IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 20, 2021, accepted December 23, 2021, date of publication January 12, 2022, date of current version January 19, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3142247

An Examination on Autoencoder Designs for Anomaly Detection in Video Surveillance

ERNESTO CRUZ-ESQUIVEL, (Member, IEEE), AND ZOBEIDA J. GUZMAN-ZAVALETA
Department of Computing, Electronics and Mechatronics, Universidad de las Américas Puebla, Cholula 72810, Mexico

Corresponding author: Zobeida J. Guzman-Zavaleta (zobeida.guzman@udlap.mx)

This work was supported in part by Consejo Nacional de Ciencia y Tecnología (CONACyT) student grant 975079, and in part by Fundación Universidad de las Américas Puebla (UDLAP) BAUI grant, and with infrastructure from Laboratorio Nacional de Supercómputo del Sureste de México LNS project 201902064C.

ABSTRACT Current anomaly detection methods for video surveillance find anomalies effectively enough; however, it comes at a high computational cost and specific hardware resources demanding. In counterpart, other video analysis tasks such as video action recognition now employ techniques that reduce the need for higher computational cost. Some of those techniques can be helpful for video anomaly detection. Therefore, this paper explores the effectiveness of the potential concepts of distillation and joint spatiotemporal training, adapted to two novel convolutional autoencoder architectures for anomaly detection in video surveillance. Our experimental results show the feasibility of reducing the computational resources requirements with smaller architectures (only $6K$ trainable parameters), competing and outperforming current methods in challenging benchmarks.

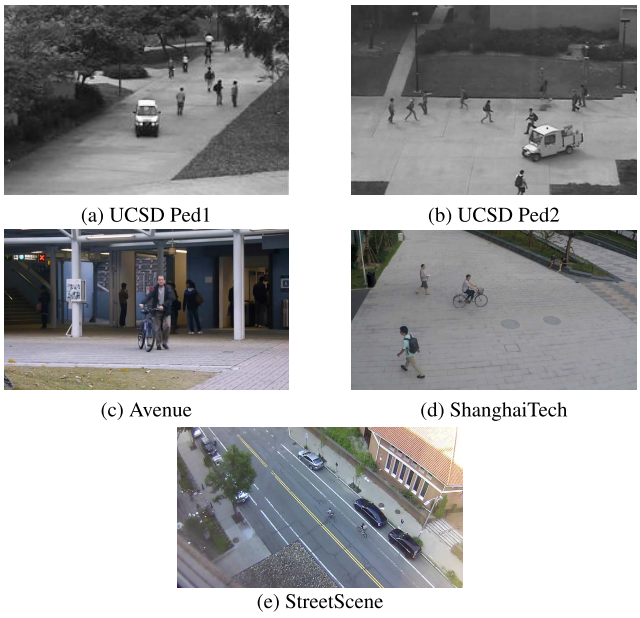
INDEX TERMS Anomaly detection, spatiotemporal features, video surveillance.

I. INTRODUCTION

Video surveillance is a popular tool in private and public security systems worldwide. Nowadays, video monitoring systems generate a high amount of data requiring an automatic analysis that intelligent surveillance systems could perform. Among other tasks, those intelligent systems should detect anomalies automatically. Anomalies are unusual context-dependent situations. In some cases, anomalies in surveillance videos could be harmless scenarios; for instance, someone bumping into another person or walking in the wrong direction. However, anomalies could be in life-threatening situations, like a severe accident or a violent crime. In the last few years, intelligent video surveillance anomaly detection has been influenced by deep learning. Autoencoder models have been a constant in this area because of their semi-supervised training capacity. Their reconstruction loss is used as an anomaly score when the system is trained with normal data. Autoencoders and other deep learning architectures have significantly improved the accuracy of anomaly detection systems (*e.g.* [1], [2], and [3]). Moreover, there exist different public datasets helping in the generalization ability of anomaly detection methods. Figure 1 shows some examples of anomaly datasets.

Nevertheless, most proposed architectures have enormous, prohibitive, or inadequate response times for systems with limited computational resources. Therefore, the motivation of this paper is the need for more efficient models that can achieve satisfactory results for the anomaly detection task in video surveillance. That is the main reason for analyzing current models for detection and other recognition tasks to propose improvements in efficiency without sacrificing accuracy in anomaly detection. For instance, some proposals in video action recognition task achieve outperforming results to extract spatial and temporal features potentially useful for anomaly detection. Such is the case of the ST-AE [8], the S3D-G [9], and the D3D [10] models. The ST-AE model uses parallel spatiotemporal autoencoders trained with grayscale frames. The S3D-G combines 2D with 3D convolutional layers with good accuracy results and less computational resources than only using the 3D convolutional layers of the popular I3D model [11]. The S3D-G model uses two networks to extract spatiotemporal features correctly. In counterpart, the D3D model uses the Knowledge Distillation technique to train a single network model to extract spatiotemporal features accurately from videos.

The associate editor coordinating the review of this manuscript and approving it for publication was Szidónia Lefkovits.



(a) UCSD Ped1 (b) UCSD Ped2

(c) Avenue (d) ShanghaiTech

(e) StreetScene

FIGURE 1. Examples of frames with anomalies in different public datasets. (a) and (b) are anomalies in the UCSD Ped1 and Ped2 datasets [4], both contain frames in grasycale with lower definition. (c) An anomaly in the CUHK Avenue dataset [5]. (d) A biker is an anomaly in the ShanghaiTech Campus dataset [6]. (e) Example of an anomaly in the StreetScene dataset [7].

The S3D-G and D3D designs are helpful to improve the speed-accuracy ratio for video analysis models. However, those ideas have not been applied to the anomaly detection task or as an auxiliary for autoencoder architectures in intelligent video surveillance. In accordance, this paper examines different strategies to design an autoencoder architecture suited for the anomaly detection task in video surveillance. Thus, the analysis of autoencoders' design focuses on ideas that have worked in similar video analysis problems. In particular, the evaluation comprises three different known strategies:

- A combination of 2D and 3D convolutional layers of a Top-Heavy model, similar to the S3D-G approach, comparing it versus only typical 3D convolutional layers.
- The incorporation of a distillation process to train a network to extract spatiotemporal features, inspired by the D3D model.
- The joint of two networks in the training process extracting spatial and temporal features in a single network as observed in the ST-AE model.

