inventions

MDPI

Article

Deep Learning Based Surveillance System for Open Critical Areas

Francesco Turchini [†] , Lorenzo Seidenari [*,†], Tiberio Uricchio [†] and Alberto Del Bimbo

Department of Information Engineeering, University of Florence, Via di Santa Marta, 3, 50139 Firenze, Italy ; francesco.turchini@unifi.it (F.T.); tiberio.uricchio@unifi.it (T.U.); alberto.delbimbo@unifi.it (A.D.B.)

[*] Correspondence: lorenzo.seidenari@unifi.it

[†] These authors contributed equally to this work.

Received: 3 August 2018; Accepted: 4 October 2018; Published: 11 October 2018

Abstract: How to automatically monitor wide critical open areas is a challenge to be addressed. Recent computer vision algorithms can be exploited to avoid the deployment of a large amount of expensive sensors. In this work, we propose our object tracking system which, combined with our recently developed anomaly detection system. can provide intelligence and protection for critical areas. In this work. we report two case studies: an international pier and a city parking lot. We acquire sequences to evaluate the effectiveness of the approach in challenging conditions. We report quantitative results for object counting, detection, parking analysis, and anomaly detection. Moreover, we report state-of-the-art results for statistical anomaly detection on a public dataset.

Keywords: anomaly detection; surveillance systems; computer vision; object detection; object tracking

1. Introduction and Related Work

The monitoring of infrastructures is a compelling need both for buildings and open areas. Indoor surveillance, with a special use case of smart buildings [1], is facilitated by the possibility of deploying and easily fusing multiple sensors. Nonetheless, wide areas may be present in buildings too, as in the case of airports or malls. The use of wide area monitoring systems with network cameras has recently increased due to the recent request for more security [2]. Many areas need constant surveillance to ensure the enforcement of rules and laws, and the decreasing cost of sensors has supported the fast deployment of network cameras. When dealing with wide areas, the use of manned surveillance, even with the use of a camera network, is extremely challenging.

Currently deployed surveillance systems are primarily based on the performance of human operators who are expected to watch, often simultaneously, a large number of screens (up to 50 [3]) that show streams captured by different cameras. The task of surveillance operators becomes more difficult and fatiguing when the number of simultaneous video streams to watch increases. Indeed, it is well known that after twenty minutes of continuous work, the operator's attention degrades significantly [4]. Typical jobs done by operators include checking the presence or absence of objects (typically people and vehicles) in security areas, ensuring the maximum capacity of a place is respected, or checking anomalous events, such as objects put in unexpected places or people in no-go areas. Hence, with the aim of reducing the mental burden and increasing the attention span, any automatic method that is able to reliably alert the operator of the presence of objects or any anomalous situation is desirable.

In the literature, limited attention has been placed on assisting security in real-time. Systems are usually classified into three generations: (1) the first CCTV based camera that only recorded video on tape; (2) digital cameras with low-level image processing (like perimeter intrusion detection, abandoned object detection, etc.) with IP-based networks; (3) multi-view intelligent surveillance systems with semantic information extraction. Recent works [5] are just starting to develop vision algorithms with the aim of reaching the third generation of surveillance systems. Denman et al. [6] argue that operational analytics, in the form of crowd counting and modelling [7], queue monitoring, or anomaly detection has received little attention in research. Even the commercial systems that do exist are few and work only with specially placed cameras. They argue that a well integrated system should be able to concurrently perform analytics and assure security. Nonetheless, current approaches still suffer from comparatively poor performance on such tasks and are prone to false alarms and missed detections [6,8–10].

In this paper, we report on a novel, real-world system that is able to perform several security analyses and report several types of analytics automatically on a video stream in challenging real-world scenarios as shown in Figure 1. Our approach is thus related to the third generation systems. The system is able to detect typical objects of interest (people, cars, trucks, motorcycles) and track their movements. It is able to count how many objects are present in real-time and how many of them enter restricted areas by passing through a virtual gate. Moreover, the system is able to detect anomalous events by exploiting a polytope-based representation.

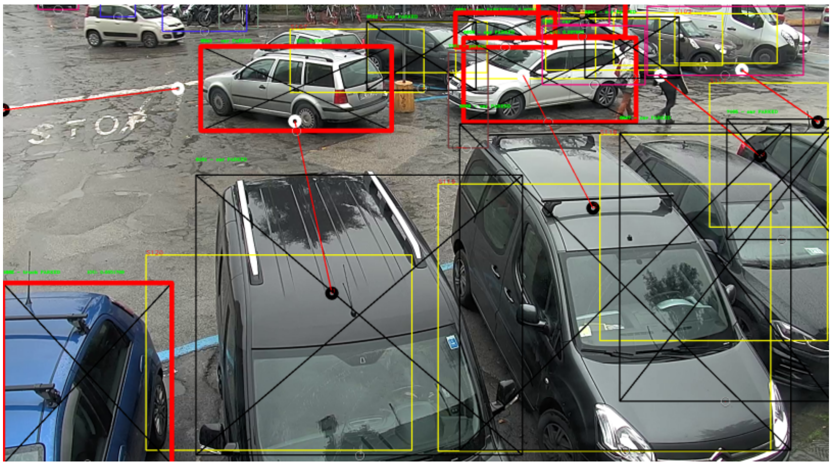

Figure 1. Examples of detections in the parking lot of Piazza Tasso. The yellow boxes denote annotated parking spots. The black crossed vehicles are in the parked state. The red boxes denote anomalies.

