Journal of the Brazilian Computer Society, 2022, 28:1, doi: 10.5753/jbcs.2022.2194

Weakly Supervised Video Anomaly Detection Combining Deep Features with Shallow Neural Networks

Silas Santiago L. Pereira [**Federal Institute of Ceará** | ***silas.santiago@ifce.edu.br***]
José Everardo Bessa Maia [**State University of Ceará** | ***jose.maia@uece.br***]

Federal Institute of Education, Science and Technology of Ceará, Av. Paranjana, 1700, Fortaleza, CE, 60740-903, Brazil

Received: 04 September 2021 • **Accepted:** 27 October 2022 • **Published:** 29 December 2022

Abstract Deep features have outgrown hand-sketched features in many applications. The availability of pre-trained deep feature extractors helps to overcome one of the deep learning main drawbacks, which is the need for large volumes of data for training. Multiple Instance Learning (MIL) has become an attractive solution for video surveillance literature once it allows working with weakly labeled bases. This work evaluates a video anomaly detection approach based on the MIL paradigm combining deep features with shallow Neural Networks. For computational efficiency, we apply Principal Component Analysis (PCA) for dimensionality reduction before classification. We performed the experiments from a set of I3D (Inflated 3D) features, which corresponds to the ShanghaiTech benchmark dataset, and the MLP and SVM shallow classifiers achieved competitive results.

Keywords: Anomaly detection, Video surveillance, Multiple Instance Learning, I3D features, Shallow Neural Networks

1 Introduction

The huge amount of video data, generated by a large number of surveillance cameras in different locations worldwide, makes human monitoring efforts more challenging. Manual surveillance can also become a tedious and erroneous activity, which motivates the need for automatic monitoring approaches. Law enforcement agencies have limitations in capturing or avoiding abnormal activities due to monitoring manual limitations. Once unusual events occur inconsistently and with low probability in real-world surveillance scenarios, the manual searching of these ones in massive video streaming is a difficult task. Various anomaly detection techniques can suffer from high false alarm rates, showing limited performance when they need to deal with real-life scenarios. Anomaly detection based on deep learning algorithms is comparatively reliable in decision making and is also able to reduce human laborious activities. Ullah *et al.* [2021].

Automated Surveillance Systems primarily aim at detecting suspicious or unusual activities, behaviors or irregular events in a scene. Such systems implicitly assume that casual activities should be anomalous because they are potentially suspicious Roshtkhari and Levine [2013]. In addition, surveillance systems provide functionalities that allow operation by autonomous systems or without human intervention. They also enable the detection and anticipation of events or object behaviors in order to generate alerts for unexpected activities. The video surveillance area is multidisciplinary, involving other research fields such as pattern recognition and analysis, signal processing, distributed systems, and communications. Video security applications have increasingly integrated advances in computer vision, signal processing, and artificial intelligence. The main purpose of applied research in video surveillance has been to move towards the interpretation of video scenes, observation, and prediction of objects in a scene based on captured information by cameras Shidik *et al.* [2019]. The use of video surveillance cameras has been widespread in a variety of places, such as streets, intersections, banks, and shopping malls.

Anomaly detection is a crucial task in many domain applications, such as intrusion detection, frauds, and video surveillance Li *et al.* [2021]. The manual processing of the information produced by surveillance cameras demands excessive manpower, which makes it relevant to automate video anomaly detection Kamoona *et al.* [2020]. Due to the subjectivity in anomalies definition, it may be difficult to specify once it depends on the location and situation and varies widely in content and duration Wan *et al.* [2020]. Context understanding is fundamental since an anomalous activity in a given occurrence can be normal in another one Rao *et al.* [2017]. A rare situation denotes an anomalous event, so it has a low occurrence probability among most normal events. This fact makes the obtention of anomalous representants a difficult task since most of the data reflect normal behavior patterns Rao *et al.* [2017]. The binary and unary classification paradigms are typically used in video anomaly detection approaches presented in the literature.

Weakly supervised learning mitigates the difficulties associated with the instance labeling task, which is a laborious stage during the construction of predictive models. For example, an individual can take a surveillance video and classify each frame as abnormal or normal, as illustrated in Figure 1. This paradigm is usually formulated as a multiple instance learning problem (MIL) Wan *et al.* [2020]. MIL is a useful approach, especially in situations where the knowledge about label categories and training examples are incomplete Ali and Shah [2008]. In MIL, there are bags which contain multiple instances instead of individual ones to represent each pattern in a dataset. A bag could have normal or anomalous labels and they are used to build a predictive model with an appropriate machine learning technique, as illustrated in Figure 2. Recent works have shown the performance efficiency of the MIL approach in detection and recognition tasks Tian *et al.* [2021].

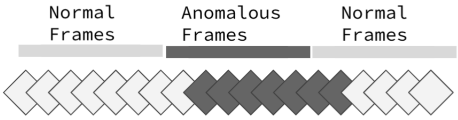

Figure 1. Frame level Video Labeling: Each frame is labeled individually as normal or anomalous

The extraction of representative characteristics from original data is an essential aspect for the adequate performance of anomaly detection methods Ribeiro *et al.* [2018]. Handcrafted features can suffer from limited adaptability, once they can be very representative in some scenarios and not perform well in other ones. Their design specification is also a limiting factor, once researchers usually project these features for a specific purpose, such as the use of motion and appearance information to represent the main characteristic of a given scene Al-Dhamari *et al.* [2020].

Deep learning approaches have achieved remarkable results in different application domains, such as image classification, object detection, voice processing, and anomaly detection Pawar and Attar [2019]. Unsupervised deep learning methods have been applied for model learning and feature extraction once distinct spatiotemporal handcrafted attributes (such as color, optical flow, and texture, commonly used in anomaly detection procedures) do not produce good generalization Kamoona *et al.* [2020]. Researchers proposed many deep learning-based classification methods and feature extraction approaches in recent years Perera and Patel [2019].

The construction of a deep learning model from scratch demands high amounts of data, processing time, and computational resources, besides the fact that annotation of massive data is costly and time-consuming. To deal with these challenges, the transfer learning concept allows the use of a CNN network built on a particular dataset for a specific task and then fine-tune this model for a new task in a different scope Al-Dhamari *et al.* [2020]. Different studies have used deep features to detect anomalous activities in videos Ullah *et al.* [2021]. Basically, researchers could implement the concept of transfer learning by updating the weights of an original pre-trained network with new training datasets by continuous backpropagation, or by using the pre-trained network as a deep feature extractor by replacing the final fully connected layer by a new classifier algorithm Al-Dhamari *et al.* [2020]. The high layer activations of convolutional neural network architectures have proved to be strong visual features so researchers have investigated the use of CNN-based representations for image and video for many tasks such as video summarization, classification, and video and image retrieval Otani *et al.* [2016].

Support vector machine (SVM) is extensively used in many applications, such as regression analysis, outlier detection, and statistical classification, and is considered an ubiquitous method in the machine learning community. SVM is able to solve linear and nonlinear classification problems by the use of various kernel functions. SVM requests very few patterns for training, and combines good training impact and good detection accuracy for new examples with the same characteristics during the test step Al-Dhamari *et al.* [2020].

