_q$ are determined by the specific architecture of the model, as we will discuss in section 4.3.

In addition, Q and P are two sets that contain the indexes of certain layers, with the layers in P contributing to the first component of $L_{FS}$ and the layers in Q contributing to the second component. Since the feature maps from the shallow layers have not learned the abstract information of the inputs, they are evaluated by their semantic closeness to the corresponding depth feature maps. By comparison, the deep layers are better at learning the high-level features. Therefore, a well-trained pseudo depth model is expected to produce the same representation as the depth model does, as the second component of $L_{FS}$ requires.

Moreover, when calculating $L_{FS}$, for each pair of the feature maps, the loss they produce is divided by the number of elements in each feature map. Specifically, the coefficient $D \times D$ regularizes the first component of $L_{FS}$ and $W \times H \times T \times C$ regularizes the second component.

Our *feature similarity loss* is justified by our experiments in section 4.3. When the pseudo depth model is supervised by $L_{FS}$, its feature maps get closer to those from the depth model, and it also obtains better and better performance in classification.

**Table 1: The 5 events in the Elevator RGBD Dataset and their corresponding descriptions and number of samples.**

| Event | Description | # Sample |
|---|---|---|
| Normal | This class contains some normal behaviors that will not threaten the safety of the elevator and passengers, e.g., intimate interaction, chatting. | 14896 |
| Passenger fainting | This class contains some passengers that seem to need rescue, e.g., falling over, lying on the floor. | 2128 |
| Quarrel | This class contains some passengers that seem to exchange angry words. | 2128 |
| Force opening the door | This class contains one or more people trying to open the elevator door at an inappropriate time that will threaten elevator safety. | 2128 |
| Violent physical conflict | This class contains some behavior like fighting or hijacking, which will threaten the safety of passengers. | 4256 |

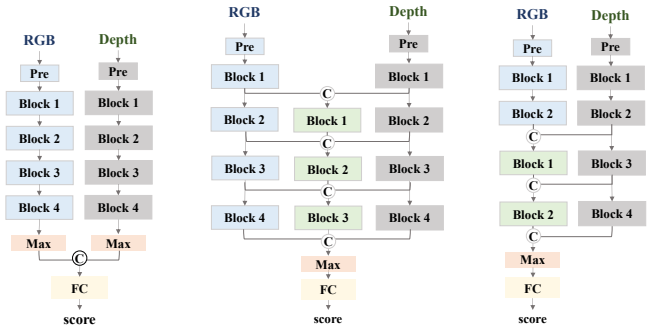
## 4 EXPERIMENT

This section evaluates our method by training three architectures on the Elevator RGBD Dataset, designed for abnormal events detection under the elevator scenario. Each of the three architectures employs a distinct fusion method, and we evaluate these RGBD models using classification accuracy and weighted F1 score. We also compare their performance against other methods for "multimodal training, unimodal inference," such as MTUT [1] and a two-stage method that utilizes depth estimation.

### 4.1 Elevator RGBD Dataset

The Elevator RGBD Dataset focuses on the abnormal events that can happen in the elevators. This dataset contains five common event classes, as listed in Table 1.

Each sample contains two types of data collected by an RGBD camera (Intel Realsense D435 [14]): the RGB video with a length of around 10 seconds and 1920×1080 resolution, and the depth video with the same length, frame rate, and resolution as the RGB video. Considering the redundant information in continuous frames, we set the frame rate as five fps. The RGB and depth data are recorded in a simulated elevator cabin with the green screen as the background, a door, and an RGBD camera at a top corner, as shown in Figure 3 (a).




a) Intermediate Late Fusion b) Intermediate Center-aligned Fusion c) Intermediate Left-aligned Fusion

**Figure 4: Three RGBD architectures we use to test our method. This figure uses the same color system as Figure 2, where the RGB (blue) blocks belong to the RGB feature extractor, and the depth (grey) blocks belong to the depth feature extractor.**

There are 266 actors participating in the recording process, and each actor performs the five events eight times as the main character in his/her series of five events following a designed script. We organize these videos and split the training, validation, and test set with the ratio of 7:1:2. Meanwhile, we control the splits to avoid that one identity presents in more than one set.

The age and gender distributions of the actors are shown in Figure 3 (b), (c). Due to the crowded space in elevators, facial contents are not essential. Moreover, since we recruit actors randomly, the distribution of actor attributes is expected to be close to the real passenger distribution.

