UTRECHT UNIVERSITY
Master Artificial Intelligence

James Godlonton 6329446

COMBINING CLASSIFIERS FOR VIDEO BASED SURVEILLANCE ANOMALY DETECTION

Thesis, 13-11-2020

First Supervisor
Albert Salah
Prof.
Second Supervisor
Wolfgang Hürst
Dr.
External Supervisor
Jianquan Liu
Dr.

Abstract

Currently ubiquitous closed circuit television surveillance systems are greatly hampered by the inability of humans to analyse the large amount of footage being produced. In order to alleviate this there has been a great deal of research into the automatic processing of surveillance video in order to detect events that may require short term emergency responses, such events are most often referred to as anomalies. In this work we aim to improve the performance of automated surveillance video anomaly detection by introducing two methods for splitting up the detection task into subsets of anomaly types for which we are then able to train independent detectors. First we propose to automatically group scenes based on which objects appear in them, allowing us to use a single detector per scene type. Second we split the problem on anomaly semantics, using independent detectors for fire and smoke detection, and traffic anomaly detection, whose outputs are then combined with the output from a generic anomaly detector to make a final detection prediction. Both of these methods allow models to be trained to detect a smaller range of anomalies which is both a simpler task and allows for a more fine grained interpretation of the reasoning behind detections. We also analyse of how much the current methods are able to distinguish an anomaly from its surrounding normal footage, as present evaluation methods fail to measure this important factor. In order to evaluate our models we run experiments using instantiations of three state of the art surveillance video anomaly detectors and a large scale crime based dataset. This is complemented with an analysis of the large scale crime based dataset used in order to highlight its shortcomings in trying to represent real world emergency detection. The results show that fire and smoke detection is the easiest task to separate out and detect independently from generic surveillance video anomaly detection, that the current state of the art methods mostly use scenic priors to detect anomalies, and there are many improvements that need to be made for the state of the art dataset.

List of abbreviations and terms

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
ATM	Automated Teller Machine
AUROC	Area Under The Receiver Operator Curve
BN-Inception	Batch Normalized Inception
C3D	Convolutional 3D Network
CCTV	Closed-Circuit Television
CNNs	Convolutional Neural Networks
CPU	Central Processing Unit
FPR	False Positive Rate
FPS	Frames Per Second
GCN	Graph Convolutional Network
IoU	Intersection Over Union
MIL	Multi Instance Learning
MLP	Multi Layer Perceptron
R-CNN	Regions With Normalized Inception
RAM	Random Access Memory
RGB	Red, Green, Blue
ROC	Receiver Opereator Curve
ROI	Region Of Interest
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TPR	True Positive Rate
TSN	Temporal Segment Network

Table of Contents

1. Introduction

The automated monitoring of closed-circuit television (CCTV) surveillance video footage has the potential to both lower emergency response times and decrease the costs involved with monitoring. This is because automated monitors can be run on standard computational hardware, promising the classical benefits of automation, that by removing the human element we are able to more consistently, quickly and cheaply perform a task. The performance of these monitoring models, in particular models aimed at achieving generalized anomaly detection in surveillance video footage, is however not yet good enough to be used in real world applications. The issue of what is good enough is not yet well defined, inappropriate evaluation measures combined with the current state of the art models' lack of interpretability leave researchers in the dark as to when they have achieved performance significant enough for real world application.

Research in the field of creating automated monitors is primarily concerned with creating computer vision based models that can identify alarm worthy events in video footage automatically. The detection of alarm worthy events is termed *anomaly detection* in literature because most early methods in the field implemented classic unsupervised anomaly detection techniques from other areas such as network analysis to overcome a lack of alarm worthy samples during training. Modern approaches have however strayed away from traditional anomaly detection techniques and better resemble an imbalanced two class classification problem, using both positive and negative samples during training. These supervised methods however still refer to the problem as anomaly detection, straying away from a traditional statistical definition of anomaly detection such as *determining test samples that do not belong to the same distribution as the normal training samples* [1] and towards a less formal definition of anomalies as *infrequent events worthy of alarm*. We have followed this trend in our work and use anomaly detection to stand for *the detection of events that involve crime or cause harm, and that require fast short term response*. We select three state of the art supervised methods from the last three years to compare, analyse and improve upon, on a state of the art crime based dataset which contains both negative and positive samples of emergencies, examples of which can be seen in Figure 1.

In order to improve the current state of the art detectors we propose to move away from using only a single generic detector and instead train multiple detectors each for a specific

(a) Road accident (b) Fighting (c) Fire

Figure 1. Examples of anomalies taken from videos of the UCF-Crime dataset by Sultani et al. [2]. Each anomaly video is annotated with a category and a labeling indicating on which frames the anomaly occurs.

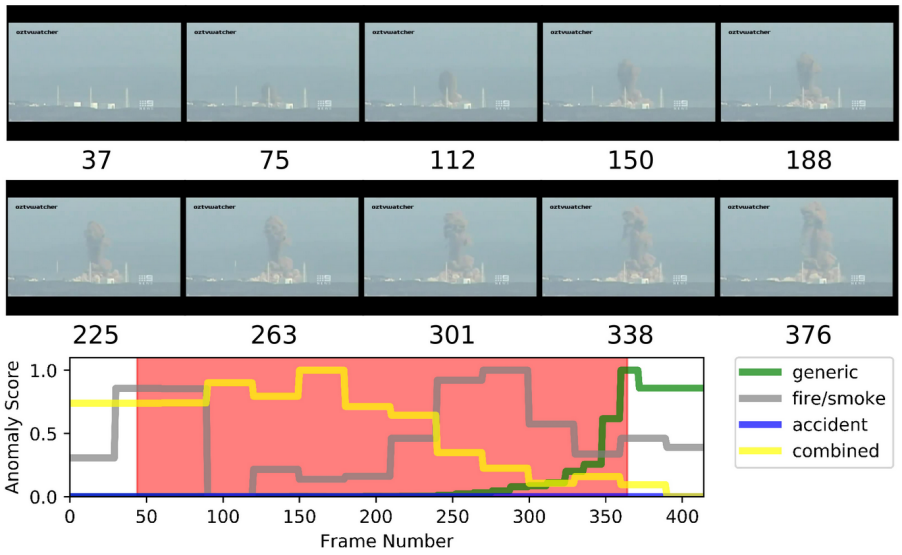


Figure 2. Timeline of a video containing a factory explosion. Generic, fire/smoke and accident scores are given by independent detectors and are combined to form the combined detection. The fire/smoke detector has enabled the combined detector to outperform the generic detector. The red segment indicates the ground truth anomaly annotation for the video provided by Sultani et al. [2].

sub problem. The outputs of these more specific detectors can then be combined with the generic detector in order to give a single overall anomaly prediction. We can see an example of this from our research in Figure 2 where we see the benefits of training a detector solely for the detection of fire and smoke.

1.1 Motivation

The primary goal of automated anomaly detection in surveillance footage is to keep people safe efficiently and effectively. Whilst anomaly detection is used to describe the field, it is important, to avoid a creep away from real world relevance, to remember that this is a useful abstraction of alarm worthy events. Alarm worthy events are those where emergency

responses and criminal investigations being aware of a situation in a timely manner is important for effectively mitigating damages, this is often the case as highlighted by the work of Sivarajasing et al. [3] where they determine the effect of increased surveillance monitoring on a reduction in the level of injuries sustained from violent altercations.

Not only is the human labour cost of human surveillance monitoring prohibitive but it is not able to reach the scale required by the recent proliferation of cameras which has not seen a corresponding increase in camera monitors. There has also been a rapid rise in household and vehicle based cameras as illustrated by Mohan et al. [4] for which paying a person to monitor is too expensive. This has greatly diminished the effectiveness of most surveillance systems as monitoring plays a vital role in the efficacy and impact of CCTV surveillance footage as corroborated by La et al.'s work in crime control and prevention [5].

As with many processes inhibited by labour costs automation provides an enticing solution. We propose to use currently available two dimensional (2D), red, green blue (RGB) surveillance footage coupled with computer vision and CCTV surveillance anomaly detection techniques to automate the process of detecting alarm worthy events. We use 2D RGB surveillance footage as a medium because it is low cost and already in place. This means that if our automation proves successful it can be seamlessly integrated into the current surveillance environment as a purely software model. This will allow for a very fast widespread adoption and a relevant impact on society. Computer vision techniques such as action classification and object recognition are chosen as preprocessing steps in our pipelines as these fields have acheived considerable successes in real world applications and provide an effective means of distilling information from the frames of a video.

There is also the potential for automation to increase our understanding of the precise nature of an event and provide evidence therefor, which is crucial in effective long term remediation. We however focus on short term response detection due to the large gap it currently has as well as the higher potential for algorithms to succeed in this area then in longer term scene understanding. The large gap in timely detection is due to the lack of live human monitors and the reason current artificial intelligence (AI) algorithms have a higher potential to succeed in timely detection is that it is a far narrower problem then the longer term scene understanding. Scene understanding is a highly ambiguous process with difficult to determine context making it a highly complex and difficult problem for current AI techniques. Anomaly detection does not however require the full understanding of a scene to achieve promising results as seen in the current state of the art works [2, 6, 7]. This is because we can distill a scene into a much smaller set of features that represent its anomaly well enough to classify a scene correctly whilst ignoring large amounts of

information that might be relevant to understanding other aspects of a scene.

1.2 Research Questions

The methods and experiments in this thesis are intended to further progress towards the overarching goal of developing a real world implementation of an automated video surveillance model to be useful in the short term mitigation and response of alarm worthy events. We do this by investigating both where the previous methods are falling short and by proposing new improved methods. In order to do this we hypothesise three areas where the current state of the art can improve and then direct one research question towards each of these hypotheses. The first hypothesis is that the currently used state of the art UCF-Crime surveillance video anomaly detection dataset [2] is not an accurate representation of the real world and is unable to effectively compare methods' abilities to detect surveillance video anomalies for real world applications. This leads us to our first research question to be answered in the literature review.

RQ1: In what way should the state of the art UCF-Crime surveillance video anomaly detection dataset be improved to better evaulate models' ability to detect video based anomalies in the real world?

The second hypothesis is that the range of alarm worth anomaly events in surveillance video footage is too diverse and cannot be distilled by a single model into the fundamental definition of what it means to be an anomaly. In particular the visual relationship between explosions, fighting and car accidents may not be strong enough for detection by a single model and therefore perhaps a collection of models individually trained for each category would be able to provide an improved performance. This gives us our second research question to be answered by experimentation on combining multiple detectors.

RQ2: What is the best way in which to successfully decompose the problem of anomaly detection in CCTV surveillance into smaller sub problems?

The third hypothesis is that the current state of the art methods are unable to correctly distinguish an anomaly from it's surrounding normal footage in the same scene. A review of the performance of the sate of the art methods has shown that anomaly predictions change most drastically between scenes rather than within one scene. This is concerning as it limits a models ability to alarm on events in a scene. In order to measure the extent of the problem we propose to answer research question three by re running the state of the art methods without anomalies present.

RQ3: To what extent do the current state of the art methods rely on scenic priors rather than anomaly semantics for obtaining their performance?

1.3 Structure of Thesis

This thesis begins with a review of recent literature in order to place the experiments and methods in their scientific context. This is followed by an in depth description of the methods used in order to enable analysis and reproduction. Finally results are reported on various approaches taken to improve performance in CCTV surveillance anomaly detection.

We start the literature review with an analysis of various trends seen in CCTV surveillance anomaly detection and how they have impacted the direction of our work, including the development of different datasets and increasing performance on them. This is followed by a description of some shortfalls in the current evaluation measures as well as the UCF-Crime dataset. The literature review is rounded off with a brief description of the fields relating to our used feature extraction techniques, that is: action classification, object detection, fire and smoke detection, and traffic accident detection. This completes the context for the entire pipeline of our methods.

In the methods section we begin with a description of the methods we used to decompose the problem of anomaly detection, by using clusters of the training data and by targeting specific anomaly types. This is followed by a description of the techniques used to remove anomalies from the test set in order to evaluate the extent to which to state of the art methods use anomaly semantics for detection. Then we describe the implementation details of the three state of the art methods tested as well as for the feature extraction techniques used across the three methods. We also describe the additional algorithms employed in order to create models focused on various sub categories of anomalous behaviour, that is, detection of fire and smoke using the Resnet50 deep neural network implementation of Olafenwa and Abimbola [8] and detection of road accidents using the unsupervised motion based method of Li et al. [9].

Finally we describe the results of using clusters to retrain multiple models with a more specific focus where we see the efficacy of having distinct models for videos with vehicles and videos without vehicles as well as the shortcomings of performing the model split based on human presence. This is then followed by results on combining the output of sub detectors trained for detecting specific anomaly indicators with a generic anomaly detector. We report results on using both fire and smoke detection as well as traffic anomaly detection, illustrating the improvement in high confidence predictions. We then analyse

how much attention each of the three state of the art methods is paying to the actual anomalous segment in a video in order to validate that they are not merely using scenic priors to achieve high average prediction scores, we show that the multi-instance learning (MIL) work of Sultani et al. [2] and the graph convolutional network (GCN) work of Zhong et al. [6] rely heavily on scenic priors. Lastly we end the thesis with some conclusions and directions for future work.

2. Literature Review

2.1 Trends in CCTV Anomaly Detection

The topic of anomaly detection in RGB CCTV footage can be grouped across four main dimensions which have progressed as the field has matured. Firstly by whether or not they make use of anomalies during training. The field begun by using methods that train with only normal data which we term semi-supervised methods and progressed towards using normal and anomaly data during training, which we term fully supervised methods. The limiting factor in this was largely the availability of data, there was not enough data to use anomalies during training until the release of the large scale UCF-Crime dataset in 2018 by Sultani et al. [2] which contains 1900 different scenes and has prompted a flurry of fully supervised methods.

Secondly by the style and computational complexity of classification model used. Models began statistical in nature and slowly progressed towards more black box deep neural networks due to an increased performance found therein and the increased availability of the computational power previously limiting neural network models. Recent work has again shown a resurgence in statistical methods as increased interpretability is again at the forefront of the machine learning consciousness.

Thirdly anomaly detection in CCTV progressed by the feature extraction models used. This begun by using pixel values from a single frame as features and has progressed into more and more complex feature extractors such as object detectors, action classifiers, or even previous anomaly detection methods as seen in the state of the art work of Lv et al. [7]. This change can be linked to the adoption of neural network models across the field of machine learning in image and video applications as neural networks allow the extraction of a meaningful representation from an intermediate layer in their network allowing for effective and simple transfer learning.

Finally there has been an increase in focus on localizing anomalies. This change comes in both evaluation and model design. In evaluated this change can be thought of as an increased importance placed on distinguishing anomalies from their surrounding normal footage in the same video, reducing false positives in order to increase real world applica-

Dataset	Year	Number of scenes	Number of frames	Percentage of frames anomaly	Best AUROC
UMN [16]	2006	3	7710	14.00%	0.996 [17]
UCSD Ped1 [18]	2010	1	14000	28.61%	0.97 [19]
UCSD Ped2 [20]	2013	2	4560	35.88%	0.99 [21]
CUHK Avenue [11]	2013	1	30652	12.46%	0.90 [17]
ShanghaiTech [22]	2018	13	317398	5.38%	0.85 [17]
Street Scene [23]	2020	1	203257	21.65%	0.61 [23]
IITB-Corridor [15]	2020	1	483566	22.39	0.67 [15]
UCF-Crime [2]	2018	1900	13327113	5.62%	0.85 [7]

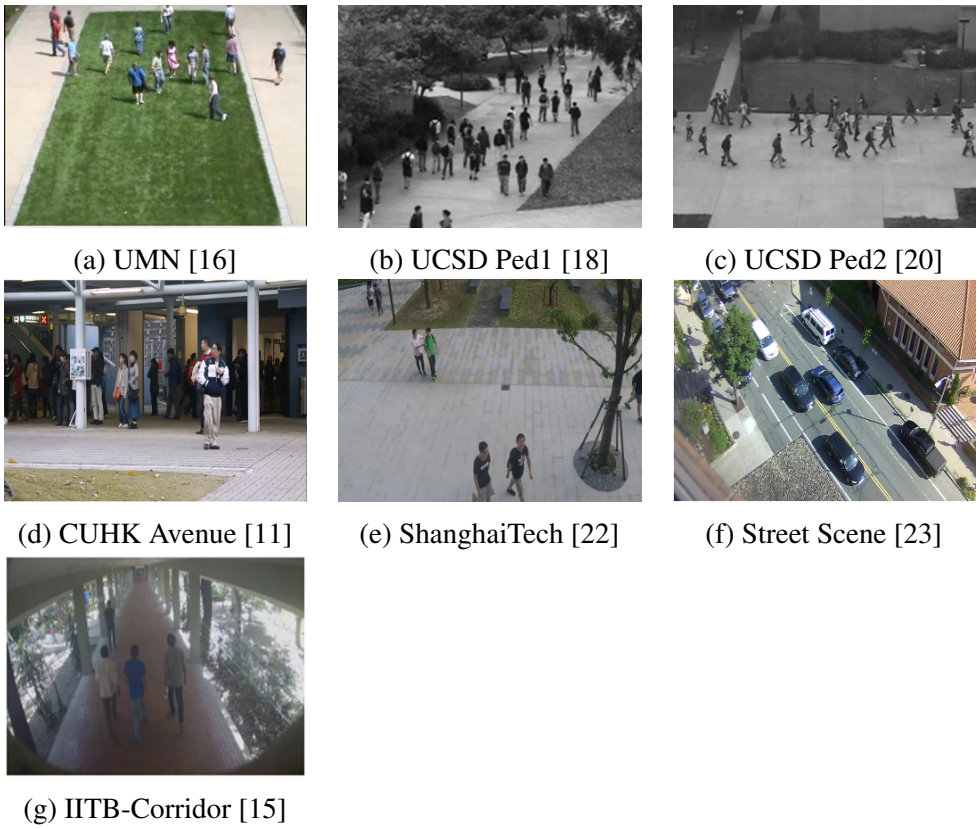
Table 1. Normal training datasets above with UCF-Crime, the only dataset using anomalies during training below. Adapted and updated from the work of Ramachandra et al. [24].

bility. In training and design, models have achieved improved results by narrowing the annotation of training videos, both in the temporal dimension by annotated the anomaly itself and not just the presence of an anomaly in a video and also spatially by giving bounding boxes for anomalies.