In our study, we consider anomalous activity detection as a binary classification problem in which deep features containing visual and temporal information are fed into a machine learning supervised estimator to make it learn about the difference between abnormal and normal activities. We applied the widely used supervised machine learning classifiers SVM (Support Vector Machine) **?** and MLP (Multilayer Perceptron) **?** algorithms. We applied the MIL approach, so the training phase uses only video labels. We construct and evaluate our approach with a previously preprocessed and publicly available set of deep features extracted with deep neural network Inflated 3D (I3D) for the benchmark dataset ShanghaiTech. We employ the Principal Component Analysis (PCA) method as a linear manifold approach for dimensionality reduction and performance enhancement of anomaly detection models.

Our proposal mitigates existing challenges in video anomaly detection. Firstly, to deal with subjectivity inherent to video anomaly definition and the difficulties in distinguishing normal and anomalous patterns (due to noise, variations, and low interclass variance), we employ transfer learning capabilities from the use of pre-processed features extracted from a deep network for better representativeness. To build predictive models robust to sparsity existing in the used video anomaly dataset, we maintain the proportionality between training and testing partitions as the original processed dataset. Since anomalous behaviors are diverse, we estimate 95% confidence intervals over randomized training data to guarantee the appropriate choice of the best-evaluated models trained over diversity on training data. Finally, labeling video data is a laborious activity for humans, mainly in massive video data. Multiple Instance Learning sounds convenient to lead with the incomplete knowledge about video training data.

The contributions of our proposed research can be summarized as follows:

- Present a performance comparison between a gradient descent-based optimization technique (MLP) and a quadratic programming optimization-based method (SVM) for the purpose of abnormal detection. We evaluate the generated models from distinct metrics and with a high confidence level of 95% for the considered performance metrics.
- Combine a non-linear SVM classifier and the representation power of I3D deep features to build a robust and competitive estimator.
- Compare our results with two previous state-of-the-art studies Wan *et al.* [2020] and Kamoona *et al.* [2020] on the application of MIL paradigm and binary classification for video anomaly detection. From the evaluation and comparison of the proposed approach with two studies in the literature, we achieve promising results.

This work extends the study published in Pereira and Maia [2021] by the inclusion of the SVM algorithm in our testbeds. Also, our experiment results were statistically reliable since we evaluated the generated models considering a high confidence level of 95% with a t-student distribution. Unlike

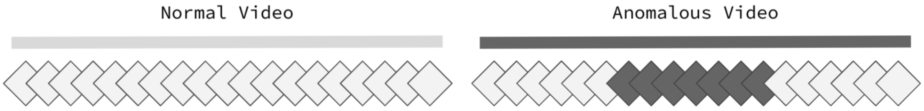

Figure 2. Normal and Anomalous Weakly Labeled Videos: An anomalous video contains at least one anomalous frame instance

previous work, this experiment configuration ensures better confidence in generated results. Besides, we considered distinct combinations of I3D deep features to verify the performance of machine learning algorithms MLP and SVM when trained with features built from RGB, Optical Flow individually, and both derived deep features.

This work is organized as follows: Section 2 presents some relevant research associated with video anomaly detection in the context of the MIL paradigm. Section 3 describes the methodology to build and evaluate our proposed approach. In section 4, the achieved results are presented and discussed. Section 5 presents the conclusions and directions for future research.

2 Related Work

This section is a brief review of some research studies related to our work to contextualize the present topic in the literature. Recent surveys which expand the coverage of this section can be found in Suarez and Naval Jr [2020] and Nayak *et al.* [2020].

In Al-Dhamari *et al.* [2020], the authors present a video anomaly detection framework based on VGGNet and BSVM. Human motion feature extraction from RGB video frames in complex and noisy surveillance scenarios is done by transfer learning. To deal with lighting effects on the performance of the proposed abnormal detection approach in the data preprocessing step, the authors apply illumination normalization using the histogram equalization technique to control lighting conditions. The Gaussian filter technique is also used to remove unwanted small objects. A convolutional neural network (CNN) based on Visual Geometry Group network 19 (VGGNet-19) pre-trained model was applied to extract high-level descriptive features, once it performs other experimented pre-trained CNN networks such as GoogleNet, ResNet50, AlexNet, and VGGNet-16. A Binary Support Vector Machine (BSVM) technique is used to establish the anomaly event detection model. The framework was evaluated on UMN and UCSD-PED1 benchmark datasets in terms of accuracy, AUC, and EER evaluation metrics. The authors reported better performance of the proposed approach compared with classical frameworks. Experiments accomplished an accuracy and AUC of 97.44% and 0.9795 for the UMN dataset, and 86.69% and 0.7987 for the UCSD-PED1 dataset, respectively. Experimental results also show that VGGNet-19 outperforms in accuracy the handcrafted descriptors, oriented gradients, background subtraction, and optical flow.

Ullah et al. (2021) present an intelligent anomaly detection framework which can operate in surveillance networks with lesser time complexity. The authors apply a deep spatiotemporal feature extraction by using a pre-trained ResNet-50 residual network (ResNet), which is built on the ImageNet dataset. A multi-layer bi-directional long short-term memory (BD-LSTM) architecture is used to boost learning capabilities in order to perform normal and anomalous pattern detection. The suggested approach is trained by a weakly supervised technique to deal with the absence of labeled data. Experiments with different benchmark datasets were performed to verify the framework functionality within complex surveillance scenarios. The authors reported a significant increase in accuracy for two of the evaluated datasets compared to state-of-the-art methods.

In Sultani *et al.* [2018], the authors proposed a video anomaly detection approach from the use of a weakly labeled training dataset. They considered weakly supervised learning by using MIL approach. Anomaly detection was treated as a regression problem in which a feature vector is mapped to an anomaly score. Anomalous and normal surveillance videos were segmented in clips so that a video (bag) contained multiple segments (instances of a bag). The segment anomaly scores were generated by a predictive model built from the training data. To build and evaluate the proposed approach, the authors considered a large-scale video anomaly dataset which is composed of multiple anomalous events. From the obtained results, the proposed approach overcomes other anomaly detection state-of-the-art approaches in performance.

Kamoona et al. (2020) suggested a deep neural network with an encoding-decoding architecture for anomaly detection in video surveillance scenarios to allow the capture of temporal and spatial information of video instances. The main contribution was to consider the temporal relations among video instances to treat them as sequential visual data instead of a set of independent instances. Using a weakly supervised learning-based framework under the MIL paradigm, They proposed a cost function which penalizes incorrect detections by maximizing the average distance between anomalous and normal predictions. The authors used the video surveillance benchmark datasets UCF-Crime and ShanguaiTech to build and evaluate the proposed approach. A set of spatiotemporal attributes are extracted from each evaluated video using a C3D network model. The performance evaluation of compared models considered ground truths at the frame level. The achieved results are competitive in relation to the state-of-the-art simulation studies.