Several works have studied the problem of detecting generic objects and anomalous situations. Many approaches derived from Krizhevsky et al. [11] targeted object detection [12,13]. Girshick et al. proposed the extraction of CNN features from regions of an image and then classification of their content [14]. Nonetheless, this methodology requires full forward network inference for every region and is a major computational burden; it is not apt to real-time systems. This burden can be moderated by computing a single fully convolutional feature and classifying rescaled sub-windows over the feature map [12,13]. Finally, single pass methods have been developed [15,16] which generate a fixed set of scored windows for each class, given an input image, to solve a regression task. Currently, the YOLO detector is one of the best performing architectures with respect to competing methods especially in terms of the speed/accuracy trade-off [15]. A further discussion on single and dual stage object detectors can be found in [17].

Anomaly detection is essentially a one-class learning problem. Such a task can be solved either by parametric [18–21] or non-parametric approaches [22,23]. This last class of methods uses an implicit model of feature distribution by storing features and looking at reciprocal distances. Non-parametric models are often fast but memory consuming. The use of a fixed-size, parametric representation moderates this issue. Dynamic textures were used by Li et al. [18] to capture unseen motion patterns. A similar idea was used by Kim et al. [19] to learn a PCA mixture that reduced the dimensionality and to learn an effective distribution model. Representation learning is challenging since no labels are available; nonetheless, it may yield good accuracy [20]. Xu et al. [20], exploited autoencoders to model video sequences.

We present an integrated, flexible system which is able to tackle two primary surveillance problems: object counting and anomaly detection with localization. Both objectives are achieved, guaranteeing real-time performances with good experimental results, keeping the system very adaptable to various contexts. Specifically, we are able to perform several tasks, some of which have not been comprehensively tackled in previous works:

- Abnormal car behaviour detection (with specific attention to off-spot parking and wrong direction movement).
- Parking duration understanding.
- Generic anomaly detection and localization for non-car objects based on statistical model learning.
- Accurately estimating the number of objects (cars, people, etc.) in the area of interest.

For all tasks, we exploit the temporal coherence of video data, which is a distinctive approach with respect to existing surveillance methods. This allows our system to increase its efficiency and lighten the workload of surveillance operators.

The remainder of the paper is organized as follows: in Section 2, we describe the whole system, which is comprised of the various analysis modules; in Section 3, we discuss two case studies where we deployed the system and performed several quantitative experiments that are reported in Section 4. Finally, in Section 5, we report the conclusions of the study.

2. Surveillance System

The proposed system has 2 main functionalities that we treat separately: (i) the detection of objects of interest, including their tracking and counting, both spatially and temporally; and (ii) the detection of anomalous areas that contain unexpected objects or suspicious elements. A brief system scheme which highlights its main modules is reported in Figure 2.

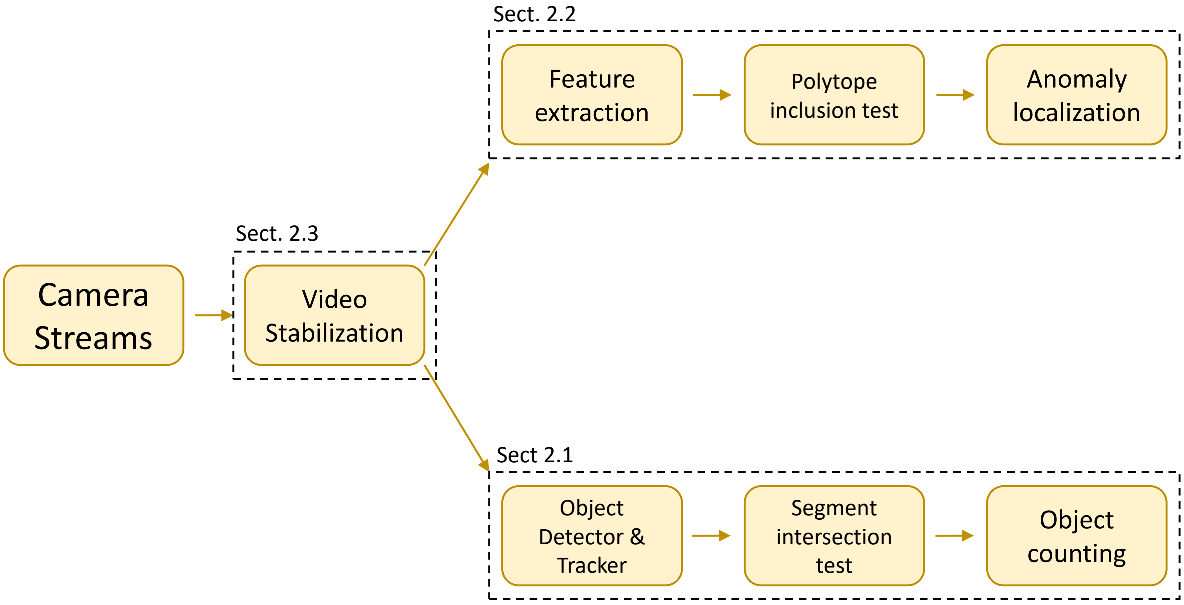

Figure 2. Scheme of our surveillance system. For each algorithm block, we report paper sections explicitly.

2.1. Object Detection, Tracking, and Counting

Regarding video streams, this functionality of the system aims to detect objects of interest and watch their movements. The system shall be able to always know how many objects are present in real-time and maintain a count of how many of them have passed through a security gate. We divided the system into three sub-modules. First, we discuss the detection of the objects; then, we describe how we track them and how we handle the security gates for counting purposes.

2.1.1. Detection

Our method exploits the YOLOv2 [24] object detection pipeline method which allows processing in real-time and can detect hundred of classes. Briefly, YOLO is a single-stage detector that processes an image in a single pass and generates a set of boxes with an associated probability for each class. Training is performed on the COCO dataset [25]. The method is based on a fully convolutional deep architecture where the last layer is designed in a way to predict a fixed number of windows using 1×1 filters to output the correct number of predictions, which is connected to the number of classes to be recognized.