This paper's innovation is the presentation of two novel techniques based on the concepts of S3D-G, D3D, and ST-AE. Two custom training losses are presented with novel techniques for their training. Two architectures were presented to test the novel presented techniques. Convolutional autoencoders are used as the base of the presented architectures.

The benefit of using a convolutional autoencoder relies on its simplified structure using a semi-supervised training process suited to detect anomalies in surveillance videos.

The organization of this document is the following. The related work for video analysis techniques is presented in Section II. Section III describes the models for the assessment. After, Section IV discuss the examination results, and finally, Section VI contains the reached conclusions.

II. RELATED WORK

Deep Learning techniques have been a constant for analyzing images and videos in the past decade. In the case of images, 2D convolutional layers demonstrated to be appropriate for extracting characteristics that help computer algorithms to understand the image content. Nevertheless, in videos, 2D convolutional layers obtain relevant characteristics only for the current frame without considering the temporal relation of past and future frames. Therefore, researchers started applying 2D convolutions to a set of consecutive frames instead of filtering individual frames to solve this problem. An example of this is the model R2D presented by Tran *et al.* in [12]. However, R2D is not remarkably effective as consecutive frames, treated as image channels, collapse after the first convolution. Therefore, Ji *et al.* in [13] added a third dimension to the convolutional layer to consider the temporal changes of the frames with application in video action recognition task. Then, Zhao *et al.* in [8] employed 3D convolutional layers for video anomaly detection task. Although a third dimension has advantages in their results, it also increases the models' complexity and requires more training epochs.

2D convolutional layers were not completely discarded, and they were applied for the Two-Stream approach for video action recognition by Zisserman and Simonyan [14]. The use of two networks for analyzing video is an idea that comes from biological motivations, and that was observed long ago by researchers [15]. Because of its experimental results, the Two-Stream approach gained popularity in video analysis tasks, mostly in video action recognition (*e.g.* [9], [11], [16]) and video surveillance anomaly detection (*e.g.* [1], [2], [17]). Two-Stream approach is combined with other techniques to improve its results, such as Gaussian Mixture in GMFC-VAE [1] and GMM-DAE [17]. Appearance and Motion DeepNet (AMDN) [18] used a third stream of pixel early fusion to increase the model's detection accuracy. However, the Two-Stream approach has two main drawbacks: the use of more than one network for inference and the need for a pre-processing technique, like optical flow, which could be costly for real-time applications.

Carrera and Zisserman improved the 2D convolutional layers of the Two-Stream model into the Inflated 3D model; that is, the I3D model [11] for action recognition. Although this model is accurate for action classification, it needs a lot of training data to achieve the desired accuracy. In cases like the implicit two-path AutoEncoder (ITAE) [19], the Two-Stream idea has been applied using two encoders and different frame update ratios to obtain spatial and temporal features. Posteriorly, Xie *et al.* [9] used the I3D as a base for two testing models, Top-Heavy I3D and Bottom-Heavy I3D. Those consist of I3D models that combine 2D convolutional layers

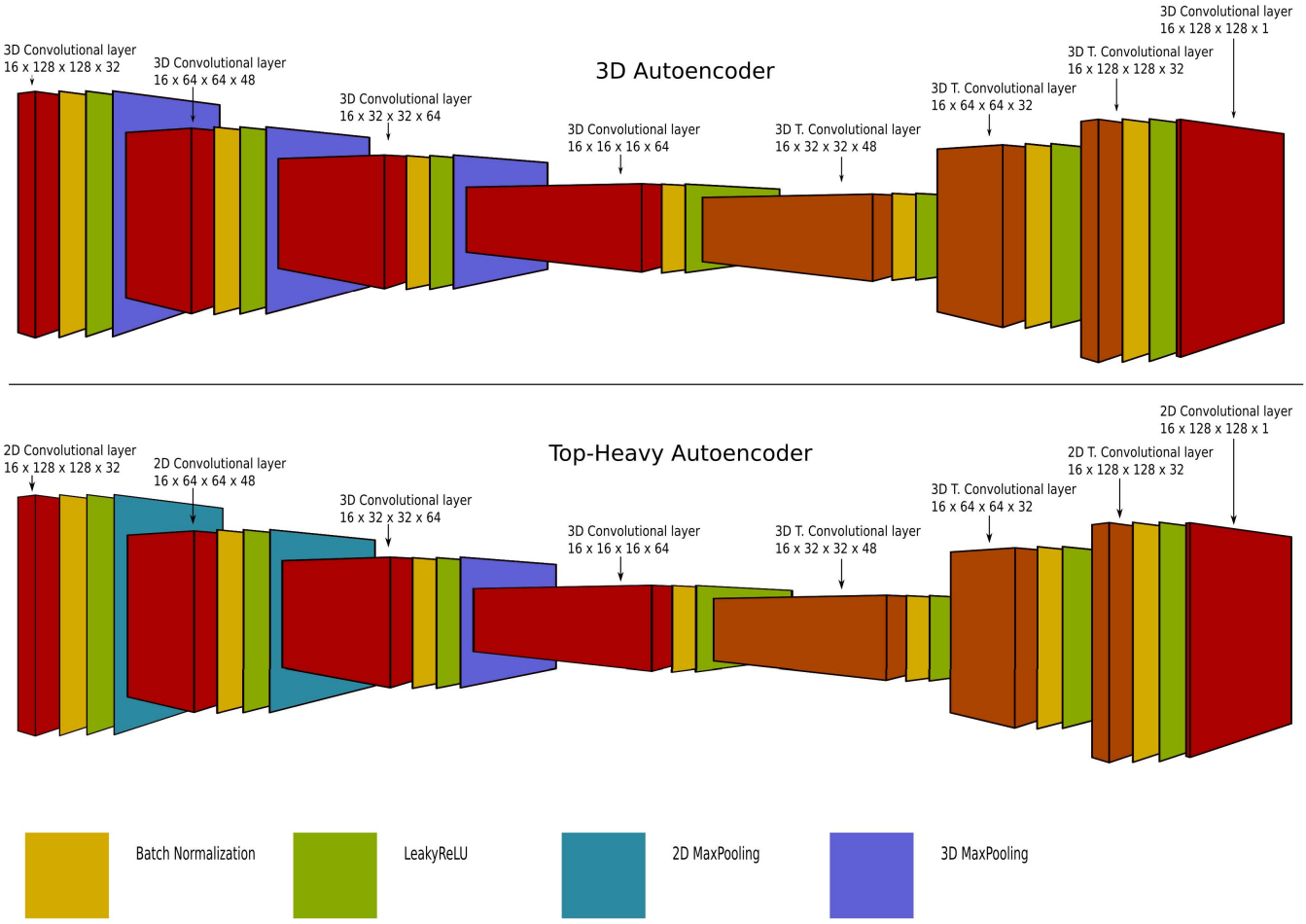


FIGURE 2. The 3D autoencoder and top-heavy autoencoder structures in the assessment.