Wan et al. (2020) presented the weakly supervised framework AR-NET (Anomaly Regression Net) for video anomaly detection. The binary classification-based approach combines Multiple-Instance Learning (MIL) for segment level classification so that the training stage considers only video-level labels. The authors also proposed and evaluated the cost functions Dynamic Multiple-Instance Learning Loss (DMIL) and Center Loss, which are suggested to learn discriminants for anomaly detection. The pre-trained neural network model Inception-v1 I3D (Inflated 3D) is used as a feature extractor which receives appearance (RGB) and motion information (optical flow) as input. The authors compared the approach with state-of-the-art literature using the challenge benchmark dataset ShangaiTech. From the obtained results, the proposed approach overcomes the compared techniques in performance. Moreover, from a subjective analysis of this approach in some challenging scenes, the authors conclude that the anomaly detection problem is still a challenge for state-of-the-art models.

The results of the last two research studies Wan *et al.* [2020] and Kamoona *et al.* [2020] were used to compare the achieved results in our proposed approach since both use the same benchmark dataset applied in our study. Similar to Wan *et al.* [2020], our study also evaluates the efficiency of binary classification and MIL approach for video anomaly detection. We also verify the impact of the use of the linear manifold approach PCA to improve the performance of the evaluated binary classifiers.

3 Methodology

This section describes the main steps for data preparation, modeling, and evaluation of our proposed video anomaly detection approach. The stages applied in our research are briefly described in the following subsections.

In our study, we consider the video anomaly detection problem in the perspective of binary classification. We describe this problem as follows: Let $X = \{x_i\}_{i=1}^{n}$ a dataset consisting of n videos. Each video x_i has duration t_i, so that $T = \{t_i\}_{i=1}^{n}$ is the temporal duration of the dataset. Let $Y = \{y_i\}_{i=1}^{n}$ be the binary labels for each video in dataset X. A predictive model receives a given video x_{test} and produces an inference as a score, probability or class Wan *et al.* [2020].

3.1 Data Preparation

The benchmark dataset ShanghaiTech Luo *et al.* [2017] was originally proposed for unary classification, once all training samples are normal instances. The dataset contains captures at the ShanghaiTech University and describes different conditions of illumination and viewpoints. There are 437 videos with 130 anomalies among 13 scenes. Figure 3 exemplifies normal and anomalous situations in the considered dataset.

In Wan *et al.* [2020], the authors adopted a split of this dataset present in Zhong *et al.* [2019] to enable binary classification and used the pre-trained deep neural network Inflated 3D (I3D) as a feature extractor. The generated features correspond to appearance (RGB) and movement (optical flow) information. The I3D feature dataset available at this link[1] reflects the previously processed anomaly patterns from the ShanghaiTech dataset described in Wan *et al.* [2020]. We used these deep features in our experiments. The processed dataset as I3D features contains 437 instances, where 330 are normal videos and 107 are anomalous videos. Each instance in this dataset corresponds to a matrix with dimension n x 2048 or n x 1024, where n is a variable number of existing segments and the number of attributes is 1024 (when only appearance features or movement information are considered individually) or 2048 (when both appearance and movement information are considered).

We evaluate our approach on three feature dispositions of the ShanghaiTech dataset: I3D features generated with only RGB (1024 I3D features from RGB data) or Optical Flow (1024 I3D features from optical flow data) and both (2048 I3D features generated from RGB and Optical flow data) as summarized in Table 1.

Table 1. Feature Dispositions of ShanghaiTech Dataset

DISPOSITION	DESCRIPTION	FEATURES
COMBINE	I3D features from RGB and Optical flow	2048
RGB	I3D features from RGB	1024
FLOW	I3D features from Optical Flow	1024

For each processed video, there is a corresponding category label (normal or anomaly) and the labels of each existing frame of this video. In our research, we applied the dataset with I3D features to build and evaluate the predictive video anomaly detection models.

We describe the data preparation for modeling and performance evaluation as follows. Initially, we load the dataset $\mathbf{D} = \{(X_i, y_i, yf_i)\}_{i=1}^{N}$ with $N = 437$ processed videos. There are 330 normal (N) video instances and 110 anomalous (A) video instances. Each instance X_i is a matrix n x 2048 or n x 1024, where n is a variable number of segments in a given video, $y_i \in \{0, 1\}$ is the video label and $yf_i = [yf_i^1, \ldots, yf_i^k]$ is a sequence of k frame labels existing in the video. In the last expression, $yf_i \in \{0, 1\}$ and k is a variable number of frames. Then, we partitioned $\mathbf{D}$ into training and test datasets, and we used 75% of data for further training with cross-validation. It is relevant to mention that features obtained from frame sequences are relevant for anomaly event detection, since it allows to consider the spatiotemporal characteristics of the video Ullah *et al.* [2021].

By following the described procedures, we build both partitions so that we maintain the same proportion of anomalous and normal instances in training and test partitions as the original processed I3D data. The training set contains 247 normal videos and 80 anomalous ones. The test set contains 83 normal videos and 27 anomalous videos.

We perform the segment level composition of training and test sets as follows. Initially, we partitioned $\mathbf{D}$ into $\mathbf{D}_N$ and $\mathbf{D}_A$ datasets, which represents the video sets of normal and anomalous classes, respectively. Then, we split $\mathbf{D}_N$ and $\mathbf{D}_A$ into $\mathbf{D}_N^{train}$, $\mathbf{D}_N^{test}$, $\mathbf{D}_A^{train}$ e $\mathbf{D}_A^{test}$. Finally, we compose the training and test partition as $\mathbf{D}_{train} = \{\mathbf{D}_N^{train}, \mathbf{D}_A^{train}\}$ and $\mathbf{D}_{test} = \{\mathbf{D}_N^{test}, \mathbf{D}_A^{test}\}$.

Then, we arrange the training and test partitions to build a segment level machine learning predictor and evaluate the generated model at the frame level, respectively. Each video in $\mathbf{D}_{train}$ is then transformed into n new segments with 2048 or 1024 attributes per segment. The new training segment partition is described as $\mathbf{S}_{train} = \{(S_i, y_i)\}$, where S_i is a video segment and y_i is the corresponding segment class, which is the same label as the video containing this clip.