## 4.2 Comparison of Different Architectures in Training Stage 1

In our experiments, we build the RGBD models by fusing the RGB and depth models with three methods: the intermediate late fusion method [36] ($RGBD_{Late}$), the intermediate center-aligned fusion method [17] ($RGBD_{Center}$), and the intermediate left-aligned fusion method ($RGBD_{left}$). Figure 4 shows the diagrams of these three architectures, and the blue, grey, and green colors represent the RGB blocks, depth blocks, and fusion blocks, respectively.

In the intermediate late fusion architecture in Figure 4 (a), the fully connected layer of the RGB and depth models are removed. The output feature maps following the global max-pooling layers are concatenated. The global max-pooling reduces each feature map in its width, height, and temporal dimension to 1, returning a feature vector with the same number of elements as the channel number. Then, the output of the pooling layer is put into a fully connected layer, which is the only trainable layer in this model.

In the intermediate center-aligned fusion model in Figure 4 (b), the feature maps from an RGB block, a depth block, and a fusion block (if applicable) are concatenated and put into the next fusion block. After the third fusion block, there are a max-pooling and a fully connected layer. The fusion blocks and the fully connected layer are trainable.

**Table 2: Specification of the blocks and FC layers in three RGBD architectures. *in*, *out*, *k*, *s*, and *p* refer to the in-channel number, out-channel number, size of kernel, stride, and padding, respectively. ResBlock refers to the 3D residual block in the Res3D network [15], and without specified otherwise, the value of stride is 2 and that of padding is 1.**

| | | $RGBD_{Late}$ | $RGBD_{Center}$ | $RGBD_{Left}$ |
|---|---|---|---|---|
| RGB Blocks | Pre | conv (in=3, out=16, k=7, s=2, p=3) max pool (k=3, s=2, p=1) | | |
| | 1 | ResBlock (32, 3), s=1 | | |
| | 2 | ResBlock (64, 4) | | |
| | 3 | ResBlock (128, 6) | | — |
| | 4 | ResBlock (128, 3) | | — |
| Depth Blocks | Pre | conv (in=1, out=16, k=7, s=2, p=3) max pool (k=3, s=2, p=1) | | |
| | 1 | ResBlock (32, 3), s=1 | | |
| | 2 | ResBlock (64, 4) | | |
| | 3 | ResBlock (128, 6) | | |
| | 4 | ResBlock (128, 3) | | |
| Fusion Blocks | 1 | — | ResBlock (128, 4) | ResBlock (128, 6) |
| | 2 | — | ResBlock (512, 6) | ResBlock (256, 3) |
| | 3 | — | ResBlock (1024, 3) | — |
| FC | | fc 256 | fc 1024 | fc 384 |

**Note:** ResBlock (n, m)=$\begin{bmatrix} 3\times3\times3 & n \\ 3\times3\times3 & n \end{bmatrix} \times m$

In the intermediate left-aligned fusion model in Figure 4 (c), the feature maps from a depth block are concatenated with the feature maps from an RGB block or a fusion block. The concatenated result is put into the next fusion block or the max-pooling layer. The output of the pooling layer is put into a fully connected layer, and the fusion blocks and the fully connected layer are trainable.

**Implementation Details**: we adopt the Res3D network [15] as the backbone in our architectures, and we shrink its size for faster inference. In all these architectures, the feature maps from block 1 are regulated by the first component of $L_{FS}$, and those from block 2, 3, 4 are regulated by the second component. That is, we set $P = \{1\}$, and $Q = \{2, 3, 4\}$. The detailed parameter settings in each architecture are specified in Table 2.

In all the experiments, we use a batch size of 16 containing 16-frames clips. The frames are picked by isometric sampling from the RGB and depth videos, with each of them resized to 135 and 240 in width and height, respectively. We use the Adadelta optimizer [43] with $\rho$ of 0.9, $\epsilon$ of $10^{-6}$, and weight decay of $10^{-3}$ and $L_{CE}$ to supervise the training process. We first pre-train the RGB and depth blocks in the RGB and depth models for 11100 iterations and put these blocks into the corresponding positions in these three architectures. Then, these three architectures are trained for 11100