We used a method trained on the 80 classes in COCO, but only a few were actually interesting for our application; the tracker updated its state for all detections, but decisions were only taken for objects that had the class of a vehicle or person.

2.1.2. Tracking

Each object is represented by its bounding box, namely its location. The tracker has a global state that monitors all objects in a single condensed state. At each frame, a set of new detections is produced to track objects. We need to update the tracker's state by associating each track to every detection, if possible, with unassociated detections starting new tracks. We do not use the object class to perform associations for two reasons. First, we want to keep track consistency, even in cases where the detector makes mistakes for a few frames. We can recover from this error by exploiting the estimated distribution of classes over the entire track. Second, there are few cases in which the detection of a different class spawns on top of another, and often, the geometry of the new detection is not consistent with the other one. This case, for example, may happen when a person exits a car and the detector is not detecting the car for some reason.

We use a greedy association multi-target tracker. At each frame, a set of detections $\mathcal{D}_t$ is generated by applying the object detector described in Section 2.1.1. Considering the possibly empty set of tracks $\mathcal{T}_{t-1}$ detected at the previous frame, we compute an association matrix $\mathbf{M}$ such that $\mathbf{M}_{ij} = \frac{d_i \cap t_j}{d_i \cup t_j}$, also known as the intersection over union measure. To track an object, we apply the function associate$(\cdot)$ described in Algorithm 1.

Algorithm 1 Data association algorithm. We associate tracks and unassociated detection if `IoU` $> \tau$ and remove a track if it is "dead" for ω frames. Matrix $\mathbf{A}$ keeps track of associations and vector $\mathbf{l}$ counts the number of frames in which a track i is not associated with any detection.

```
FUNCTION associate ($\mathcal{T}_{t-1}, \mathcal{D}_t$)
Data: $\mathcal{T}_{t-1} : \{t_1 \dots t_n\}, \mathcal{D}_t : \{d_1 \dots d_m\}, \mathbf{M}_{ij} = \frac{d_i \cap t_j}{d_i \cup t_j}$
Result: $\mathcal{T}_t$
while $\max_{ij} \mathbf{M}_{ij} > \tau$ do
    if not $\mathbf{A}_{ij} \wedge \mathbf{M}_{ij} > \tau$ then
        $\langle \hat{i}, \hat{j} \rangle \leftarrow \arg\max_{ij} \mathbf{M}_{ij}$;
        $t_{\hat{i}} \leftarrow d_{\hat{j}}$ $\mathbf{A}_{\hat{i}:} \leftarrow$ TRUE;
        $\mathbf{A}_{:\hat{j}} \leftarrow$ TRUE;
/* Unassigned detections initialize new tracks. */
$\mathcal{T}_t \leftarrow \mathcal{T}_{t-1} \cup \{d | \mathbf{A}_{ij} =$ TRUE$\}$;
/* Remove tracks not assigned for $\omega$ frames. */
$\mathcal{T}_t \leftarrow \mathcal{T}_{t-1} \setminus \{t_i | \mathbf{l}_i > \omega\}$
```

To cope with the issues raised by real-world scenarios, in which a classification algorithm should output a prediction after observing a set of possibly noisy detections, we introduce our approach for video sequences. We propose a smart strategy to aggregate predictions based on an estimated quality which we use as a weight. In the following, we introduce an image quality measure and the final decision model to incorporate multiple frames into a single decision for a track extracted using our tracking algorithm.

Consider a set of decisions $\mathcal{Y}_a : \{y_a(1) \dots y_a(T)\}$ for an attribute a performed on samples of a track $\mathcal{T}$. We compute a measure of quality for each sample considering $p(\mathcal{C}_k|b)$ which is the probability of class $\mathcal{C}_k$ for a box b. We further normalize weights, taking into account the track length $w'_t = w_t/|\mathcal{T}|$. Since we want to estimate categorical variables, we compute weighted histograms of attribute value counts. $b_i^a = \sum_{k=1}^{N} w_k$ is the bin that accounts for attribute a and has a value i; decisions on the track are taken as $\hat{y}_{\mathcal{T}} = \arg\max_i b_i$.

2.1.3. Counting

Object tracking yields a set of tracks representing the paths followed by objects on the scene. Each path is constituted by a sequence of points. Counting is performed by testing the segment intersection between a given line of passage (gate) and the sequences of points on each track.

We address gate traversal by objects as a segment intersection problem of the second degree, also known as the orientation test [26]. The robustness of such sweeping algorithms was analyzed in [27]. Given two segments $A = (s1, e1)$ and $B = (s2, e2)$, we can understand if they intersect by testing the orientations of the ordered triplets formed by the four points $(s1, e1, s2, e2)$. If $(s1, e1, s2)$ and $(s1, e1, e2)$ have different orientations and $(s2, e2, s1)$ and $(s2, e2, e1)$ have different orientations, then we have an intersection between A and B. The special case of collinear points is handled by fitting a line though the given points and verifying that the angle is around 0 or 180 degrees within a certain tolerance.

When we have an intersection, we also understand the direction in which the intersection occurs by looking at the orientations of the triplets. We define a positive direction for positive counting, and thus, a negative direction for negative counting.

2.2. Anomaly Detection

The second functionality is the ability to alert the operator of an anomalous presence in the video stream. We consider a fully supervised approach to tackle anomaly detection and localization as a one-class classification problem. We use the Polytope Ensemble technique [28] as the modeling method. This method represents an approximation of the space containing the input feature samples with a set of polytopes. Among the polytope convex class considered, the convex hull has the geometric structure that is best tailored to model this kind of data distribution.