In order to enable evaluation at frame level, we describe the test segment partition by $\mathbf{S}_{test} = \{(S_i, yfs_i)\}$, where

[1] https://github.com/wanboyang/Anomaly_AR_Net_ICME_2020

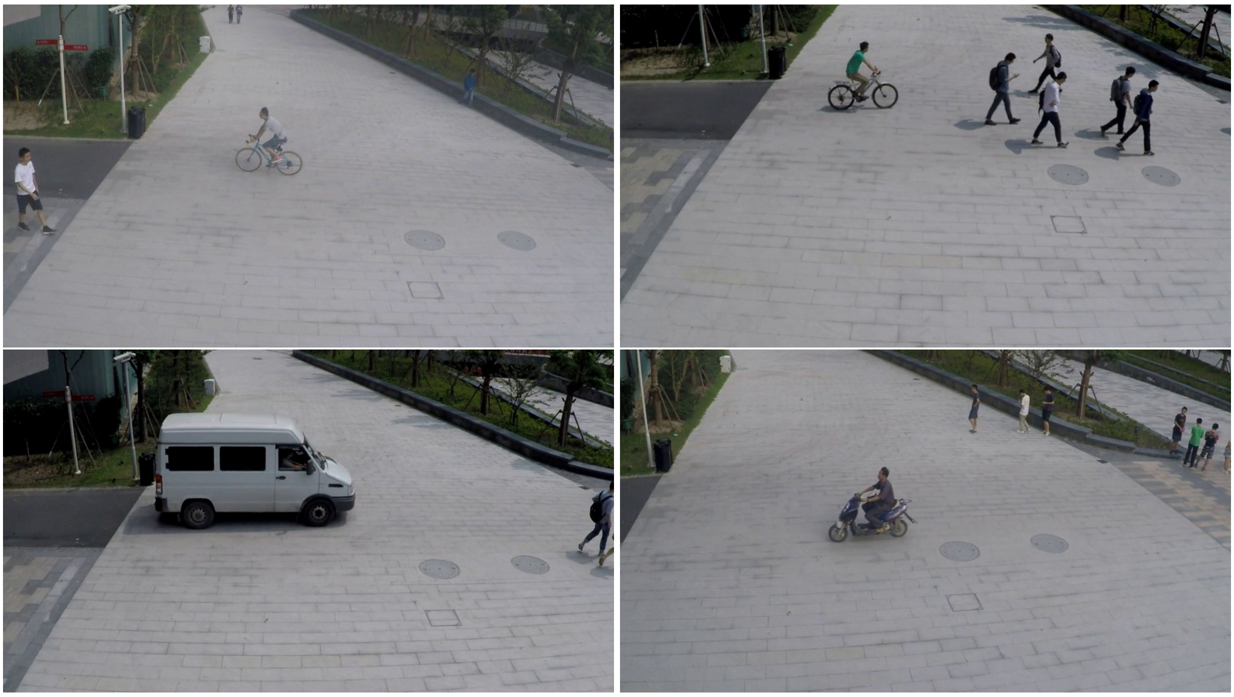

Figure 3. Anomalous Event Examples in ShanghaiTech Dataset Available in Zhong *et al.* [2019]: Vehicles entering a Pedestrian Zone

yfs_i is a sequence of 16 frame labels obtained from yf_i. The training set is composed of 14532 segments (12579 normal and 1953 anomalous segments, respectively). The test set is composed of 4983 segments (4422 normal and 561 anomalous segments), respectively. The algorithm 3.1 summarizes the described procedures for data preparation.

[1] Load the dataset with 437 videos (330N and 107A), video labels and frame labels Select 247 normal and 80 anomalous videos for training partition Select 83 normal and 27 anomalous videos for test partition Convert the training video partition ($247N, 80A$) in a training segment partition with 14532 segments ($12579N, 1953A$) Convert the test video partition ($83N, 27A$) in a test segment partition with 4983 segments ($4422N, 561A$)

Similarly to Wan *et al.* [2020], we approach the video anomaly detection task as a weakly supervised problem followed by the binary classification, so we consider only video labels during the construction phase of a predictive model. In the MIL approach, a positive video (bag) contains at least one instance of a positive event and a negative bag consists of only negative examples.

3.2 Modeling and Evaluation

With the prepared, numerically represented, and partitioned video data, we can build, evaluate and compare I3D feature-based models for the anomaly detection problem. To perform the experiments, we use the implementations in the libraries *Scikit-learn* Pedregosa *et al.* [2011], *Keras*[2] e *tensorflow*[3], available for Python language [4]. The choice of these libraries is due to the efficiency and rapid prototyping in the modeling and performance evaluation of machine learning models. We executed all the experiments in a cloud computing service (Google Colab).

3.2.1 Principal Component Analysis (PCA)

We use Principal Component Analysis (PCA) **?** to reduce the input space with 2048 I3D features to 624 and 205 principal components which explain approximately 95% and 85% of total data variance in training data, respectively. In our research, we used the same number of components 624 and 205 related to COMBINE disposition for RGB and FLOW dispositions of I3D features. According to Zhao *et al.* [2020] feature extraction techniques, where PCA is one of the most widely used, allow the obtention of more compact representations of the data which include essential information for decision making.

PCA is a non-supervised approach that aims to rotate the axes of a data matrix representation space, maintaining the orthogonality so that when data are projected on these new axes, the explained data variance is maximized in decreasing order for some sequence of principal component directions. PCA components can be computed from different structures such as covariance or data correlation matrices. This applied to SVD (Singular Value Decomposition) Saha *et al.* [2009]. The SVD performs the following decomposition for a centered data matrix:

$$M_{m\ x\ p} = U_{m\ x\ m} \cdot D_{m\ x\ p} \cdot V^T_{p\ x\ p} \qquad (1)$$

where $U \in \mathbf{R}^{m\ x\ m}$ and $V \in \mathbf{R}^{p\ x\ p}$ are orthogonal matrices for which their columns correspond to eigenvectors of $M \cdot M^T$ e $M^T \cdot M$, respectively. The diagonal matrix D is composed of singular elements, which are the squared roots of eigenvalues λ_i of $M \cdot M^T$ or of $M^T \cdot M$. These eigenvalues are generally ordered so that $\lambda_i \geq \lambda_{i+1}$, for $i = 1, 2, \ldots, p-1$.

[2] https://keras.io

[3] https://www.tensorflow.org

[4] https://www.python.org/

We used K-fold Cross-Validation in the training data to select the best predictive model over the $k = 5$ iterations per experiment. According to James *et al.* [2013], the number of folds in k-fold cross validation such as k=5 and k=10 have been shown empirically appropriate for the bias-variance trade-off associated with the choice of k parameter. These values are associated with the generation of test error rate estimates which do not suffer from excessively high bias or high variance.

For the SVM technique, we selected the model with the lowest balanced accuracy among the k iterations of cross-validation for evaluation with the test segment dataset. For the experiments with MLP neural networks, we perform this selection by the use of the Root Mean Squared Error (RMSE) evaluation metric which is described by:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{d_i - f_i}{\sigma_i}\Big)^2} \tag{2}$$

The following steps summarize the procedures for model training and evaluation: (1) Load the segment level training set; (2) Apply the cross-validation procedure to train and evaluate k predictive models with the training and test partitions obtained in each iteration; (3) Select the best model among the k iterations of the cross-validation. We used the metrics BACC and RMSE to select the SVM and MLP-based models, respectively. Finally, (4) we evaluate the best model with the test set. Since we build the generated models at segment level, we mapped each predicted test segment output to the corresponding output frame sequence (each video clip corresponds to a sequence of 16 frames) to allow evaluation at frame level and comparison with state-or-art literature.

