International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 35, No. 5 (2021) 2152003 (27 pages)

DOI: 10.1142/S0218001421520030

A Comparative Study of Transfer Learning Approaches for Video Anomaly Detection

Matheus Gutoski*, Manassés Ribeiro, Leandro T. Hattori, Marcelo Romero,
André E. Lazzaretti and Heitor S. Lopes

Laboratory of Bioinformatics and Computational Intelligence
Federal University of Technology – Paraná (UTFPR)
Av Sete de Setembro 3165, Curitiba (PR), 80230-901, Brazil
**matheusgutoski@alunos.utfpr.edu.br*

Received 15 May 2019
Accepted 29 August 2020
Published 9 December 2020

Recent research has shown that features obtained from pretrained Convolutional Neural Network (CNN) models can be promptly applied to a variety of problems they were not originally designed to solve. This concept, often referred to as Transfer Learning (TL), is a common practice when labeled data is limited. In some fields, such as video anomaly detection, TL is still an underexplored subject in the sense that it is not clear whether the architecture of the pretrained CNN model impacts on the video anomaly detection performance. In order to clarify this issue, we perform an extensive benchmark using 12 different pretrained CNN models on ImageNet as feature extractors and apply the features obtained to seven video anomaly detection benchmark datasets. This work presents some interesting findings about video anomaly detection using TL. The highlights of our findings were revealed by our experiments, which have shown that a simple classification process using One-Class Support Vector Machines yields similar results to state-of-the-art models. Moreover, a statistical analysis suggests that architectural differences are negligible when choosing a pretrained model for video anomaly detection, since all models presented similar performance. At last, we present an in-depth visual analysis of the Avenue dataset, and reveal several aspects that may be limiting the performance of state-of-the-art video anomaly detection methods.

Keywords: Transfer learning; anomaly detection; dataset analysis; deep learning.

1. Introduction

The increasing concern with public security, allied to the decreasing cost of hardware, has made surveillance cameras omnipresent in private and public spaces. However, the number of available human observers has not grown in the same proportion as the number of surveillance cameras. In this scenario, the effectiveness of the footage is greatly hindered, since the human endeavor required for effectively

*Corresponding author.

observing it is still too high. The main drawback of human-based video surveillance is that the security footage is often used in a reactive, rather than in a proactive way. That is, footages are often analyzed after a fact has occurred, e.g. for identifying the author of a prohibited act. In such a scenario, an automatic online anomaly detection system could detect misconduct immediately, alerting the appropriate authorities and allowing for quick corrective action. Hence, automatic video anomaly detection is a subject of great importance to public security and has drawn great attention from the Computer Science community.[9]

Recently, Deep Learning (DL) methods have achieved the state-of-the-art results on many image-related problems, including abnormal event detection and recognition in video sequences,[15,21] face recognition,[7] object tracking and segmentation,[26] video prediction and analysis,[30] among countless other applications.

Despite the excellent performance of the above-mentioned DL methods, performing automatic video anomaly detection still remains a difficult task, mainly because anomalous events in videos are ill-defined and strongly context dependent. For instance, an agitated crowd can be considered normal or anomalous, depending on the context of the scene.[4]

Since contextual knowledge is not always included in the training data, it may be necessary to acquire it from an external source. One way to tackle this problem is to incorporate knowledge acquired at a different, yet similar, task. For instance, a network trained for classifying images of objects may have acquired important knowledge about shapes, textures, and colors during its training phase. Thus, somehow this previously acquired knowledge may be useful for solving tasks that include similar objects. This concept is often referred to as Transfer Learning (TL).[14,19]

In the recent literature, very few works have explored TL in the context of anomalous event detection in surveillance videos,[4,21] achieving state-of-the-art performance and encouraging further research.

Both of the above-cited works employ pre-trained Convolutional Neural Network (CNN) models as feature extractors. However, the experiments performed by the authors only explored a small number of models and benchmark datasets. Reference 21 employs the VGG-f[1] model as a feature extractor and argues that deeper models, such as GoogLeNet,[24] ResNet[3] and VGG-verydeep[1] may achieve better anomaly detection performance. Hence, a more extensive benchmarking of pre-trained CNN models, ranging from relatively shallow to very deep, is important for better understanding their performance on video anomaly detection tasks.

Reference 4 also points out another important issue regarding a particular benchmark dataset (Avenue), which is one of the most popular video anomaly detection dataset. It is argued that some static objects that appear in the test set were incorrectly labeled as normal. This raises an important question: is the performance of current video anomaly detection methods limited due to incomplete or mislabeled datasets? A possible way to tackle this question is by performing a thorough analysis

of the classification results and attempt to understand the reason behind the misclassification of frames.

This work has two main parts. First, we perform video anomaly detection by employing TL as a way to extract feature representations from surveillance video data. Frames are then classified as normal or anomalous using One-Class Support Vector Machines. Our method is significantly simpler than other state-of-the-art methods in the literature, since we do not split frames into patches or use any complex form of temporal learning. Instead, we employ a simple moving average filter over time to minimize the impact of noise. Other factors that make our approach much simpler are: (1) Easy implementation using current Deep Learning and Machine Learning frameworks; (2) Low computational power requirements, since we employ pre-trained Deep CNN models. The only training required in our method is performed by the One-Class SVM; (3) Low computational time, since the full algorithm can be executed in just a few minutes on a moderate machine. Nonetheless, we show that our method can achieve results that are comparable to other state-of-the-art methods for anomalous event detection at the frame level. The main problem we tackle with this approach is the lack of general guidelines for choosing a feature extraction model. The second part of this work focuses on a visual analysis of the misclassification cases in the Avenue dataset so as to have insights on the classification limits imposed by the data, not the methods.

The highlights of this paper are as follows:

- A simple yet effective method for performing anomaly detection in security videos using TL, achieving results comparable to other state-of-the-art approaches;
- An extensive benchmarking and performance evaluation of 12 CNN models on seven video anomaly detection benchmark datasets;
- A visual-level analysis of the results obtained in the Avenue dataset,[11] which points to possible problems that may be limiting the classification performance of anomaly detection models.

This paper is organized as follows: Section 2 presents theoretical aspects about TL. Section 3 presents a detailed explanation of the method used for performing video anomaly detection. Section 4 presents the experiments and results. Section 5 presents the visual analysis of the results, and finally, Sec. 6 presents the final remarks of this work.

2. Transfer Learning

Machine learning algorithms work under the premise that the training and test data belong to the same distribution, which is a strong assumption for real-world problems.[14,19] Contrariwise, TL assumes that these distributions may be different, and a robust model may achieve satisfying results when applied to completely new problems.[10]

Reference 14 defines TL as follows: given a source domain D_s, a source task T_s, a target domain D_t and a target task T_t, TL aims at improving the learning of a target predictive function $f_t(\cdot)$ in D_t by using the previously acquired knowledge in D_s and the knowledge in T_s, given that $D_s \neq D_t$ and $T_s \neq T_t$.

TL is commonly used when facing two main problems: insufficient computing power to train a large model or the lack of a substantial amount of labeled training data. The computing power problem is softened by not having to train the model from scratch. In its turn, the lack of labeled data also becomes a smaller problem, since it was shown that a fine-tuning process of the last layer with a small amount of data can lead to a satisfactory performance on classification problems.[29] Specifically, CNNs can be used as a feature extractor by forwarding a new image throughout the network and capturing its latent representation at any hidden layer. Hence, labeled data is not required in this approach, given that the CNN model in question was originally trained to solve a similar problem. According to Ref. 14 formalism, our approach is classified as feature-representation-transfer under inductive TL.