with 3D convolutional layers in different orders. The first layers are 2D convolutions in the Top-Heavy model, while the last layers are 3D convolutions. In Bottom-Heavy, 3D convolutional layers come first, ending with 2D convolutions. Top-Heavy I3D obtained better results than Bottom-Heavy and closer results to the original I3D, so it was used as base for their S3D-G model [9]. However, S3D-G had difficulties to extract spatiotemporal features by using a single network. Therefore, the Two-Stream approach improved the results of S3D-G in video action recognition. The S3D-G ideas can be used in architectures that use 3D convolutions to process spatio-temporal features.

Moreover, to improve the feature extraction problem, Stroud *et al.* [10] presented the D3D model. D3D is an action classification model based on the S3D-G architecture, and it uses the Knowledge Distillation technique for its training. The design of Knowledge Distillation transmits knowledge from large size networks to smaller size networks. In D3D, the distillation goes from the temporal stream knowledge to the spatial stream. As a result, a single network extracts spatiotemporal features similar to a Two-Stream network. For training a D3D model, the temporal network has to be pre-trained; that is, an S3D-G model with the optical flow is used normally for the temporal stream learning. Then, the fitted Temporal S3D-G becomes an auxiliary for the Spatial S3D-G stream training. The learning process of the spatial network uses its misclassification error combined with the misclassification error of the already trained Temporal S3D-G. Finally, in the inference process, only the Spatial S3D-G is used. Even D3D model demonstrates that the distillation procedure can be applied to train spatiotemporal models, it has not been applied to other video tasks besides the action recognition.

A. VIDEO SURVEILLANCE ANOMALY DETECTION

To adapt the Two-stream approach from video action recognition task to anomaly detection in video surveillance, works in the state-of-the-art use autoencoder structures instead of Convolutional Neural Networks (CNNs). Using autoencoders allows training them using only the normal data, which is essential due videos with anomalies are scarce. More complex techniques try to improve the stochastic understanding of the video surveillance data, such as Variational Autoencoders [1]; but these might also present some difficulties for

their training [20]. Other generative methods, like the Adversarial Discriminator [2], use a pair of Generative Adversarial Networks (GANs) that learn to generate optical flow frames from RGB color scale frames and RGB frames from optical flow frames. The anomalies in the surveillance videos are found when the models are unable to correctly generate the optical flow or the RGB color scale frames. Although Adversarial Discriminator's approach obtains close to 1 AUC-ROC results for the UCSDPed 1 and 2, training two complete GANs in parallel demands a high computational resources availability. More recent techniques for video surveillance anomaly detection include the use of social interaction for generating the temporal stream [21] and the use of heatmaps to improve result analysis [22].

The potential strategies presented in S3D-G, Top-Heavy model, and in the D3D distillation technique have not been proved in video surveillance anomaly detection nor in combination with autoencoders. Nevertheless, a similar approach to the D3D was tested by Zhao *et al.* in the Spatiotemporal Autoencoder (ST-AE) model training [8] for video surveillance anomaly detection. Their ST-AE model learned using a parallel training: a reconstruction and a predictive stream. ST-AE uses one encoder and two decoders. It is expected that the decoder learns to extract both type of features and only one of the ST-AE streams is used for inference. The training used in ST-AE model was not tested for other configurations like optical flow and grayscale streams combination.

III. DESCRIPTION OF DISTILLATION AND JOINT SPATIOTEMPORAL TRAINING CONCEPTS

This work explores and adapts concepts with outperforming results in other video analysis tasks by two assessments: 1) analyzing the convenience of a Top-Heavy design, including a combination of 2D and 3D convolutional layers against only 3D convolutions in autoencoder architectures. 2) Comparing two different training processes, the one that uses a novel distillation technique with the anomaly score versus the one that incorporates two autoencoders with the spatial and the temporal features specialization. In the following subsections, we include a description of each strategy and its consequent incorporation in the autoencoder design.

A. THE TOP-HEAVY DESIGN

The base of the Top-Heavy autoencoder design in this assessment is the Top-Heavy I3D model tested by Xie *et al.* [9]. A Top-Heavy autoencoder includes 2D and 3D convolutional layers in the encoder and 3D and 2D transpose convolutional layers in the decoder. The benefit of combining the different dimensions of convolutions is to obtain better results while needing less training data, which is essential as video surveillance anomaly datasets are smaller than the ones available in other tasks, like video action recognition. Also, 2D convolutional layers are faster than 3D convolutional layers and less prone to overfit. Therefore, the Top-Heavy autoencoder is easier to train, faster, and smaller than a standard 3D autoencoder. The architecture in this study is smaller than the observed in the Top-Heavy I3D. Figure 2 shows the structure of the 3D autoencoder and the Top-Heavy autoencoder. First, the Top-Heavy encoder contains two 2D convolutional layers and two 3D convolutional layers. Then, the decoder includes one 3D transpose convolution layer and two 2D transpose convolutional layers. Leaky ReLU is the activation function for the convolutional and the transpose convolutional layers. At the end of the decoder, there is a 3D convolutional layer with the sigmoid function as activation. Finally, a batch normalization layer is applied to reduce the training time. The full description of the Top-Heavy autoencoder is in Table 1.

B. ANOMALY SCORE DISTILLATION IN TRAINING

D3D model can not be applied to detect anomalies in surveillance videos. The reason is the difference between the output of CNNs and Autoencoders. The video action recognition task is a multiclass classification, where the CNN must output the correct class for any input that represents the same frame in any color model. According to that idea, the misclassification error of a CNN trained with the optical flow helps to train a network with grayscale inputs. That is, the input frames are the same, and only their pre-processing differs. However, the case is different in the anomaly detection task; therefore, this paper presents a distillation procedure for training Autoencoders that can extract spatiotemporal features from video data, which we call Anomaly score distillation. The architecture was designed to apply the proposed distillation technique. The Anomaly score distillation procedure is illustrated in Figure 3.