3.2.2 Multilayer Perceptron (MLP)

The definition of the suitable neural network topology has been based traditionally on trial and error, heuristics, and pruning or constructive techniques Stathakis [2009]. In our experiments, we considered Multilayer Perceptron (MLP) neural network architectures with one, two, and three layers and with 64, 128, 256, 512, 1024 and 2048 neurons in each hidden layer. For a MLP network with a single hidden layer, the output of each neuron in a hidden layer is specified by

$$S_i{}^{L2} = D(\max(0, W_{L1} \cdot S_i + b_{L1})) \tag{3}$$

where $D(\cdot)$ corresponds to Dropout regularization with a rate of 0.70, which will discard some entries during the training stage to prevent model overfitting.

The Rectified Linear Unit (RELU) activation function is specified by $y = \max(0, x)$ and was used for the hidden layer neuron activation. The network output is described by

$$s_i = 1/(1 + \exp(W_{L2} \cdot S_i + b_{L3})) \tag{4}$$

which is represented by a neuron with a sigmoid activation function. We consider a batch size of 512 and 1000 epochs in the network training stage for the evaluated MLP architectures. The Adaptative Moment Estimation (ADAM) optimization technique was employed as a variant of the gradient descent method.

The cost function Binary Cross Entropy was employed in the training phase of the neural network binary classification models and is described by

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{5}$$

where $y_i \in \{0, 1\}$ e $p(y_i)$ is the probability of occurrence for pattern i.

3.2.3 Support Vector Machines (SVMs)

Support vector machines (SVMs) are one of the most accurate and robust techniques in well-known data mining methods Wu and Kumar [2009]. According to Al-Dhamari *et al.* [2020], binary SVM is the most frequently used method to solve different classification problems and it is able to yield a good generalization performance by implementing the principle of structural risk minimization. SVM constructs a hyperplane that works as an optimal decision surface to distinguish data points in the space. The aim is to project data points onto higher dimensional space where the different categories become linearly separable. SVM offers better generalization ability than other empirical risk minimization-based classification techniques since it is based on structural risk minimization Al-Dhamari *et al.* [2020].

For a linearly-separable binary classification task, the aim is to find an optimal hyperplane $H = w^T x + b$ with a maximal separation margin $\rho = \frac{2}{||w||}$, where w and b are the weight vector and bias, respectively Wu and Kumar [2009]. This is a constrained optimization problem calculated in function of support vectors and expressed by using Lagrange multipliers. For linearly inseparable problems, the kernel trick technique is a common approach which consists to choose an appropriate kernel function to produce a non-linear input data transformation in order to make the problem linearly separableWu and Kumar [2009].

Different kernel functions like radial basis function (RBF), linear and polynomial can be used with SVM, which is usually required to solve quadratic programming (QP) problem Al-Dhamari *et al.* [2020]. According to Amraee *et al.* [2018], the number of parameters in the RBF kernel is less when compared to the other kernels, which influences the complexity of model selection. The RBF kernel has the parameter σ which controls the performance of SVM Al-Dhamari *et al.* [2020]:

$$K(x_i, x_j) = exp(-||x_i - x_j||/2\sigma^2) \tag{6}$$

Once instances cannot be entirely separated, a common approach is to apply soft margin optimization by introducing slack variables to allow the existence of noisy data. The objective function is expressed by:

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \tag{7}$$

which is subject to n restrictions $y_i(w^T x_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$, $i = 1, ..., n$ and C is a regularization parameter to control the trade-off between the number of inseparable data points and machine complexity. This parameter give us a trad-off between bias and variance Jabeen *et al.* [2019].

3.3 Evaluation Metrics

We consider six different performance metrics to report the achieved results of each model configuration. Let *TP* be the number of true positives, *FP* the number of false positives, *TN* the number of true negatives, and *FN* the number of false negatives. For each method and for each experiment, we computed the following 6 metrics for performance evaluation:

1. Area under Curve (AUC)
2. False Positive Rate ($FPR = \frac{FP}{FP+TN}$) or False Alarm Rate (FAR)
3. Balanced Accuracy ($BACC = \frac{TPR+TNR}{2}$)
4. Precision ($PREC = \frac{TP}{TP+FP}$)
5. Recall ($REC = \frac{TP}{TP+FN}$)
6. F1-score ($F1 = \frac{PREC \cdot REC}{PREC+REC}$)

In all our experiments, we compute the confidence levels for the considered performance metrics considering a high confidence level of 95% over 10 executions per experiment.

Confidence intervals are often used in model validation and provides an interval estimate of a population parameter. Assuming that an unknown population standard deviation σ is more realistic, this value σ is estimated using the sample standard deviation s and Student t distribution Petty [2012]. The confidence interval for the population mean μ is given by the following expression:

$$\left[\overline{x} - t_c \frac{s}{\sqrt{n}}, \overline{x} + t_c \frac{s}{\sqrt{n}}\right] \tag{8}$$

where t_c represents the critical value for *Student* t distribution for a confidence level c. Another form to represent confidence interval is *point estimate* ± *margin of error*. The point estimate is the sample mean $\overline{x}$ when we consider the confidence interval for the population mean μ.

4 Results and Discussion

The main goal of this study is to detect anomalies effectively. Our work defines model effectiveness as a measure of algorithm performance in terms of considered selected evaluation metrics (such as accuracy, false alarm rate, and area under ROC curve) for anomaly detection machine learning-based models. In this sense, an effective model will be the one with the best generalization capability for performance evaluation experiments on the test data. In this section, we detail the choice of hyperparameters values used in SVM and MLP algorithms. We also present a comparison with cutting-edge methods in order to validate our proposed approach.

As detailed in the previous section, the dataset consists of 437 videos, being 330 normal videos and 107 anomalous ones. For the experiments, we maintained this proportionality between training and testing data. For the training set, we used 247 normal videos and 80 anomalous videos, leaving 83 normal videos and 27 anomalous videos for testing. After this split, we convert the videos into segments. In training, we used cross-validation. We repeated the experiments 10 times by randomly selecting test sets.

For each of the six considered metrics, the results in this section describe the lower and upper bounds of confidence intervals using the notation *sample mean* ± *margin of error*. In all the experiments, we consider a high confidence level of 95%. Since the generated models were complex and had a long running time, we performed only 10 executions per model in order to estimate confidence intervals.

We repeat the following procedure ten times to estimate a 95% confidence interval: Firstly, we randomize the training segments dataset. Secondly, we find and select the best model in the iterations of the cross-validation step. Thirdly, we load the test segment partition and evaluate the found model. Then, we save the performance results. After this procedure, we compute the confidence intervals for the considered evaluation metrics.

4.1 Selection of Hyperparameters

We use the experiments in this section to identify the best parameters for the considered models with different settings. In order to determine the most suitable number of hidden neurons in each MLP architecture, we evaluate the network for one, two, and three hidden layers with 64, 128, 256, 512, 1024 and 4048 neurons per hidden layer. We take into account all I3D features in this experiment (2048 features for COMBINE disposition and 1024 features for RGB or FLOW disposition). We could verify that a neural network topology with 2048 neurons per hidden layer and only one single layer achieves better performance for the three different dispositions of I3D features. Specifically, with a network with one single layer and 2048 neurons, we achieve a confidence interval for AUC of 0.9118 ± 0.0033, 0.8874 ± 0.0014 and 0.8404 ± 0.0070 and a confidence interval for FPR of 0.0676 ± 0.0036, 0.0772 ± 0.0041 and 0.0562 ± 0.0049, for COMBINE, RGB and FLOW I3D features dispositions, respectively.