While the data used to train a model play an important role in obtaining robust features, the architecture of the model itself may also impact on the features. In general, deeper models have shown better performance than shallow models on image classification tasks, such as the ILSVRC.[3,24] This performance gain may be attributed to the high-level features, learned at the deeper layers of the network. However, it is not clear if such features are ideal for tasks outside of the scope of the original problem, since they may be overly adapted to it. To circumvent this issue, one can extract features from a layer in the middle of the network. Nonetheless, this solution can be computationally costly, since the dimensionality in the middle layers is usually much higher than those of the final layers. For instance, they can reach hundreds of thousands in models such as GoogleNet and ResNet152. Another option is to extract the features from the final layers of shallowest networks. This may produce lower-level features while keeping dimensionality on a reasonable range of a few thousands.

Some experimental works in the recent literature have shown that the depth in which the features are extracted may affect the final classification results.[21] However, to date, there are still no general guidelines for selecting the most appropriate model or choosing the most appropriate layer to perform feature extraction. Hence, trial-and-error has been the current approach.

3. Method

In this work, we use models trained for classifying images of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).[16] The dataset contains approximately 14 million images and 1000 different classes. Such models are easily accessible, given the growing amount of pre-trained models available at the repositories of major DL frameworks. An overview of the proposed method is shown in Fig. 1. Each part will be detailed in the following sections.

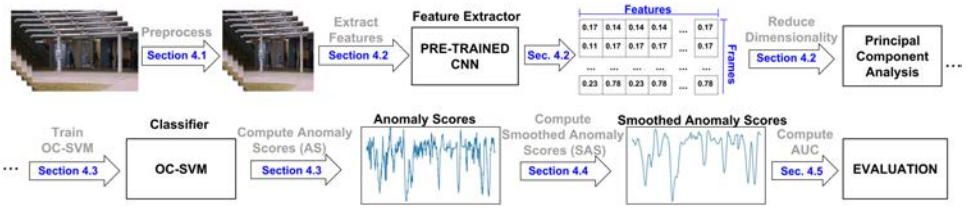

Fig. 1. Overview of the proposed video anomaly detection method.

3.1. *Data preprocessing*

All video anomaly detection benchmark datasets used in this work are publicly available. All videos were discretized into a sequence of frames covering the full video duration. Each frame was previously labeled as normal or anomalous by the original creators of the datasets.

The first preprocessing step was to resize the frames to 224×224 pixels using an interpolation algorithm, similar to Ref. 21. The second step was to normalize the pixels between $[-1, 1]$. Since the networks take RGB inputs and some of the datasets provide grayscale frames, we replicate the grayscale channel three times.

3.2. *Feature extraction*

In this work, 12 CNNs were used as feature extractors, as follows: AlexNet[8]; the original GoogLeNet model and Inception v3, which is an improvement upon the original model[24]; the ResNet model variations[3]; the VGG model variations[1]; and the more recent DenseNet.[5] All of the model's weights can be found in either Caffe, TensorFlow,[20] or Keras model repositories.

Feature extraction was done by forwarding each frame throughout the network and capturing the information at the last pooling layer. No fine tuning process is done before the feature extraction.

It is important to note that the dimensionality at the last pooling layer is significantly smaller when compared to any other convolutional or pooling layer in the network, making the process computationally feasible. To further reduce the dimensionality, we employ the Principal Component Analysis (PCA) algorithm. The PCA model is fit by using the features obtained from the training set. This model is then used to reduce the dimensionality of the test set. We aim at minimizing the number of principal components, under the constraint that at least 98% of the variance has to be preserved. This reduces both the dimensionality from thousands to, in some cases, less than a hundred features, and the computational effort of the classification process, described in the following section.

3.3. *Classification using OC-SVM*

Once the features are extracted, the classification process is performed by an OC-SVM. Therefore, only normal examples are used in the training phase.

In order to perform the classification, we compute the distance of each test data point $\mathbf{z}$ to the decision border of the hypersphere by using Eq. (1). We used this distance as an anomaly score (AS), which is considered anomalous if it is less than zero:

$$\mathrm{AS} = \sum_i \alpha_i K(\mathbf{z}, \mathbf{x}_i) - \frac{1}{2}\left[1 + \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - R^2\right], \tag{1}$$

where $\boldsymbol{\alpha}$ are the Lagrange multipliers, K is the Gaussian kernel function, R is the radius of the hypersphere, and $\mathbf{x}_i$ and $\mathbf{x}_j$ are a pair of support vectors. Since we employ the Gaussian kernel, the formulation of the SVDD model is equivalent to the OC-SVM proposed in Ref. 18, as pointed out by Ref. 25.

3.4. *Moving average filter*

In videos, anomalies generally occur over time, which means that temporal factors are relevant for the classification task, as shown by Ref. 31. Moreover, several prior works have used temporal descriptors to perform video anomaly detection.[22,28] Hence, a simple moving average filter is used to remove some noise in the classification process.

For each frame i on a continuous video sequence, a Smoothed Anomaly Score SAS_i is proposed, which is the smoothed AS_i (Eq. (1)). The SAS_i is computed by using a moving average filter, according to Eq. (2), where s is the size of the moving average mask and i is the current frame.

$$\mathrm{SAS}_i = \frac{1}{s} \sum_{j=0}^{s-1} \mathrm{AS}_{i+j}. \tag{2}$$

This strategy ensures that the current frame SAS_i is influenced by forthcoming frames. Since most of the datasets used in this work provide a sequence of different videos that have time gaps between them (cuts and changes of scenario), the SAS_i is calculated over continuous (uncut) video sequences only.

In a real-life application, this filter has the downside of forcing the anomaly detection system to operate s frames behind the real-time footage. However, for most applications, this small time gap will be neglectable.

3.5. *Evaluation*

For evaluating the results, we use the Area Under the Receiver Operating Characteristic Curve (AUC). This measure ensures that the classifier is evaluated at many different thresholds, i.e. the distance cutting points above which a sample is considered anomalous.

4. Experiments and Results

All experiments done in this paper were run on a computer with an Intel Core i7 processor at 3.30 GHz, Nvidia Titan X GPUs, and minimal installation of Ubuntu 18.04 LTS. Despite using GPUs for forwarding frames throughout the Deep Learning models, this task could be easily accomplished using regular CPUs.

4.1. *Benchmark datasets*

Seven video anomaly detection benchmark datasets were used for evaluating the performance of the proposed method:

UTFPR-HSD1 and **UTFPR-HSD2**[a] are video anomaly detection datasets aimed at detecting traffic anomalies on busy highways. They were shot at Curitiba, Brazil, by the Federal University of Technology, Paraná. In order to avoid traffic jams in rush hours, long trucks are restricted during certain times of the day. Therefore, scenes containing at least one long truck are considered anomalous, whereas scenes containing any other vehicles are considered normal. UTFPR-HSD1 and UTFPR-HSD2 differ due to the differences in camera angle, elevation, illumination and location. The UTFPR-HSD1 dataset is composed of 6 602 frames in the training set, and 1 660 frames in the test set. In its turn, UTFPR-HSD2 contains 5 640 frames in the training set and 1 986 frames in the test set. These datasets are fully labeled in such a way they can be used for other purposes such as multi-label classification.