Our method can be considered semi-supervised. We train the system only with samples extracted from video sequences depicting regular circumstances (negative examples). Interestingly, upon system deployment, a large amount of normal data is readily available via an initial acquisition. To keep the system up to date, polytopes can be estimated by acquiring new descriptors over time. This setting does not require human supervision. Learned models cannot be directly transferred from one scene to another, but the system training is the only cost of the novel acquisition of descriptors.

2.2.1. Constructing the Model

Suppose we have a set of points $\boldsymbol{X} = \{x_1, \dots, x_m\}$ for training. The convex hull of $\boldsymbol{X}$ is defined as

$$\boldsymbol{C}(\boldsymbol{X}) = \left\{ \sum_{i=1}^{|X|} \gamma_i x_i | x_i \in \boldsymbol{X};\ \sum_i \gamma_i = 1, \gamma_i \geq 0 \forall i \right\}. \tag{1}$$

Given a new sample, we can classify it as abnormal if it does not belong to the convex hull.

2.2.2. Robust Convex Hull

Directly using the computed convex hull may overfit the model to the training set and decrease its performance in practical conditions. As a result, we increase the robustness of the model by modifying the structure of the convex hull. We shift the vertices of the convex hull farther or closer to the centroid, similarly to [28]. In detail, given the centroid of the polytope c_i and its set of vertices $\boldsymbol{V} \subset \boldsymbol{X}$, we compute an expanded polytope by correcting the vertices such that

$$\boldsymbol{V_\alpha} = \{v + \alpha \frac{(v - c_i)}{||v - c_i||}, v \in \boldsymbol{V}\}. \tag{2}$$

Hence, the new corrected polytope is enlarged or shrunken from the original polytope by an α factor. When α is negative, the model is more sensitive. The opposite applies when α is positive.

2.2.3. Ensembles of Polytopes

We consider an ensemble of size T with convex hulls to further increase the robustness of the algorithm. Using the popular Improved Dense Trajectories algorithm [29], we extract both motion and appearance descriptors. We employ the trajectory coordinates HoG, HoF, and MBH. We denote the size of the destination subspace as d and the size of the feature f as D_f for each convex hull, and for each feature, we generate a random projection matrix R_i^f with a norm of 1 and a size of $d \times D_f$. The original data is then projected as

$$\boldsymbol{X}_{\boldsymbol{R}_i^f} = \{R_i^f x, \forall x \in \boldsymbol{X}\}. \tag{3}$$

Each convex hull of the ensemble will have a unique shape, computed on each $\boldsymbol{X}_{\boldsymbol{R}_i^f}$. At every iteration of the algorithm, we then derive a different random projection matrix. Thanks to the expanded polytope described in the previous section, by varying α, we can obtain a set of ensembles of different levels of sensitivity. Anyway, this step is optional, since by scaling the distance between the vertices and the centroid, we can directly compute the shrinking factors (Equation (2)). The main advantage of this approach is that by using a fixed size ensemble of polytopes, the method can be considered parametric, since for a dataset of arbitrary size, it only stores the estimated low-dimensional polytopes. This reduces the memory footprint of the approach compared to non-parametric methods which require all the data to be stored.

2.2.4. Anomaly Localization

After training the model, we proceed to localizing anomalies in new images. Given a test image, we first extract the features and test each one for inclusion in the ensemble of convex hulls. We detect anomalies in individual trajectory descriptors by considering their coordinates. Hence, the system can propose several areas of the video frames as being anomalous. A local trajectory is marked anomalous if every descriptor x_f is outside all the model polytopes:

$$x_f \notin C^f(\boldsymbol{X}_{\boldsymbol{P}_i^f}) \; \forall f, i. \tag{4}$$

This is applied to all of the considered features (trajectories, HoG, HoF, MBH); hence, the system requires a strong indication that the trajectory is anomalous. An anomalous detection remains as such for the entire extent of the trajectory descriptor (the default is 15 frames).

Taking every anomalous trajectories together as a proposal, we then filter spurious detections by clustering them together. Let us denote the set of trajectories $\boldsymbol{T_a} = \{t_1, t_2, \ldots, t_N\}$ that were marked as anomalous. Each trajectory t_i is a sequence of M points, $t_i = \{p_{i1}, \ldots, p_{iM}\}$ lasting M video frames. At each frame f, the set of points of the active anomalous trajectories is

$$\boldsymbol{P_a} = \{p_{in} \in t_i | n = f, \; t_i \in \boldsymbol{T_a}\}. \tag{5}$$

Using the K-means algorithm on such points, we localize the potentially abnormal areas of the frame. The result of K-means is a partition $\boldsymbol{S_a}$ of K Voronoi cells that contains the anomalous points set $\boldsymbol{P_a}$:

$$\boldsymbol{S_a} = \{S_1, \ldots, S_K | S_1 \cup \cdots \cup \, S_K = P_a, \, S_{k_1} \cap \, S_{k_2} = \varnothing \, \forall \, k_1, k_2\}. \tag{6}$$

Assuming that spurious false positive detections may generate small clusters, we discard any anomaly proposal S_k that has a smaller cardinality than a fixed threshold in order to guarantee a reliable detection. The remaining partitions constitute the final anomaly mask for the frame, which is defined by the computed polygon described by its points.