The designed architecture includes two novelties:

1) Combination of 2D and 3D convolutional layers in an Autoencoder model. The Top-Heavy order of the convolutional layers is applied.
2) A novel distillation procedure for Autoencoder architecture training. The Anomaly score distillation procedure uses Autoencoder reconstruction loss instead of misclassification error.

As observed in Figure 3, the temporal network is trained using optical flow of consecutive video frames. The trained temporal network obtains the temporal reconstruction loss for each of the optical flows in the database. The temporal reconstruction loss, L_t, is defined by Equation (1), which is the mean absolute error between the original optical flow frames $x^{(i)}$ and the reconstructed optical flow frames of the temporal stream $f_t(x^{(i)})$.

$$L_t(\theta) = \frac{1}{n}\sum_{i=1}^{N}\left|f_t(x^{(i)}) - x^{(i)}\right| \tag{1}$$

The temporal reconstruction loss contains information on the data's temporal characteristics. Frames with high temporal reconstruction loss contain numerous anomalous temporal characteristics; low temporal reconstruction loss indicates that temporal characteristics are behaving as expected. Therefore, the temporal reconstruction loss is an index of a temporal anomaly for each frame to help the

TABLE 1. The top-heavy autoencoder network structure: Convolutions and MaxPooling layers configuration in encoder and decoder.

Encoder

Layer	Input	Kernel size	Kernel number	Stride/pad	Output
Input	16 x 128 x 128	-	-	-	16 x 128 x 128 x 1
Convolution 1	16 x 128 x 128 x 1	1 x 3 x 3	32	1/1/1	16 x 128 x 128 x 32
MaxPooling 1	16 x 128 x 128 x 32	1 x 2 x 2	-	1/2/2	16 x 64 x 64 x 32
Convolution 2	16 x 64 x 64 x 32	1 x 3 x 3	48	1/1/1	16 x 64 x 64 x 48
MaxPooling 2	16 x 64 x 64 x 48	1 x 2 x 2	-	1/2/2	16 x 32 x 32 x 48
Convolution 3	16 x 32 x 32 x 48	3 x 3 x 3	64	1/1/1	16 x 32 x 32 x 64
MaxPooling 3	16 x 32 x 32 x 64	2 x 2 x 2	-	2/2/2	8 x 16 x 16 x 64
Convolution 4	8 x 16 x 16 x 64	3 x 3 x 3	64	1/1/1	8 x 16 x 16 x 64

Decoder

Layer	Input	Kernel size	Kernel number	Stride/pad	Output
Input	8 x 16 x 16 x 64	-	-	-	8 x 16 x 16 x 64
Transpose Convolution 1	8 x 16 x 16 x 64	3 x 3 x 3	48	2/2/2	16 x 32 x 32 x 48
Transpose Convolution 2	16 x 32 x 32 x 48	3 x 3 x 3	32	1/2/2	16 x 64 x 64 x 32
Transpose Convolution 1	16 x 64 x 64 x 32	1 x 3 x 3	32	1/2/2	16 x 128 x 128 x 32
Transpose Convolution 1	16 x 128 x 128 x 32	1 x 3 x 3	1	1/1/1	16 x 128 x 128 x 1

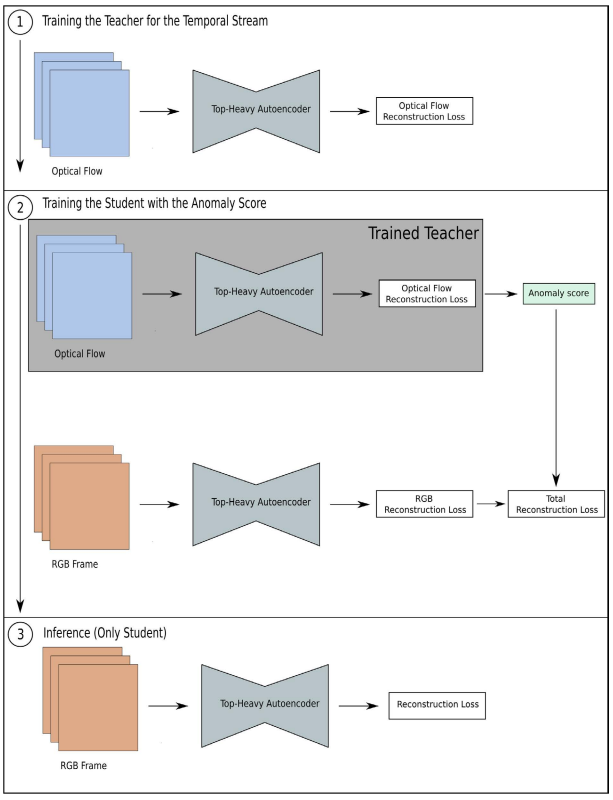


FIGURE 3. Anomaly score distillation procedure. First step: the teacher network is trained using optical flow. Second step: the trained teacher network distills knowledge to the student network. Third step: student network can extract spatiotemporal features in the inference.

spatial network training. Thus, the anomaly score transfers the temporal knowledge to the spatial stream. According to Equation (2), the anomaly score $s(\theta, x^{(i)})$ is the squared difference between the temporal reconstruction loss L_t and the spatial reconstruction loss L_s in the dataset $x^{(i)}$.

$$s(\theta, x^{(i)}) = (L_t(x^{(i)}) - L_s(x^{(i)}; \theta))^2 \tag{2}$$

The anomaly score increases when there exists a substantial difference between the spatial and temporal reconstruction loss. The highest values of the anomaly score are when the temporal network considers a frame highly anomalous, but the spatial network considers it normal, or when the temporal network considers a frame normal and the spatial network considers it highly anomalous. The anomaly score, s, changes according to the temporal reconstruction loss and the spatial reconstruction loss. Therefore, the anomaly score is an absolute anomaly measure, including the temporal and spatial characteristics of the video.

The total loss or distilled loss for the spatial network, $L(\theta)$, is calculated by adding the spatial reconstruction loss and the anomaly score according to Equation (3). The purpose is to train the system to detect anomalous temporal characteristics in the spatial stream. The loss of the spatial network, L_s, is updated with the back-propagation procedure. The anomaly score, s, changes according to the temporal reconstruction loss and the spatial reconstruction loss. In contrast, the temporal reconstruction loss is fixed for each of the frames in the training iterations.

$$L(\theta) = \sum_{i=0}^{N-1} (L_s(x^{(i)}; \theta) + s(x^{(i)}; \theta)) \tag{3}$$

C. JOINT SPATIOTEMPORAL TRAINING

Zhao *et al.* trained their ST-AE model using an auxiliary prediction decoder [8]. The ST-AE combines an encoder and two decoders, one for reconstruction and another for prediction.