For the Binary SVM with RBF kernel, we experiment with different values for parameter C in SVM optimization. According to Wu and Kumar [2009], the parameter C acts as a regularization parameter and its value can be determined either analytically or experimentally. For these experiments, we reduced each dataset disposition (COMBINE, RGB and FLOW, respectively) to 205 principal components with the PCA method. With an SVM model with RBF kernel, an appropriate parameter C of 0.1, 1.0 and 0.5, we achieve a confidence interval for AUC of 0.9194 ± 0.0051, 0.8755 ± 0.0006 and 0.8766 ± 0.0011 and a confidence interval for FPR of 0.0054 ± 0.0002, 0.0626 ± 0.0003 and 0.0021 ± 0.0001 for COMBINE, RGB and FLOW I3D dispositions, respectively.

4.2 Performance Evaluation

This subsection presents the main performance results obtained from the evaluation of our proposed approach for video surveillance anomaly detection. Table 2 shows the considered model configurations in our performance evaluation. For each disposition of I3D features, we used the appropriate values of parameter C and network topology found in the previous subsection.

Table 2. Evaluated Model Settings

	METHOD	DISP	INPUT	PARAMS
SVM-COMB-2048F	SVM	COMBINE	2048	C = 0.1
SVM-COMB-624F	SVM	COMBINE	624 (PCA)	C = 0.1
SVM-COMB-205F	SVM	COMBINE	205 (PCA)	C = 0.1
SVM-RGB-1024F	SVM	RGB	1024	C = 1.0
SVM-RGB-624F	SVM	RGB	624 (PCA)	C = 1.0
SVM-RGB-205F	SVM	RGB	205 (PCA)	C = 1.0
SVM-FLOW-1024F	SVM	FLOW	1024	C = 0.5
SVM-FLOW-624F	SVM	FLOW	624 (PCA)	C = 0.5
SVM-FLOW-205F	SVM	FLOW	205 (PCA)	C = 0.5
MLP-COMB-2048F	MLP	COMBINE	2048	1L, 2048N
MLP-COMB-624F	MLP	COMBINE	624 (PCA)	1L, 2048N
MLP-COMB-205F	MLP	COMBINE	205 (PCA)	1L, 2048N
MLP-RGB-1024F	MLP	RGB	1024	1L, 2048N
MLP-RGB-205F	MLP	RGB	624 (PCA)	1L, 2048N
MLP-RGB-624F	MLP	RGB	205 (PCA)	1L, 2048N
MLP-FLOW-1024F	MLP	FLOW	1024	1L, 2048N
MLP-FLOW-624F	MLP	FLOW	624 (PCA)	1L, 2048N
MLP-FLOW-205F	MLP	FLOW	205 (PCA)	1L, 2048N

In Table 3 we summarize the performance values for the MLP and SVM models. From the results, we can observe the lower and upper bounds for the 95% confidence interval of the mean of each evaluated metric. We display the results for each metric as *sample mean (error estimate)*. We evaluated each model 10 times for confidence interval estimation. The evaluated models achieved approximate results for the considered performance metrics. The SVM models obtained the best results for AUC and FPR metrics. A high AUC shows better model performance and a lower FAR corresponds to a more consistent robustness of an anomaly detection technique.

The SVM models SVM-COMB-2048F, SVM-COMB-624F, and SVM-COMB-205F models trained with appearance and motion-based deep features obtained better results for the AUC and FPR metrics. The models SVM-COMB-624F and SVM-COMB-624F reflect the application of the PCA technique for data dimensionality reduction as a preliminary step to training a non-linear SVM. Although the AUC values for MLP models MLP-COMB-2048F, MLP-COMB-624F, and SVM-FLOW-205F are approximate to SVM, the percentage of false alarms was high for these experimented MLP models. We observe that the confidence intervals of models SVM-COMB-2048F and SVM-COMB-624F overlap for AUC, which indicates no difference between these two AUC means. In terms of FPR, we also observe that the SVM-COMB-2048F model overcomes the SVM-COMB-624F model with a small difference since they have the same sample FPR error estimate and distinct FPR sample means.

The Receiver Operating Characteristics (ROC) graphic Fawcett [2006] is a useful metric to compare the performance among different algorithms and presents the balance between False Positive Rate (FPR) and True Positive Rate (TPR). The performance of a given classification model is represented by a point in bi-dimensional space in the ROC graphic. In Figures 4, 5 and 6 we present the ROC curves for an arbitrary execution of ten rounds in each evaluated model and in each dataset disposition (COMBINE, RGB and FLOW).

The ROC curve allows accommodating uncertainty by making it possible to visualize all performance possibilities of the estimators Fawcett [2006]. We can observe that the performance results of the different models are approximate, and there are regions in the curve where a given test value has a high positive rate and a lower false positive rate.

Table 3. Performance Evaluation of Binary Classification Models for a Confidence Level of 95%

	AUC	FPR	BACC	PREC	REC	F1
SVM-COMB-2048F	**91.94**	**0.33**	**54.03**	**56.25**	**8.39**	**14.59**
	(0.17)	**(0.02)**	**(0.31)**	**(1.62)**	**(00.63)**	**(1.00)**
SVM-COMB-624F	**91.94**	**00.37**	**52.97**	**46.32**	**06.32**	**11.10**
	(0.17)	**(0.02)**	**(0.43)**	**(2.59)**	**(0.89)**	**(1.45)**
SVM-COMB-205F	**91.69**	**00.51**	**56.25**	**57.14**	**13.01**	**21.14**
	(0.19)	**(0.07)**	**(0.54)**	**(1.81)**	**(1.13)**	**(1.46)**
SVM-RGB-1024F	90.39	03.94	80.56	45.89	65.07	53.81
	(0.12)	(0.18)	(0.33)	(1.10)	(0.65)	(0.83)
SVM-RGB-624F	90.65	04.34	79.13	42.59	62.60	50.68
	(0.15)	(0.19)	(0.25)	(1.01)	(0.54)	(0.71)
SVM-RGB-205F	90.47	03.90	78.67	44.69	61.24	51.65
	(0.25)	(0.18)	(0.55)	(1.13)	(1.13)	(0.92)
SVM-FLOW-1024F	88.07	00.23	56.75	75.45	13.74	23.23
	(0.23)	(0.02)	(0.40)	(2.07)	(0.80)	(1.17)
SVM-FLOW-624F	88.21	00.26	57.01	73.69	14.29	23.91
	(0.27)	(0.04)	(0.39)	(2.09)	(0.80)	(1.07)
SVM-FLOW-205F	87.57	00.28	57.38	73.43	15.04	24.95
	(0.27)	(0.03)	(0.27)	(1.77)	(0.57)	(0.76)
MLP-COMB-2048F	91.35	07.12	82.30	34.11	71.71	46.21
	(0.31)	(0.27)	(0.55)	(0.79)	(1.19)	(0.76)
MLP-COMB-624F	90.77	06.89	80.74	33.77	68.38	45.20
	(0.33)	(0.26)	(0.67)	(0.79)	(1.41)	(0.83)
MLP-COMB-205F	91.26	07.84	81.78	31.88	71.41	44.06
	(0.30)	(0.38)	(0.81)	(0.83)	(1.80)	(0.85)
MLP-RGB-1024F	82.54	05.35	71.42	31.70	48.20	38.21
	(0.86)	(0.37)	(0.47)	(1.31)	(1.01)	(1.00)
MLP-RGB-205F	83.28	05.78	71.83	30.50	49.45	37.71
	(0.54)	(0.22)	(0.81)	(0.44)	(1.81)	(0.69)
MLP-RGB-624F	84.63	05.42	72.26	32.28	49.95	39.12
	(0.59)	(0.51)	(0.76)	(1.37)	(1.96)	(0.66)
MLP-FLOW-1024F	87.50	08.20	78.29	28.87	64.79	39.93
	(0.46)	(0.29)	(0.43)	(0.72)	(0.87)	(0.73)
MLP-FLOW-624F	88.61	08.42	80.06	29.49	68.53	41.23
	(0.33)	(0.33)	(0.35)	(0.70)	(0.83)	(0.65)
MLP-FLOW-205F	88.97	07.58	80.29	31.68	68.16	43.21
	(0.21)	(0.50)	(0.44)	(1.22)	(1.17)	(1.02)