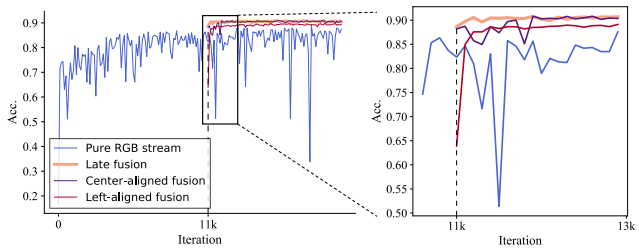



**Figure 5: Comparison of validation accuracy of the RGB model with those of the intermediate late fusion mode ($RGBD_{Late}$), the intermediate center-aligned fusion model ($RGBD_{Center}$), and the intermediate left-aligned model ($RGBD_{Left}$).**

**Table 3: Comparison of the test accuracies of different architectures and methods.**

| Method | Acc. | # Parameters |
|---|---|---|
| RGB model | 83.12 | 8.77M |
| Depth model | 83.04 | 8.63M |
| RGB+Depth | 86.71 | 17.54M |
| $2RGB_{Late}$ | 86.28 | 17.55M |
| $RGBD_{late}$ | 89.08 | 17.54M |
| $2RGB_{Center}$ | 83.80 | 266.01M |
| $RGBD_{Center}$ | 88.37 | 266.00M |
| $2RGB_{Left}$ | 85.74 | 25.79M |
| $RGBD_{Left}$ | 87.85 | 25.78M |

iterations.

**Experiment Results**: Figure 5 shows the validation accuracies of these three architectures and the RGB model during the training process, and Table 3 lists their test accuracies.

To validate that the increased number of model parameters does not cause the improved performance, we also try the two-stream RGB models by replacing the depth blocks with RGB blocks in three RGBD architectures ($2RGB_{Late}$ , $2RGB_{Center}$, $2RGB_{Left}$). These two-stream RGB models are trained from scratch for 22200 iterations. We also try simply adding the output scores from the RGB and depth models to see whether the performance is enhanced (RGB + depth). The results and the number of model parameters are also listed in Table 3.

By simply adding the output scores of the RGB and depth models, we enhance the performance, which becomes better than that of the three two-stream RGB models. The three RGBD models perform better than the corresponding two-stream RGB models and the simple addition result, which validate the effectiveness of the depth information and our architectures.

## 4.3 Generation and Evaluation of the Pseudo Depth

As discussed in section 3.2, we build a pseudo depth model with the same 3D CNN structure as the RGB model and train it with our proposed *feature similarity loss*. To evaluate the pseudo depth

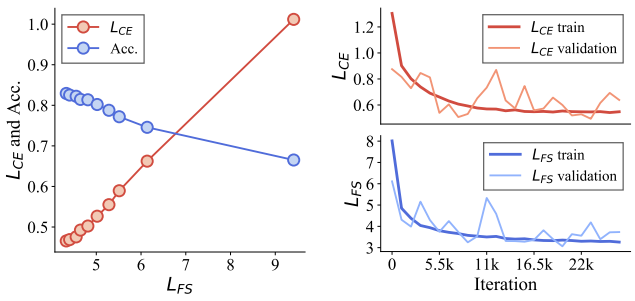



**Figure 6: Correlation between $L_{FS}$, $L_{CE}$, classification accuracy, and training iteration of the pseudo depth model. Left: $L_{CE}$ and Acc. vs. $L_{FS}$. Right: $L_{CE}$ vs. iteration and $L_{FS}$ vs. iteration.**

model's classification capability, we adopt the pre-trained depth classifier with the extracted pseudo depth features as inputs. If the classification loss of this assembled pseudo depth model decreases, the pseudo depth feature extractor is considered to be improving its performance in representation learning.

In the inference stage, we replace the depth blocks with pseudo depth blocks to build RGBP architectures ($RGBP_{Late}$, $RGBP_{Center}$, $RGBP_{Left}$) and change the inputs into the RGB videos.

**Implementation Details:** we apply $L_{FS}$ on each of the feature maps from the four blocks of the depth model and the pseudo depth model, with their $\lambda$ set to 1, 2, 3, 5, respectively. The depth model is frozen, and the learning rate of the pseudo depth model is set to 1. In the inference stage, after the RGBP is assembled, we can directly use it for inference. In our experiment, $RGBP_{Center}$ is directly used, and the fusion model of $RGBP_{Late}$ and $RGBP_{Left}$ are finetuned for 1110 and 4440 iterations, respectively, both having a learning rate of $10^{-3}$.

**Experiment Results:** When training a pseudo depth model supervised by a well-trained depth model via $L_{FS}$ alone, we observe its positive correlation with $L_{FS}$ and its negative correlation with the classification accuracy. That is, on the training set, as $L_{FS}$ decreases, the classification loss of the pseudo depth model decreases, and its classification accuracy improves, as shown in Figure 6. Therefore, we justify that our definition of $L_{FS}$ provides effective supervision.