The **Avenue** dataset[11] currently serves as a benchmark for video anomaly detection. Normal events are defined by people walking in different directions. Anomalous events occur when people run, throw objects or loiter. The training dataset contains 15 328 frames, and the test dataset contains 15 324 frames. Recently, a modified version of the Avenue dataset was introduced by Ref. 4, named **Avenue17**. In this version, the train set remains the same, while the test set is smaller because some videos were removed. The author argues that some objects in these videos were not annotated properly. This version contains 15 328 frames in the training set and 10 622 frames in the test set. Further discussion about this subject can be found in Sec. 5.

UCSD Ped1 and **UCSD Ped2**[12] are video anomaly detection datasets captured by stationary cameras in a pedestrian walkway. Walking pedestrians are considered normal events, and anomalies occur when vehicles, skateboarders, bicycles, and wheelchairs pass throughout the scene. The Ped1 dataset contains 6 800 frames in the training set and 2 000 frames in the test set. In its turn, the Ped2 dataset contains 2 550 frames in the training set and 2 010 frames in the test set.

The **UMN** dataset[13] contains footage of crowd behavior in three different locations. Normal events are defined by people walking randomly, while anomalies occur

[a] UTFPR-HSD1 and UTFPR-HSD2 are Highway Surveillance Datasets available at https://github.com/bioinfolabic/UTFPR-HSD.

when the crowd evades the scene by running outwards. The training set contains 5 122 frames and the test set contains 1 382 frames.

4.2. *Experiments*

The experiments presented in this section aim at answering important questions that arise in TL applications: (i) Which model should be chosen for the video anomaly detection problem? (ii) How does the performance of TL is compare with other methods in the literature?

As an attempt to answer these questions, we evaluate the performance of 12 different pretrained CNN models (in the object recognition task in images), and used them as feature extractors for the video anomaly detection problem. Moreover, results are compared to the state-of-the-art methods and a simple baseline. The baseline method uses the well-known Histogram of Oriented Gradients (HOG) with its default parameters as a feature extractor. HOG was used as feature extractor in the very same way as the CNN models, as shown in Fig. 1.

The experiments follow the description provided in Sec. 3. The OC-SVM is sensitive to its control parameters, hence, we used a factorial experiment to find the best set of parameters, with the following values: $\{0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 0.9\}$ and $\{0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1.0\}$ for the kernel and the regularization parameter, respectively. We repeat this process for every model and dataset for fairness in the comparison, i.e. all models are compared using their best combination of parameters. Regarding the Moving Average Filter, we also perform a parameter search for determining the best value for s. The set of parameters includes the arbitrarily chosen values $\{2, 3, 5, 7, 10, 15, 20, 30, 50, 75, 100, 150, 200\}$. Only the results obtained by the best set of parameters are reported in the following section.

4.3. *Results*

The experimental results for all datasets are presented in Table 1.

While many of the models present similar results for a given dataset, some of them stand out. AlexNet, for instance, outperformed all other models on the UCSD Ped1 dataset, which is one of the most challenging among the datasets used in this work. This is surprising, due to the simplicity of AlexNet when compared to some of the deeper models. Another case is the VGG-M network applied to the UCSD Ped2 dataset, which achieved significantly higher performance than the other models. A more recent model, DenseNet, has presented the best results in three out of seven datasets. However, it still performed below VGG-M in average.

An important issue is raised when all datasets are considered: are there significant differences in the performance of the models? The average anomaly detection performance, shown in Table 1, indicates that the difference in performance between the models is quite small. Notwithstanding, this small difference may not be convincing without a statistical analysis. Therefore, the Friedman test[2] was used to measure

Table 1. Results obtained for each model and dataset measured by the AUC.

Dataset	AlexNet	GoogleNet	Incep. v3	ResNet10	ResNet50	ResNet152	VGG-S	VGG-F	VGG-M	VGG-16	VGG-19	DenseNet
UTFPR-HSD1	0.918	0.962	0.936	0.892	0.947	0.913	0.948	0.885	0.942	0.891	0.936	**0.972**
UTFPR-HSD2	0.775	0.898	0.936	0.842	0.950	0.872	0.907	0.921	0.914	0.922	0.783	**0.956**
Avenue	0.813	0.829	0.762	0.819	0.732	0.821	0.803	0.836	0.820	**0.847**	0.827	0.833
Avenue17	0.886	0.847	0.824	0.847	0.848	0.864	0.876	0.882	0.880	**0.904**	0.868	0.865
UCSD Ped1	**0.719**	0.674	0.629	0.651	0.686	0.680	0.636	0.642	0.677	0.688	0.664	0.630
UCSD Ped2	0.746	0.731	0.759	0.566	0.513	0.617	0.786	0.784	**0.893**	0.662	0.612	0.821
UMN	0.911	0.960	0.983	0.946	0.983	0.981	0.981	0.981	0.988	0.776	0.950	**0.992**
Average	0.824	0.843	0.832	0.794	0.808	0.821	0.848	0.847	**0.873**	0.812	0.805	0.867

whether there is a significant difference between the performance of each model considering all datasets. This test does not assume that the data follow a normal distribution. Hence, it is adequate for analyzing our results. Apply the results shown in Table 1 (with swapped rows and columns) to the Friedman test equation:

$$\chi_r{}^2 = \frac{12n}{p(p+1)} \sum_{j=1}^{p} \left\{ \bar{r}_j - \frac{1}{2}(p+1) \right\}^2, \tag{3}$$

where n is the number of rows, p is the number of columns, and $\bar{r}_j$ is the mean rank of the jth column of a new matrix obtained by ranking the standard deviations of each row of Table 1. This computation yields a $\chi_r{}^2$ value of 14.87. The p-value for the test is given by $\mathbf{P}(\chi^2_{p-1} \geq \chi_r{}^2)$, which results in a p-value of 0.188. Given that $0.188 \geq 0.05$, the null hypothesis is not rejected,[2] indicating that there is no significant difference between the models. In other words, there is no evidence that any particular model extracts better features in comparison to other models for video anomaly detection in the particular scenario studied in this work. This suggests that the architecture of the model may not be a decisive factor for selecting a feature extractor for this specific problem. Moreover, the similarities in performance may be partially explained by the fact that all models were trained using the same data, i.e. the ImageNet dataset.

Table 2 compares the baseline feature extractor (HOG) and the state-of-the-art methods for each dataset in terms of AUC. TL performed much better than HOG as feature extractor, considering the best result obtained for each dataset. Moreover, our approach performs very similar to the state-of-the-art methods on most datasets.

Our experimental results have shed some light on both of the questions posed at the beginning of this section, as follows: (i) Since no significant difference was found

Table 2. Anomaly detection performance comparison measured by the AUC. (left) The results of our approach using HOG and TL (right) the state-of-the-art found in the literature.

Dataset	Baseline (HOG)	TL (best)	State-of-the-art	Method	Reference
UTFPR-HSD1	0.851	**0.972**	—	—	—
UTFPR-HSD2	0.552	**0.956**	—	—	—
Avenue	0.798	0.847	**0.878**	Narrowed Motion Clusters	Ref. 6
Avenue17	0.863	**0.904**	**0.904**	Narrowed Motion Clusters	Ref. 6
UCSD Ped1	0.616	0.719	**0.927**	Local Spatio-Temporal Anomalies	Ref. 17
UCSD Ped2	0.701	0.893	**0.908**	Appearance and Motion DeepNet	Ref. 27
UMN	0.873	0.992	**0.997**	Online Growing Neural Gas	Ref.23
Average	0.750	0.895	—	—	—

between the performance of the models over all datasets, it may be the case that differences regarding the architecture of the models (such as their depth) are not decisive factors for video anomaly detection. Hence, a good choice may be a simple model, since the computational effort is much lower than using other deeper and more complex models. (ii) Features obtained using the TL approach yield significant results on video anomaly detection tasks. Our experiments have shown that such features outperforms classical methods such as HOG, and perform similar to the state-of-the-art methods. Precomputed features for the best results and sample evaluation code can be found at `https://github.com/bioinfolabic/Transfer-Learning-Issues-in-Video-Anomaly-Detection`.