The prediction decoder is for the training stage exclusively; the adjustment of the model weights employs the reconstruction loss of both reconstruction and prediction branches. The spatial stream is directly affected by the temporal stream each iteration, but the temporal stream also receives feedback from the spatial stream. Both branches learn the spatiotemporal features in the process. Finally, with the ST-AE model trained, only the spatial branch is used for inference.

In this work, we present the Joint spatiotemporal Training technique. The novelty of the Joint spatiotemporal Training technique is a loss function that teaches a single-stream network the extraction of optical flow and grayscale features. The proposed loss function updates spatial and temporal streams in parallel. Although both streams are trained to obtain spatiotemporal features, the spatial stream is preferred for inference because of the lack of pre-processing of its input.

The Joint spatiotemporal Training technique was applied it to train two complete Top-Heavy autoencoders, a temporal and a spatial one. The spatial network receives either grayscale or RGB color frames as input, while the temporal network receives optical flow frames. First, the reconstruction loss of both autoencoders was combined and used to update both networks' weights simultaneously. Then, the spatial Top-Heavy autoencoder is used for inference as it does not need pre-processing its input. Figure 4 shows the Joint spatiotemporal training in combination with the Top-Heavy architecture.

$$L(\theta_1, \theta_2) = L_s(\theta_1) + (L_s(\theta_1) * L_t(\theta_2)) \tag{4}$$

Equation (4) shows the total loss, $L(\theta_1, \theta_2)$, for the Joint spatiotemporal training technique. The loss of the spatial network, $L_s(\theta_1)$ and the temporal network, $L_t(\theta_2)$, are multiplied. The multiplication of both reconstruction losses is added to the spatial reconstruction loss. Therefore, the spatial reconstruction loss has more weight than the temporal reconstruction loss. Both losses are updated for each iteration using back-propagation.

IV. DATASET AND METRICS IN THE ASSESSMENT

This Section describes the datasets for training and testing for the proposed architecture and its variances. Additionally, we present the evaluation metrics to validate the results.

For our experiments, we used four open datasets for anomaly detection: UCSD [4], CUHK Avenue [5], ShanghaiTech Campus [6], and StreetScene [7]. Figure 1 shows some frame examples from the datasets. The description of those datasets are as follows, but summarized in Table 2:

- UCSD dataset contains two different scenarios, Ped1 and Ped2, divided into video clips from pedestrian walkways. The anomalies included in this context are mainly unknown objects such as bikes and small cars. Ped1 contains 6,800 training frames and 7,200 testing frames with a 238×158 pixels resolution. Ped2 contains 2,550 training frames and 2,010 testing frames with a 360×240 pixels resolution. Ped1 and Ped2 are available only in grayscale.
- CUHK Avenue dataset is also a pedestrian dataset. The anomalies in this context are mainly unexpected pedestrian behavior and some unknown objects like bikes. The Avenue dataset has 15,328 training frames and 15,324 testing frames with a resolution of 640×360 pixels.
- ShanghaiTech Campus dataset contains 13 different scenes with different camera angles and lighting conditions. This dataset has pedestrian paths, and anomalies are mostly non-pedestrians like bikers or someone pushing a stroller. The ShanghaiTech dataset contains 274,515 training frames and 42,883 testing frames with a resolution of 856×480 pixels.
- StreetScene dataset has one single scene with different daylight conditions, a double-direction lane with two pedestrian paths. Cars, bikes, and pedestrians are considered normal. Anomalies are strange behavior of the objects in the scene. This dataset contains 56,847 training frames and 146,445 testing frames with a resolution of 1280×720 pixels.

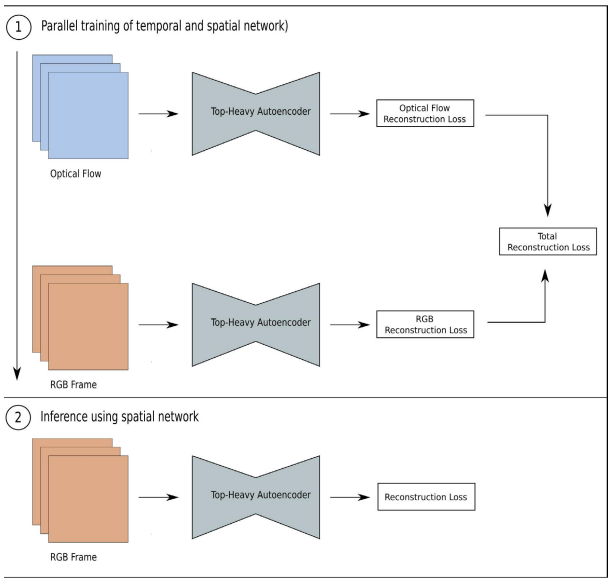


FIGURE 4. Joint spatiotemporal training process. First step: temporal and spatial network are trained simultaneously. Second step: spatial network is used for extracting spatiotemporal features in the inference.

We combined the four training datasets to alleviate overfitting and imbalanced data problems induced by an insufficient number of videos [23]. Based on the formats of the videos, we separate these datasets into two groups: grayscale and RGB. We kept clips from different datasets in separate subfolders for fair comparisons between different benchmarks. Nevertheless, those subfolders are considered part of the more extensive dataset. The grayscale training dataset combines the four provided grayscale datasets (USCD with scenarios Ped1 and Ped2, Avenue, ShanghaiTech, and StreetScene) and the grayscale versions of the testing folders of Avenue, ShanghaiTech, and StreetScene. Video frame

TABLE 2. Number of training, testing and total frames for the selected datasets.

Dataset	Training frames	Testing frames	Total frames	Frames' size
UCSD Ped1	6,800	7,200	14,000	238x158
UCSD Ped2	2,550	2,010	4,560	360x240
CUHK Avenue	15,328	15,324	30,652	640x360
ShanghaiTech	274,515	42,883	317,398	856x480
StreetScene	56,847	146,445	203,292	1280x720

preprocessing includes downsizing to 128 × 128 pixels' resolution apart from the grayscale conversion. The combined grayscale dataset has 356,040 frames in total. The RGB training dataset combines the training folders from only three datasets (Avenue, ShanghaiTech, and StreetScene) containing 346,690 frames with a resolution of 128 × 128 pixels in the RGB color model. Both training datasets include data augmentation using a horizontal flip. In sum, the grayscale training dataset contains 712,080 frames, while the RGB training dataset contains 693,380 frames.