2.3. Camera Stabilization

Camera setups in outdoor environments may require stabilization. Slight side-wise motion can be caused by the wind or vibrations from other sources. To make the grounding of image plane detection with real-world coordinates consistent, we stabilize images iteratively using homographies. We assume that cameras may oscillate and rotate around a main setup with small displacements. Given two frames I_t and I_{t+1}, first, we run a high efficiency corner detector [30] on I_t, keeping the first 500 keypoints of the image. Then, on the subsequent frame I_{t+1}, we compute, for each detected keypoint, a displacement using the Pyramidal Lucas & Kanade algorithm [31]. We use RANSAC [32] to fit an homography between I_t and I_{t+1} by computing $\mathbf{H}_{t+1,t}$. We apply a bilinear interpolation warp to I_{t+1}, thus rectifying the image and removing camera motion. Border artifacts may arise, but, in practice, we have found that the object detector is robust.

3. Case Studies

We tested our system in two different scenarios, both representing real use cases. The first one is parking lot surveillance, while the second one is the monitoring of a pier for element counting purposes.

3.1. Pier

This case deals with high risk sensitive areas in which a large number of people, also using vehicles, may access sterile areas. In our case, the main goal is to verify that vessel manifests are correctly communicated by companies and that the number of entities disembarking matches that of the manifest. A camera is mounted at a height of 20 m facing the pier where ferries dock. The elevated position of the camera reduces occlusions when both cars/trucks and people are disembarking. We define a counting zone and count the elements that go through it. This application exploits the novel 5G connectivity standard which we are testing on the site. The cameras send h264 HD frames over fiber to a 5G modem which is connected to a scalable remote server farm which is upgraded when multiple streams are fed to the server.

3.2. Parking Lot

This problem relates to enforcing law and monitoring safety in an unrestricted parking lot. Municipalities need to enforce parking fares in all parking lots. In certain cases, it is unfeasible to restrict access with physical barriers to areas. In these situations, plate reading cameras are placed in order to monitor entrances. Our system is integrated with the plate reading cameras and tracks vehicles, recognizing parking events and also mapping the read plate onto the corresponding parking spot.

This setup comprises two cameras mounted on the opposite sides of a parking lot in Florence. As camera height is restricted to 4–5 m, we need two points of view to monitor the entire area without incurring occlusion problems. Our objective is to verify the correct parking of cars in the slots and to detect wrong accesses to the area (i.e., cars entering in the opposite way). We raise alarms when vehicles are parked in an illegal area, when vehicles proceed in the wrong direction, and when a person loiters for an extended amount of time, in case a generic anomaly (statistical model of Section 2.2) is detected. We use a server co-located with cameras using two NVIDIA GTX 1080 (one per stream). The system communicates with a cloud based platform using a 3G connection through lean JSON messages. We also upload snapshots from cameras when anomalies are raised.

4. Results

4.1. Object Counting

We tested our counting approach on an in-house dataset acquired at the pier of Bari. The dataset consists of 28,416 frames acquired during the disembark of a large cruise ship. For safety and privacy reasons we cannot release the dataset's imagery.

We performed two kinds of annotations on this dataset: temporal annotation and spatial annotation. For temporal annotation, we fixed a virtual gate and manually annotated the number of people, cars, trucks, and motorbikes passing through it. At each frame, for each class, we annotated the total of elements that crossed the gate in that frame. This was done for the whole extent of the videos. Regarding the spatial counting task, we annotated the positions of people, cars, trucks. and motorbikes which werein the scene. This was done on 284 frames taken at intervals of 100 frames.

As we can see in Table 1, our system has good precision in detecting people; however, it has low recall, mainly due to dark areas where people are not detected and to the occlusions that happen when people are in a crowd. However, in the areas where detection is feasible, we correctly locate the majority of people on the scene. For cars, we obtain both good precision and recall as cars exit in a queue; thus, they are easier to detect and occlusions rarely occur. For trucks, our system shows the opposite behaviour with respect to the person class. In fact, the detection precision for tracks is only around 60%, while the recall is over 90%. This can be explained by the detection errors that occur for complex vehicles, for example, trucks with trailers which are detected as two separate vehicles. For motorbikes, the low recall results are ascribed to the low detection performances of YOLO for this specific class of objects.

Table 1. Spatial counting.

Class	Precision	Recall	F1
Person	0.81	0.56	0.66
Car	0.76	0.88	0.81
Truck	0.62	0.91	0.74
Moto	0.76	0.29	0.42

In Table 2, we report the temporal counting results. We can see that, especially in Sequence 2 and Sequence 3, which are very complex, we achieve very good results with a low error rate for total counting for the person, car, and truck classes. For Sequence 1, we observe an underestimation of the number of people present in the scene. This is due to the exclusive presence of people who are free to move and form a crowd. This behaviour result in occlusions which prevent correct detection. This does not happen in Sequence 2 and Sequence 3, where people move in a more organized way, as there is a contemporary presence of people and vehicles. For the motorbike class, we face the same problems as for spatial counting. The detector is not effective with this class and often localizes the person on the motorbike rather than the motorbike itself. This helps for the people counting task, but on the other hand, it clearly degrades performance in the motorbike counting task.

Table 2. Temporal counting on Sequences 1, 2, and 3.

Sequence	Class	Count	GT
1	Person	29	36
	Car	-	-
	Truck	-	-
	Moto	0	6
2	Person	79	83
	Car	22	21
	Truck	6	5
	Moto	-	-
3	Person	24	20
	Car	1	1
	Truck	10	10
	Moto	1	6

4.2. Parking Lot Analysis

As another benchmark for our tracking system, we tested the accuracy of parked vehicle detection on the public parking described in Section 3.2. We annotated a total of 36 spots of which 10 were viewed in both streams.