5. Visual Analysis

Most of the current research in the video anomaly detection field relies on popular benchmark datasets for performance evaluation. However, very few works have addressed the solidity of the human made annotations that indicate whether an event is normal or anomalous.[4] This is an important issue, since mislabeled data may prevent methods from correctly classifying some anomalous events. Notwithstanding, this may partially explain the gap between the current state-of-the-art performance and the ideal 100% classification accuracy.

In some datasets, anomalies are well-defined, such as in the UCSD Ped1 and Ped2 datasets (objects such as bicycles, skateboards, vehicles and wheelchairs establish the anomaly), the UMN dataset (a panic situation establishes the anomaly), and in the UTFPR-HSD datasets (long trucks clearly characterizes the anomalies). However, in some datasets such as Avenue, anomalies are ill-defined since they depend on context and human behavior. This makes the task of annotating the data much trickier, and leaves some aspects of abnormality open to interpretation. Notwithstanding, this may lead to uncertainty when evaluating anomaly detection methods.

The Avenue dataset is one of the most used in anomaly detection studies. However, no method has, yet, achieved 100% of classification accuracy. Recently, Ref. 4 raised some issues about the correctness of the annotation of frames. This lead to the creation of another "cleaned" dataset, Avenue17. Therefore, we performed a deep analysis of all frames of the original Avenue dataset, so as to unveil possible problems not previously known. In order to perform this analysis, we use the SAS (Eq. (2)) obtained by using the VGG-16 features. We choose this model because it presented the best performance on the Avenue dataset. Within our analysis, we identify four categories of problems that occur in the Avenue dataset that may cause misclassification.

We define the first category as **continuity interruption**. This problem happens when the ground truth changes abruptly because the agent that causes the anomaly briefly leaves and then re-enters the scene. This type of problem happens for many reasons, such as the occlusion of a person when running behind a pillar, or an object

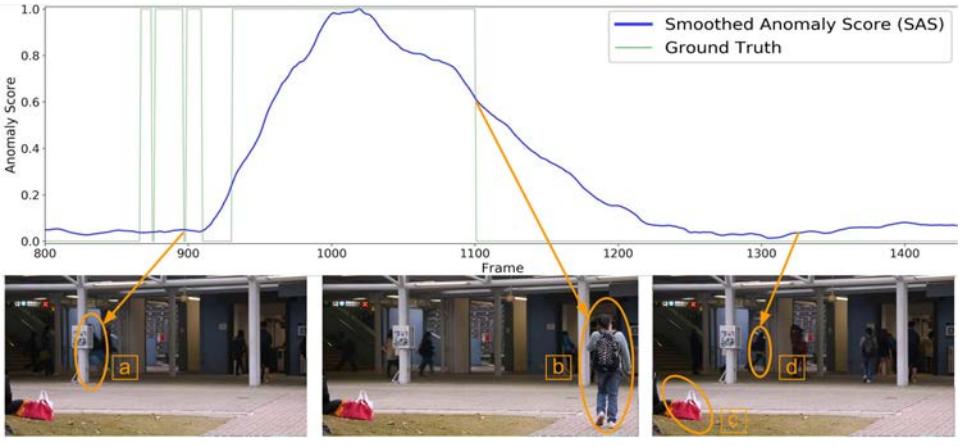

Fig. 2. Visual Analysis of Avenue dataset and the Four Categories of Annotation Problems. The letters in the figure indicate the categories as follows: (a) Continuity interruption: The next couple frames are considered normal since the person running is behind the pillar. Notice the oscillations in the ground truth. (b) Abrupt change in the ground truth: The next frame is considered anomalous, despite being strictly similar to the current frame, which is considered normal. (c) Static object mislabeling: Some static objects are not included in the training videos, yet, they are considered normal. (d) Labeling inconsistency: The person running is considered normal, despite being considered an anomalous event in multiple other videos. Refer to the text for more details.

leaving the frame momentarily after being thrown upward. This issue occurs due to the way the data was annotated, which is at pixel level. That is, the frame-level annotations are generated by the user as follows: a frame is considered anomalous if at least one pixel was annotated as such. This causes continuity interruptions in the ground truth, i.e. it oscillates very rapidly within a small time window. We argue that a certain continuity should be considered when the frame-level ground truth is generated, since anomalous events generally occur over time. Figure 2(a) shows a case where the ground truth oscillates rapidly because the person running passes behind pillars. This is also a case where our model fails to recognize the person running as an anomaly. This may be partially explained by the fact that some train videos contain people running.

The second category is defined as **abrupt change in the ground truth**. That is, the ground truth changes from normal to anomalous in the middle of a continuous event. For instance, a person walking away from the camera ceases to be considered anomalous after crossing an invisible proximity threshold. An example from this type of problem can be seen in Fig. 2(b). Note that the SAS follows a gradual decrease as the person walks away from the camera. This type of problem is not exclusive to the Avenue dataset, since the problem of defining the exact point in which an anomaly ceases (or starts) to exist is common among all datasets.

The third category is defined as **static object mislabeling**. This category of problems was originally observed by Ref. 4. It refers to anomalous static objects or

Table 3. Summarization of the problems identified and their indexes in the Avenue dataset test videos. Each case is shown in the Appendix Section.

Problem	Video ID	Initial Frame	Final Frame
Abrupt change in GT	1	930	10102
	6	175	630
	11	240	360
	15	400	600
	19	1	248
Labeling inconsistency	1	1290	1439
	13	230	285
	17	33	56
Continuity interruption	1	74	124
		390	440
		863	910
	9	548	550
	10	580	805
	11	95	115
	12	655	675
	13	465	475
	14	410	420
Static object mislabeling	1;2;8;9;10	entire video duration	
	6	830	850
	11	1	15
	11	125	145
	12	578	593
		800	820

Note: GT: Ground Truth.

people that were labeled as normal. We support the idea of the authors, since, in our point of view, static objects that were not included in the training set should be considered anomalous objects when they appear in the test set. The solution to this problem was simply removing the videos in which this type of problem occurred, which lead to the creation of the Avenue17 dataset. Figure 2(c) illustrates an instance of this type of problem.

The fourth category is defined as **labeling inconsistency**. In some cases, the labels completely ignore some ongoing events that should logically be considered anomalous. For instance, people running were considered anomalous in some situations, but normal in others. Figure 2(d) shows an example of this type of problem. In this situation, frames where a person appears running were labeled as normal. In this case, our model did not detect the person running as an anomaly, but it was considered a hit from the AUC perspective. Notice that label errors may cause both inadequate increases and decreases in terms of AUC, hindering the evaluation consistency. Finally, Table 3 presents the video ID in which each of the problems described before occur. The frame window in which the problem takes place is also included, indicating the number of the frame when the problem starts (Initial frame)

and when it ends (Final frame). Each table entry is shown graphically in the appendix section.