Additionally, we created a TV-L1 Optical Flow [24] version of the existing Grayscale and RGB datasets. For each pair of frames (skipping the first frame for each clip), the TV-L1 obtained a pair of frames (u, v). The TV-L1 optical flow version has 1,422,392 frames for the grayscale dataset and 1,385,184 frames for the RGB dataset. The summarized description of the four final pre-processed datasets is in Table 3. We used the hold-out strategy for the evaluation of the trained models, using the specific frames' sets of training and test provided by each dataset. For this, each dataset kept testing folders separated. We employed the same pre-processing techniques for training and test datasets with the temporal matching between the video frames and their corresponding TV-L1 optical flow frames.

TABLE 3. Number of total frames for the generated training datasets.

Dataset	Training frames
Grayscale dataset	712,080
RGB dataset	693,380
TV-L1 Optical Flow (Grayscale dataset)	1,422,392
TV-L1 Optical Flow (RGB dataset)	1,385,184

A. EVALUATION METRICS

The models' evaluation uses four metrics: AUC-ROC, EER, inference time, and trainable parameters. The Area Under the Curve Receiver Operating Characteristic (AUC-ROC) is a binary classification evaluation metric that measures the performance of the video anomaly detection task. The ROC curve has the True Positive Rate in the y axis and the False Positive Rate in the x axis. The best result for the AUC is 1, meaning all the inputs were classified correctly, while 0.5 means the classifier is acting randomly. The Equal Error Rate (EER) is the point where false positives and false negatives are equal. The lower the EER, the better the performance of the system. The inference time helps us to measure the model speed. We present the average inference time per 1000 frames for each model. Parameters modified and defined by backpropagation during training are considered trainable; thus, the number of trainable parameters is the metric for the model size. The model size will directly affect the memory requirements but not necessarily inference time or capacity.

V. ARCHITECTURES' DESIGN COMPARISON

According with the description in Section III, two architectures are part of this proposed examination: 1) a *3D autoencoder* and 2) a *Top-Heavy autoencoder*. Table 4 lists the number of training parameters and inference time for the methods in our assessment. As expected, the 3D autoencoder is bigger than the Top-Heavy autoencoder that uses 2D convolutional layers. Additionally, there is no substantial difference in size between the grayscale and RGB architectures. The only difference is the number of channels of the input layer, but then the channels from the RGB collapse, so it becomes precisely the grayscale frame's architecture. The difference in the inference time for the Top-Heavy Autoencoder and the 3D Autoencoder, is 0.06 ms. The training of the models was using a Nvidia Tesla V100. The inference time was measured using a system with an Nvidia GeForce RTX 2070 SUPER GPU card. The smaller GPU card is used for inference to test the behavior in commercial systems. The inference time includes the frame pre-processing without the costly optical flow generation; it took as long as 10 ms for processing a pair of images.

TABLE 4. Comparison of trainable parameters and inference time of the different architectures.

Models	Parameters	Inference time
3D autoencoder	10,752	2.55 ms
Top-Heavy autoencoder	6,144	2.49 ms

A. EXPERIMENTAL SETUP

The architectures and the datasets of the previous sections were developed using Tensorflow 2.1.3 and OpenCV 3.4.8. Learning rate was fixed at 0.00001 with an Adam Optimizer for all the models. All the architectures were trained with 30 epochs. Models were trained using a Nvidia Tesla V100 card with CUDA 10.2. The inference times were calculated using an Nvidia GeForce RTX 2070 SUPER GPU card with CUDA 10.2.

B. ARCHITECTURES' PERFORMANCE COMPARISON

The comparisons ran as follows. First, we compare the Top-Heavy autoencoder versus the 3D autoencoder. Then, the results in the first comparison guide us to evaluate the best autoencoder using two different training methods, the *Anomaly Score* and the *Joint spatiotemporal* training.

1) TOP-HEAVY VS. 3D CONVOLUTIONS AUTOENCODERS

The comparison of the Top-Heavy autoencoder versus a 3D autoencoder included both grayscale and RGB datasets. Each input in the assessment corresponds to a video clip of 16 frames. Hence, the input size is 16 × 128 × 128 × 1 for the grayscale dataset, while 16 × 128 × 128 × 3

TABLE 5. AUC-ROC and EER results of the 3D vs. top-heavy autoencoders for each dataset.

	3D autoencoder		Top-Heavy autoencoder	
Grayscale-datasets	**AUC-ROC**	**EER**	**AUC-ROC**	**EER**
UCSD Ped1	72.5	30.1	71.4	31.3
UCSD Ped2	86.4	15.9	89.0	13.8
Avenue	84.2	21.7	83.7	21.2
ShanghaiTech	85.9	19.8	86.6	19.1
StreetScene	57.6	44.2	57.7	42.9
	3D autoencoder		**Top-Heavy autoencoder**	
RGB-datasets	**AUC-ROC**	**EER**	**AUC-ROC**	**EER**
Avenue	86.7	19.5	86.1	19.7
ShanghaiTech	85.6	19.0	87.0	18.7
StreetScene	57.6	44.1	57.8	43.2

TABLE 6. AUC-ROC and EER results of the top-heavy autoencoder for each dataset using two training techniques: the anomaly score and the joint spatiotemporal.

	Anomaly score		Joint spatiotemporal	
Grayscale-datasets	**AUC-ROC**	**EER**	**AUC-ROC**	**EER**
UCSD Ped1	71.1	31.2	72.8	29.9
UCSD Ped2	87.3	14.1	86.6	14.6
Avenue	83.3	21.8	82.5	22.6
ShanghaiTech	87.1	18.1	86.9	18.2
StreetScene	57.3	42.7	55.1	42.8
	Anomaly score		**Joint spatiotemporal**	
RGB-datasets	**AUC-ROC**	**EER**	**AUC-ROC**	**EER**
Avenue	86.3	19.3	85.2	19.3
ShanghaiTech	87.0	17.6	86.6	18.2
StreetScene	57.6	42.1	57.4	42.4

for the RGB dataset. Table 5 presents the measurements using AUC-ROC and EER for each dataset. The Top-Heavy autoencoder slightly outperforms the 3D autoencoder in three datasets (UCSD Ped2, ShanghaiTech, and StreetScene). However, the difference in the results of the 3D autoencoder and Top-Heavy autoencoder models is minimal. The average absolute difference of the AUC-ROC between both models in grayscale and RGB datasets is 1.2. Moreover, the Top-Heavy autoencoder uses only 57% of the trainable parameters than the 3D autoencoder uses, as observed in Table 4. Therefore, it is possible to substitute a 3D Autoencoder with the lighter Top-Heavy Autoencoder without largely affecting the final results.