The feature learning process is visualized in Figure 7. We present the feature maps (averaged over channels and temporal dimension) output by the four blocks in the pseudo depth model after 100, 1100, and 11100 iterations. As the number of iterations increases, the feature maps look more and more similar to that of the depth model in general and look different from that of the RGB model.

Table 4 lists the test accuracies of these three RGBP architectures and their corresponding RGBD architectures. $RGBP_{Late}$, $RGBP_{Center}$, and $RGBP_{Left}$ improve the accuracy with respect to the RGB model by 4.71%, 3.40%, and 3.30%, respectively. Moreover, they improve the F1 score with respect to the RGB model by 4.95%, 3.00%, and 3.49%, respectively. Therefore, all the RGBP architectures have a noticeable better performance than the RGB model and the corresponding two-stream RGB architectures.

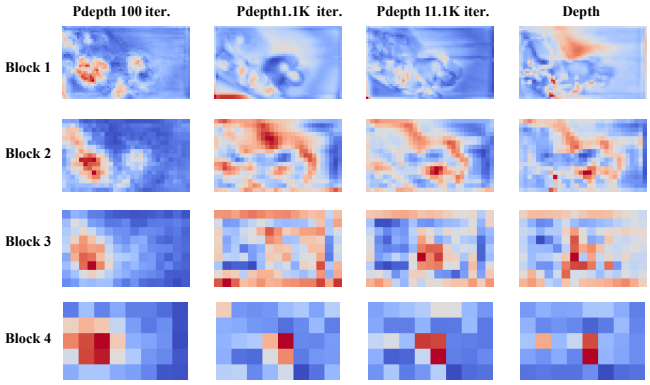



**Figure 7: Visualization of the feature maps from each block of the pseudo depth model after 100, 1100, and 11100 iterations, and the feature maps from the depth and RGB models.**

## 4.4 Comparison with Other Frameworks

We compare our RGBP framework with the other two frameworks shown in Figure 1 (a), (b). We use the same pre-trained RGB and depth models to implement MTUT [1] by ourselves, and we train it with the standard Adadelta optimizer on our Elevator RGBD Dataset. We choose the state-of-the-art depth estimation model [29] on the NYU-Depth V2 dataset [34] and combine it with our three RGBD architectures. The inputs of this depth estimation model are RGB videos, and its output predictions are put into the depth streams of our three RGBD architectures. Then, these three RGBDs models are finetuned for 2200 iterations with a learning rate of $10^{-4}$, whose results are shown in the rows of $DE_{Late}$, $DE_{Center}$, and $DE_{Left}$, respectively. Table 5 lists the performance of these two frameworks on the Elevator RGBD Dataset and the best performance of our RGBP framework. It shows that our RGBP framework improves the accuracy by 1.78% with respect to the MTUT model and 2.43% with respect to $DE_{Late}$. Moreover, our RGBP framework improves the F1 score by 2.20% with respect to the MTUT model and 2.73% with respect to $DE_{Late}$.

As shown in Figure 1 (a), the $DE_{Late}$, $DE_{Center}$, and $DE_{Left}$ use the depth data for training and the estimated depth data for inference. Therefore, part of the final prediction error is contributed by the error coming with the estimated depth data. As shown in Table 5, the performance of $DE_{Late}$ and $DE_{Center}$ are relatively more robust to the error in the estimated depth. One reason accounting for this is that the four RGB blocks in $RGBD_{Late}$ and $RGBD_{Center}$ provide better representations of the input data, which thus neutralize the error in the estimated depth inputs. However, in $RGBD_{Left}$, the features from the depth model are concatenated with those from the RGB model as early as after the second block. Therefore, the error in the depth inputs can influence a large portion of the RGBD model, resulting in poor performance of the $DE_{Left}$.

## 5 CONCLUSION

In this work, we present the RGBP framework, a new method for "multimodal training and unimodal inference," which takes RGB data alone as input while can make a prediction based on both the RGB features and the predicted depth features. The proposed method aims at utilizing rich training data to improve the model performance in the case of limited data, which is valuable in many practical applications. In future work, the proposed technique for generating pseudo depth features from RGB data is expected to applications beyond the current elevator scenario and abnormal event recognition domain.