6. Conclusions

In this work, we proposed a simple method based on TL and OC-SVMs for performing anomaly detection in videos. TL was used as a way for extracting features from video frames. Our results suggest that the features extracted this way contain valuable information that can discriminate between normal and anomalous events. Despite the simplicity of the method, we have shown that CNNs pretrained on ImageNet yield powerful feature extractors, leading to nearly state-of-the-art performance on video anomaly detection.

Additionally, we have performed a benchmark evaluation of 12 different CNN models. Our results suggest that there is no significant difference in performance between the models when applied to video anomaly detection, despite of the architectural differences. While some models perform best on specific datasets, their performance tends to be even when averaging across all datasets. This may be partially explained by the *No Free Lunch* theorem, which states that the performance of optimization algorithms are equal when averaged across all possible problems. Hence, the choice of a model that performs feature extraction for video anomaly detection may be as simple as picking the model with fewer parameters, since it is computationally cheaper.

Although TL methods have achieved good practical results, there are still open issues regarding the method. It is not clear what is being transferred between problems/domains. Similarly, there are still no guarantees regarding the extensibility of TL, it has evolved much more in the practical than in the theoretical grounds. These issues shall be addressed in the future.

Our final contribution in this paper is a criticism to the current evaluation strategy used in recent video anomaly detection research. We performed an in-depth visual analysis of a popular benchmark dataset (Avenue), and concluded that some ground truth annotations are not fair to classification methods. This may explain the reason why state-of-the-art methods were, to date, unable to classify the entirety of the dataset correctly. Since the ground truth in this dataset is dependent on subjective human interpretation, maybe the intrinsic features of the data, not the methods, are the main factor that set an upper-bound to the overall classification performance. This observation raises some issues. First, it is possible that current video anomaly datasets, in general, have inconsistencies in the ground truth. Second, it is clear the need for a representational formalism to deal with the subjectiveness of human interpretation of visual scenes. Third, more robust evaluation methods are needed, capable of being less sensitive to the complex nature of the semantic noise, which is intrinsic to real-world data.

Acknowledgments

Author M. Gutoski would like to thank CNPq for the scholarship number 141983/2018-3; M. Ribeiro thanks the Catarinense Federal Institute and IFC/CAPES/Prodoutoral for the scholarship; M. Romero thanks the Organization of the American States, the Coimbra Group of Brazilian Universities and the Pan American Health Organization; L. T. Hattori thanks CAPES for the scholarship; Author H. S. Lopes would like to thank to CNPq for the research grant number 440977/2015-0. All authors would like to thank Fundação Araucária and NVIDIA for the donation of Titan-XP GPUs used in the experiments.

Appendix

In this section, we present a graphical analysis of the problems shown in Table 3. Figures are presented in the same order as shown in the table.

Abrupt change in ground truth

We define abrupt changes in ground truth as events that cease to be considered normal or anomalous abruptly in the middle of a continuous event.

This is the case in Fig. A.1, where a girl walks from the front to the back of the scene. In this case, the anomaly is defined by the proximity of the girl to the camera. Hence, as she walks away, the scene becomes less and less anomalous.

The problem arises because of the discrete nature of the annotations. At some point, it is necessary to change the annotation from "completely normal" to

Fig. A.1. Abrupt change in GT (Video 1). Here, the ground truth changes from anomalous to normal when the girl crosses an invisible proximity threshold.

Fig. A.2. Abrupt change in GT (Video 6). Here, the ground truth changes from normal to anomalous when the man reaches a certain proximity to the camera.

"completely anomalous" without any transitional state in between. The same happens in Figs. A.2–A.5.

Labeling inconsistency

We define labeling inconsistencies as events that are considered normal in some situations and anomalous in other situations with no apparent reason.

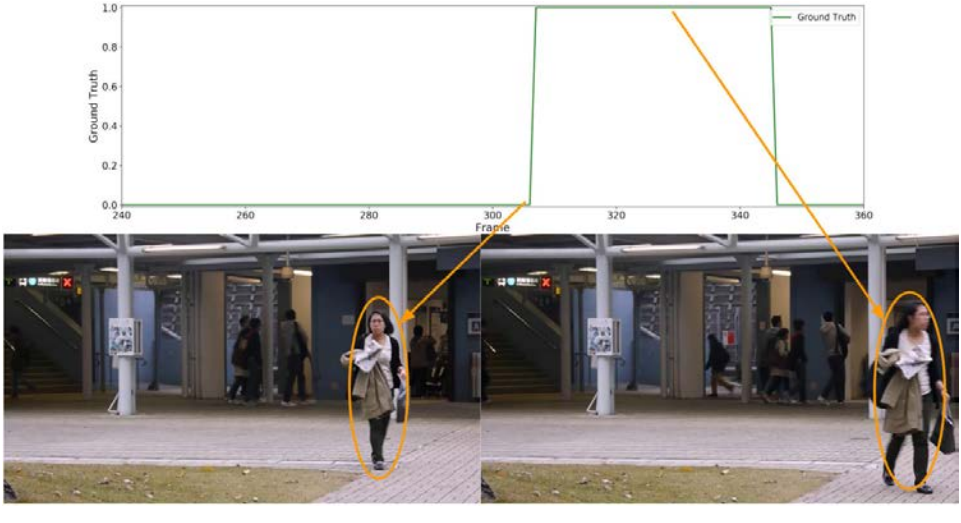

Fig. A.3. Abrupt change in GT (Video 11). The problem is the same as in Fig. A.2.

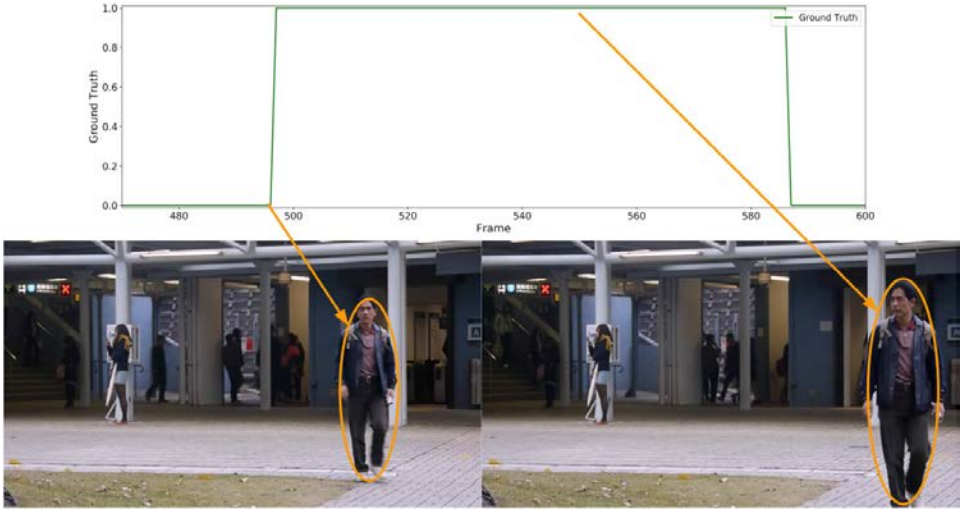

Fig. A.4. Abrupt change in GT (Video 15). The problem is the same as in Fig. A.2.

For instance, in Fig. A.6, people running in the background were considered anomalies as shown in (a) and (b). However, some other people running in the same video were not considered anomalous, as seen in (c) and (d).