2) ANOMALY SCORE DISTILLATION VS. JOINT SPATIOTEMPORAL TRAINING

We selected the Top-Heavy autoencoder due to its performance. Then, using the same dataset, we compared the two developed training techniques: the *Anomaly score distillation* and the *Joint spatiotemporal*. Regarding the results using the Grayscale dataset and its TV-L1 Optical Flow, the Top-Heavy spatial autoencoders received as input a clip of frames with the shape of $16 \times 128 \times 128 \times 1$; dropping the first frame of each folder to match the number of frames of the optical flow dataset version. The temporal Top-Heavy autoencoders received as input a clip of optical flow frames, u and v frames, with the shape of $16 \times 128 \times 128 \times 2$. One pair of temporal and spatial networks were trained using the Anomaly score distillation technique. The second pair of temporal and spatial networks were trained using the Joint spatiotemporal training technique.

The same procedure was repeated with the RGB dataset. Two spatial Top-Heavy autoencoders were trained using as input the RGB dataset frames, $16 \times 128 \times 128 \times 3$. Two more Top-Heavy autoencoders were trained using the optical flow version of the RGB dataset. These temporal networks had input frames with the shape $16 \times 128 \times 128 \times 2$. Spatial and temporal networks were separated into two pairs, one trained with the Anomaly score distillation technique and the other with the Joint spatiotemporal training technique.

Table 6 shows that the training technique Anomaly score has better results in most of the cases. The only exception is with the UCSD Ped1 dataset, where Joint spatiotemporal training beats the Anomaly score. The difference between both training methods AUC-ROC has an average of 1.12, concluding that the training process with the Anomaly score is a better option when not many available computational resources are available.

C. COMPARISON OF THE ANOMALY SCORE TRAINED AUTOENCODER DESIGN AGAINST RELATED WORKS

Completing the assessment in this paper, we include a comparison of the proposed design with related works using the Grayscale dataset in the Table 7. The Grayscale dataset version was selected to include the popular UCSD Anomaly detection dataset in the comparison. Table 7 lists the AUC-ROC results of the proposed design, our Top-Heavy autoencoder using the Anomaly Score in the training step for grayscale frames (**AS GS**, for short in the table) against other generative techniques for video anomaly detection task. The related methods in the table reported results with the same benchmarks as our assessment in their papers, those are:

1) AMDN: The Appearance and Motion DeepNet model [18] uses Autoencoders combined with a modified Two-Stream Network that adds an auxiliary third stream to help improve detection results.
2) HOG-HOF CAE: Histrogram of Oriented Gradients-Histogram of Optical Flow Convolutional Autoencoder [23] uses an Autoencoder and the Two-Stream Approach for its model and it is important to use as comparison as it uses older *Hand-crafted* techniques, HOG and HOF.
3) STAE-grayscale: The SpatioTemporal Autoencoder [8] is constructed with one encoder and two decoders of 3D convolutional layers, it uses a parallel training of the decoders which is important to compare against distillation technique.
4) OF-ConvAE-LSTM: The Optical Flow Convolutional Autoencoder Long Short Term Memory model [25] uses Long Short Term Memory network for the extraction of temporal features, which has been used for extracting temporal information in some other tasks

TABLE 7. Comparison of AUC-ROC and EER values of the proposed autoencoder design against related works using different datasets.

	Trainable	UCSD Ped1		UCSD Ped2		Avenue		ShanghaiTech		StreetScene	
Models	**Parameters**	**AUC-ROC**	**EER**	**AUC-ROC**	**EER**	**AUC-ROC**	**EER**	**AUC-ROC**	**EER**	**AUC-ROC**	**EER**
AMDN	38,707	92.1	16.0	90.8	17.0	-	-	-	-	-	-
HOG-HOF CAE	281,600	81.0	27.9	90.0	21.7	70.2	25.1	-	-	-	-
STAE-grayscale	10,752	92.3	15.3	91.2	16.7	77.1	33.8	-	-	-	-
OF-ConvAE-LSTM	22,373	92.4	-	92.9	-	89.5	-	-	-	-	-
Adversarial D.	153,650	96.8	7.0	95.5	11.0	-	-	-	-	-	-
GMFC-VAE	22,272	94.9	11.3	92.2	12.6	83.4	22.7	-	-	-	-
GMM-DAE	7,040	-	-	96.5	-	89.3	-	81.2	-	-	-
ITAE + GM	50,427	-	-	97.3	-	86.0	-	73.0	-	-	-
Temporal Cues	-	-	-	-	-	86.9	-	73.2	-	-	-
Spatial + temporal	-	-	-	84.5	-	80.3	-	-	-	-	-
GS (ours)	**6,144**	71.1	31.2	**87.3**	**14.1**	**83.3**	**21.8**	**87.1**	**18.1**	**57.3**	**42.7**

like Time-Series analysis [26], [27], and it is also a recurrent tool for video anomaly detection models.

5) Adversarial D.: The Adversarial Discriminator model [2] has obtained the best results in the UCSD dataset and it has a robust Two-Stream architecture using Generative Adversarial Networks.
6) GMFC-VAE: The Gaussian Mixture Fully Convolutional - Variational Autoencoder [1] keeps the standard Two-Stream Network approach and it uses a Variational Autoencoder to improve its feature extraction capacity.
7) GMM-DAE: The Gaussian Mixture Model - Deep Denoising Autoencoder [17] uses two-stream approach as base. To improve the late fusion of both streams, a Gaussian Mixture Model is applied to the latent space of each autoencoder.
8) ITAE + GM: The implicit two-path autoencoder + Generative Modeling [19] uses different frame rates for each of the streams, appearance or motion. To learn normality an explicit likelihood generative model is added.
9) Temporal Cues: The Temporal Cues from Socially Unacceptable Trajectories for Anomaly Detection [21] uses social interaction to develop temporal stream to improve the spatial stream results.
10) Spatio + temporal: The Anomaly Detection in Videos Using Two-Stream Autoencoder with Post Hoc Interpretability [22] uses two-stream approach to obtain spatiotemporal features and analyzes the obtained features using heat maps.