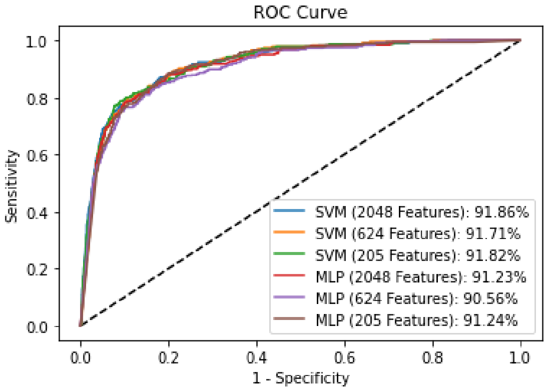

Figure 4. ROC Curves for MLP and SVM Models Trained and Evaluated from I3D Features (RGB + Optical Flow)

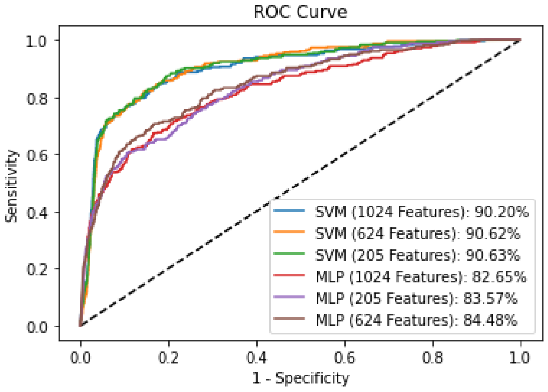

Figure 5. ROC Curves for MLP and SVM Models Trained and Evaluated from I3D Features (RGB)

We also compared our proposed approach with the state-of-the-art results available in Wan *et al.* [2020] and Kamoona

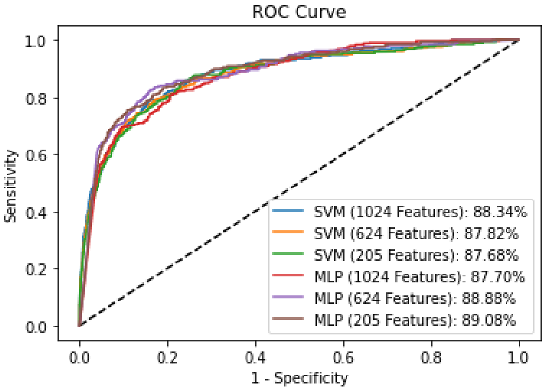

Figure 6. ROC Curves for MLP and SVM Models Trained and Evaluated from I3D Features (RGB + Optical Flow)

et al. [2020] studies, as described in Table 4. A confidence interval for a mean gives us a range of plausible values for the population mean for the considered evaluation metrics. For the model SVM-COMB-2048F, we have 95% of confidence where the interval 0.9194 ± 0.0017 contains the true value of the mean AUC and that the interval 0.0033 ± 0.0002 contains the true value of mean FPR.

Table 4. Comparison of Results

Approach	AUC (%)	FPR (%)
Wan *et al.* [2020] (MIL Loss + Center Loss)	91.24	0.10
Wan *et al.* [2020] (MIL Loss)	89.10	0.21
Kamoona *et al.* [2020]	89.67	-
Proposed Approach (SVM-COMB-2048F)	**91.94**	**0.33**

We can observe that some of the MLP models are able to achieve comparable performance results to the state-of-the-art in terms of AUC. However, this model has yet a high false alarm rate, when compared to the literature results achieved by Wan *et al.* [2020]. A possible explanation is due to the high capacity (i.e., high complexity) of these literature models, being capable to perform more complex detection tasks and generate few false alarms.

In Table 5, we compare anomaly detection results based on different dispositions of I3D features. We again display the achieved AUC and FAR rates available in Wan *et al.* [2020] for different feature extractors are for comparison to our approach. We chose the binary non-linear SVM models SVM-COMB-2048F, SVM-RGB-624F, and SVM-FLOW-205F as they are the best in AUC in COMBINE, RGB, and FLOW dispositions of I3D features, respectively For the results achieved with the proposed approach, we display the sample means of AUC and FPR, as detailed in the confidence intervals showed in Table 3. For the model SVM-RGB-624F-PCA, we have 95% of confidence that the interval 0.9065 ± 0.0015 contains the true value of the mean AUC and that the interval 0.0434 ± 0.0019 contains the true value of mean FPR. Similarly, for the model SVM-FLOW-205F, we have 95% of confidence that the interval 0.8757 ± 0.0027 contains the true value of the mean AUC and that the interval 0.0028 ± 0.0003 contains the true value of mean FPR. Furthermore, considering only the AUC metric, we observe that the achieved AUC in Wan *et al.* [2020] is out of the 95% confidence interval of the model SVM-COMB-2048F which indicates our approach is promising.

Table 5. AUC and FAR for Distinct Dispositions

		Wan *et al.* [2020]	**Proposed Approach**
COMBINE	AUC (%)	91.24	**91.94**
	FPR (%)	0.10	0.33
RGB	AUC (%)	85.38	**90.65**
	FPR (%)	0.27	4.34
FLOW	AUC	82.34	**87.57**
	FPR (%)	0.37	0.28

Our approach presents competitive results in AUC for the three dataset dispositions. Although the generated MLP models are slightly higher in FPR, the SVM models achieved approximate results. For the SVM-based models, we could observe a relation between that the use of optical flow-based features and low FPR rate, when compared to the SVM models built with only RGB-based features. We also observe the performance gains when we combine video appearance and motion information by simple concatenation, which indicates that these concatenated data patterns became more representative in the SVM algorithm. We conclude that the combination of a non-linear SVM classifier and the representation power of I3D deep features is robust and competitive as compared to the state-of-the-art approaches for the considered evaluated dataset.