In Fig. A.7, the man holding papers was considered normal up to the point where he turns to the camera. This is not in accordance with the annotation procedure, since there are no other instances in the dataset where this kind of action was considered anomalous. The simple act of facing the camera was considered

Fig. A.5. Abrupt change in GT (Video 19). The problem is the same as in Fig. A.2.

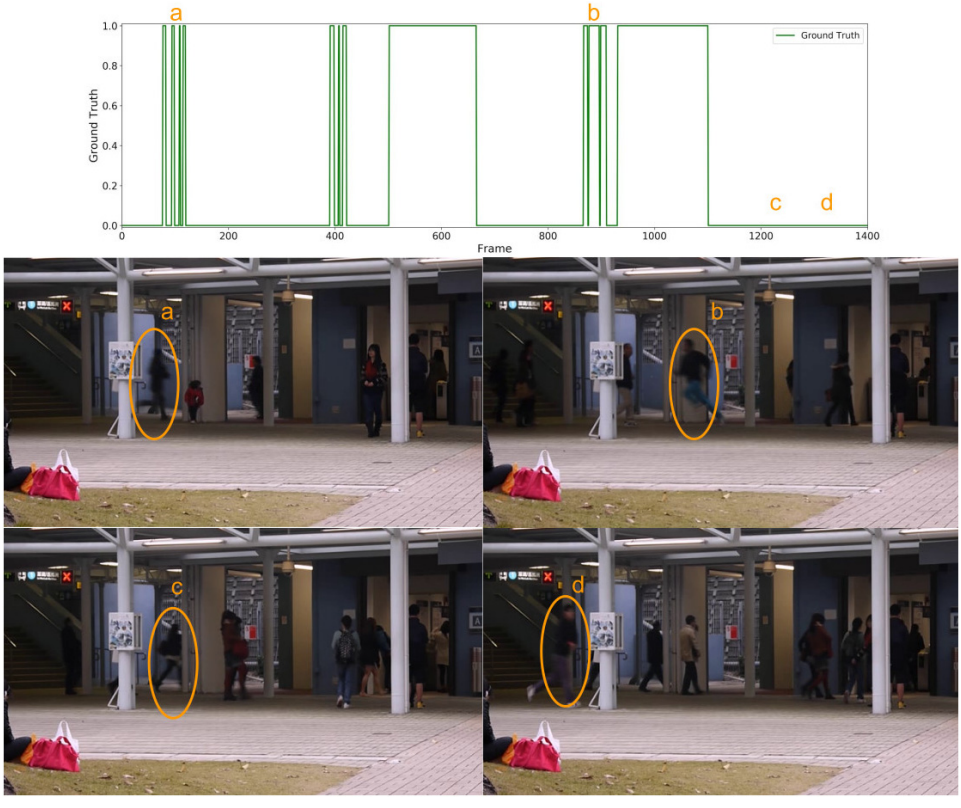

Fig. A.6. Labeling inconsistency (Video 1). In (a) and (b), people running were considered anomalous, while in (c) and (d), people running were considered normal.

anomalous, even though the scene looks very similar to what it was a few frames earlier. We consider this to be a very weak reason for defining this action as an anomaly.

In Fig. A.8, there seems to be a delay when updating the ground truth. The subject highlighted in the scenes was intended to be considered anomalous while dancing, which occurred from frames 0 up to about 45. Then, the subject stops dancing and stands still, which was considered a normal event (a person standing). The problem occurs in the approximate window of frames 45 through 58, where the ground truth remains anomalous while the subject is standing still. According to the procedure proposed, i.e. dancing is considered anomalous and standing still is considered normal, this window of frames should be labeled as normal.

Continuity interruption

We define continuity interruption as anomalous events that get interrupted because of occlusion or the anomalous object leaving the frame momentarily.

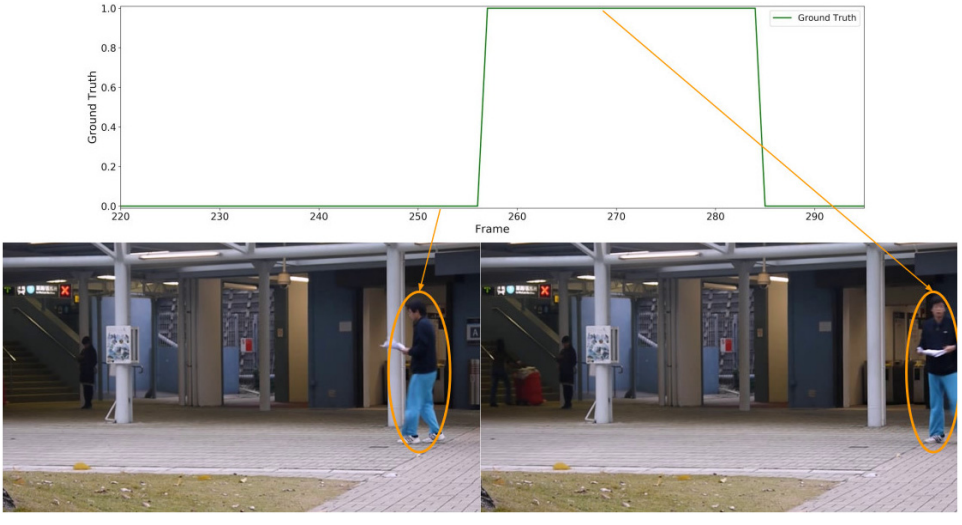

Fig. A.7. Labeling inconsistency (Video 13). The scene changes from normal to anomalous because of the action performed by the actor, which was facing the camera. However, this is an isolated instance in the dataset and seems to be a very weak reason for defining an anomaly.

Since we are dealing with anomaly detection in videos, some kind of temporality should be taken into account when defining anomalies. It is more natural to assume that, once an event started, it will have a certain continuity over time. That is, even if the agent that causes the anomaly briefly leaves the scene or gets occluded (for very

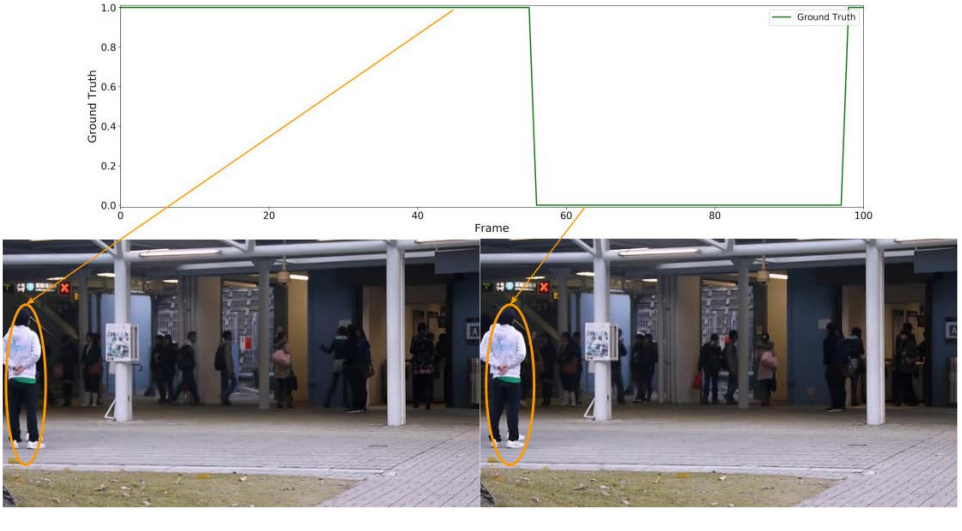

Fig. A.8. Labeling inconsistency (Video 17). Subject highlighted is considered anomalous in one frame and anomalous in other frames, while performing exactly the same action.