2.1.1 Semi-supervised vs Supervised Training Methods

Initially works in the field focused on modeling the distribution of normal data without any anomalous data entries during training [10, 11, 12, 13, 14, 15]. This choice was largely made due to the lack of tagged anomaly datasets. The small number that were available were reserved for testing. These methods are known as semi-supervised machine learning methods and work by modelling the distribution of normal events in some manner and then classifying samples as anomalous when they appear to not be from this distribution. This leaves us with *out of distribution* as our approximation of what it is to be an anomaly.

The main difficulty with the out of distribution approximation is that what is out of distribution depends heavily on the context of a scene. For example cycling is normal on a cycle path but not on a crowded walkway. This is perhaps why these methods have largely been used on very narrow datasets where the train and test sets came from the same or a very limited number of similar scenes. This eases the changing context issue by keeping scenic context consistent throughout training and testing. These methods are able to reach reasonable accuracy, as seen in Table 1, by specializing for a small number of scenes. We aim to take advantage of this insight by using multiple models to reduce the context diversity required by a single model on a diverse dataset.



(a) UMN [16] (b) UCSD Ped1 [18] (c) UCSD Ped2 [20]

(d) CUHK Avenue [11] (e) ShanghaiTech [22] (f) Street Scene [23]

(g) IITB-Corridor [15]

Figure 3. Images showing the lack of diversity in anomaly detection datasets that use only normal data during training. This highlights the problem that in the semi-supervised CCTV surveillance anomaly detection domain performance across many datasets may not imply good generalizability of a model as the datasets are all very similar. The datasets are similar because collection of an anomaly dataset often happens on a university campus whereby walking paths are the easiest subject matter.

It is also important to highlight here a weakness in the testing of these approaches. Because they are all tested by an application to a single or few scene datasets and these datasets are all fairly similar, the results of these methods show us little about their ability for general CCTV surveillance anomaly detection. For example, almost all the scenes tested contain streets or paths, people walking and good lighting. An illustration of this similarity can be seen in Figure 3. What this means is that even though methods developed may perform well over multiple datasets they are in actuality only tested on a very narrow range of scenes. This means that the generalizability, a very important element for real world applicability, of these models is unproven. In our research we aim to address this by testing our model on the UCF-Crime dataset [2], a very diverse dataset of crime related videos with varying contexts and definitions for anomaly.

More recent methods aim to use examples of anomaly and normal data in training in order to determine the boundary that separates the two, these are referred to as fully-supervised

methods. The primary dataset for validating these is the UCF-Crime dataset which during training is annotated at the presence of anomaly in a video level, leaving the anomaly segment of the video to the model to determine. The models used can take many forms such as multi-instance learning by Sultani et al. [2] or noisy label cleaning by Zhong et al. [6] in order to address the issue of not having temporal anomaly labels. The state of the art methods all take the approach of training a single anomaly detection model to use across all scenes.

It is important to note that training single and multi scene models has been split along the lines of semi-supervised and fully-supervised methods unnecessarily. For example, there is no method applying semi-supervised learning to the UCF-Crime dataset and no fully-supervised method applied to the single scene test tests. This is largely due to the lack of anomalies for a single scene. It is important however to recognize that finding anomalous examples within the same scene is not the only way to provide fully supervised training to a model tested on a single scene. We for example will use samples of anomalies found in similar scenes to specialize our model by training for specific clusters of anomalies and this could potentially be done for every scene.

2.1.2 Towards Complexity and Back Again

The field of anomaly detection in surveillance footage has closely followed trends in machine learning as a whole. In particular it has of late seen in increasing skepticism towards overly complicated deep learning pipelines for their lack of explainability and clarity on what elements of the pipeline effect the performance as highlighted in the work of Lipton et al. [25] published in 2019 where they emphasize the importance of identifying sources of empricial gains. Initial works on CCTV surveillance anomaly detection started with fairly small datasets and by using statistical methods for detecting anomalies. This can be seen in two highly influential papers, both Mehran et al. [10] (2009) and Lu et al. [11] (2013) look to make statistical calculations that depend on the underlying distribution of normal data and then utilize the negative contributions of new samples to these calculations as a sign of an anomaly.

These methods are highly efficient with the latter reaching a processing throughput of 150 frames per second (FPS). They also provide a decent detection performance of 0.96 on UMN and 91.8 on PED1 respectively. The performance of these methods is however only tested on these very simple and uniform datasets. Intuition tells us that that the simplicity of these models would struggle to capture the normal distribution as the complexity and nuance of both normal and anomalous samples increases.

Based on this intuition and spurred on by a boom in the field researchers began to use deep learning methods to detect anomalies. Methods such as autoencoders [12, 26, 17] and generative adversarial networks [22] were used for detection using only normal data. More generic dual class methods such as MIL and convolutional neural networks (CNNs) [6, 2, 7] were used for datasets with anomalies present in training. This focus proved successful with many of those developed still being considered state of the art today [17, 7]. These methods often include extensive pipelines that continue to increase in complexity. This is followed by a corresponding increase in running time and opaqueness of classification rationale which has traditionally been accepted on the promise of increased detection performance.

Recently however we see a backlash towards increasing complexity and black box models for being unclear as to where their increased performance comes from with hyperparameter tuning and pre processing often being the real needle movers as shown in the work of Melis et al. [27]. Championing the resurgence in prioritizing analyzability, interpretability and computational efficiency is a recent work by Doshi et al. [28] based off normal only data. They use an optical flow calculation and an object detector to extract a set of relevant features such as the mean and variance of flow or the likelihood that an object is of a certain class. A nominal set of normal samples is then used to represent the normal distribution. The distance of a sample from the nominal set is then calculated using euclidean distance. The distance is then thresholded against a running decision statistic for classification. This method is not only fast but is also able to give fine grained explanations by showing us what dimension in the feature representation contributed most to the high euclidean distance required to be an anomaly. They were able to achieve comparable results to the current state of the art with a 0.72 frame level area under the receiver operator curve (AUROC) on the ShanghaiTech dataset whilst providing both a fast and explainable solution that requires less training data then deep neural network models.

Another way in which we can reduce complexity is in reducing the scope of our dataset. To this end Vilamala et al. [29] and Martinez et al. [30] use subsets of the UCF-Crime dataset [2] representing violence and shoplifting respectively in order to focus in on these domains. Martinez et al. go even further then narrowing the scope to a subset of the anomalous videos by narrowing the shoplifting detection to a subset of each anomaly. To this end they focus only on detecting the behaviour leading up to shoplifting. The intention here is to proactively stop the anomaly from occurring whilst allowing the model to capture a less diverse normal distribution.

2.1.3 Preprocessing

The state of the art methods all use some form of pre-processing. This varies in complexity and often aims to take advantage of transfer learning from other domains such as object detection and action classification.

Definition 1: *(Transfer Learning)* To improve the performance of a learning task on a target domain by utilizing the knowledge learned from a different source domain or learning task [31].

Transfer learning from other computer vision tasks has proven to be a useful way of reducing training time and increasing performance a seen in previous anomaly detection works [2, 6, 7, 17]. We therefore describe some of the pre processing techniques the current state of the models use here and then investigate various potential domains for pre processors in later sections. The current state of the art single and dual class methods both use models pre trained for another task for pre processing video segments [17, 6].

In the work of Ionescu et al. they use an object detector following the methods of Lin et al. [17]. This produces a bounding box around objects with which they crop the image and then directly input the image contents from the box into an auto encoder network. They are using transfer learning here to narrow the attention of their method up front leaving only likely anomaly areas to be computed. The second preprocessing they do is to compute gradients between the objects in the current frame from the same objects in the $t - 3$ and $t + 3$ frames. Object tracking is easily done via overlapping bounding boxes as the frames are close together. Anomalous behaviours always contain motion so this again is a useful way of focusing the attention of their model onto relevant features in a scene.

In Zong et al's [6] work on graph convolutional networks we see an example of utilizing transfer learning not just as a pre processing step but including it in the training process, refining the models that were trained for a different purpose in order to be more directly applicable to the current task. In each experiment they utilize one of two well known activity recognition approaches C3D by Tran et al. [32] and I3D by Wang et al. [33] trained on the Sports-1M [34] and Kinetics-400 [35] datasets respectively as the initial input to their pipeline. This provides a synthesized representation of the spatio-temporal aspects of segments within a scene. The resulting anomaly classifications on each segment is further fed back to the initial model and is used to update the pre-trained weights.

The retraining of the underlying action classifier is particularly unique and whilst this is able to reduce the overall size of network required by utilizing all memory during training

it has some potential drawbacks. Firstly neural networks have a tendency to forget as illustrated by Kirkpatrick et al. [36] and so what useful feature extraction was in each model may be lost. Secondly it increases the opaqueness of a classification decision by reducing the compartmentalization of the pipeline leaving no understandable intermediate stages for investigation. The latest work of Lv et al. [7] take it one step further by using the trained model of Zhong et al. [6] as their pre processing step, allowing them to train their model to specifically only work on the localization of anomalies, improving performance even further.

2.1.4 Going local

Recently there have been a number of attempts at localizing anomalies in surveillance footage [37, 13, 38, 39, 7]. The central idea is that anomalies do not take place in the entire video, sometimes taking up very little space within a scene. This adds complexity to the problem as approaches have to understand anomalies at various different scales with varying amount of background information. The focus on localization is growing from both a model design perspective and an evaluation perspective. The works of Chong et al. [13], Lv et al. [7] and Liu et al. [38] attempt to use further localization in their models to make performance gains, whilst the works of Landi et al. [39] and Gianchandan et al. [37] focus on measuring localization during evaluation, emphasizing the importance of correct localization for real world predictability.

The simplest methods utilizing localization for performance gain can be seen in the work of Chong et al. [13] and Lv et al. [7] In the work of Chong et al. they split the space and time related convolutions of an autoencoder into different stages. This forces the model to encode/decode these elements separately and therefore allows the final model to learn a normal representation for these aspects separately. Lv et al. achieves the current best AUROC of 0.85 on the UCF-Crime dataset [2] by focusing on the difference between segment representations and insuring that the max of these differences is high in anomaly videos whilst the average is low in order better localize anomalies, this is also the method we implement in this work.

Liu et al. [38] further this by forcing models to consider the anomaly region within a frame. They do this by providing bounding boxes for the entire UCF-Crime dataset and then designing a training approach that forces a model to use weights from the anomaly area in order to output an anomalous score. This is done by using an attention map output of the model to form a predicted region of an anomaly and using the IOU between the predicted region and the ground truth bounding box as a factor within the loss function. With this approach they were able to achieve an AUROC of 0.82 on the UCF-Crime dataset. An

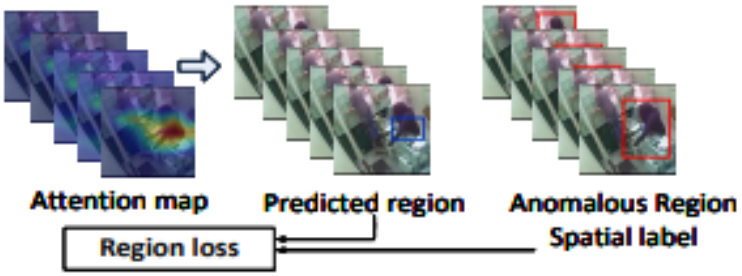


Figure 4. Region loss as implemented by Liu et al., image taken from their work [38]. The attention map from the neural network activation is used to define a predicted anomaly region which is intersected with the annotated anomaly region in order to calculate a region loss factor.

illustration of the region loss can be seen in Figure 4.

The approaches that adopt localization during evaluation treat anomalies as tubes in sequences of frames, the models then predict anomaly tubes, with the intersection over union (IoU) between prediction and annotated tube be used for evaluation. For this formulation Landi et al. [39] develop the UCF Crime2Local dataset, a bounding box annotated subset of UCF-Crime consisting of 100 anomaly and 200 normal videos. They perform their feature extraction and convolutional based classification on spatio-temporal tubes from videos. These tubes are essentially cropped squares of different sizes taken over a number of frames. Using these annotations they were able to acheive 0.75 AUROC on their new dataset. This was followed up by Gianchandan et al. [37] who re-implemented the weakly supervised multi instance learning work from Sultani et al. [2] by changing the instances in each bag from segments of a video to spatio-temporal tubes. On the UCF Crime2Local dataset they were able to achieve 0.68 AUROC. Although this is less then the 0.75 state of the art it is only utilizing weak labels at a video level compared with Landi et al's work where they make full use of bounding boxes during training.

2.2 Evaluation measures

There are various evaluation methods used in literature well summarized in an article by Ramachandra et al. [24] where they focus on the area of single scene anomaly detection. The central metric for evaluation is the area under the receiver operator curve for frame level detection. This is defined as the area under the curve representing the $True\ positive\ rate\ vs\ False\ positive\ rate$ where $True\ positive\ rate = \frac{\#TruePositive}{\#Positive}$ and $False\ positive\ rate = \frac{\#FalsePositive}{\#Negative}$. This method has been adopted across the

literature and can be thought of as calculating the $TPR\ vs\ FPR$ at all thresholds of what is classified as an anomaly. See Figure 5 for an illustration.

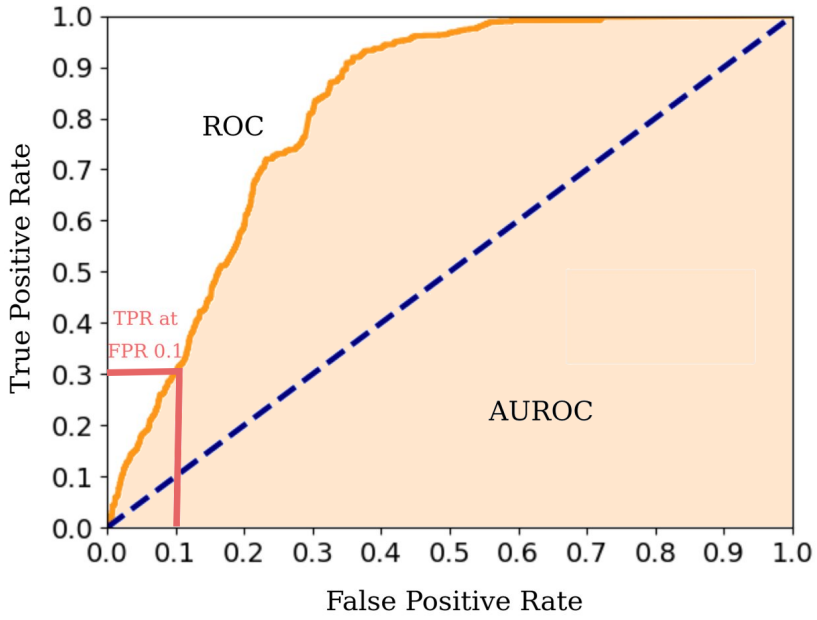


Figure 5. The receiver operator curve (ROC) plots the true positive rate (TPR) against the false positive rate (FPR). The light orange area under the receiver operator curve (AUROC) for frame level anomaly classification is the metric used to evaluate performance in CCTV surveillance anomaly detection literature. The TPR at FPR 0.1 in red is our proposed improvement to this metric as real world applications have to operate at a low FPR.

The key variation in the literature comes in when we look at what we consider a true positive or a false positive. In the most basic case researchers consider frame level detection. That is when an algorithm gives an anomaly score to an entire frame and frames are annotated as either anomalous or not. This method is relatively simplistic and allows for an easy annotation task and is therefore preferred in the biggest anomaly detection datasets such as UCF-Crime [2]. The frame based approach does however suffer from the weakness that it ignores spatial localization, making it unclear as to what is being detected as an anomaly. This is particularly concerning for busy scenes where many activities may be going on.

Mahadevan et al. [18] address the spatial element by introducing pixel level detection. This does however require that the dataset be pixel level (bounding boxes) annotated. This is not only more costly but suffers from annotation ambiguity as to where the spatial boundary of an anomaly is. To address this ambiguity they still consider AUROC at a frame level, judging a frame to be a true positive if the detected anomaly pixels overlap with at least 40% of the annotated anomaly and a false positive if at least one pixel is

determined anomalous in a frame with no anomalies.

Whilst adding a temporal element to the detection it doesn't guarantee a spatial element as a simple post processing step of extending a single anomalous pixel detection to the entire frame will directly increase true positives. To address this Sabokrou et al. [40] introduce the constraint that of the detected pixels at least 10% of them should overlap with an actual anomalous region. This still however doesn't correctly evaluate frames with multiple anomalies.

To address the issue of multiple anomalies per frame Ramachandran et al. [23] consider anomalies at a per anomaly level rather then a per frame level . They do this by using IoU for spatial refinement and anomaly tracking to consider a single anomaly over time. For this a connected region of anomalous pixels over a given threshold is considered a single detected region. If the region has an IoU large enough over a certain percentage of an annotated anomaly's track then it is considered a true positive. Note however that this approach requires not only bounding boxes but tracking IDs for each anomaly.

Whilst the more modern evaluation methods provide a more accurate measure of detection ability they depend on a more detailed annotation scheme, including bounding boxes and tracking ids, that is not available to us in the UCF-Crime dataset. To improve evaluation without the additional annotation Lv et al. [7] propose only evaluating the AUROC scores on the anomaly videos, in order to avoid boosting results by correctly classifying long normal videos. We take this a step further and propose to use the TPR at an FPR of 0.1 as the evaluation metric. The reason for this is that the number of normal samples in a dataset far out number the number of anomaly samples, meaning that unless there is a very low FPR there will be far more normal samples tagged as anomaly then anomaly samples tagged as anomaly rendering the results unusable for real world application. This approach will help us focus on performance increases in high confidence predictions that are useful for real world application.

2.3 UCF-Crime Dataset

Here we present an in depth analysis of the current state of the art multi scene anomaly detection dataset UCF-Crime published by Sultani et al. [2] in 2018. This is the dataset on which we test our methods due to its diversity, size and previous use. The UCF-Crime dataset is 1900 videos totalling 124 hours of footage decomposed into 14 categories. A single category for normal videos and 13 categories labeling 13 different anomalies.

The UCF-Crime dataset is the largest multi scene dataset with a considerable increase in

diversity provided by the 1900 unique scenes compared to the 13 unique scenes used in the next highest ShanghaiTech dataset [22]. Diversity is important as it provides a more realistic comparison to the real world and prevents overfitting. It is also the only dataset using tagged anomalies in training that has been well tested in the literature. Providing tagged videos in training allows for the use of a greater variety of supervised methods that are able to learn representations of anomalies rather then just normal behaviours. This is able to increase performance. This makes it a good choice in order to benchmark our methods. Whilst the dataset's size and diversity promote it as an accurate representation of the real world it still has a number of shortcomings which should be avoided when creating a new CCTV surveillance video anomaly detection dataset. We therefore analyse these shortcomings in order to answer *RQ1: In what way should the state of the art UCF-Crime surveillance video anomaly detection dataset be improved to better evaulate models' ability to detect video based anomalies in the real world?*.