We processed more than 8 h of video recorded in a challenging situation with a high degree of occlusions. Parking spots were annotated with the following procedure. First a set of bounding boxes was manually annotated. Then, we ran our system and associated all parked car detection boxes with spot regions. Finally, we set the final spot bounding boxes as the weighted mean of all associated boxes. We used this procedure to produce a more robust annotation. We measured the accuracy of our system in estimating the occupancy of parking spots correctly. We report three metrics. First, we describe how accurately our system can flag spots as occupied or free. Secondly, we report the precision and recall for the detection of abnormal behavior such as out of spot parking or driving in the wrong direction. As shown in Table 3, recall was close to 80%, meaning that a large number of all infringements can be identified and reported. Regarding the parking spot occupancy analysis, it has to be noted that the few errors arose in cases involving occlusions or when odd sized vehicles were parked.

Table 3. Performance of our system in a challenging parking lot scenario.

Spot Accuracy	Abnormal Parking Recall	Abnormal Parking Precision
93.88%	79.27%	60.75%

Regarding the reduction in terms of the operator's workload, note that we automatically processed 7 h 40 m of video, detecting a total of 241 anomalies. Manual checking of such 5 s snippets requires a total of 20 min for a single operator which is much less than continuously observing the whole stream of almost 8 h. This means that the operator's attention is sporadically summoned and must be kept for a very short time frame.

4.3. Anomaly Detection

We performed experiments on the UCSD Pedestrian dataset [18]. This dataset is made of two sets of videos, namely Ped1 and Ped2, that depict real pedestrian areas. All the non-pedestrian entities appearing in the scene are considered anomalous patterns. We followed the experimental protocol for this dataset which considers frame-level and pixel-level criteria [18].

For the frame-level evaluations, we considered detections without spatial information: a frame is considered anomalous if at least an abnormal detection occurs for that frame, regardless of its location.

In this case, fortuitous guesses are likely to happen, so we can predict a frame correctly with a spatially incorrect detection.

Pixel-level evaluation is a more accurate analysis of algorithm behavior. In this setting, ground truth pixel masks are available for comparison with the algorithm's output. A detection is considered a true positive if there is at least 40% pixel overlap between the ground truth and the predicted mask. On the contrary, a detection is considered a false positive in cases where anomalies are predicted in non-anomalous frames or if the overlap with ground truth masks is lower than 40%. We show the Receiver Operating Characteristic (ROC) curves of TPR and FPR and the Rate of Detection (RD) with a varying level of system sensitivity. The system sensitivity is modified by changing the α value in Equation (2).

Figures 3 and 4 show that our method limits false positive detections, especially at low sensitivity levels, at the frame level. We detect and localize less than 20% false positives, while we spot more than 50% true positives at lower sensitivity values on Ped1. The behaviour is better on Ped2, where we correctly detect and localize more than 50% true positive anomalies, incurring less than 5% false positives. The false positive rate increases if we raise the system sensivity to unseen patterns, but still maintains good robustness. We report the Rate of Detections for all considered methods for both datasets and both criteria (if available) in Table 4. Our system obtains a frame-level result which is comparable to the state-of-the-art and outperforms all existing methods for pixel-level evaluation, exploiting a very simple operating mechanism which does not include object detectors or neural networks, offering great flexibility with respect to tuning possibilities and feature fusion capabilities. Frame-level anomaly detection is an easier task since this metric does not penalize lucky guesses; indeed, a method may locate an anomaly at the frame level by wrongly labeling a certain set of normal pixels. Pixel level anomaly detection is a more complex and certainly a more useful task. As an example, it allows high-resolution details of abnormal regions to be acquired either via a master-slave camera system or directly via digital PTZ. Note that system initialization for a new camera setup comes at a very low cost. Considering the example of the UCSD Ped1 and Ped2 datasets, it can be seen that the system can be trained over 5–10 min of normal data, and it is able to recognize and localize abnormal patterns.

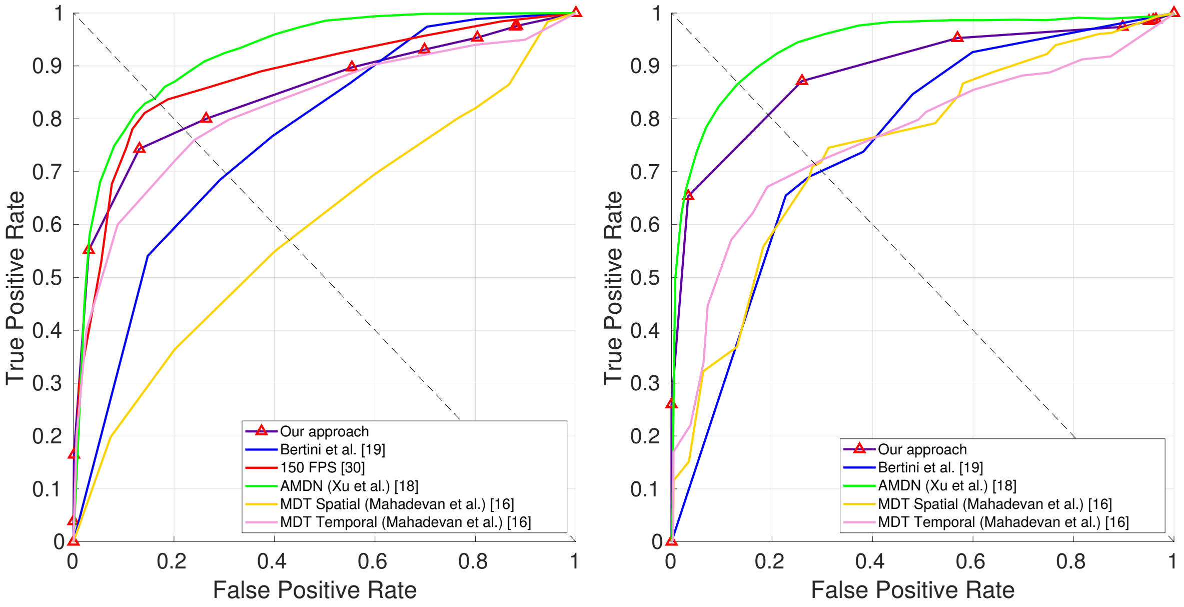

Figure 3. Frame-level TPR–FPR curves comparing our approach with various well-known methods in the Ped1 (**left**) and Ped2 (**right**) settings.