Fig. A.9. Continuity interruption (Video 1). The ground truth rapidly oscillates because of very brief occlusions caused by the pillars.

few frames), the whole window in which the event occur should be considered anomalous.

For instance, in Fig. A.9, the event in which a person appears running rapidly oscillates between normal and anomalous because of brief occlusion caused by the pillars. This kind of annotation only makes sense in a frame-by-frame analysis, which

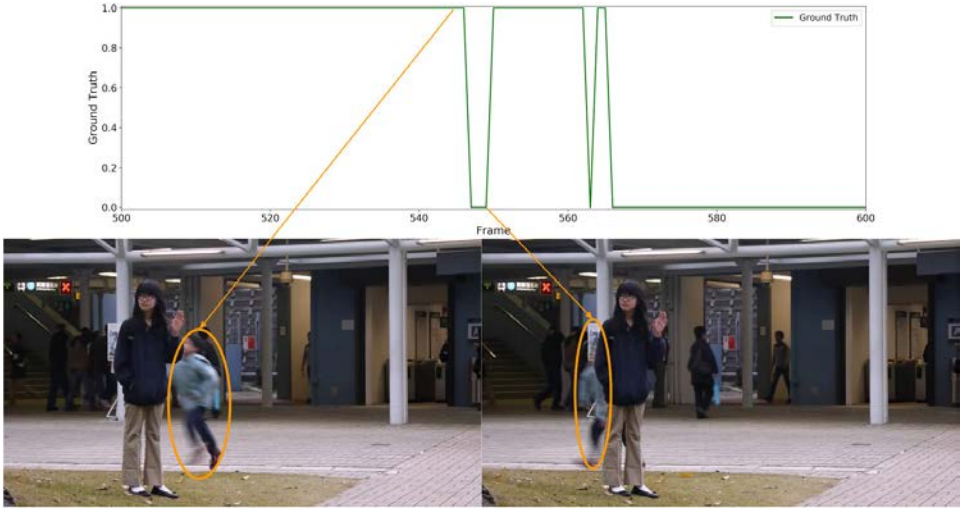

Fig. A.10. Continuity interruption (Video 9). The ground truth rapidly oscillates because of very brief occlusion of the boy caused by the girl.

Fig. A.11. Continuity interruption (Video 10). The ground truth rapidly oscillates when the backpack briefly leaves the scene.

defeats the purpose of having methods that take temporality into account. The same kind of problem occurs in Fig. A.10, where the boy gets briefly occluded by the girl.

Figures A.11–A.13 present the same scenario, where the backpack being thrown in the air defines the anomaly. For a very brief moment, the backpack leaves the frame, causing a continuity interruption in the ground truth. This happens several times throughout the videos.

Fig. A.12. Continuity interruption (Video 11). Same problem as in Fig. A.11.

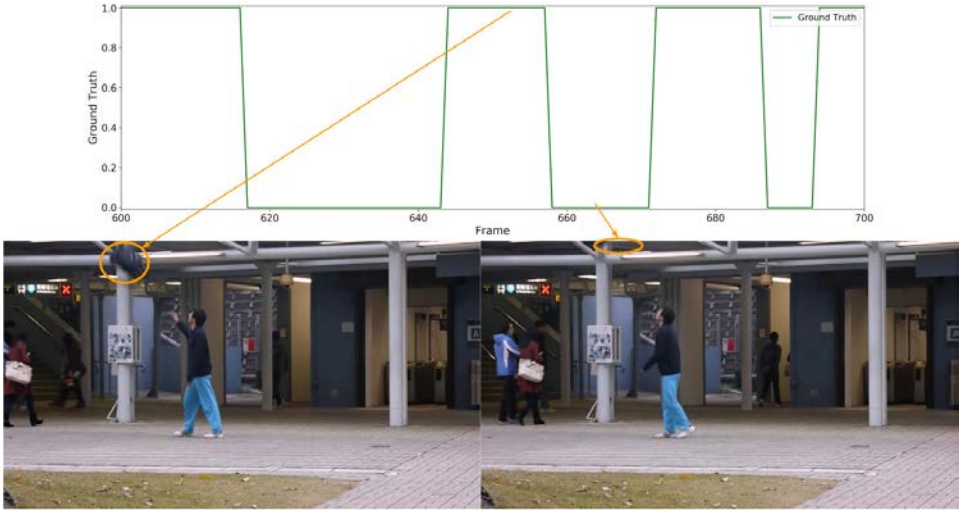

Fig. A.13. Continuity interruption (Video 12). Same problem as in Fig. A.11.

Figures A.14 and A.15 are very similar to the backpack scenario, except that this time the objects in question are sheets of paper. Again, the ground truth oscillates very rapidly as the sheets of paper leave the scene from above.

Static object mislabeling

Most of the problems presented in this section were already reported by Ref. 4. Nevertheless, we reinforce them.

Fig. A.14. Continuity interruption (Video 13). The ground truth rapidly oscillates when the sheets of paper briefly leave the scene.

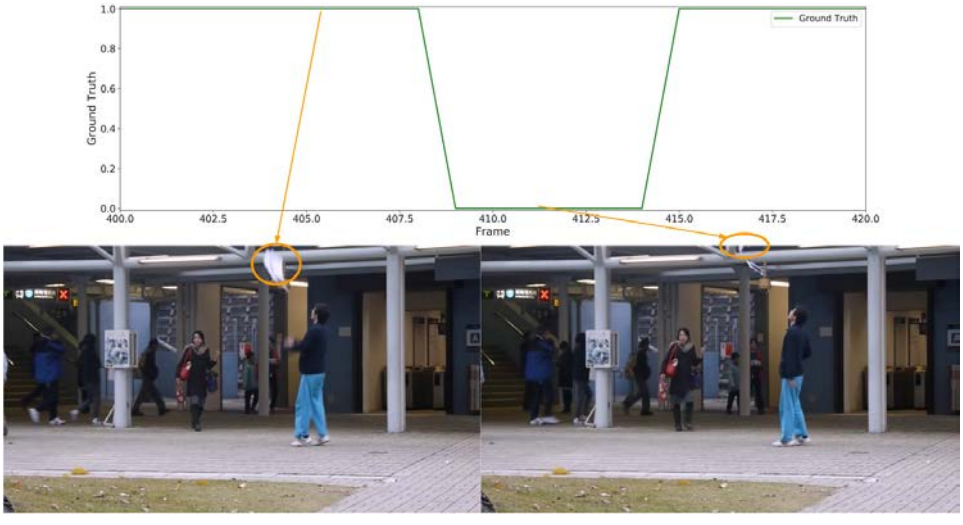

Fig. A.15. Continuity interruption (Video 14). Same problem as in Fig. A.14.

Static object mislabeling refer to objects that are considered normal but should be considered anomalous, since they were never seen in the training videos. Since one class classifiers rely on normal events to properly classify anomalous events, the static objects and agents shown in this section should be classified as anomalous.

Fig. A.16. Static object mislabeling (Video 1, 2). Despite objects similar to the bags never appearing in the training set, these objects were considered normal during the entire video duration.

Fig. A.17. Static object mislabeling (Video 8–10). Despite there not being any similar instance in the training set (person standing still in the foreground), the girl is considered normal for the entire duration of the video.