2.3.1 Anomaly Categories

We find that the 13 anomaly categories carry little value in the realm of anomaly detection as they are too nuanced for a model to correctly learn to discriminate between them. They are also unnecessary for determining the presence of an anomaly. In accordance with this approaches using this dataset since its release have all ignored the presence of the 13 categories focusing solely on anomaly vs normality. The categories are complex because differences between them are highly semantic and sometimes completely unclear. For example the difference between abuse, fighting and assault requires an understanding of the relationship and power dynamics between the actors involved in the physical altercation. Further complexity comes in the form of ambiguity when traffic accident or arson events result in an explosion, however are not categorized as explosion since each video has only one label. The disconnect between the given categories and the information on which the current anomaly detection techniques base their decisions makes reasoning about decisions difficult. In order to address this we provide a more generic classification of the videos.

In Table 2 we present a more generic, low level, description of the categories in this dataset. We describe the dataset in more basic terms not defined by a judicial system but rather by the identifying features of a scene. This allows us to assess the dataset at a level more closely related to the features outputted from a pre-processing step. From this we can see that the dataset contains four central types of anomalies.

1. Violent action related
2. Fire and Smoke related
3. Vehicle related

UCF-Crime Class	Simplified Elements
Abuse	Human-human violence Human-animal violence
Arrest	Human-human violence
Arson	Bright Light Smoke
Assault	Human-human violence
Burglary	Abnormal Vehicle Behavior Weapon present
Explosion	Bright Light Smoke
Fighting	Human-human violence
Road Accidents	Abnormal Vehicle Behavior
Robbery	Human-Human violence Weapon present
Shooting	Human-human violence Weapon present
Shoplifting	Human hiding object Human grab object and run
Stealing	Human-object violence Human-object unusual interaction
Vandalism	Human-object violence

Table 2. UCF-Crime class simplification

4. Shoplifting and Stealing

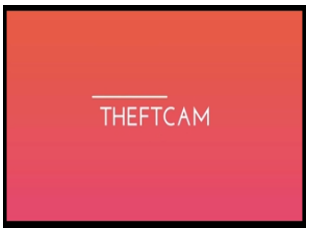
The benefits of this break down are multi faceted, firstly it allows the development of model aspects targeted at more specific events. This leaves the models with less to learn as we have distilled high level domain knowledge within them. In the most basic case we can have an individual detector for each of these core elements and develop a kind of cascading classifier as seen in the work of Mathias et al. [41]. Secondly the decomposition of classification into the detection of these features is a step towards more explainable AI. To this end an algorithm can report not just the anomaly score of a segment but also what basic elements where responsible for its decision. Explainable AI is of importance for meaningful application as with important decisions explanation can inform human oversight and create a higher trust in detections. We do not however completely remove the black box element of the algorithm, we simply move it further down the semantic tree and closer to basic action classification or image recognition which we are then able to more easily understand.

Interesting to note is the abnormal vehicle behavior description for burglary. This is because the burglary class contains a large number of automated teller machine (ATM) theft whereby a truck was used to pull an ATM off the wall by attaching it to the ATM with a chain. Category 4 is the most difficult as it only contains shoplifting and stealing which are highly semantic in nature and often aren't discernible to the human eye. For example one stealing video shows a person getting into a car and driving off. It is unclear from only the video that this is not the owner of the car.

2.3.2 Precence of Artifacts

There is a considerable presence of digital video artifacts that were not removed when assembling this dataset. These should be accounted for when training and testing any models. This is important since these artifacts not only reduce the accuracy with which the dataset represents the real world but may provide unintended correlation with anomalies which the model can learn, boosting performance improperly. Below we list the most common artifacts and in Figure 6 we highlight this by showing examples of frames that are annotated as normal and yet not representative of CCTV footage.

1. Many videos have entry or exit sequences as well as watermarks.
2. Many videos have multiple camera switches or cameras that pan and/or change their zoom.
3. Many videos have large black borders in order to make video size consistent.



(a) Digital intro sequence

(b) Camera switch

(c) Black border

Figure 6. Examples of non CCTV artifacts in the UCF-Crime dataset [2].

(a) Explosion Aftermath

(b) Explosion Aftermath

Figure 7. Examples of frames incorrectly labeled as normal in the UCF-Crime dataset [2].

2.3.3 Issues in Annotation

The annotation methodology for this dataset has not been reported by Sultani et al. [2]. Of particular concern is the test set where videos are annotated at a per frame level. This presents problems in particular because of the number of frames mislabeled as normal. For example the authors appear to tag frames in a video as an anomaly only if they fit the description of the event label for the anomaly. Cases where this is particularly misleading is videos whereby the event has a lasting after effect, for example when tagging explosions. Whilst correctly annotating the actual explosion frames e.g. bright flashing light as anomalous, they then label everything afterward as normal. However an explosion often leaves a scene in a drastically abnormal state, making the labeling of this as normal during test time incorrect. We can see clear examples of frames incorrectly labeled as normal in Figure 7.

2.3.4 Issues in Test Set Composition

Of the 1900 videos in the UCF-Crime dataset 290 are provisioned for testing, 150 normal and 140 anomaly videos. It is important to note that the majority of an anomaly video is made up of normal footage meaning that anomaly behaviours do in fact occur at a much lower rate then normal behaviour making up 6.7% of the test set, correlating better with the real world then a balanced dataset. What is however of concern is that of there are a

few very long videos that dominate the test set, inflating results considerably.

We look at the number of frames in the whole test set compared to the top 4 longest videos from the test set in Table 3. We can see that the longest 4 videos whilst making up for 1.4% of the videos account for 24.6% of the frames. This problem is further exacerbated by the state of the art methods because they divide each video into 32 segments for classification. This is present in the works of Sultani et al. [2], Zhong et al. [6] and Lv et al. [7] all of which are considered state of the art. What this does is to unduly give the model information about the length of the video since the number of frames in a segment are determined by dividing the video into 32 evenly sized segments. Because the length of an anomaly doesn't increase within the video, correctly classifying these videos as negative (normal) will considerably reduce the FPR whilst retaining the same TPR, increasing the AUROC measure. Furthermore the mentioned methods all use activity recognizers as backbone feature extractors. These are designed for short term activity recognition in the order of seconds and so will likely produce representations with less informative activity recognition features on longer videos as they are run on segments as long as 2 minutes each. These representations may then look closer to no activity (normal behaviour) then others, unfairly creating a normal classification.

Videos	Normal Frames	Anomalous Frames	Total
All	1036974	74834	1111808
Longest 4	273479	210	273689

Table 3. Comparison of the number of frames in the test set and the 4 longest videos from the test set of the UCF-Crime dataset [2]. Clearly these 4 predominantly normal videos have a disproportionate representation in the test set.

In order to asses the magnitude of the problem we compare the approaches of Sultani et al. [2], Zhong et al. [6] and Lv et al. [7] on the full test set and on the corrected subset with the longest 4 videos removed in Figure 8. As we can see, the addition of these 4 videos accounts for a large increase in the AUROC score which is highest in the methods of Lv et al. [7] and Zhong et al [6], the different amounts that different methods are effected is most concerning as it may change the conclusions of previous comparisons. To further confirm that it is these long normal videos that are responsible for the difference, we plot histograms of the number of frames in each anomaly score bucket for the works of Sultani et al. [2] and Lv et al. [7] in Figure 9. From this it clear that the method of Lv et al. [7] especially pushes normal scores towards zero as intended by their sparsity and temporal modules enabling them to better take advantage of long normal videos.

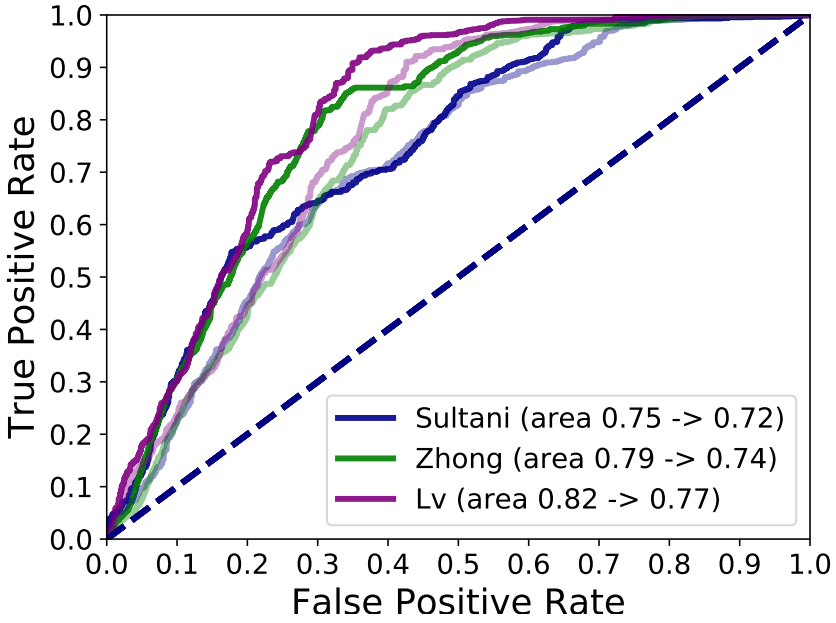
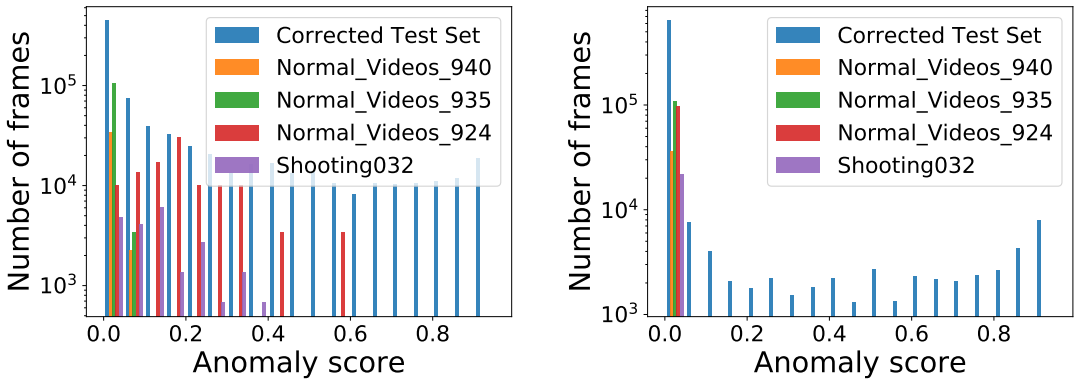


Figure 8. ROC Curves comparing the test sets with and without the 4 videos over 20000 frames. The large increase in correctly classified normal frames allows for a lower FPR at the same TPR resulting in an increased area under the curve. The increased area comes from improvements in segments with greater then 0.2 FPR rendering them usesless for real world application.



(a) Sultani et al. [2]

(b) Lv et al. [7]

Figure 9. Comparison of the number of frames in each anomaly score bucket for the works of Sultani et al. [2] and Lv et al. [7] on the UCF-Crime test set [2], separating the four longest videos to form the corrected test set. Important to note is the log scale necessary as the bulk of scores are pushed to 0, especially in the work of Lv et al. [7].

2.4 Action Classifiers

Many of the state of the art anomaly detection methods use transfer learning in order to leverage successes from other video related domains. In particular the three state of the art methods we evaluate make use of action classifiers for feature extraction. This involves inputting each video segment into a neural network based action classifier and then using the activation at an internal layer to represent the video segment. The two classifiers we discuss in further detail in the methods section are Convolutional 3D Network (C3D) used in the work of Sultani et al. [2] and Temporal Segment Network (TSN) with a Batch Normalized Inception (BN-Inception) backbone used in the work of Lv et al. [7].

2.5 General Object Detection

Many of the present anomaly detection techniques use general object detection in pre processing. This is used either as an attention mechanism for example in the work of Ionescu et al. [17] by cropping the frames to objects/humans detected or as a feature representation extractor as in Doshi et al's [28] work. General object detection is the task of localizing objects within a scene and determining their class label. Object detection has a long history in computer vision being one of the most researched areas in the field. For a full review of the last twenty years see Zou et al's [42] survey. Early successes in object detection were largely based on hand made features created to represent images in a salient and efficient manner. The issue of localization was dealt with predominantly by considering a large number of different windows over the image [43, 44]. Since 2014 object detection has been the poster child for deep artificial neural networks due to the rapid increase in performance seen since Girshick et al. [45] first proposed the application of regions with CNN features (R-CNN) to the task.

Deep neural network pipelines at first used a two stage approach separating the detection into region proposal and region classification steps [45, 46]. This however has since been surpassed by single stage techniques [47, 48, 49], which perform localization and classification all in a single model. The benefit of this is a large speed up in classification time. Of the current methods Redmon's YoloV3 [47] is the fastest and so is chosen for our experimentation as we will perform experiments on a single consumer grade central processing unit (CPU) and therefore have to prioritize computational efficiency.

2.6 Fire and Smoke detectors

The presence of arson and explosions in the UCF-Crime dataset [2] presents a move away from detecting only human based anomalies. These classes lead us to the field of fire and smoke detectors as it is highly succesful and may prove useful in pre-classifying these non human related examples, allowing other implemented models to remain more human centric. According to Cetin et al. [50] there has been a lot of research into fire and smoke detection due to the high damage caused by fires and the fast reaction time of video footage analysis as opposed to chemical based detectors. Chemical detectors are slow as they require the smoke to reach them for detection. Detecting the presence of fire or smoke varies in its complexity however even simple heuristics have been used and could be useful as a fast and effective was of filtering many non-human related scenes.

The bulk of research focuses on forest or wide area fire detection because the need is greater due to no humans being present to call emergency services. The work on wide area fire detection is transferable to a CCTV scenario and there is already some research specific to CCTV footage. Hashem et al. [51] use the industry standard inputs of color and motion for fire detection to first detect potential fire regions based on an RGB range and then use the RGB difference over a third of a second to represent motion. Finally a support vector machine (SVM) is trained on these inputs to detect the presence of a fire. This method not only performs detection well reaching an accuracy of 95.32 on the Mivia fire detection dataset [52] but also runs fast with an average processing time of 0.43s per frame. This illustrates the simplicity of fire detection and the potential for using a pre trained detector as a low cost pre processing step. Even simpler heuristics have been applied with success such as that smoke has R, G, B values close together and fire has RGB values such that R > G > B [50] providing an even more efficient and interpretable approach. We however choose to implement a more heavy weight neural network approach since the penalty to computation is minimal when compared to other aspects of our model and the neural network gives improved performance whilst allowing for the extraction of representations at intermediate layers.

When training our fire and smoke detection model we use the Fire-Smoke Dataset provided by Abimbola and Olafenwe [8] augmented with frames from explosion and arson anomaly videos in UCF-Crime [2]. The original dataset is divided into three classes: fire, smoke and neutral with 900 images in each class. We add an additional 618 images from the UCF-Crime dataset in order to adapt the original dataset to our use case, since the original dataset is far more diverse in subject matter and zoom then our CCTV footage ranging from e-cigarrete smoke to factory explosions.

2.7 Traffic Anomaly Detection

The field of traffic anomaly detection in surveillance footage is fairly new and can be traced back to Track 2 of the NVIDIA AI City Challenge 2018 [53]. Before this approaches looked at other variables such as road design and traffic throughput in order to try and predict the probability of accidents occurring. This however is not useful for our approach as scenic prior anomaly probabilities are unable to inform emergency services as to when a response is required.

Interestingly the work on traffic anomaly detection in surveillance footage has closely followed the back end of the complexity trend seen in anomaly detection in general. This is clearly illustrated as the top method from the anomaly track of the NVIDIA AI 2018 challenge by Xu et al. [54] trained a deep neural network action classifier to detect anomalies where as the top approach in the 2020 challenge by Li et al. [9] avoided large black box neural networks by taking advantage of a popular approximation of anomalies in traffic surveillance as *vehicles stopped where they should not be* and then using well established object detection and tracking methods to determine vehicles and their trajectories with thresholds for what is stopping for an anomaly amount of time.

Traffic anomaly detection has also been able to take advantage of its reduced scope to effectively localize anomalies to relevent parts of an image or regions of interest (ROI). Vehicle movement is often used in order to mask regions of an image without any vehicle movement in the video as ineligible for anomaly detection. This helps focus the attention of methods on only the road in the footage, this is especially useful in the heuristic approaches where stopped cars on pavements and in parking lots would cause false positives. This can be complemented with the removal of vehicles that never stop in the footage as they obviously do not fit the anomaly approximation.

For our research we adapt the work of Li et al. [9], the winner of the vehicle anomaly detection track of the NVIDIA AI CITY Challenge 2020 [55] by loosening various heuristic constraints such as object detection confidence threshold, vehicle standing still limit, and backtracking limit in order to compensate for the lower quality and more diverse nature of the UCF-Crime dataset [2]. These relaxations are acceptable because the output from this method isn't the final result and so further refinement is still possible by the classification combination approach that combines the generic anomaly detector with the output from the vehicle anomaly detector.

3. Methods

We test two different approaches for splitting anomaly detection in CCTV surveillance footage into smaller subproblems that can be addressed by separate classifiers which can then be combined for an improved performance. The first method is to split the problem based off the objects present in different scenes, the idea is that object presence may indicate different types of scenes that bias towards certain types of anomalies. In order to do this we cluster the training videos based on objects detected in each video. A baseline classifier is then retrained for each cluster, this means that our collection of classifiers differs not in model structure but rather in what dataset they were trained on. The combination strategy is to assign each test video to a cluster and then use the model for that cluster to classify each segment of that video. The second approach aims to further utilize transfer learning by using models designed for different anomaly types rather then simply retraining the same model for different scenes. An extreme learning machine is then trained to combine the models outputs. The specific anomalies we target are fire and smoke detection as well as traffic anomaly detection.