5 Conclusion and Future Work

In this work, we present and evaluate a binary classification approach that combines dimensionality reduction with PCA and the MIL paradigm for the video surveillance anomaly detection problem. We use a set of I3D features that corresponds to the processed benchmark dataset ShanghaiTech for evaluation and comparison to our approach with state-of-the-art results. We also observe that the application of the PCA technique was useful to build competitive models by reducing the input space from 2048 features to 624 and 205 principal components, which explains a significant amount of the variance in training data. For SVM models built with deep features based on the combination of motion and appearance, the proposed approach achieved comparable results to the state-of-the-art literature in terms of AUC and approximate results in terms of FPR. Our approach is therefore robust to distinguish between abnormal and normal patterns.

Future directions include the evaluation of other approaches for feature representations and exploration of the potential of deep neural network architectures for video surveillance tasks in the context of weakly supervised machine learning. We aim to study the underlying characteristics of deep features as well as the explanations for their good performance in different video surveillance studies. We also plan additional experiments with other video anomaly datasets in order to mitigate possible difficulties when using large volumes of data.

Declarations

Authors' Contributions

All authors contributed to the writing of this article, read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data can be made available upon request.

References

Al-Dhamari, A., Sudirman, R., and Mahmood, N. H. (2020). Transfer deep learning along with binary support vector machine for abnormal behavior detection. *IEEE Access*, 8:61085–61095. DOI: 10.1109/ACCESS.2020.2982906.

Ali, S. and Shah, M. (2008). Human action recognition in videos using kinematic features and multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 32(2):288–303. DOI: 10.1109/TPAMI.2008.284.

Amraee, S., Vafaei, A., Jamshidi, K., and Adibi, P. (2018). Abnormal event detection in crowded scenes using one-class svm. *Signal, Image and Video Processing*, 12(6):1115–1123. DOI: 10.1007/s11760-018-1267-z.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874. ROC Analysis in Pattern Recognition. DOI: 10.1016/j.patrec.2005.10.010.

Jabeen, S., Saleem, S., Azam, A., and Khan, U. G. (2019). Scene recognition of surveillance data using deep features and supervised classifiers. In *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)*, pages 1–6. IEEE. DOI: 10.23919/ICACS.2019.8689001.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer. DOI: 10.1007/978-1-4614-7138-7.

Kamoona, A. M., Gosta, A. K., Bab-Hadiashar, A., and Hoseinnezhad, R. (2020). Multiple instance-based video anomaly detection using deep temporal encoding-decoding. *arXiv preprint arXiv:2007.01548*. DOI: 10.1016/j.eswa.2022.119079.

Li, T., Wang, Z., Liu, S., and Lin, W.-Y. (2021). Deep unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3636–3645.

Luo, W., Liu, W., and Gao, S. (2017). A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 341–349.

Nayak, R., Pati, U. C., and Das, S. K. (2020). A comprehensive review on deep learning-based methods for video anomaly detection. *Image and Vision Computing*, page 104078. DOI: 10.1016/j.imavis.2020.104078.

Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., and Yokoya, N. (2016). Video summarization using deep semantic features. In *Asian Conference on Computer Vision*, pages 361–377. Springer. DOI: 10.1007/978-3-319-54193-$8_2$3.

Pawar, K. and Attar, V. (2019). Deep learning approaches for video-based anomalous activity detection. *World Wide Web*, 22(2):571–601. DOI: 10.1007/s11280-018-0582-1.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pereira, S. S. and Maia, J. B. (2021). Uma abordagem baseada em redes neurais, multiple instance learning e pca para detecção de anomalias em videovigilância. In *Anais do XLVIII Seminário Integrado de Software e Hardware*, pages 123–130. SBC. DOI: 10.5753/semish.2021.15814.

Perera, P. and Patel, V. M. (2019). Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463. DOI: 10.1109/TIP.2019.2917862.

Petty, M. D. (2012). Calculating and using confidence intervals for model validation. In *Proceedings of the Fall 2012 Simulation Interoperability Workshop*, pages 10–14.

Rao, T. N., Girish, G., and Rajan, J. (2017). An improved contextual information based approach for anomaly detection via adaptive inference for surveillance application. In *Proceedings of International Conference on Computer Vision and Image Processing*, pages 133–147. Springer. DOI: 10.1007/978-981-10-2104-$6_1$3.

Ribeiro, M., Lazzaretti, A. E., and Lopes, H. S. (2018). A study of deep convolutional auto-encoders for anomaly detection in videos. *Pattern Recognition Letters*, 105:13–22. DOI: 10.1016/j.patrec.2017.07.016.

Roshtkhari, M. J. and Levine, M. D. (2013). An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. *Computer vision and image understanding*, 117(10):1436–1452.

Saha, B. N., Ray, N., and Zhang, H. (2009). Snake validation: A pca-based outlier detection method. *IEEE Signal Processing Letters*, 16(6):549–552. DOI: 10.1109/LSP.2009.2017477.

Shidik, G. F., Noersasongko, E., Nugraha, A., Andono, P. N., Jumanto, J., and Kusuma, E. J. (2019). A systematic review of intelligence video surveillance: Trends, techniques, frameworks, and datasets. *IEEE Access*, 7:170457–170473. DOI: 10.1109/ACCESS.2019.2955387.

Stathakis, D. (2009). How many hidden layers and nodes? *International Journal of Remote Sensing*, 30(8):2133–2147. DOI: 10.1080/01431160802549278.

Suarez, J. J. P. and Naval Jr, P. C. (2020). A survey on deep learning techniques for video anomaly detection. *arXiv preprint arXiv:2009.14146*. DOI: 10.48550/arXiv.2009.14146.

Sultani, W., Chen, C., and Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488.

Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J. W., and Carneiro, G. (2021). Weakly-supervised video anomaly detection with contrastive learning of long and short-range temporal features. *arXiv preprint arXiv:2101.10030*. DOI: 10.1109/ICCV48922.2021.00493.

Ullah, W., Ullah, A., Haq, I. U., Muhammad, K., Sajjad, M., and Baik, S. W. (2021). Cnn features with bi-directional lstm for real-time anomaly detection in surveillance networks. *Multimedia Tools and Applications*, 80(11):16979–16995. DOI: 10.1007/s11042-020-09406-3.

Wan, B., Fang, Y., Xia, X., and Mei, J. (2020). Weakly supervised video anomaly detection via center-guided discriminative learning. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE. DOI: 10.1109/ICME46284.2020.9102722.

Wu, X. and Kumar, V. (2009). *The top ten algorithms in data mining*. CRC press.

Zhao, H., Lai, Z., Leung, H., and Zhang, X. (2020). *Feature Learning and Understanding: Algorithms and Applications*. Springer Nature. DOI: 10.1007/978-3-030-40794-0.

Zhong, J.-X., Li, N., Kong, W., Liu, S., Li, T. H., and Li, G. (2019). Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1237–1246.