**Table 4: Performance of different RGBP architectures and the comparison with corresponding RGBD architectures. The *Avg Time* represents the averaged inference time on Nvidia GTX-1080Ti.**

| Method | Acc. | F1 score | Avg Time | FLOPs |
|---|---|---|---|---|
| RGB model | 83.12 | 82.69 | 7.31ms | 3.97G |
| $RGBD_{Late}$ | **89.08** | **88.87** | 14.23ms | 7.22G |
| $RGBP_{Late}$ | **87.83** | **87.64** | 14.95ms | 7.94G |
| $RGBD_{Center}$ | 88.37 | 87.97 | 38.99ms | 28.17G |
| $RGBP_{Center}$ | 86.52 | 85.69 | 43.34ms | 28.89G |
| $RGBD_{Left}$ | 87.85 | 87.53 | 15.66ms | 7.58G |
| $RGBP_{Left}$ | 86.42 | 86.18 | 15.57ms | 8.29G |

**Table 5: Comparison of our RGBP frameworks with other frameworks on the Elevator RGBD Dataset.**

| Method | Acc. | F1 score | Avg Time | FLOPs |
|---|---|---|---|---|
| MTUT | 86.05 | 85.44 | 7.31ms | 3.97G |
| $DE_{Late}$ | 85.40 | 84.91 | 1068.64ms | 604.84G |
| $DE_{Center}$ | 85.17 | 84.64 | 1093.20ms | 625.79G |
| $DE_{Left}$ | 70.56 | 69.72 | 1069.87ms | 605.19G |
| $RGBP_{Late}$ (Ours) | **87.83** | **87.64** | 14.95ms | 7.94G |

## 6 ACKNOWLEDGMENTS

This research was funded by Kunshan Government Research (KGR) Funding in AY 2020/2021.

## REFERENCES

[1] Mahdi Abavisani, Hamid Reza Vaezi Joze, and Vishal M Patel. 2019. Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition With Multimodal Training. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 1165–1174.

[2] Roberto Arroyo, J Javier Yebes, Luis M Bergasa, Iván G Daza, and Javier Almazán. 2015. Expert video-surveillance system for real-time detection of suspicious behaviors in shopping malls. *Expert systems with Applications* 42, 21 (2015), 7991–8005.

[3] Cem Yusuf Aydogdu, Souvik Hazra, Avik Santra, and Robert Weigel. 2020. Multi-modal cross learning for improved people counting using short-range FMCW radar. In *2020 IEEE International Radar Conference (RADAR)*. IEEE, 250–255.

[4] Aaron F. Bobick and James W. Davis. 2001. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence* 23, 3 (2001), 257–267.

[5] Oren Boiman and Michal Irani. 2007. Detecting irregularities in images and in video. *International journal of computer vision* 74, 1 (2007), 17–31.

[6] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. 2019. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8001–8008.

[7] Nicola Conci and Leonardo Lizzi. 2009. Camera placement using particle swarm optimization in visual surveillance applications. In *2009 16th IEEE international conference on image processing (ICIP)*. IEEE, 3485–3488.

[8] Blanca Delgado, Khalid Tahboub, and Edward J Delp. 2014. Automatic detection of abnormal human events on train platforms. In *NAECON 2014-IEEE National Aerospace and Electronics Conference*. IEEE, 169–173.

[9] Oscar Deniz, Ismael Serrano, Gloria Bueno, and Tae-Kyun Kim. 2014. Fast violence detection in video. In *2014 international conference on computer vision theory and applications (VISAPP)*, Vol. 2. IEEE, 478–485.

[10] Yuan Gao, Hong Liu, Xiaohu Sun, Can Wang, and Yi Liu. 2016. Violence detection using oriented violent flows. *Image and vision computing* 48 (2016), 37–41.

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2414–2423.

[12] VK Gnanavel and A Srinivasan. 2015. Abnormal event detection in crowded video scenes. In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (Ficta) 2014*. Springer, 441–448.

[13] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. 2019. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3828–3838.

[14] Anders Grunnet-Jepsen, John N Sweetser, Paul Winer, Akihiro Takagi, and John Woodfill. 2018. Projectors for Intel® RealSense™ Depth Cameras D4xx. *Intel Support, Interl Corporation: Santa Clara, CA, USA* (2018).

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 6546–6555.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[17] Di Hu, Chengze Wang, Feiping Nie, and Xuelong Li. 2019. Dense multimodal fusion for hierarchically joint representation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3941–3945.