In this chapter we describe the two classifier combination approaches in more detail, followed by a description of how we tested state of the art methods with anomalies artificially removed in order to investigate how much the current methods use the semantics of a segment to produce its anomaly score. We then describe the state of the art methods used as baseline detectors and feature extractors in the classification combination strategies, namely the MIL anomaly detection approach by Sultani et al. [2] used for retraining clusters, as input to the extreme learning machine and in the analysis of anomaly semantics use, the GCN anomaly detection approach of Zhong et al. [6] used in the analysis of anomaly semantics use and as the backbone for the context encoding approach of Lv et al. [7] that we use in the analysis of the UCF-Crime dataset, as an input to the extreme learning machine, and in the analysis of anomaly semantics use. Finally we describe the action classifiers C3D by Tran et al. [32] and TSN by Wang et al. [33] used as backbones in the MIL and GCN approaches respectively, the YoloV3 detector by Redmon [47] used for object detection in the clustering approaches and in the traffic anomaly detection method, the Resnet50 fire and smoke detection model implemented by Abimbola and Olafenwe [8] used as input to the extreme learning machine, and the unsupervised traffic anomaly approach by Li et al. [9] also used as input to the extreme learning machine.

3.1 Retraining for Specific Clusters

We begin with an analysis of the clusters produced by KMeans clustering in order to determine whether distinct clusters exist, what their general descriptions are, and which cluster splits are most well defined for retraining individual classifiers. We suspect that the more distinct the clusters are the more chance they provide a useful decomposition of the anomaly detection in CCTV surveillance problem, furthermore if we can provide well defined definitions for each cluster it increases the interpretability of the individual models trained. For clustering we use an 80 dimensional vector representing the objects detected within each video from the 80 classes in the COCO object detection dataset [56]. In order to reduce computational load we first represent each video by a tuple of 5 frames, $(f_1, f_{\frac{1}{4}t}, f_{\frac{1}{2}t}, f_{\frac{3}{4}t}, f_t)$, where t is the total number of frames in a video. The YoloV3 [57] object detector is then run on each of the five frames, providing a number of detections $d^i \in D$ each with a class category $c^i \in (1, 2, ..., 80)$ and a confidence score $s^i \in [0, 1]$. For a single frame we then add up the confidences of object predictions in each category to represent the frame as an 80 dimensional object vector and then average the vectors representing the five frames in order to represent a video as an 80 dimensional object vector as seen in Equation 3.1.

$$V = \frac{1}{5}\sum_{n=1}^{5}\sum_{d_i \in D^n} \hat{e}_{c^i} \cdot s^i \tag{3.1}$$

Using the object vectors for each video we cluster the dataset using the kmeans clustering implementation in the Scikit-learn Python library [58]. We analyse the clusters using both 2 class and 10 class kmeans clustering as the 2 class approach may provide more balanced clusters making retraining more stable and the 10 class will be able to give us a better idea of the overall scenic variety. Kmeans clustering is an iterative approach to clustering that takes as input the number of clusters required, the vector representations of each sample in the dataset, and the current clusters, which we randomize initially. It then tries to iteratively improve clusters by re classifying samples in order to minimize the average distance between each sample within a single cluster as seen in Equation 3.2 where $S = \{S_1, ..., S_k\}$ are clusters of video vectors $(V_1, ..., V_n)$ and μ_i is the average of all vectors in cluster S_i.

$$argmin_S \sum_{i=1}^{k}\sum_{V \in S_i} ||V - \mu_i||^2 \tag{3.2}$$

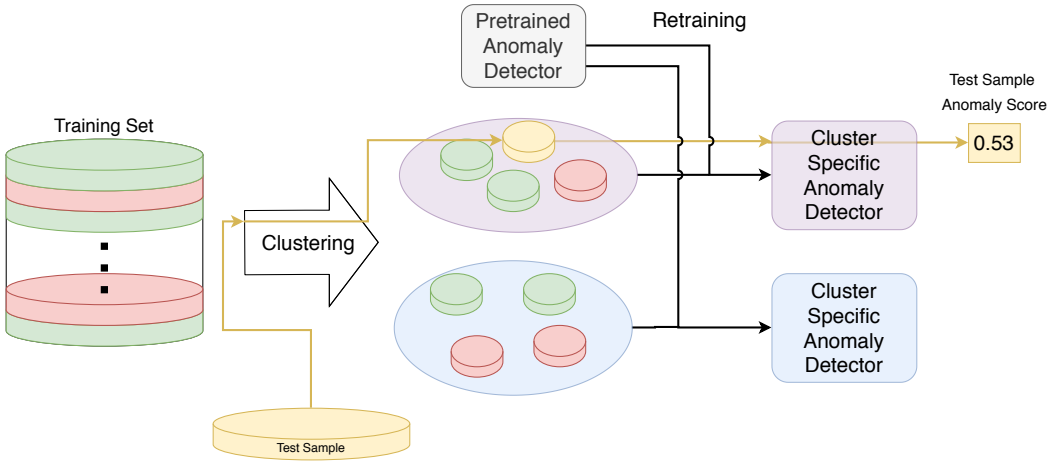


Figure 10. The overall method for retraining using specific clusters. The videos in the training set are first clustered according to which objects are present in each video. A pre trained anomaly detector is then retrained using the data from each cluster. At test time a test video is assigned to a cluster and each sample for that cluster uses the model trained on that cluster for classification. In the example the yellow sample is assigned to the purple cluster which means the purple anomaly detector is used for classification.

We can then assign each test sample to a cluster by representing each cluster as the average of all the training set object vectors in it and then calculating which cluster the test sample's object vector is closest to using Euclidean distance. An example of the retraining per cluster approach for 2 class clustering can be seen in Figure 10. For the combination approach we select the most relevant clustering splits seen in the cluster analysis and then retrain a trained baseline detector, the state of the art MIL method of Sultani et al. [2], for each cluster by using only the training samples assigned to that cluster. For testing each test video is assigned to a cluster and the model trained on that cluster is used to assign an anomaly score to each segment of that test video.

3.2 Combining Specialized Detectors

This method uses the same classifiers for all test samples by training an extreme learning machine to combine the outputs of detectors designed for specific anomaly types with a baseline detector. We wish to combine classifiers from other domains as they directly make use of the semantics of the anomaly, insuring a relevant interpretation of predictions and limiting the use of scenic priors. For each video segment we combine the outputs from baseline classifiers B for which we use the works of Sultani et al. [2] and Lv et al. [7] with the output of a Resnet50 fire and smoke detector by Olafenwa and Abimbola [8] denoted F, and the outputted score from a traffic anomaly detector by Li et al. [9] denoted T by concatenating them into a single vector representation, the optimum model structure for combining a baseline method with the two more specific detectors can be seen in Figure 11.

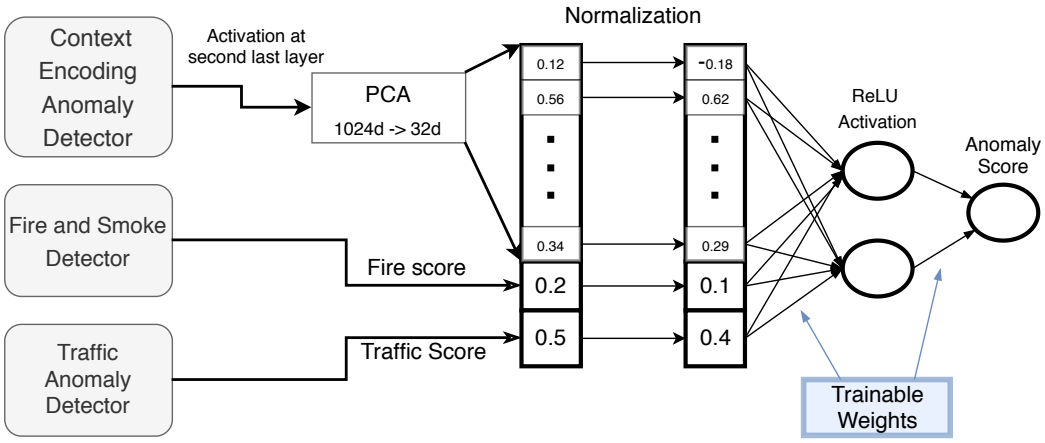


Figure 11. Optimal method for combining fire and smoke detection by Abimbola and Olafenwa [8] and traffic anomaly detection by Liu et al. [9] with the general surveillance anomaly detection via context encoding by Lv et al. [7]. The second last layer activation of Lv et al's model is first reduced to 32 dimensions before being concatenated with the fire and traffic scores. Each dimension in the now 34 dimension vector is scaled to a mean of 0 and a variance of 1 according to the mean and standard deviation of the values in the training set. The normalized representation is then fed through a multi-layer perceptron with a single, size 2, internal layer with ReLU activation in order to output the final anomaly score.

The different detectors segment videos at different scales with the baseline detectors dividing videos into 32 evenly sized segments, the fire detector making a single prediction per 1s of footage and the traffic anomaly detector providing a single score for all frames within a detected anomaly. In order to consolidate these outputs we use the fire detection segmentation and consider segments of 1s. For each video with a total number of frames n and a frame rate of 30 frames per second we represent it as a sequence $V = (C(0), C(30), ..., C(t), C(t+30), ..., C(30 \cdot \lfloor \frac{n}{30} \rfloor))$ and for training consider ground truth anomaly scores to be $V_a = (max(a_0, .., a_{30}), max(a_{30}, ..., a_{60}), ..., max(a_{30 \cdot \lfloor \frac{n}{30} \rfloor}, ..., a_n))$, a_i is 1 if the frame i has an annotated ground truth of anomaly and 0 if it has an annotated ground truth of normal and $C(t) = (B(t), F(t), T(t))$ where $B(t)$ is the baseline representation for the segment in which frame t falls, $F(t)$ is the fire score for the 1s segment starting at frame t and $T(t)$ is the maximum traffic anomaly score for frames in the range $(t, t+1, ..., t+30)$.

For the baseline scores $B(t)$ we test both the final anomaly score and the activation at the second last layer. When testing the second last layer activation of Lv et al's context encoding method [7] we additionally perform a principle component analysis to reduce the activation vector from 1024 dimensions to 32 dimensions. This is important to reduce sparsity and redundancies in the activation representation, this is not needed for the work of Sultani et al. [2] as the activation is already 32 dimensions. Once we have obtained the concatenated representation $C(t)$ for all 1s segments in the dataset we normalize each

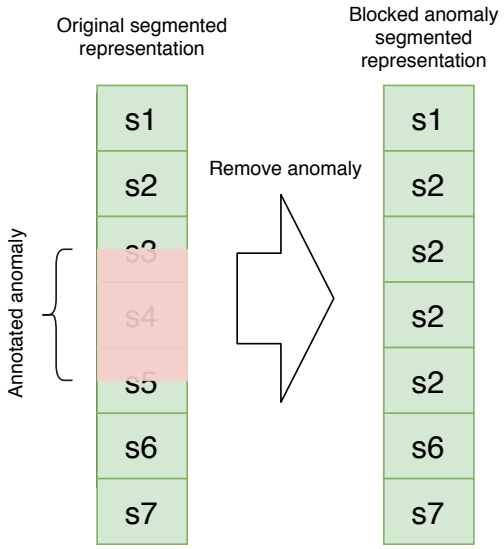


Figure 12. In order to remove anomalous segments we replace them with the last fully normal segment before the annotated anomaly starts.

dimension to have a mean of 0 and a variance of 1 accross the training dataset. This is done because dimensions come from different classifiers and so their absolute differences are not comparable. When testing the representations are normalized using the mean and variance obtained in the normalization of the training set and scores at a 1s segment level are converted to a frame level by giving each frame in the segment the same score.

We train two different extreme learning machines to classify the concatenated representations, namely a generic stochastic gradient descent (SGD) regressor using Huber loss and a multi layer perceptron (MLP) regressor with a ReLU activation at its intermediate layer in order to capture non linear relationships. The reason regressors are used instead of classifiers is so that the model produces anomaly scores which can be compared at different thresholds as required by the AUROC evaluation method seen in state of the art literature. We first perform a hyperparameter search using a validation set before finally testing only the best performing parameters on the test set so as to avoid fitting for the test set.

3.3 Analysing the Use of Semantics

In this study we analyze how much attention the baseline methods pay to the semantics of anomalies. In order to do this we replace the anomaly segment of each video in the test set with the normal segment just before the anomaly starts as seen in Figure 12. A model that bases its anomaly score on the semantic information of an anomalous event should see a large performance drop due to this change. A model that shows little difference in anomaly detection is likely to be using scene based priors to obtain its predictions. A decision based

off scenic priors does little to address the real world use case whereby we want to shorten response times to alarm worthy events. This is because it is only useful in telling us which scenes have higher anomaly rates, but does not provide information on when anomalies are actually happening. We re test three state of the art methods on this altered test set, the MIL method of Sultani et al. [2], the GCN method of Zhong et al. [6], and the context encoding method of Lv et al. [7].

3.4 State of the Art

In order to evaluate our methods we obtain instances of three anomaly detection models that have achieved state of the art performance on the UCF-Crime dataset [2]. We begin with a third party implementation of the the MIL approach of Sultani et al. [2] as the original baseline for the UCF-Crime dataset [2]. Next we obtain a pre trained model from the state of the art graph convolutional method of Zhong et al. [6] as they obtain a considerable performance increase over Sultani et al's [2] methods, boosting AUROC from 0.74 to 0.82. Finally we implement the recent work of Lv et al. [7] as they further increase performance and build directly off the graph convolutional approach of Zhong et al. [6]. The work of Lv et al. [7] can be thought of as a post processing optimization on the outputs of Zhong et al's [6] work. The overall performance of each of our instantiations can be seen in Figure 13. All three methods had to be adapted to be used on a CPU rather then the original GPU use, due to hardware availability. The method of Zhong et al. [6] had to be down sampled due to computation required which may have resulted in its worse performance and propagated this performance decrease to the method of Lv et al. [7].

3.4.1 Multi Instance Learning Approach

In the work which originally published the UCF-Crime dataset Sultani et al. [2] propose to move away from the previous norm of treating anomaly detection as out of distribution detection and rather propose to use a segment wise regression estimation. The regression estimation aims to estimate an anomaly score between 0 and 1 by using weakly labelled videos, that is, videos labeled as anomaly at the video level. If a video contains an anomaly it gets an anomaly label 1 and if it does not it gets a normal label 0. Most notable is that temporal annotation is not provided during training and in order to overcome this they frame the problem as a multi instance learning task. An overview of their approach can be seen in Figure 14. For this approach we use a third party PyTorch [59] implementation by Kosman [60]. The reason for this is that the adaption of an implementation to be run on a CPU is more easily completed using the PyTorch implementation rather then the Theano [61] based implementation published by Sultani et al. [2].

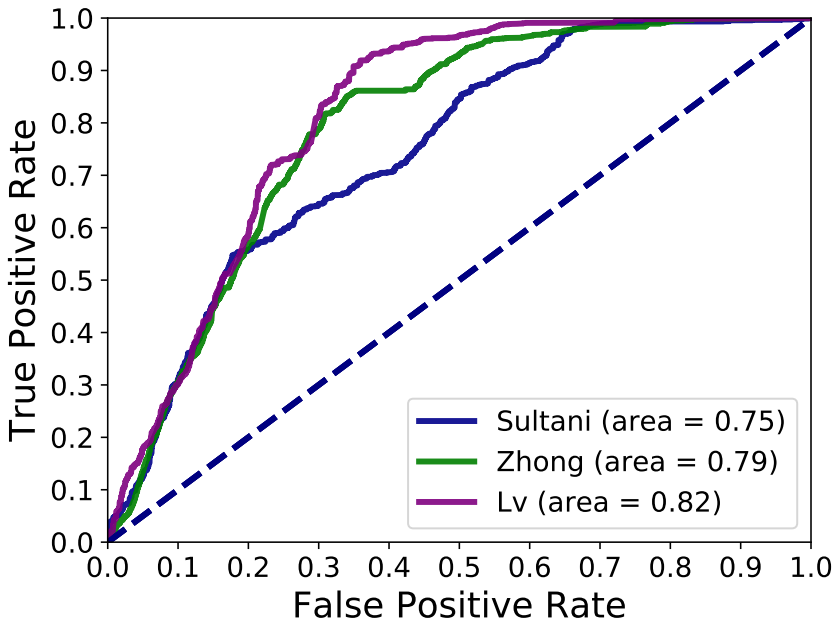


Figure 13. Receiver operator curve for the three implemented methods. Original AUROC scores are as follows: Sultani 0.74, Zhong 0.82 and Lv 0.84 [2, 6, 7].

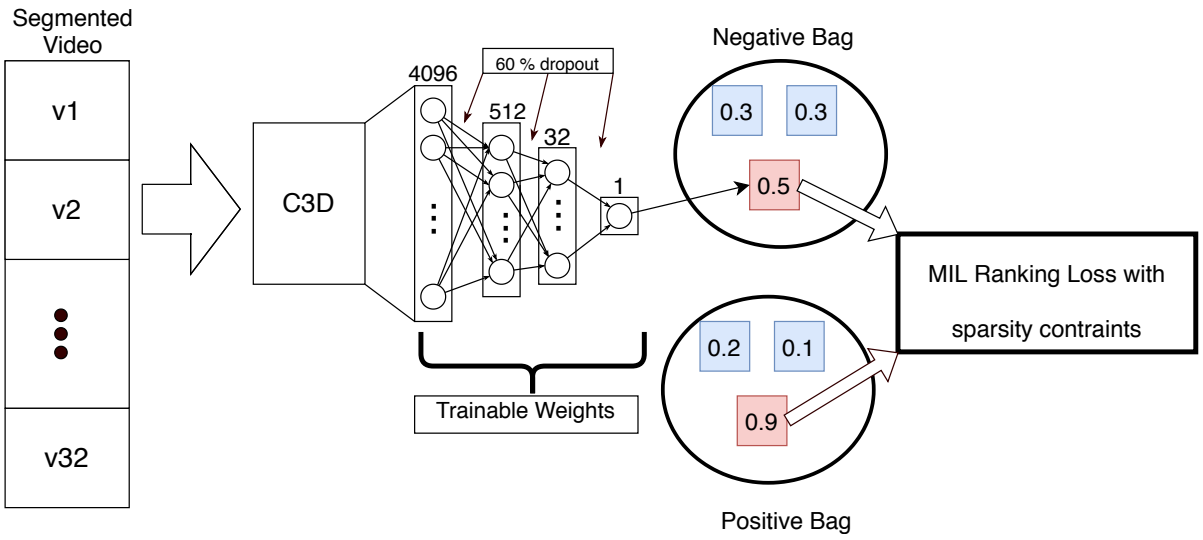


Figure 14. Flow diagram of the multi instance learning approach adapted from the work of Sultani et al. [2]. Videos are segmented into 32 segments and fed through the C3D action classifier in order to obtain a 4096 dimension vector to represent each segment. This representation is then fed into a 4 layer, trainable, fully connected neural network in order to produce an anomaly score for each segment. For the loss function two bags of segments, each bag representing one video, are compared, using only the highest anomaly score in each bag. When comparing two bags one represents a normal video and one represents an anomaly video.