In sum, the results for our model in UCSD Ped1 and UCSD Ped2 are the lowest compared to the previous work. This is related to the poor extraction of temporal features in the system. As observed in the testing, UCSD Ped1 contains more anomalies that can only be extracted by learning the normal temporal features. Skateboards in UCSD Ped1, for example, are hard to be identified by the human eye by their shape but are noticed because of skaters' movement pace. The difference between UCSD Ped1 and Ped2 in all cases is higher than 15%. The result difference between both UCSD sets can be explained by the differences in the datasets. The models presented in this work need some improvements to perform better with datasets that contain more temporal anomalies (UCSD Ped1 and StreetScene). The best results were obtained in the datasets with a higher quantity of spatial anomalies and their results were improved with the application of the presented techniques (UCSD Ped2, Avenue or Shanghaitech).

The anomalies in the Avenue dataset happen closer to the camera, making them easier to identify by spatial feature extraction. Even better, the results of our proposed model with the ShanghaiTech dataset outperform the results of all the related works. This is because the characteristics of the anomalies found in the ShanghaiTech dataset are evident and therefore detected by correct spatial feature extraction.

Even though our proposed model does not have high AUC-ROC results in UCSD and Avenue dataset, we need to highlight some essential characteristics. First, the proposed model does not need optical flow computations for the inferences and requires smaller memory resources than related works. Second, the proposed design has fewer trainable parameters than the related works. For instance, the closest model in size to our proposed model is the GMM-DAE; however, that model needs optical flow pre-processing and dynamic flow frames creation, taking more processing time than ours.

VI. CONCLUSION

In this work, we evaluated potential concepts adapted for the anomaly detection task for intelligent video surveillance. The examination included the comparison of a Top-Heavy design versus 3D convolutional layers in autoencoders. Our experimental results showed that the Top-Heavy Autoencoder uses only the 57% of the trainable parameters compared to the 3D autoencoder. We found an average absolute difference of the AUC-ROC between both models of 1.2. The reduced number of trainable parameters of the Top-Heavy autoencoder design improved the AUC-ROC results compared to the 3D autoencoder results. Then, this examination also aimed to select the best training strategy between the methods Anomaly Score and Joint spatiotemporal. The AUC-ROC difference between Anomaly Score and Joint spatiotemporal training methods has an average of 1.12. The results show a slight improvement in the temporal features extraction. Therefore, the best method for training depends on the need for the specific project due to the necessity of joint training models

simultaneously or training the teacher model first and then the student. Finally, the extraction of more defining temporal features must be addressed for future work.

REFERENCES

[1] Y. Fan, G. Wen, D. Li, S. Qiu, M. D. Levine, and F. Xiao, "Video anomaly detection and localization via Gaussian mixture fully convolutional variational autoencoder," *Comput. Vis. Image Understand.*, vol. 195, Jun. 2020, Art. no. 102920, doi: 10.1016/j.cviu.2020.102920.

[2] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1896–1904.

[3] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2718–2726.

[4] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.

[5] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 2720–2727.

[6] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly Detection–A new baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6536–6545.

[7] B. Ramachandra and M. J. Jones, "Street scene: A new dataset and evaluation protocol for video anomaly detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2569–2578.

[8] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X.-S. Hua, "Spatio-temporal AutoEncoder for video anomaly detection," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 1933–1941.

[9] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 305–321.

[10] J. C. Stroud, D. A. Ross, C. Sun, J. Deng, and R. Sukthankar, "D3D: Distilled 3D networks for video action recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Oct. 2020, pp. 614–623.

[11] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2017, pp. 4724–4733.

[12] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," 2017, *arXiv:1711.11248*.

[13] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[14] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 1, 2014, pp. 568–576.

[15] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends Neurosci.*, vol. 15, no. 1, pp. 20–25, Dec. 1992.

[16] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.

[17] Y. Ouyang and V. Sanchez, "Video anomaly detection by estimating likelihood of representations," 2020, *arXiv:2012.01468*.

[18] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," 2015, *arXiv:1510.01553*.

[19] M. Cho, T. Kim, I.-J. Kim, and S. Lee, "Unsupervised video anomaly detection via normalizing flows with implicit latent features," 2020, *arXiv:2010.07524*.

[20] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-VAE," 2018, *arXiv:1804.03599*.

[21] N. Madan, A. Farkhondeh, K. Nasrollahi, S. Escalera, and T. B. Moeslund, "Temporal cues from socially unacceptable trajectories for anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2150–2158.

[22] J. Feng, Y. Liang, and L. Li, "Anomaly detection in videos using two-stream autoencoder with post hoc interpretability," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–15, Jul. 2021.

[23] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2016, pp. 733–742.

[24] J. Sánchez Pérez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 optical flow estimation," *Image Process. Line*, vol. 3, pp. 137–150, Jul. 2013.

[25] E. Duman and O. A. Erdem, "Anomaly detection in videos using optical flow and convolutional autoencoder," *IEEE Access*, vol. 7, pp. 183914–183923, 2019.

[26] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," in *Neural Nets WIRN Vietri-01*. London, U.K.: Springer, 2002, pp. 193–200.

[27] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Netw.*, vol. 116, pp. 237–245, Aug. 2019.

ERNESTO CRUZ-ESQUIVEL (Member, IEEE) received the B.Sc. degree in mechatronic engineering from the Universidad de las Américas Puebla, in 2018, where he is currently pursuing the Ph.D. degree in intelligent systems. He worked as an artificial vision specialist in the steel industry, from 2018 to 2019. His research interests include computer vision, surveillance video, and anomaly detection.

ZOBEIDA J. GUZMAN-ZAVALETA received the bachelor's degree from the Benemérita Universidad Autónoma de Puebla (BUAP), in 2004, and the M.Sc. and Ph.D. degrees from the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), in 2008 and 2017, respectively, all studies on computer science. She is a Computer Scientist interested in finding relevant and abstract information from complex data, such as video, images, and audio. She has worked as a Visiting Researcher at INAOE, in 2017; and a Visiting Professor at IBERO, Puebla, from 2016 to 2018, and CINVESTAV-Tamaulipas, Mexico, in 2018. Currently, she has been an Associate Professor with the Universidad de las Américas Puebla (UDLAP), since 2019. She is a member of the National Council of Researchers in Mexico (SNI) and the Asociación Mexicana de Computación (AMEXCOMP).