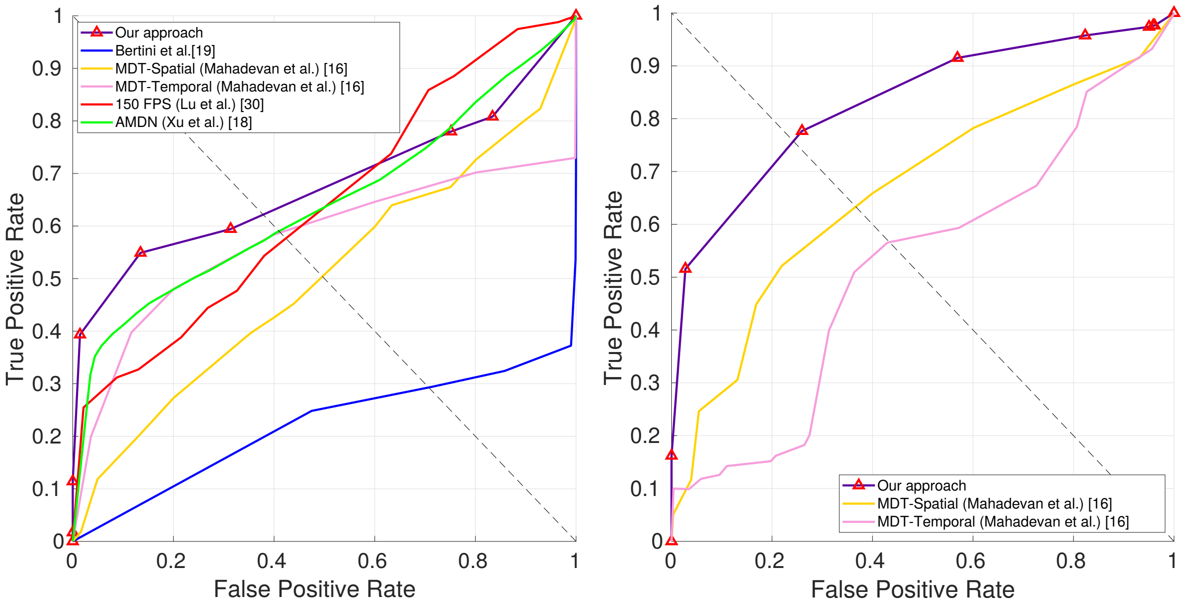

Figure 4. Pixel-level TPR-FPR curves comparing our approach with various well-known methods in the Ped1 (**left**) and Ped2 (**right**) settings.

Table 4. Rate of Detection (RD) comparison of our method versus various well-known state-of-the-art techniques on the Ped1 and Ped2 (where available) settings with frame-level and pixel-level criteria.

Method	**Ped1**		**Ped2**	
	Frame	**Pixel**	**Frame**	**Pixel**
Ours	78.1	**62.2**	80.7	**75.7**
Xu et al. [20]	78.0	59.9	**83.0**	-
MDT Spatial [18]	56.2	54.2	71.3	63.4
MDT Temporal [18]	77.1	48.2	72.1	56.8
150 fps [33]	**85.0**	59.1	-	-
Bertini et al. [22]	66.0	29.0	68.0	-
Mehran et al. [34]	63.5	40.9	65.0	27.6
Kim et al. [19]	64.4	23.2	64.2	22.4
Adam et al. [35]	61.1	32.6	54.2	22.4

5. Conclusions

In this paper, we described a third generation surveillance system based on deep learning that can be applied to complex surveillance scenarios. Our system is able to detect a wide range of events of interest such as vehicles moving in the wrong direction, an excessive amount of people entering a certain area, and so on. Regarding all abnormalities that can not be defined with location and direction based patterns, we rely on a statistical based anomaly detector. To the best of our knowledge, such a comprehensive integrated solution for wide area surveillance has not been presented previously. Existing methods usually either focus on statistical anomaly detection or on parking occupancy assessment. Our proposed system is able to keep track of all moving objects in a parking lot to detect mispositioned vehicles, wrong direction accesses, and spot occupancy with high reliability even in complex scenarios. Regarding generic abnormal behaviors, we presented a method reporting state-of-the-art results in the challenging anomaly localization task.

The deployed systems are connected with remote interfaces that need to be updated in real-time; therefore, all algorithms were developed to abide this constraint. A few limitations may be highlighted. First, is the greedy nature of our tracker data association strategy. While, in practice, we have seen this to work very well, some situations with long-term target occlusions leading to ID switching may be solved. A main drawback of non-greedy trackers is the computational complexity which, in turn, may require extensive resources to be used in real-time scenarios. Second, at the moment, our method supports multiple cameras, but decisions are taken separately on each one. On one hand, this setting allows us to install new cameras without complex multi-camera calibration; on the other hand, a multi-camera tracker may improve system accuracy in cases where occlusions are present.

Author Contributions: Conceptualization, L.S., T.U. and A.D.B.; Software, F.T., L.S. and T.U.; Writing—Original Draft, F.T., L.S., T.U. and A.D.B. Funding acquisition, A.D.B.