First each video is divided into 32 segments $V = \{s_1, s_2, ..., s_{32}\}$. Then 16 RGB frames are used to represent each segment and are fed through the C3D action classifier in order to extract a 4096 dimension representation for each segment. A 4 layer fully connected neural network is trained to give each 4096 dimension vector a score between 0 and 1, 0 being most normal and 1 being most anomalous.

The loss function used to train the 4 fully connected layers takes as input the anomalous scores for all segments in 2 videos. One of the videos has the anomaly label and one of the videos has the normal label. If we take two videos, one anomalous $V_a = \{s_1^a, s_2^a, ..., s_{32}^a\}$ and one normal $V_n = \{s_1^n, s_2^n, ..., s_{32}^n\}$ and for each segment an anomaly score $f(s_i)$ the loss function then compares only the maximum anomaly score given to a segment in each each video as seen in Equation 3.3. In addition to this temporal smoothness and sparsity constraints are added for the anomalous video, in order to insure that the anomaly output corresponds to the behaviour we would expect from a correct anomaly labeling. The temporal smoothness loss in Equation 3.4 insures that anomaly scores do not change too rapidly by punishing such changes and the sparsity constraint in Equation 3.5 insures that the bulk of the video gets a low score by adding the sum of all the anomaly scores to the final loss. The final loss function, a weighted sum of the three components can be seen in Equation 3.6.

$$l(V_a, V_n) = \max\big(0, 1 - \max_{i \in \{1,..,32\}} f(s_i^a) + \max_{i \in \{1,..,32\}} f(s_i^n)\big) \tag{3.3}$$

$$l_{smo}(V_a) = \sum_{i=1}^{31} \big(f(s_i^a) - f(s_{i+1}^a)\big)^2 \tag{3.4}$$

$$l_{spa}(V_a) = \sum_{i=1}^{32} f(s_i^a) \tag{3.5}$$

$$l_{final}(V_a, V_n) = l(V_a, V_n) + \gamma_1 l_{smo}(V_a) + \gamma_2 l_{spa}(V_a) \tag{3.6}$$

3.4.2 Label Noise Cleaner Approach

The second method we evaluate is the work of Zhong et al. [6] whereby they attempt to make even better use of pre trained action classifiers by retraining various action classifiers

to directly predict anomaly scores. To evaluate this method we use a pretrained model as provided on their GitHub repository [62]. This model is a trained instantiation of a TSN with BN-Inception classifier as designed by Wang et al. [33]. We use the RGB modality as the combination of TSN with BN-Inception and RGB input performed best, as reported by Zhong et al. [6]. The parameters of the model provided by Zhong et al. [6] specify using a stack depth of 1 and a step of 5. This means that videos are split into segments of 5 frames each and the middle frame for each one is processed by the action classifier in order to obtain an anomaly score. In order to overcome the computational limitations of using a single CPU we downsample the segmentation, splitting each video into 50 frame segments whilst retaining the stack depth of 1. The downsampling produces a 10x reduction in computational load however also results in an AUROC performance drop from 0.82 reported by Zhong et al. [6] to 0.79.

In order to provide segmentwise labels to the action classifier the method of Zhong et al. [6] alternates the training of an action classifier with that of a label noise cleaner so that iteratively the action classifier gets better at classifying anomalies and the labels for the anomalous segments, fed back to the action classifier, get cleaner. The noise cleaner does not clean labels for segments in normal videos, this is because their labels are noiseless since we know that the label for all segments is normal. For anomalous videos the label noise cleaner takes as input the anomaly score given to each segment by the action classifier and cleans it via the high confidence (low variance) predictions. These cleaned predictions are then fed back to the action classifier for the next training iteration. The label cleaner and action classifier are alternatively trained until finally only the final re trained action classifier is used for testing.

The noise cleaner takes the form of a graph convolutional network made up of 2 types of graphs combined via average pooling with sigmoid activation in order to produce better labels. For each video the graph representation takes the form $G = (V, E, X)$ where V is the vertex set such that each segment of the video is one vertex. E is the edge set whose strength represents either feature similarity or temporal distance, depending on the graph, and X is the set of feature vectors for the segments in V.

Weights $A^F_{(i,j)} \in \mathbb{R}^{\mathbb{N}\times\mathbb{N}}$ in the feature similarity graph are determined via the dot product between segment representations as seen in Equation 3.7 and the weights in the temporal graph are simply based off the difference in their sequence position as seen in Equation 3.8. The output for each vertex is then produced by combining its anomaly score with the anomaly scores of its neighbours weighted in accordance to their adjacency strength.

$$A^F_{(i,j)} = exp(X_i \cdot X_j - \max(X_i \cdot X)) \tag{3.7}$$

$$A^F_{(i,j)} = exp(-||i - j||) \tag{3.8}$$

The loss function for the noise cleaner is made up of direct and indirect supervision components $l = l_D + l_I$. The training of the action classifier follows the original method as described in the original work by Wang et al. [33]. The direct component l_D is where they consider the confidence of predictions by only considering the loss of the top K most confident predictions. Confidence is determined by making 10 predictions for each segment with different crops and determining which segment has the lowest variance over these predictions. The direct supervision loss function for the high confidence segments can be seen in Equation 3.9 where $\{y_i\}^N_{i=1}$ are the probabilities assigned by the action classifier, $\{p_i\}^N_{i=1}$ are the probabilities assigned from the noise cleaner and H is the set of high probability predictions for the current video.

$$l_D = -\frac{1}{|H|}\sum_{i \in H}\big(y_i ln p_i + (1 - y_i) ln(1 - p_i)\big) \tag{3.9}$$

The indirect supervision term tries to smooth the predictions of the network over many training steps by punishing predictions that vary from the average prediction as seen in Equation 3.10 where $\bar{p_i}$ is the discounted average prediction for this segment over the last few iterations and N is the total number of segments for the current video.

$$l_I = \frac{1}{N}\sum_{i=1}^{N}|p_i - \bar{p_i}| \tag{3.10}$$

Finally at test time the predictions from the action classifier $\{y_i\}^N_{i=1}$ for each video are taken directly, making the deployment and use of this classifier convenient and prompting its use in the third method we test.

3.4.3 Context Encoding Approach

The third method we evaluate is the work of Lv et al. [7] whereby they target the correct localization of anomalies in order to improve detection. They do this using features

extracted by the last layer of the retrained action classifier developed in the label noise cleaning method of Zhong et al. [6]. We only implement the localization branch of their approach whilst noting that in their study they also utilize data augmentation to produce more samples such as cropping videos, introducing blur and blocking parts of videos. We do not implement this augmentation as the data preparation and preprocessing takes prohibitively long with little benefit, as seen in their ablation studies where the additional data augmentations improve the overall AUROC score from 84.44 to 85.38. We are also able to reach competitive performance of 0.82 without the additional augmented data and being based off representations from a Zhong et al. [6] action classifier with 0.03 reduced AUROC performance.

In order to better localize anomaly detections they propose an improved method for considering the context which surrounds a segment by representing each segment as a learned linear combination of its k nearest neighbours on each side. In practice k is found to be optimal at 2, with most of the benefit saturated and greater neighbourhood sizes being computationally prohibitive. We however use a neighbourhood size of $k = 1$ as the increase in input size for the network from 3×1024 to 5×1024 provides a fully connected network that exceeds our random access memory (RAM) limitations and hence greatly slows processing time. The change from a window size of 2 to 1 according to the original work of Lv et al. [7] only reduced the performance from a video level AUROC of 0.9565 to 0.9501 and so is an acceptable simplification. We do however note concern that they used a video level AUROC for the window size comparison instead of the standard frame level AUROC, raising a concern over the effective frame level AUROC difference as these measures are not directly comparable.

See Figure 15 for an overview of the context encoding approach. When considering a single video the approach splits it into 32 evenly sized segments. In order to overcome the different segmentation approach taken in the work of Zhong et al. [6] the last layer vector representations are aggregated for a segment. Whilst the aggregation approach is not specified in the original paper we average each dimension for the segment, also noting that our downsampling means that we have 10x less action classifier representations to aggregate for each of the 32 segments. Given a sequence of vector representations of segments from a video $X = (s_1, ..., s_{32})$ a representation for each segment is then obtained as seen in Equation 3.11 where W_j and W_0 are matrix transformations with learnable parameters.

$$\hat{s_t} = \sum_{j=-1,1} W_j s_{t+j} + W_0 s_t + b \tag{3.11}$$

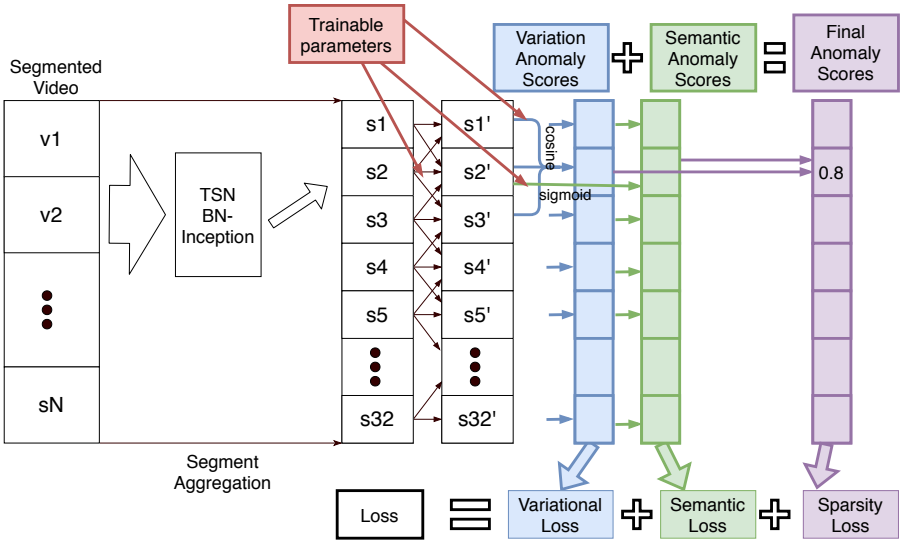


Figure 15. An overview of the context encoding approach taken by Lv et al. [7]. First each video segment is run through the pre trained TSN BN-Inception action classifier as trained by Zhong et al. [6] and represented by the activation of the second last layer of this model. Segments are then aggregated so that each video has 32 segments. Each segment is then represented as a linear combination of its own and its neighbour's vector representations. After this a segments semantic score is extracted directly via sigmoid activation and a variation score is extracted via a cosine comparison with its neighbours. Semantic and variation scores are added to get the final anomaly score, which is used to calculate the sparsity loss during training and performance during testing.

Using this representation the method then targets two sources of information for producing an anomaly score. The first source is the direct semantic information in the transformed feature representation, determined via a fully connected layer with a sigmoid activation function as seen in Equation 3.12. The second source is the contextual information gained by comparing a segment's representation to that of its neighbours. This can be considered a second order analysis of local variations as it is performed on the representation produced via the combination strategy of Equation 3.11. In order to compare segments the cosine distance between their vector representations is used as seen in Equation 3.13.

$$A_{sem}(\hat{s}_t) = \sigma(W_{sem}\hat{s}_t + b_{sem}) \tag{3.12}$$

$$A_{var}(\hat{s}_t) = (2 - \cos(\hat{s}_{t-1}, \hat{s}_t) - cos(\hat{s}_t, \hat{s}_{t+1}))/4 \tag{3.13}$$

These scores are added together to get the final anomaly score $A_{final} = A_{sem} + A_{var}$ for each segment. In order to overcome the lack of segmentwise labelings and produce a meaningful loss function they consider the video anomaly score to be the maximum

difference between segment scores in the videos, the idea here being that the normal and anomaly segments of an anomaly video will produce a high difference whilst all segment scores in a normal video should remain low. We formally define the video wide score in Equation 3.14.

$$S(X) = \max_{i,j=1,...32} |A(\hat{s}_i) - A(\hat{s}_j)| \tag{3.14}$$

Through equation 3.14 we obtain video level anomaly scores for semantic S^{sem} and variation S^{var} separately. The batch losses l_{sem} and l_{var} for each of these are computed via a simplistic averaging strategy as seen in Equation 3.15 whereby the first 30 videos have a ground truth of anomaly and the last 30 have a ground truth of normal, leaving us with a total batch size of 60.

$$l(\{X^i\}|_{i=1}^{60}\}) = max\{0, 1 - \frac{1}{30}\sum_{i=1}^{30} S(X^i) + \frac{1}{30}\sum_{i=31}^{60} S(X^i)\} \tag{3.15}$$

Finally the losses for the two information sources are combined with a sparsity loss similar to that of Sultani et al. [2] as defined in Equation 3.5 and weighted by a small β to produce the final loss l_{final} as seen in Equation 3.16. By optmizing for this loss Lv et al. [7] are able to more effectively localize anomalies resulting in an increase in overall AUROC score from the 82.12 seen in the work of Zhong et al. [6] to 84.44.

$$l_{final} = l_{sem} + l_{var} + \beta l_{spa} \tag{3.16}$$

3.5 Feature Extraction Techniques

3.5.1 Action Classifiers

We use instances of two action classifiers, Convolutional 3D Network (C3D) used in the work of Sultani et al. [2] and Temporal Segment Network (TSN) with a Batch Normalized Inception (BN-Inception) backbone used in the work of Zhong et al. [7].

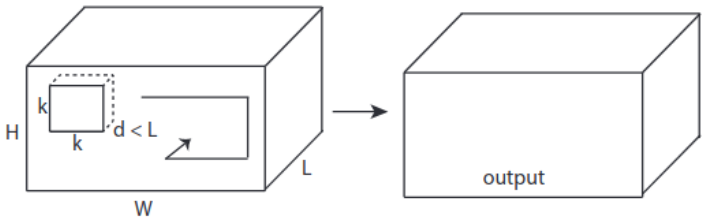


Figure 16. Example of 3D convolution adapted from the work of Tran et al. [32]. The 3D convolutions produce from a cube another cube representation, preserving the temporal information in the input. In the C3D model the input from a segment is 16 frames deep and each convolution kernel is 3 deep.

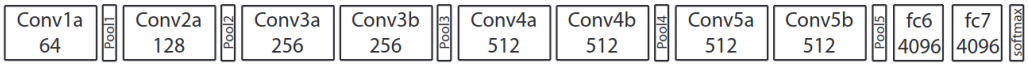


Figure 17. Architecture of C3D, taken from the work of Tran et al. [32]. Each Convolutional layer uses 3x3x3 convolutional kernels with the denoted number of filters. The max pooling layers use 222 kernels and the fully connected layers fc6 and fc7 have 4096 outputs each. Ordinarily the 4096 output from fc7 is used as a representation for a video.

Convolutional 3D Network

C3D is a simplistic and popular feature extraction network as designed by Tran et al. [32]. It was designed for generic video feature extraction with the main contribution being the additions of 3D convolutions in order to retain temporal interactions from a scene. To this end convolutions are not just taken as a square with a width and height but rather as a cube, including a square from multiple frames at once, as seen in Figure 16. The implementation used by Sultani et al. [2] takes as input 16 frames and at the fc7 layer as seen in Figure 17 produces a 4096 dimensional vector representing these frames. The network used is trained on the Sports-1m dataset [34]. Whilst the C3D network is dataset agnostic the Sports-1m dataset [34] is the most commonly used.

The Sports-1m dataset as released by Karpathy et al. [34] contains over a million YouTube videos containing 487 different sporting actions. The videos are of different lengths which is good as diversity promotes the learning of generic features however the dataset is collected as a set of YouTube links and so it is difficult to obtain a single static version of the dataset. For this reason the same weights of the same pre trained network are used for experimental validity across studies. Narrowing the domain of feature extraction to sports runs the risk of ignoring important features salient to generic action classification but not sports. It may also over represent aspects less important in generic video detection, for example focusing too much on the nature of the green part in a video to discern between sports pitches. The dataset does however have a great number of classes which may help to force the second last layer towards a generic action representation.

Temporal Segment Network with BN-Inception

TSN is an approach designed by Wang et al. [33] in order to incorporate long distance temporal relationships in video footage whilst alleviating the high computational cost of deep learning methods. To this end it is an overarching method to intelligently combine outputs from underlying short range detectors. The underlying detection used in the work of Zhong et al. [6] is the BN-Inception model. The component wise nature of this final implementation allows for individual components to be swapped out as newer more successful models are developed.

A sample of the Temporal Segment Network used by Zhong et al. [6] can be seen in Figure 18. Firstly a video is divided into a set number of segments $\{S_1, S_2, ..., S_k\}$. Each segment is then represented by a small snippet from within it producing a sequence of snippets $\{T_1, T_2, ..., T_k\}$ each corresponding to a segment S_i. Snippets are then fed into any number of action recognition backbones to produce a sequence of class labelings $\{F_b(T_1), F_b(T_2), ..., F_b(T_k)\}$ from each backbone. In the given example the two backbones whilst having the same ConvNet structure take as input two different modalities, RGB images and warped optical flow images, in order to gain information from both the structure of the scene and how it changes. This means that the example has two backbones F_{RGB} and F_{OF}. From here the overall classification is aggregated across all the segments for each backbone classifier individually, producing a class labeling for the entire video from each backbone $C_b = G(\{F_b(T_1), F_b(T_2), ..., F_b(T_k)\})$ where G in the current implementation is an averaging of the score for each class. Finally the class labels from each backbone, or in the example C_{RGB} and C_{OF} are fused, most often using a weighting strategy $C_{final} = \lambda \cdot C_{RGB} + (1 - \lambda) \cdot C_{OF}$.

The BN-Inception or Inception-V2 model is one first produced in the work of Ioffe and Szegedy [63] as an example implementation of their novel batch normalization using an underlying inception model as developed by Szegedy et al. [64]. The idea with batch normalization is to reduce the effects of co varying model parameters by normalizing the activations between each layer. The idea is that co varying parameters slow neural network convergence as each subsequent layer's parameters are highly dependent on the absolute values of the previous layer's. What Ioffe and Szegedy propose is to normalize the activation of each node across a mini batch to a mean of 0 and variance of 1. What this means is that each layer only needs to consider the activation by comparison to other activations in the batch. This is a more consistent task as the input for each layer now comes from the same distribution with a mean of 0 and variance of 1 in every iteration.