The bags shown in Fig. A.16 and the girl in the foreground shown in Fig. A.17 are novel in the test dataset, that is, there is no similar instance in the training set.

Figures A.18–A.20 show instances where a backpack on the ground was considered normal. In our view, this should also be considered anomalous, since there is no instance like this in the training set.

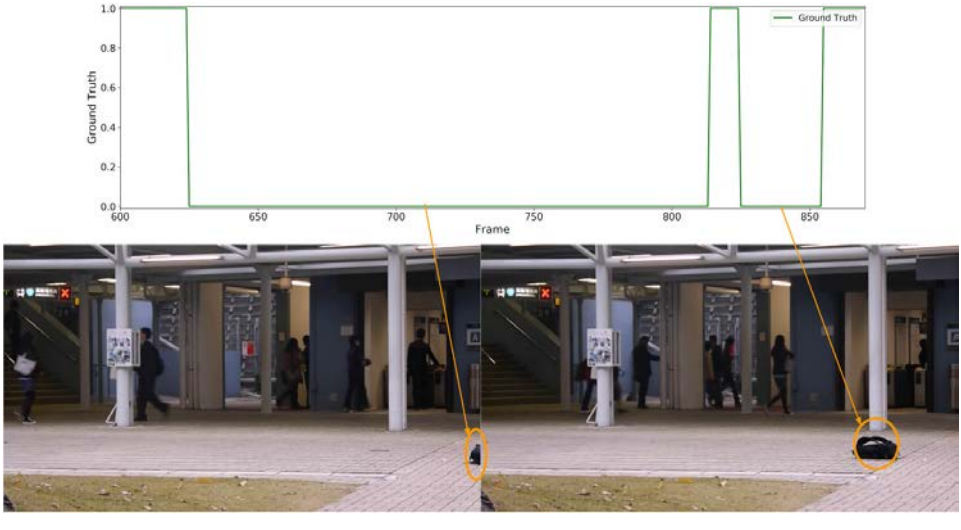

Fig. A.18. Static object mislabeling (Video 6). The backpack on the ground is considered normal, even though there is no instance similar to this in the training set.

Fig. A.19. Static object mislabeling (Video 11). Same problem as in Fig. A.18.

Fig. A.20. Static object mislabeling (Video 12). Same problem as in Fig. A.18.

References

1. K. Chatfield, K. Simonyan, A. Vedaldi and A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, arXiv:1405.3531.
2. M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *J. Am. Statistic. Assoc.* **32**(200) (1937) 675–701.
3. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, arXiv:1512.03385.

4. R. Hinami, T. Mei and S. Satoh, Joint detection and recounting of abnormal events by learning deep generic knowledge, arXiv:1709.09121.
5. G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, Densely connected convolutional networks, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Hawaii Convention Center, Honolulu, Hawaii, USA, 2017), pp. 4700–4708.
6. R. T. Ionescu, S. Smeureanu, M. Popescu and B. Alexe, Detecting abnormal events in video using narrowed motion clusters, arXiv:1801.05030.
7. D. K. Jain, Z. Zhang and K. Huang, Multi angle optimal pattern-based deep learning for automatic facial expression recognition, *Pattern Recognit. Lett.* **139** (2017) 157–165.
8. A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in *Proc. 25th Int. Conf. Neural Information Processing Systems*, Vol. 1 (Harrahs and Harveys, Lake Tahoe, Stateline, Nevada, USA, 2012), pp. 1097–1105.
9. T. Li, H. Chang, M. Wang, B. Ni, R. Hong and S. Yan, Crowded scene analysis: A survey, *IEEE Trans. Circuit. Syst. Video Technol.* **25**(3) (2015) 367–386.
10. J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue and G. Zhang, Transfer learning using computational intelligence: A survey, *Knowl-Based Syst.* **80**(1) (2015) 14–23.
11. C. Lu, J. Shi and J. Jia, Abnormal event detection at 150 fps in MATLAB, in *Proc. IEEE Int. Conf. Computer Vision* (IEEE Press, Piscataway, NJ, 2013), pp. 2720–2727.
12. V. Mahadevan, W.-X. Li, V. Bhalodia and N. Vasconcelos, Anomaly detection in crowded scenes, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE Press, Piscataway, NJ, 2010), pp. 1975–1981.
13. R. Mehran, A. Oyama and M. Shah, Abnormal crowd behavior detection using social force model, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 2 (IEEE Press, Piscataway, NJ, 2009), pp. 935–942.
14. S. J. Pan and Q. Yang, A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.* **22** (10) (2010) 1345–1359.
15. M. Ribeiro, A. E. Lazzaretti and H. S. Lopes, A study of deep convolutional auto-encoders for anomaly detection in videos, *Pattern Recognit. Lett.* **105** (2018) 13–22.
16. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei–Fei, Imagenet large scale visual recognition challenge, *Int. J. Comput. Vision* **115**(3) (2015) 211–252.
17. V. Saligrama and Z. Chen, Video anomaly detection based on local statistical aggregates, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE press, Piscataway, NJ, 2012), pp. 2112–2119.
18. B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond* (MIT Press, Cambridge, 2001).
19. L. Shao, F. Zhu and X. Li, Transfer learning for visual categorization: a survey, *IEEE Trans. Neural Netw. Learn. Syst.* **26**(5) (2015) 1019–1034.
20. M. Simon, E. Rodner and J. Denzler, Imagenet pre-trained models with batch normalization, arXiv:1612.01452.
21. S. Smeureanu, R. T. Ionescu, M. Popescu and B. Alexe, Deep appearance features for abnormal behavior detection in video, in *Proc. Int. Conf. Image Analysis and Processing* (Springer International Publishing, Cham, 2017) pp. 779–789.
22. W. Sultani, C. Chen and M. Shah, Real-world anomaly detection in surveillance videos, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Salt Lake City, Utah, USA, 2018), pp. 6479–6488.
23. Q. Sun, H. Liu and T. Harada, Online growing neural gas for anomaly detection in changing surveillance scenes, *Pattern Recognit.* **64**(Suppl. c) (2017) 187–201.

24. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Rethinking the inception architecture for computer vision, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (IEEE press, Piscataway, NJ, 2016), pp. 2818–2826.
25. D. M. J. Tax, One-class Classification, PhD thesis, Advanced School for Computing and Imaging, Technische Universiteit Delft (2001).
26. Y. Wang, Z. Luo and P.-M. Jodoin, Interactive deep learning method for segmenting moving objects, *Pattern Recogn. Lett.* **96** (2017) 66–75.
27. D. Xu, E. Ricci, Y. Yan, J. Song and N. Sebe, Learning deep representations of appearance and motion for anomalous event detection, arXiv:1510.01553.
28. D. Xu, R. Song, X. Wu, N. Li, W. Feng and H. Qian, Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts, *Neurocomputing* **143** (2014) 144–152.
29. J. Yosinski, J. Clune, Y. Bengio and H. Lipson, How transferable are features in deep neural networks?, in *Proc. 27th Int. Conf. Neural Information Processing Systems*, Vol. 2 (MIT Press, Cambridge, MA, USA, 2014), pp. 3320–3328.
30. T. Yu, L. Wang, H. Gu, S. Xiang and C. Pan, Deep generative video prediction, *Pattern Recognit. Lett.* **110** (2018) 58–65.
31. 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, MM '17* (Association for Computing Machinery, New York, NY, USA, 2017), pp. 1933–1941.