Funding: This research was partially supported by Project TESEO, funded by Regione Toscana POR-FESR 2014–2020, Grant No. 3389.30072014.068000017. This research was partially supported by Leonardo S.p.a.

Acknowledgments: We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Manic, M.; Amarasinghe, K.; Rodriguez-Andina, J.J.; Rieger, C. Intelligent buildings of the future: Cyberaware, deep learning powered, and human interacting. *IEEE Ind. Electron. Mag.* **2016**, *10*, 32–49. [CrossRef]
2. Martinel, N.; Micheloni, C. Re-identify people in wide area camera network. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, USA, 16–21 June 2012; pp. 31–36.
3. Troscianko, T.; Holmes, A.; Stillman, J.; Mirmehdi, M.; Wright, D.; Wilson, A. What happens next? The predictability of natural behaviour viewed through CCTV cameras. *Perception* **2004**, *33*, 87–101. [CrossRef] [PubMed]
4. Haering, N.; Venetianer, P.L.; Lipton, A. The evolution of video surveillance: An overview. *Mach. Vis. Appl.* **2008**, *19*, 279–290. [CrossRef]
5. Valera, M.; Velastin, S.A. Intelligent distributed surveillance systems: A review. *IEE Proc. Vis. Image Signal Process.* **2005**, *152*, 192–204. [CrossRef]
6. Denman, S.; Kleinschmidt, T.; Ryan, D.; Barnes, P.; Sridharan, S.; Fookes, C. Automatic surveillance in transportation hubs: No longer just about catching the bad guy. *Expert Syst. Appl.* **2015**, *42*, 9449–9467. [CrossRef]
7. Sabeur, Z.; Doulamis, N.; Middleton, L.; Arbab-Zavar, B.; Correndo, G.; Amditis, A. Multi-modal computer vision for the detection of multi-scale crowd physical motions and behavior in confined spaces. In *International Symposium on Visual Computing*; Springer: Berlin, Germany, 2015; pp. 162–173.
8. Hashemzadeh, M.; Farajzadeh, N. Combining keypoint-based and segment-based features for counting people in crowded scenes. *Inform. Sci.* **2016**, *345*, 199–216. [CrossRef]
9. Yang, B.; Cao, J.; Wang, N.; Zhang, Y.; Zou, L. Counting challenging crowds robustly using a multi-column multi-task convolutional neural network. *Signal Process. Image Commun.* **2018**, *64*, 118–129. [CrossRef]
10. Sultani, W.; Chen, C.; Shah, M. Real-world Anomaly Detection in Surveillance Videos. *arXiv* **2018**, arXiv:1801.04264.
11. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Stateline, NV, USA, 3–8 December 2012.
12. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Washington, DC, USA, 7–13 December 2015.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015.
14. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014.

15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Las Vegas Valley, NV, USA, 26 June–1 July 2016.
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016.
17. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
18. Li, W.; Mahadevan, V.; Vasconcelos, N. Anomaly detection and localization in crowded scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 18–32. [PubMed]
19. Kim, J.; Grauman, K. Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Miami Beach, FL, USA, 20–25 June 2009; pp. 2921–2928.
20. Xu, D.; Ricci, E.; Yan, Y.; Song, J.; Sebe, N. Learning deep representations of appearance and motion for anomalous event detection. *Comput. Vis. Image Understand.* **2015**, *156*, 117–127. [CrossRef]
21. Alom, M.Z.; Bontupalli, V.; Taha, T.M. Intrusion detection using deep belief networks. In Proceedings of the Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 15–19 June 2015; pp. 339–344.
22. Bertini, M.; Del Bimbo, A.; Seidenari, L. Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Comput. Vis. Image Understand.* **2012**, *116*, 320–329. [CrossRef]
23. Breitenstein, M.D.; Grabner, H.; Van Gool, L. Hunting nessie-real-time abnormality detection from webcams. In Proceedings of the International Conference on Computer Vision Workshops (ICCVW), 27 September–4 October 2009; pp. 1243–1250.
24. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. *arXiv* **2016**, arXiv:1612.08242.
25. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
26. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*; MIT Press: Cambridge, MA, USA, 2009.
27. Boissonnat, J.; Preparata, F. Robust Plane Sweep for Intersecting Segments. *SIAM J. Comput.* **2000**, *29*, 1401–1421. [CrossRef]
28. Casale, P.; Pujol, O.; Radeva, P. Approximate polytope ensemble for one-class classification. *Pattern Recognit.* **2014**, *47*, 854–864. [CrossRef]
29. Wang, H.; Oneata, D.; Verbeek, J.; Schmid, C. A robust and efficient video representation for action recognition. *Int. J. Comput. Vis.* **2016**, *119*, 219–238. [CrossRef]
30. Mair, E.; Hager, G.D.; Burschka, D.; Suppa, M.; Hirzinger, G. Adaptive and generic corner detection based on the accelerated segment test. In Proceedings of the European Conference on Computer Vision (ECCV), Heraklion, Crete, Greece, 5–11 September 2010; pp. 183–196.
31. Shi, J.; Tomasi, C. Good features to track. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
32. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
33. Lu, C.; Shi, J.; Jia, J. Abnormal event detection at 150 fps in matlab. In Proceedings of the International Conference on Computer Vision (ICCV), Darling Harbour, Sydney, 1–8 December 2013; pp. 2720–2727.
34. Mehran, R.; Oyama, A.; Shah, M. Abnormal crowd behavior detection using social force model. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 935–942.
35. Adam, A.; Rivlin, E.; Shimshoni, I.; Reinitz, D. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 555–560. [CrossRef] [PubMed]

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).