As an example take an input batch of k samples $B = \{S_1, S_2, ...S_k\}$ and 2 intermediate

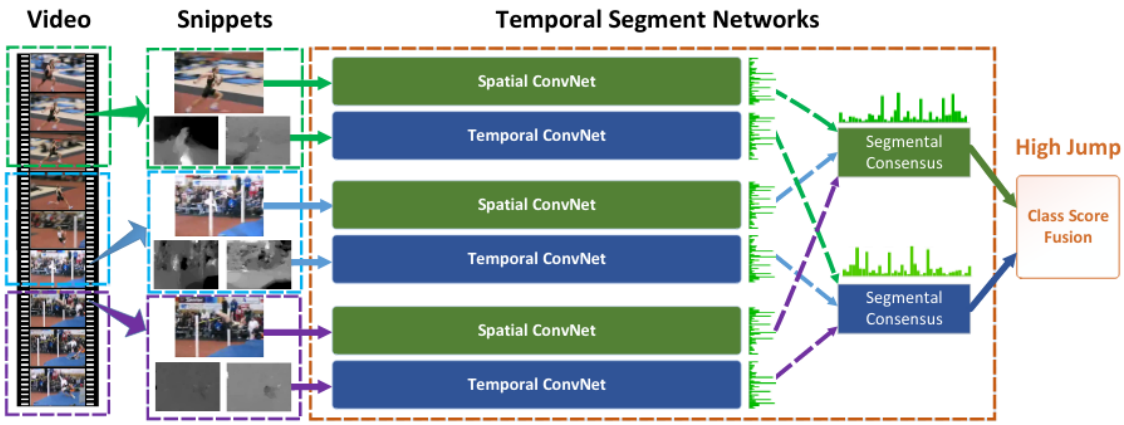


Figure 18. Temporal Segment Network as applied to a high jump video. This network uses two ConvNets [] one for each modality, RGB and warped optical flow, as its backbone short term networks. The scores for each network are then aggregated across the different segments to provide a modal consensus after which the model consensus's are combined in a weighting to produce the final class labels.

layers in the network L_i and L_{i+1}. L_i is made up of a number of nodes N_f^i and we take one as N for simplicity. Each sample in B produces an activation at N giving us k activations $\{x_1, x_2, ...x_k\}$ to be fed forward to layer L_{i+1}. Batch normalization normalizes the distribution of these activation's to have a mean of 0 and a variance of 1 as can be seen in equation 3.17. The normalized value $Norm(x_j)$ is fed forward to the subsequent network L_{i+1}. Important to note is that the expected value of x, $E(x)$ and the variance of x, $var(x)$ are calculated at a batch level which means that each sample in the batch is fed through the network in parallel.

$$Norm(x_j) = \frac{x_j - E(x)}{\sqrt{var(x) - \epsilon}} \tag{3.17}$$

The inception model is a popular deep neural network with specialized inception modules as developed by Szegedy et al. [64]. The aim of these inception modules it to take convolutions at different scales in order to account for scale variance between samples without making the network too deep and hence computationally expensive. It is a way in which to widen the network for better performance rather then to deepen it. The implementation used was pre trained on the Kinetics-400 dataset. The Kinetics-400 dataset is a state of the art action classification dataset first released in a paper by Carreira and Zisserman [35]. The Kinetics-400 dataset continues to be expanded and at present contains 700 action classes with at least 600 samples each. Each sample has a single class label and is 10s long. Due to its diversity of action classes the dataset is ideal for learning a generic representation of actions. The temporal scale of actions is however kept constant, which is

why additional mechanisms such as segmentation and aggregation are necessary in order to adapt the pretrained network to the UCF-Crime [2] domain.

3.5.2 YoloV3 Object Detection

We use the PyTorch [60] YoloV3 [47] implementation by Linder-Norén as found on his GitHub repository [57], this is a PyTorch instantiation of Redmon's YoloV3 object detector [47]. We use the YoloV3-tiny version as it is a smaller and a faster network then other state of the art methods with pre trained weights from the website of Redmon [65]. The tiny version is smaller because it uses less stacked convolutional layers then the original network and an input size of 416x416 pixels rather then 608x608 pixels. We perform the object detection on every 10th frame of each video, this provides a bounding box, class label from the 80 COCO dataset classes [56], confidence score and class confidence score for each object detected.

3.5.3 Resnet50 Fire Detection

In order to extract features targeted at detecting explosions and arson we train a deep neural network to detect fire and smoke. The network produces scores s_{fire}, s_{smoke} and $s_{neutral}$ such that their sum is 1. We then use the score $s_{anom} = s_{fire} + s_{smoke}$ as a feature representation for the fire and smoke aspect of videos. This representation is then combined with the state of the art anomaly detectors by concatenating them onto the feature representations produced by the second last and final layer of these state art anomaly detectors and then training simplistic linear regressors to correctly detect anomalies. Specifically we are targeting an improvement to detections on explosion and arson videos and high confidence predictions without worsening performance on other categories.

The segmentation of videos in the base anomaly detectors and the fire and smoke detector are different and therefore need to be combined in a meaningful way. The work of Sultani et al. [2] and Lv et al. [7] both split each video into 32 segments. The fire and smoke detector uses a single image to represent each 1 second of video. What we then do is to match up each 1 second representation with its corresponding 1/32nd segment and use that combination to represent the 1 second of footage, resulting in a segmentation of each video into 1 second chunks.

Category	Fire Smoke Dataset	UCF-Crime Augment
Fire	900	237
Smoke	900	186
Neutral	900	195

Table 4. The number of images from each category used for training the fire and smoke detector. The Fire Smoke Dataset is taken from the work of Abimbola and Olafenwe [8] and the UCF-Crime augment is images extracted from explosion and arson videos in the UCF-Crime dataset problished by Sultani et al. [2].

Dataset augmentation

To train the Resnet50 model we adapt the Fire-Smoke dataset of Abimbola and Olafenwe [8] to the UCF-crime setting in the following way. We use only the explosion and arson anomaly videos first dividing each video into a normal segment and an explosion or arson segment according to the annotation provided by Liu et al. [38] and then extracting frames from each for classification. The annotation provides a start time t_1 and an end time t_2 for the explosion or arson event in each video. We extract every 15th frame from the beginning segment $\{f_i\}_{i=10}^{i=t_1-10}$ for the normal images and then every 15th frame from the anomalous segment $\{f_i\}_{i=t_1+10}^{t_2-10}$ for fire and smoke images. A buffer of 10 frames is given to each segment to compensate for annotation ambiguity. The non explosion segment after t_2 is not used as it has been unreliably annotated because there is often explosion aftermath after this annotation, containing smoke and debris which should not be considered normal.

Finally the fire and smoke images are manually sorted into a fire set and a smoke set, removing any ambiguous images containing neither fire nor smoke. The neutral dataset is also manually pruned for any images that appear to contain either fire or smoke. We end up with a training set of 2700 Fire Smoke Dataset [8] images and 618 UCF-Crime images distributed as seen in Table 19. For an idea of the difference between the two datasets see Figure 19. Most notably the Fire Smoke Dataset is more diverse not only containing images extracted from CCTV footage and its images are also of a higher quality.

Detection Model

For the fire and smoke detection we retrain an implementation of the Resnet50 neural network by Abimbola and Olafenwa [8] where they implement a training and testing framework for fire and smoke detection following the description of residual networks found in the influential work of He et al. [66]. We retrain the model using our augmented dataset for 20 additional epochs. Deep residual networks as presented by He et al. [66] are neural networks with many layers and short cut connections in order to mitigate optimization degradation that occurs in deep networks. The short cut connections are

(a) Fire, 1.0 (b) Smoke, 0.82 (c) Neutral, 0.76

(d) Fire, 0.6 (e) Smoke, 0.68 (f) Neutral, 0.62

Figure 19. Sample images of the dataset used to train the Resnet50 fire and smoke detector and their Resnet50 classification results showing class and confidence. Top: images from the Fire Smoke Dataset of Abimbola and Olafenwa [8], Bottom: images extracted from explosion and arson videos in the UCF-Crime dataset [2]. Important differences between the two sources are that the UCF-Crime dataset is less diverse in subject matter and zoom and has a lower resolution.

identity mappings that skip a number of layers within a network. These allow the learned weights of a network to optimize a residual of the current activation rather then the activation itself.

The use of a deep network structure been shown to improve classification ability as shown by Szegedy et al. [64]. Their network however required careful parameter and network architecture choices whereas combining a deep neural network with residual learning provides us with a robust classification learner. The used Resnet50 model follows the structure used in the original paper as seen in Table 5 which illustrates the depth achieved by this network allowing it to learn complex representations.

3.5.4 Road Accident Detection

In order to produce traffic anomaly detections we adopt the approach of Li et al. [9]. The reason this approach was chosen is that it is the current state of the art coming first in the NVIDIA AI CITY 2020 traffic anomaly detection challenge [55], but also because it uses deterministic algorithms based on a heuristic interpretation of accident detection as determining *vehicles that are stopped in the wrong place*. Furthermore the approach is highly modularized again enhancing interpretability by simplifying the task for black box detectors into interpretable and well understood goals such as object detection and object

layer name	output size	layers
conv1	112×112	7×7, 64, stride2
conv2_x	56×56	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv2_x	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv3_x	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv4_x	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
avg_pool	1×1	average pool, 1000-d fc, softmax

Table 5. Network architecture of Resnet50. Building blocks that make up one layer are in brackets. Skip connections used to mitigate optimization degradation are used between ever layer. This network design follows the original work of He et al. [66].

tracking. For our implementation we adapt the parameters of the traffic anomaly detection implementation published by Li et al.[9] on their Github page [67]. We also swap out the Faster R-CNN [68] object detector, preferring the YoloV3 detector [47] as we already have an implementation available and it less computationally intensive. Object trajectories are calculated using the deepsort implementation published by Wojke et al. [69] and available on their Github repository [70].

The approach starts off by detecting still vehicles on the road. It does this by first masking non road parts of videos by excluding parts of the frame that do not have a vehicle trajectory within them in any part of the video. The masked footage is then used to create a background version of each video which excludes any pixels with flow above a particular threshold. This masked version of the background is then fed through an object detector in order to detect potential candidates for stopped cars. Once a candidate has been selected its box level and pixel level tracks in the original video are then combined via union in order to provide annotation for the full anomaly, including the part when the vehicle is moving. We describe each step in more detail below with a full list of the relaxed parameters available in Appendix 1.

Hypothetical Abnormal Mask

The final mask is based on the intersection of two sub masks we change this to the union of the two submasks as the detections from YoloV3 were not good enough in many cases

creating too many masks that removed entire videos. The first mask is determined by analysing where in a scene pixels are changing. To this end if the pixel difference exceed a threshold between two frames they are considered to contain moving objects, all pixels exceeding this threshold are not masked in order to retain potential abnormality area. The second is determined by masking areas that do not contain a vehicle trajectory from deepsort. The deepsort trajectories are filtered by a minimum length, travel distance and bounding box size in order to remove false, side road and parking lot trajectories.

Background Modeling

Background modeling is performed using the MOG2 background extraction algorithm by Zoran [71] to remove moving vehicles. This is used in the forward direction to predict candidate anomalies by finding still vehicles and is used in the backward direction in order to refine where anomaly predictions start. The reason the different directions are used is that the criterion for removing a pixel from the extracted background is weaker then that for adding it. The result is that when we run the MOG2 extraction backward the anomaly object appears sooner in the video then when we run the extraction forward, providing a better starting point for the anomaly.

Anomaly Tracking

Box level tracking uses the vehicle bounding boxes found in the background of the video. Boxes in cosecutive frames are combined to form a single tube if their IoU is over a certain threshold otherwise a new tube is started. Tubes that end and start within a threshold of each other are combined as they are considered to be a part of the same anomaly. The pixel level tracking follows the method developed in the work of Bai et al. [72] where they iteratively update pixel level matrices $V_{undetected}, V_{detected}, V_{score}, V_{state}, V_{start}$ and V_{end} which keep track of when pixels are within a detected bounding box or not and combine the anomaly status of one pixel accross many frames if the difference is less then a particular threshold. Suspicious pixels can then be combined to form the detected anomaly.

Obtaining object trajectories

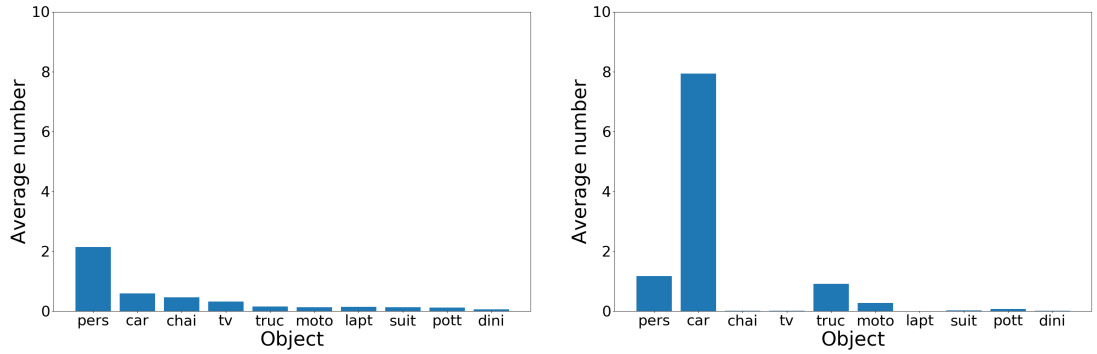
In order to provide object trajectories we run the YoloV3 [47] detector pre trained on the COCO object dataset [56] on every 10th frame in the UCF-Crime dataset. This produces bounding boxes for objects from 80 different categories as well as a confidence score for each detection. The following vehicle categories are then considered for road accident detection: bicycle, car, motorbike, bus and truck. The detections for these categories are fed through the deepsort tracking implementation in order to provide tracks of objects through the video

4. Results

4.1 Analysing Clusters

This is the first of two experiments we use to investigate the hypothesis that by retraining a baseline model for specific clusters of data we can improve the detection results. This addresses *RQ2: What is the best way in which to successfully decompose the problem of anomaly detection in CCTV surveillance into smaller sub problems?* as we retrain the base classifier for each cluster providing a collection of models used for classification. First we attempt to cluster the videos based on what objects are detected within them. The aim here is to develop an understanding of the data and to look for potential dimensions on which to cluster the data for re training.

We start by 2 class kmeans clustering the UCF-Crime [2] training set. The hypothesis here is that the data should be easily separable into two very different rough categories such as indoor vs outdoor or road vs non road. Looking at the object averages for each class in Figure 20 we can see that this indeed the case as the second class has on average 8 cars present and so indeed represents street scenes as opposed to the more diverse first class. We then cluster the videos using 10 class kmeans in order to investigate if there exist further notable splits in the data. The classes are described by their general them in Table 6.



(a) (b)

Figure 20. Distribution of most detected objects by YoloV3 [57] detected across 2 kmeans classes.

Class Description	Video Count
No detections	725
1-2 people no cars	334
1-2 cars with no people	262
Streets with 5-10 cars	152
1 or 2 people with chairs	130
small groups of people	102
streets with lots of cars	59
crowds of only people	39
streets with lots of people	38
rooms with many chairs	18

Table 6. Different clusters in UCF-Crime data

We note here that there is a large class of videos in which no detections were made. This rules out using techniques built only upon object detection for the task of anomaly detection because in many videos the objects are not clear enough for detection. Furthermore the bulk of the classes mentioned in Table 6 are determined by the number of cars and the number of people in the scene. This points towards the potential in using these two object categories as a building block for more specific classifiers.

Next we look at the distribution of anomalies within these clusters. This is done in order to analyse whether or not anomaly videos correspond in some way to the objects detected within them. Looking at the histogram in Figure 21 we can see that anomaly videos appear in a much greater frequency in the cluster in which no objects are detected. Having looked at the videos we suspect that is due to the lower quality of the footage. This perhaps speaks to a bias where videos of crimes, explosions and other noteworthy events are uploaded to YouTube regardless of video quality however when uploading normal CCTV footage one has more choice and therefore the overall quality of the videos is better. From these results we conclude that objects are in general not a good way to cluster the dataset into fine grained categories as most objects appear seldomly in the dataset. Cars and people may however be used as the detection thereof appears to be robust and they appear frequently within the dataset.

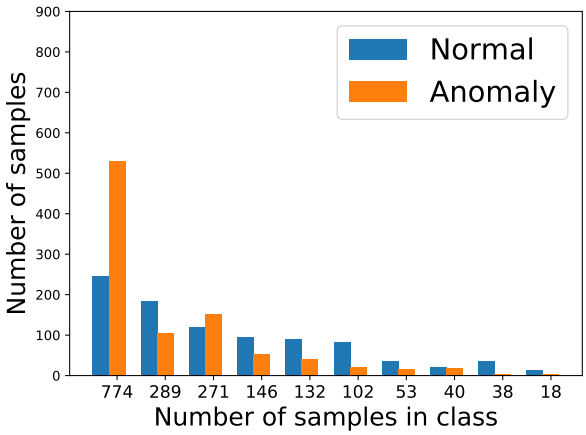


Figure 21. Distribution of anomalies across 10 kmeans clusters.

4.2 Retraining Using Object Clusters

In this experiment we continue the work of Section 4.1 by re training the MIL anomaly detection model of Sultani et al. [2] using meaningful clusters. Based off the results in Section 4.1 we start with the clusters defined by vehicle and person presence and then expand this to kmeans clustering. For the presence and absence of vehicles the dataset is split fairly evenly. Most of the videos however contain people. We therefore also perform the clustered retraining on the presence/absence of more then two people in the detections to check the efficacy of this method on a more balanced dataset. The size of each cluster over the training dataset can be seen in Table 7.

Cluster	#Anomaly	#Normal	Total
No Car	485	532	1017
Car	454	392	746
No people	140	108	248
People	799	816	1615
< 2 people	306	189	495
> 2 people	633	735	1368

Table 7. Number of scenes in each cluster. Objects are detected using the YoloV3 [47] object detector over 5 frames: first, last, middle and the quartiles for each video. The presence or absence of cars and people are used to separate clusters, with the additional split of >2 people chosen as it more evenly splits the clusters.