[18] Ahmad Jalal, Yeon-Ho Kim, Yong-Joong Kim, Shaharyar Kamal, and Daijin Kim. 2017. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern recognition* 61 (2017), 295–308.

[19] Chunhua Jia, Wenhai Yi, Yu Wu, Hui Huang, Lei Zhang, and Leilei Wu. 2020. Abnormal activity capture from passenger flow of elevator based on unsupervised learning and fine-grained multi-label recognition. *arXiv preprint arXiv:2006.15873* (2020), arXiv–2006.

[20] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1725–1732.

[21] Roberto Leyva, Victor Sanchez, and Chang-Tsun Li. 2014. Video anomaly detection based on wake motion descriptors and perspective grids. In *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 209–214.

[22] Li Liu and Ling Shao. 2013. Learning discriminative representations from RGB-D video data. In *Twenty-third international joint conference on artificial intelligence*.

[23] Amira Ben Mabrouk and Ezzeddine Zagrouba. 2017. Spatio-temporal feature using optical flow based distribution for violence detection. *Pattern Recognition Letters* 92 (2017), 62–67.

[24] Amira Ben Mabrouk and Ezzeddine Zagrouba. 2018. Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications* 91 (2018), 480–491.

[25] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. 2010. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1975–1981.

[26] Hajananth Nallaivarothayan, Clinton Fookes, Simon Denman, and Sridha Sridharan. 2014. An MRF based abnormal event detection approach using motion and appearance features. In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 343–348.

[27] Bingbing Ni, Yong Pei, Pierre Moulin, and Shuicheng Yan. 2013. Multilevel depth and image fusion for human activity detection. *IEEE transactions on cybernetics* 43, 5 (2013), 1383–1394.

[28] Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno García, and Rahul Sukthankar. 2011. Violence detection in video using computer vision techniques. In *International conference on Computer analysis of images and patterns*. Springer, 332–339.

[29] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 2019. 3D Ken Burns effect from a single image. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–15.

[30] Dhanesh Ramachandram and Graham W Taylor. 2017. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine* 34, 6 (2017), 96–108.

[31] Aravinda S Rao, Jayavardhana Gubbi, Sutharshan Rajasegarar, Slaven Marusic, and Marimuthu Palaniswami. 2014. Detection of anomalous crowd behaviour using hyperspherical clustering. In *2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 1–8.

[32] Nida Rasheed, Shoab A Khan, and Adnan Khalid. 2014. Tracking and abnormal behavior detection in video surveillance using optical flow and neural networks. In *2014 28th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 61–66.

[33] Guang Shu, Gaojing Fu, Peng Li, and Haiyu Geng. 2014. Violent behavior detection based on svm in the elevator. *International Journal of Security and Its Applications* 8, 5 (2014), 31–40.

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*. Springer, 746–760.

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

[36] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. 2005. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*. 399–402.

[37] Jing Wang and Zhijie Xu. 2015. Crowd anomaly detection for automated video surveillance. (2015).

[38] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*. Springer, 20–36.

[39] Tian Wang and Hichem Snoussi. 2014. Detection of abnormal visual events via global optical flow orientation histogram. *IEEE Transactions on Information Forensics and Security* 9, 6 (2014), 988–998.

[40] Ping Xiao, Maylor KH Leung, and Kok Cheong Wong. 1996. Eleview: An Active Elevator Monitoring Vision System.,.. In *MVA*. 253–256.

[41] Dan Xu, Rui Song, Xinyu Wu, Nannan Li, Wei Feng, and Huihuan Qian. 2014. Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. *Neurocomputing* 143 (2014), 144–152.

[42] B Yogameena and K Sindhu Priya. 2015. Synoptic video based human crowd behavior analysis for forensic video surveillance. In *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*. IEEE, 1–6.

[43] Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).

[44] Hao Zhu, Man-Di Luo, Rui Wang, Ai-Hua Zheng, and Ran He. 2021. Deep audio-visual learning: A survey. *International Journal of Automation and Computing* (2021), 1–26.

[45] Songhao Zhu, Juanjuan Hu, and Zhe Shi. 2016. Local abnormal behavior detection based on optical flow and spatio-temporal gradient. *Multimedia Tools and Applications* 75, 15 (2016), 9445–9459.

[46] Yujie Zhu and Zengfu Wang. 2016. Real-time abnormal behavior detection in elevator. In *Chinese Conference on Intelligent Visual Surveillance*. Springer, 154–161.