Car vs No Car

We present the results of retraining on two clusters, the first is videos with no cars present and the second is videos with cars present. The mixed strategy involves using the baseline

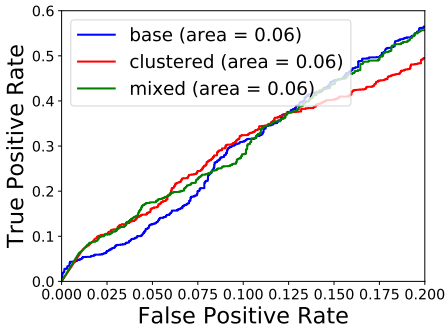
	Results per cluster		Results on all	
Model	Cluster 1	Cluster 2	Clustered	Mixed
Base	0.73	0.76	0.72	0.75
NoCar/Car	0.71	0.75		
Base	0.64	0.76	0.74	0.74
NoPeople/People	0.69	0.75		
Base	0.7	0.76	0.75	0.74
< 2 People/ > 2 People	0.75	0.75		

Table 8. AUROC Scores of re training on different clusters (base 0.75). We compare results on individual clusters for the baseline model and the model retrained on that cluster. Cluster 1 represents the cluster without the precense of the object('s) and cluster 2 the cluster with the object('s). The clustered result on all uses the model belonging to the given cluster in order to produce an anomaly score. The mixed approach uses the baseline model for videos belonging to the cluster without the presence of a given object and the retrained cluster for videos belonging to the cluster with the described object.

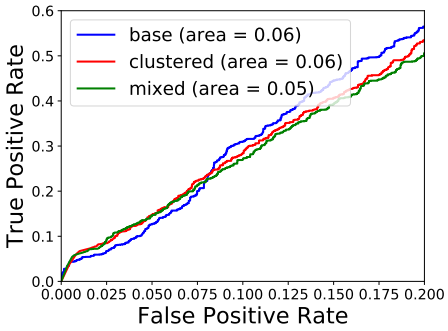
detector for videos without vehicles and the clustered detector for videos with vehicles. The reason for this is that we expect the diversity in the nocar dataset to be far greater as it includes any scene not taken from the streets and any scene where YOLOv3 [47] is unable to make detections. As the dataset becomes more diverse it suffers more from a small training set as over fitting is punished due to an increasing discrepancy between the training and test set. For this reason in the mixed strategy we use the baseline detector trained on all training samples for the no car class.

We see in Table 8 that the overall AUROC improves for the mixed technique but not for the purely clustered technique. However when we look at the section of the ROC curve with $FPR \in [0, 0.2]$ in Figure 22, the most applicable range for real life applications, we dont not see any significant improvement in performance. This illustrates that at the very low FPR the re training does not improve results. This discrepancy between the important low FPR range and the overall curve implies that the retraining is improving the low confidence classifications rather then the high confidence ones. This is not useful for real world application as using low confidence anomalies will produce too many false positives.

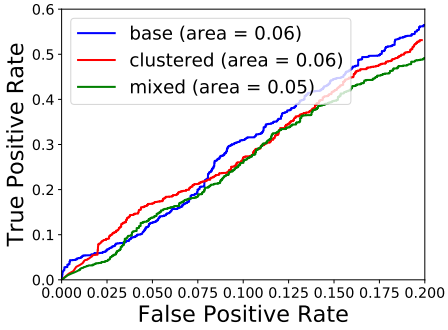
We suspect that the improvement only at low confidence means the retraining is simply fitting to biases present in the car anomaly videos rather then the anomalies themselves. This hypothesis is further confirmed by looking at the results on the individual clusters in Table 8 where we see that the models tested on the specific No Car and Car clusters individually worsen results. This means that it is the change in absolute anomaly score produced between the two models that have improved results rather then enhanced detection of anomalies. The no car model learns to give overall lower anomaly scores and the car



(a) NoCar vs Car



(b) NoPerson vs Person



(c) <2People vs >2People

Figure 22. ROC curves of retraining on different clusters at the most important subsection where FPR < 0.2. Base is the MIL detection method of Sultani et al. [2], clustered uses two models produced by taking a training MIL model and then retrained on a specific cluster and mixed is retraining one model on the cluster that contains the detected object and using the baseline detector for the other cluster. The improvement over the baseline for the clustering and mixed approaches at a very low FPR shows the ability of the clustered retraining of Sultani's MIL model [2] to improve high confidence predictions.

	Results per cluster		Results on all	
Model	Cluster 1	Cluster 2	Clustered	Mixed
Base	0.23	0.43	0.32	0.28
NoCar/Car	0.22	0.50		
Base	0.20	0.33	0.29	0.27
NoPeople/People	0.13	0.29		
base	0.24	0.33	0.27	0.26
< 2 People/ > 2 People	0.27	0.28		

Table 9. TPR at FPR=0.1 of re training on different clusters (base 0.31) following the same scheme as Table. 8

model learns to give absolutely higher anomaly scores. The overall results then improve as the car cluster is biased towards having a higher rate of anomaly occurrences as seen in Table 7. Looking at the anomaly scores we see that this is indeed the case as the average score for the car cluster increased by 0.014 from the base method to the clustering method and the no car cluster only increased by 0.005.

People vs No People

For the case of people vs no people the dataset becomes far more skewed as seen in the scene counts of Table 7. This is because the class with people detected is far larger then the one without. In order to address this we introduce a second clustering method where one class is scenes with more then 2 people and another is scenes with less then 2 people. The idea here is that we want to distinguish between scenes with crowds and scenes without crowds as these may contain different types of anomalies.

Interesting to note in Table 8 and Table 9 is that the no people cluster's performance increases considerably more then the less then 2 people cluster, suggesting it has a more well defined anomaly type. Furthermore the cluster defined by having more then two people performs better then that defined by having more then no people, this suggests that the group of scenes with on average between 0 and 2 people detected performs badly, perhaps because it is the most diverse. Neither of the splits however appear to improve the overall results.

Fine grained clustering

Here we explore the hypothesis that finer grained clusters will allow models to specialize further. For this we compare kmeans as well as agglomerative clustering techniques. Agglomerative clustering cannot be done iteratively and so is not applicable to real world scenario's since we have to analyse the entire dataset at once in order to determine the clusters. Regardless we use it as a comparison for the kmeans approach as it may produce more meaningful clusters. Both methods are tested for 6 and 10 clusters as this gives us a wide range of clusters without making individual clusters too small.

For the clustered models we use a model of Sultani et al's method [2] trained for 500 epochs and then trained on individual clusters for 100 and 300 additional epochs respectively. For the baseline we compare the results with Sultani et al's [2] model trained for 600 and 800 epochs respectively. We can see the overall results in Table 10. The first thing to notice is that whilst the TPR at FPR=0.1 varies quite considerably the AUROC remains fairly consistant. This illustrates that improving high confidence predictions comes at a cost of low confidence predictions balancing to give a similar AUROC.

Method	Epochs	TPR	AUROC
KMeans 6	100	0.30	0.75
KMeans 6	300	0.32	0.75
KMeans 10	100	0.27	0.72
KMeans 10	300	0.25	0.72
Aggl 6	100	0.27	0. 74
Aggl 6	300	0.29	0.73
Aggl 10	100	0.30	0.74
Aggl 10	300	0.30	0.72

Table 10. TPR at FPR=0.1 and overall AUROC scores for the Kmeans and Agglomerative clustering strategies. Epochs represent how many additional epochs the baseline method was retrained for. The baseline method acheived a TPR of 0.3 and an AUROC of 0.75 meaning that only the Kmeans6 method with 300 retrain epochs acheives improved results at FPR = 0.1.

The increasing TPR at high confidence when we have more clusters appears to further suggest that allowing for more specificity i.e. smaller clusters allows models to specialize further. The hypothesis that greater class imbalance would worsen results due to increased overfitting appears to be false. Looking at the improvement as our two classes got more imbalanced suggests the opposite, that having smaller classes allows for greater specificity in our specialized model for that cluster.

4.3 Combining Specialized Detectors

In this experiment we aim to answer *RQ2: What is the best way in which to successfully decompose the problem of anomaly detection in CCTV surveillance into smaller sub problems.* by combining classification models designed for simpler sub tasks, fire and smoke detection as well as traffic anomaly detection with our baselines classifiers. For fire and smoke detection we use the Resnet50 model trained to detect fire and smoke in images and for traffic anomaly detection we use the traffic anomaly detection of Liu et al. [9].

We first split the original training set from UCF-Crime into a training and validation set with a make up as seen in Table 11 in order to determine optimal parameters for the combination strategy without directly optimizing for the test set. We test both a stochastic gradient descent regressor (SGD) and a multi layer perceptron regressor (MLP). We keep the models simple in order to insure they utilizes the already synthesized information present in the outputs of the other, more complex, underlying detectors. We test across different concatenated input modalities from the set $\{Sult32, Sult1, Lv32, Lv1\} \times \{Fire1, Acc1, Fire1Acc1\}$ where $Sult32$ is the vector extracted from the activation of the last layer in the MIL approach of Sultani et al. [2], $Sult1$ is the final output from Sultani et al's MIL approach , $Lv32$ is the 32 dimensional vector obtained by performing a singular value decomposition

	Training	Validation	Test
Anomaly	639	171	140
Normal	632	167	150

Table 11. The number of videos of the anomaly or normal category in each split of the UCF-Crime dataset [2]. Training on training set and testing on validation set is used to optimize parameters.Training on training plus validation set and testing on test set is used to report final results.

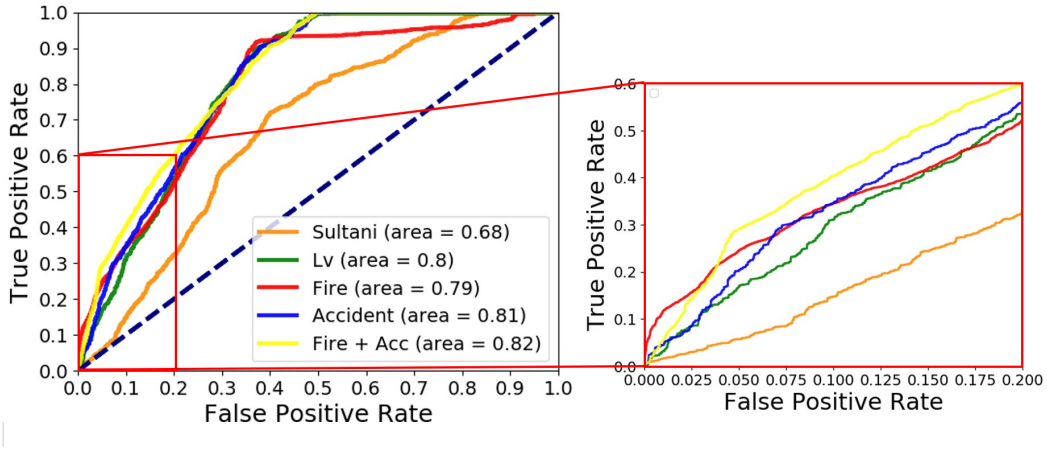


Figure 23. ROC curve of the results from the optimal parameters for the different datasets. Fire, Accident and Fire + Acc are the results produced by combining these detectors with the best baseline method. The zoomed in version of the graph clearly illustrates the benefit of the combined detectors at a low FPR or high confidence.

principle component analysis on the 1024 dimensional last layer of the approach by Lv et al. [7]. $Lv2$ is the vector containing the *semantic score* and *variation score* produced by Lv et al. [7]. $Fire1$ is the sum of the fire and smoke scores produced by the Resnet50 detector. $Acc1$ is the outputted score by the traffic anomaly detector and $Fire1Acc1$ is the concatenation of the two. In Figure 23 we can see the improvements of adding fire and smoke, and traffic anomaly detections to the baseline methods. The results shown are the optimal parameter settings for each input on the validation set, the optimal parameters can be seen in Table 12 and the full TPR at FPR 0.1 results can be see in Appendix 2.

Clearly the results on the validation set show a marked performance improvement by adding fire and smoke detection and traffic anomaly detection. The addition of accident detection gives a higher overall AUROC of 0.81 whilst the addition of fire and smoke detection performs better at a very low FPR < 0.025. This shows that fire detection performs better on very high confidence classifications but also provides a lot of low confidence false positives. The ability of the combination of the two to achieve even better performance implies that they improve detections over different videos, which is what we expect and desire given that the Explosions and Arson categories are distinct from the Road Accident and Burglary category.

Parameter	Optimal setting
Model	Multi-layer perception
Input representation	Lv32Fi1Acc1
Data Scaling	Normalize mean=0, var=0
Loss function	Huber
Learning rate scheme	Linear decrease
Initial learning rate	0.01
Final learning rate	0.00001
Iterations	30
Number of intermediate layers	3
Size of layers	34x2x1
Optimizer	Adam
Sample weight	$\frac{\#AllSamples}{2 \cdot \#CurrentClassSamples}$

Table 12. The optimal parameters for combining fire and smoke, and traffic anomaly detection with baseline anomaly detectors by Sultani et al. [2] and Lv et al. [7]. A block search was used to determine optimal parameters on a validation set so as to avoid optimizing for the test set.

Next we train the models, using the optimal parameters for each input on the full training plus validation set and test it on the test set. The results of this can be seen in Figure 24. From this it is clear that the addition of fire detection has improved performance with a TPR of 0.3402 at FPR 0.1 compared with 0.3347 for the best baseline method. The additions of traffic anomalies however has worsened performance, obtaining 0.2309 TPR at FPR 0.1. This is concerning as in the validation set the addition of traffic anomaly detection provided the most improvement. This may be because the parameter search on the validation set fit the model to one that suited traffic anomalies specifically in the validation set. This would mean that there is a major difference between traffic anomalies in the test and validation sets.

To explore this we look at the mean length of traffic related anomalies in each set, we consider anomalies of the Road Accident and Burglary classes to be traffic related, this is because burglary is vehicles pulling ATM's from shops. In the validation set 42 out of 338 videos are vehicle related anomalies and in the test set and 36 out of 290 videos have vehicle related anomalies. In the test set however the average length of a vehicle related anomaly is 459 frames where as in the validation set it is considerably longer at 952 frames. This makes it likely that a reduced performance by the underlying traffic anomaly detector by Li et al. [9] is the reason for the drop in performance. This is because the traffic anomaly detection method anchors detections by finding a still vehicle in an incorrect position and then building the vehicles trajectory from this. Given a shorter anomaly detection it is less likely that an anchor for the anomaly can be found.

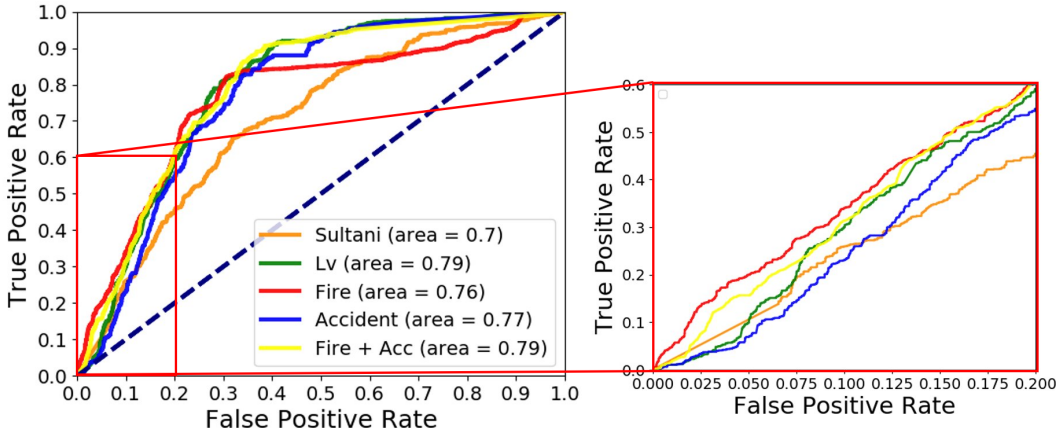


Figure 24. ROC curve of test set performance from the optimal parameters determined via the validation set. Fire, Accident and Fire + Acc are the results produced by combining these detectors with the best baseline method. The zoomed in version of the graph clearly illustrates the benefit of the fire detector at a low FPR or high confidence. The traffic anomaly detection model reduces performance worse.

4.4 Analysing the Use of Semantics

The performance of Sultani et al's [2] and Zhong et al's [6] methods drop surprisingly little when anomaly segments are removed during testing. In Figure 25 we can see the results for the three anomaly detection baselines. This clearly shows that the method of Lv et al. [7] achieved its goal of better localizing anomalous and indeed pays more attention to the semantics of the anomaly itself rather then focusing on scenic priors. This is shown due to the dramatic drop in performance that this method produces when the anomalous segments are removed, Lv et al. [7] drops 0.16 where as the works of Sultani et al. [2] and Zhong et al. [6] only drop 0.02 and 0.03 respectively. By illustrating how little the state of the art methods actually consider the anomaly itself this result suggests that the current state of the art methods could benefit considerably by drawing better attention to anomalies by considering localization in their model design.

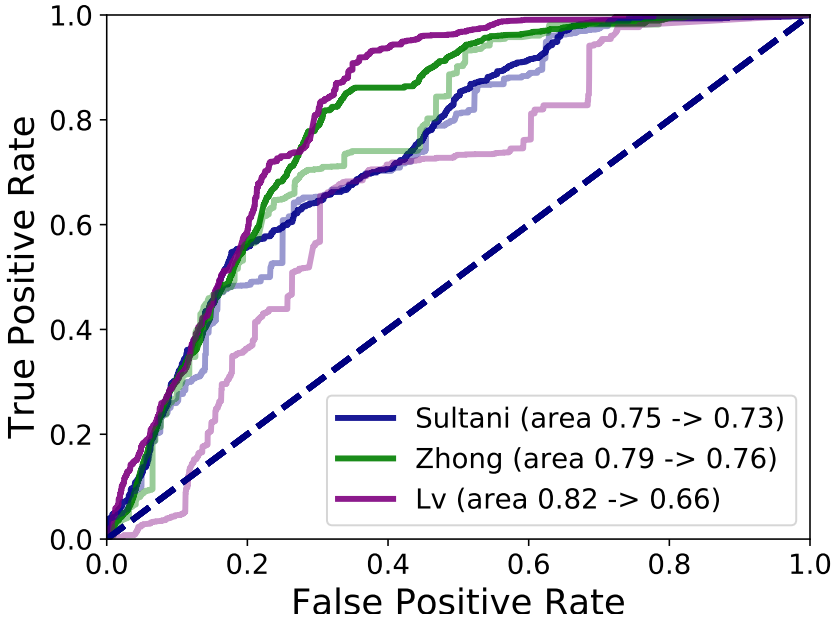


Figure 25. ROC curves comparing baseline tests and tests without anomalous segments for the works of Sultani et al. [2], Zhong et al. [6] and Lv et al. [7]. The transparent curve and in all cases worse result is the method run on the same test set with the anomalous segments in anomaly videos blocked out by normal segments from those videos.

5. Conclusions and Future Work

Through an investigation into the UCF-Crime dataset we have found various shortcomings that can be addressed in future work through the creation of a new CCTV surveillance anomaly detection dataset answering *RQ1: In what way should the state of the art UCF-Crime surveillance video anomaly detection dataset be improved to better evaulate models' ability to detect video based anomalies in the real world?*. Most notably, a new dataset with more consistent video lengths in order to avoid dominance by few videos and a cleaner dataset with less digitally edited images and scene cuts. This will allow the evaluation of methods to be more applicable to the real world by removing segments that may provide undue indications of anomaly presence, giving a clearer idea of when we have achieved CCTV surveillance anomaly detection at a level high enough for real world implementation.

We have shown the benefits of using a fire detection classifier in combination with a baseline classifier for detection of CCTV surveillance anomalies on the UCF-Crime dataset [2]. We have also shown the limited ability of more fine grained models based off clustering scenes by object presence and vehicle anomaly detection to improve performance. This along with the dataset analysis helps to answer *RQ2: What is the best way in which to successfully decompose the problem of anomaly detection in CCTV surveillance into smaller sub problems?*. We would suggest training classifiers to detect fire and smoke, traffic anomalies, violence, and shoplifting individually and then to combine the outputs from each detector, rather then training on subsets of the data.

For future work on the UCF-Crime dataset [2] we recommend exploring detectors for the remaining two categories of anomaly, namely *violence related* and *shoplifting and stealing*. There is already research into these fields and successfully implemented these methods will mean that every category of anomaly in UCF-Crime has a detector specialized for it. Further improvements must also be made to the traffic anomaly detection method in order for its contribution to be more generalizable.

The AUROC evaluation method, used to compare state of the art works, has been shown to be unable to correctly evaluate real world applicability. This is largely because the AUROC measure emphasizes detections at all confidence levels when for real applications

only the performance at a high confidence matters due to the high prevalence of negative samples (normal footage). The concern over the AUROC measure is further confirmed by the small decrease in performance seen when anomaly features are replaced by normal ones, meaning that this measure cannot distinguish between when models are using scenic priors for classification and when they are using the semantics of an anomaly itself. This is an important distinction as the detections of anomalies using their semantics is of far greater importance for a reactive surveillance application such as emergency services. The anomaly masking experiment was able to answer *RQ3: To what extent do the current state of the art methods rely on scenic priors rather than anomaly semantics for obtaining their performance?* by showing how little the current methods change when anomaly segments are removed, suggesting that the current state of the art methods rely mostly on scenic priors rather then anomaly semantics for detection.

In general future work can continue the paradigm shift from a single generic anomaly detector towards using a collection of more specific detectors by either targetting improvement in one of the specific detectors such as a shoplifting detector or by developing novel ways to decompose the the CCTV surveillance anomaly detection problem. The compartmentalization of the problem will then allow independent research in these areas to be easy integrated into a single detection pipeline.

Bibliography

[1] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery". In: *International conference on information processing in medical imaging*. Springer. 2017, pp. 146–157.

[2] Waqas Sultani, Chen Chen, and Mubarak Shah. "Real-world anomaly detection in surveillance videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6479–6488.

[3] Vaseekaran Sivarajasingam, Jonathan P Shepherd, and Kyle Matthews. "Effect of urban closed circuit television on assault injury and violence detection". In: *Injury Prevention* 9.4 (2003), pp. 312–316.

[4] Anup Mohan, Kent Gauen, Yung-Hsiang Lu, Wei Wayne Li, and Xuemin Chen. "Internet of video things in 2030: A world with many cameras". In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2017, pp. 1–4.

[5] Nancy G La Vigne, Samantha S Lowry, Joshua A Markman, and Allison M Dwyer. "Evaluating the use of public surveillance cameras for crime control and prevention". In: *Washington, DC: US Department of Justice, Office of Community Oriented Policing Services. Urban Institute, Justice Policy Center* (2011).

[6] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1237–1246.

[7] Hui Lv, Chuanwei Zhou, Chunyan Xu, Zhen Cui, and Jian Yang. "Localizing Anomalies from Weakly-Labeled Videos". In: *arXiv preprint arXiv:2008.08944* (2020).

[8] 0layemi Abimbola and Moses Olafenwa. *DeedQuestAI Fire-Smoke-Dataset*. `https://github.com/DeepQuestAI/Fire-Smoke-Dataset`. 2019.

[9] Yingying Li, Jie Wu, Xue Bai, Xipeng Yang, Xiao Tan, Guanbin Li, Shilei Wen, Hongwu Zhang, and Errui Ding. "Multi-granularity tracking with modularlized components for unsupervised vehicles anomaly detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 586–587.

[10] Ramin Mehran, Alexis Oyama, and Mubarak Shah. “Abnormal crowd behavior detection using social force model”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 935–942.

[11] Cewu Lu, Jianping Shi, and Jiaya Jia. “Abnormal event detection at 150 fps in matlab”. In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 2720–2727.

[12] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. “Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 1705–1714.

[13] Yong Shean Chong and Yong Haur Tay. “Abnormal event detection in videos using spatiotemporal autoencoder”. In: *International Symposium on Neural Networks*. Springer. 2017, pp. 189–196.

[14] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. “Learning deep representations of appearance and motion for anomalous event detection”. In: *arXiv preprint arXiv:1510.01553* (2015).

[15] Royston Rodrigues, Neha Bhargava, Rajbabu Velmurugan, and Subhasis Chaudhuri. “Multi-timescale Trajectory Prediction for Abnormal Human Activity Detection”. In: *The IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 2626–2634.

[16] *Unusual crowd activity dataset of University of Minnesota*. URL: `http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi`.

[17] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. “Object-centric auto-encoders and dummy anomalies for abnormal event detection in video”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7842–7851.

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

[19] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. “Abnormal event detection in videos using generative adversarial nets”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, pp. 1577–1581.

[20] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. “Anomaly detection and localization in crowded scenes”. In: *IEEE transactions on pattern analysis and machine intelligence* 36.1 (2013), pp. 18–32.

[21] Hung Vu, Tu Dinh Nguyen, Trung Le, Wei Luo, and Dinh Phung. “Robust anomaly detection in videos using multilevel representations”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 5216–5223.

[22] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. “Future frame prediction for anomaly detection–a new baseline”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6536–6545.

[23] Bharathkumar Ramachandra and Michael Jones. “Street Scene: A new dataset and evaluation protocol for video anomaly detection”. In: *The IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 2569–2578.

[24] Bharathkumar Ramachandra, Michael J Jones, and Ranga Raju Vatsavai. “A Survey of Single-Scene Video Anomaly Detection”. In: *arXiv preprint arXiv:2004.05993* (2020).

[25] Zachary C Lipton and Jacob Steinhardt. “Troubling trends in machine learning scholarship”. In: *Queue* 17.1 (2019), pp. 45–77.

[26] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. “Learning temporal regularity in video sequences”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 733–742.

[27] Gábor Melis, Chris Dyer, and Phil Blunsom. “On the state of the art of evaluation in neural language models”. In: *arXiv preprint arXiv:1707.05589* (2017).

[28] Keval Doshi and Yasin Yilmaz. “Any-Shot Sequential Anomaly Detection in Surveillance Videos”. In: *arXiv preprint arXiv:2004.02072* (2020).

[29] Marc Roig Vilamala, Liam Hiley, Yulia Hicks, Alun Preece, and Federico Cerutti. “A pilot study on detecting violence in videos fusing proxy models”. In: *2019 22th International Conference on Information Fusion (FUSION)*. IEEE. 2019, pp. 1–8.

[30] Guillermo A Martınez-Mascorro, José R Abreu-Pederzini, José C Ortiz-Bayliss, and Hugo Terashima-Marın. “Suspicious Behavior Detection on Shoplifting Cases for Crime Prevention by Using 3D Convolutional Neural Networks”. In: *arXiv preprint arXiv:2005.02142* (2020).

[31] Sinno Jialin Pan and Qiang Yang. “A survey on transfer learning”. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[32] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. “Learning spatiotemporal features with 3d convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[33] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. “Temporal segment networks for action recognition in videos”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.11 (2018), pp. 2740–2755.

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

[35] Joao Carreira and Andrew Zisserman. “Quo vadis, action recognition? a new model and the kinetics dataset”. In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6299–6308.

[36] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[37] Urvi Gianchandani, Praveen Tirupattur, and Mubarak Shah. “Weakly-Supervised Spatiotemporal Anomaly Detection”. In: *University of Central Florida Center for Research in Computer Vision REU* (2019).

[38] Kun Liu and Huadong Ma. “Exploring Background-bias for Anomaly Detection in Surveillance Videos”. In: *Proceedings of the 27th ACM International Conference on Multimedia*. 2019, pp. 1490–1499.

[39] Federico Landi, Cees GM Snoek, and Rita Cucchiara. “Anomaly Locality in Video Surveillance”. In: *arXiv preprint arXiv:1901.10364* (2019).

[40] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, and Reinhard Klette. “Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes”. In: *IEEE Transactions on Image Processing* 26.4 (2017), pp. 1992–2004.

[41] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. “Face detection without bells and whistles”. In: *European conference on computer vision*. Springer. 2014, pp. 720–735.

[42] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. “Object detection in 20 years: A survey”. In: *arXiv preprint arXiv:1905.05055* (2019).

[43] Paul Viola and Michael J Jones. “Robust real-time face detection”. In: *International journal of computer vision* 57.2 (2004), pp. 137–154.

[44] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. IEEE. 2005, pp. 886–893.

[45] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[46] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[47] Joseph Redmon and Ali Farhadi. “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).

[48] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. “Ssd: Single shot multibox detector”. In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[49] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[50] A Enis Çetin, Kosmas Dimitropoulos, Benedict Gouverneur, Nikos Grammalidis, Osman Günay, Y Hakan HabiboÇ§lu, B UÇ§ur Töreyin, and Steven Verstockt. “Video fire detection–review”. In: *Digital Signal Processing* 23.6 (2013), pp. 1827–1843.

[51] Mahdi Hashemzadeh and Alireza Zademehdi. “Fire detection for video surveillance applications using ICA K-medoids-based color model and efficient spatio-temporal visual features”. In: *Expert Systems with Applications* 130 (2019), pp. 60–78.

[52] Pasquale Foggia, Alessia Saggese, and Mario Vento. “Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion”. In: *IEEE transactions on circuits and systems for video technology* 25.9 (2015), pp. 1545–1556.

[53] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C. Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty, Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, Jenq-Neng Hwang, and Siwei Lyu. “The 2018 NVIDIA AI City Challenge”. In: *Proc. CVPR Workshops*. 2018, pp. 53–60.

[54] Yan Xu, Xi Ouyang, Yu Cheng, Shining Yu, Lin Xiong, Choon-Ching Ng, Sugiri Pranata, Shengmei Shen, and Junliang Xing. “Dual-mode vehicle motion pattern learning for high performance road traffic anomaly detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 145–152.

[55] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. "The 4th AI City Challenge". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020, pp. 2665–2674.

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[57] Erik Linder-Norén. *PyTorch-YOLOv3*. `https://github.com/eriklindernoren/PyTorch-YOLOv3`. Accessed: 2020-07-01.

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

[59] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[60] Eitan Kosman. *Pytorch implementation of Real-World Anomaly Detection in Surveillance Videos*. `https://github.com/ekosman/AnomalyDetectionCVPR2018-Pytorch`. Accessed: 2020-10-10.

[61] Theano Development Team. "Theano: A Python framework for fast computation of mathematical expressions". In: *arXiv e-prints* abs/1605.02688 (May 2016). URL: `http://arxiv.org/abs/1605.02688`.

[62] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H. Li, and Ge Li. *Graph Convolutional Label Noise Cleaner: Train a Plug-and-play Action Classifier for Anomaly Detection*. `https://github.com/jx-zhong-for-academic-purpose/GCN-Anomaly-Detection`. Accessed: 2020-09-01.

[63] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[64] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[65] Joseph Redmon. *YOLO: Real-Time Object Detection*. https://pjreddie.com/darknet/yolo/. Accessed: 2020-07-01. 2018.

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[67] Yingying Li, Jie Wu, Xue Bai, Xipeng Yang, Xiao Tan, Guanbin Li, Shilei Wen, Hongwu Zhang, and Errui Ding. *Multi-granularity tracking with modularlized components for unsupervised vehicles anomaly detection*. https://github.com/PaddlePaddle/Research/tree/master/CV/AICity2020-Anomaly-Detection. Accessed: 2020-10-10.

[68] Shaoqing Ren, Kaimeng He, Ross Girshik, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* (2015), p. 9199.

[69] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. "Simple Online and Realtime Tracking with a Deep Association Metric". In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962.

[70] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. *Simple Online and Realtime Tracking with a Deep Association Metric*. https://github.com/nwojke/deep_sort. Accessed: 2020-08-01. 2017.

[71] Zoran Zivkovic. "Improved adaptive Gaussian mixture model for background subtraction". In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 2. IEEE. 2004, pp. 28–31.

[72] Shuai Bai, Zhiqun He, Yu Lei, Wei Wu, Chengkai Zhu, Ming Sun, and Junjie Yan. "Traffic Anomaly Detection via Perspective Map based on Spatial-temporal Information Matrix." In: *CVPR Workshops*. 2019, pp. 117–124.

Appendices

Appendix 1 - Relaxed Traffic Anomaly Detection Parameters

Algorithm	Parameter	Li et al [9]	Ours	Meaning
Background Modeling	Frame rate	30	6	Frame rate for extracting background
Tube construction	Score threshold	0.5	0.3	Confidence at which to include detections in tube construction
	Mask threshold	0.5	0.3	IOU with mask required for detection to be included.
	Link IOU Threshold	0.4	0.3	The IOU threshold to combine two detections into a tube.
	IOU Threshold	0.8	0.6	The IOU threshold to combine two detections into a tube going backward
	Time Threshold	100	10	The minimum length of a tube
	Skip Threshold	10	41	Upper limit on consecutive frames without detection for tube.
Box Tracking	Span Threshold	3000	40	Cross Tube Fusion threshold
	Merge Threshold	7000	7000	Temporal fusion threshold
Pixel Level Tracking	Frame Rate	10	10	
	Len Time Thresh	40	5	Minimum abnormal duration (seconds)
	Suspicious Time Period	20	3	Minimum suspiciou abnormal duration (seconds)
	Detection Threshold	3	1	The normal-suspicious state transition threshold
	No Detection Threshold	3	3	The suspicious/abnormal-normal state transition threshold
	Anomaly Score Theshold	0.7	0.3	Anomely score threshold
	Bounding Box Threshold	0.7	0.3	Detection confidence threshold
	Traceback Threshold	400	400	Backtrack time threshold
	IOU Threshold	0.1	0.1	IOU score threshold
	Ration Thres	0.6	0.6	Relaxed constraint satisfaction ratio

Table 13. The changes to parameter values in order to make the road accident detection from Li et. al's [9] work more applicable to the UCF-Crime dataset [2]

Appendix 2 - Fire and Smoke, and Traffic Anomaly Detection Validation Set Results

Model		MLP			SGD		
Initial Learning Rate		0.1	0.01	0.001	0.1	0.01	0.001
Base Detector	Representation						
Su	Sult1	0.0	0.134	0.134	0.134	0.134	0.134
	Sult1Fi1	0.1528	0.1121	0.1169	0.1169	0.1169	0.1169
	Sult32	0.0	0.1675	0.1982	0.1352	0.1333	0.1356
	Sult32Fi1	0.132	0.0	0.1288	0.1337	0.1373	0.1341
Lv	Lv2	0.1993	0.0	0.2242	0.2108	0.211	0.2108
	Lv2Fi1	0.0	0.3236	0.3452	0.2101	0.2108	0.2129
	Lv32	0.0	0.2885	0.2664	0.2936	0.3041	0.3011
	Lv32Fi1	0.2959	0.2698	0.314	0.3007	0.2901	0.3011

Table 14. TPR at FPR 0.1 for different methods of combining fire and smoke detection with the baseline anomaly detection works of Sultani et al. [2] and Lv et al [7].

Model		MLP			SGD		
Initial Learning Rate		0.1	0.01	0.001	0.1	0.01	0.001
Base Detector	Representation						
Su	Sult1	0.0	0.0	0.134	0.134	0.134	0.134
	Sult1Acc1	0.1084	0.0976	0.134	0.1009	0.1016	0.1016
	Sult32	0.1622	0.1986	0.1475	0.1333	0.1352	0.1327
	Sult32Acc1	0.0	0.1271	0.0271	0.1365	0.1267	0.131
Lv	Lv2	0.0	0.0	0.0	0.211	0.2112	0.2108
	Lv2Acc1	0.2203	0.2463	0.2301	0.2144	0.2123	0.2127
	Lv32	0.2565	0.2845	0.2829	0.2934	0.291	0.2946
	Lv32Acc1	0.3263	0.3141	0.3457	0.3098	0.2922	0.3121

Table 15. TPR at FPR 0.1 for different methods of combining traffic anomaly detection with the baseline anomaly detection works of Sultani et al. [2] and Lv et al [7].

Model		MLP			SGD		
Initial Learning Rate		0.1	0.01	0.001	0.1	0.01	0.001
Base Detector	Representation						
Su	Sult1	0.0	0.0	0.0	0.134	0.134	0.134
	Sult1Fi1Acc1	0.0	0.2111	0.1269	0.1273	0.121	0.1248
	Sult32	0.1202	0.0223	0.1167	0.1337	0.1359	0.1322
	Sult32Fi1Acc1	0.0	0.1137	0.1192	0.1345	0.1328	0.1268
Lv	Lv2	0.224	0.0	0.2227	0.211	0.2112	0.2112
	Lv2Fi1Acc1	0.0	0.2677	0.3477	0.2086	0.2127	0.2156
	Lv32	0.285	0.2828	0.3106	0.2959	0.3078	0.3012
	Lv32Fi1Acc1	0.3279	0.4035	0.3925	0.328	0.3213	0.3044

Table 16. TPR at FPR 0.1 for different methods of combining fire and smoke detection and traffic anomaly detection with the baseline anomaly detection works of Sultani et al. [2] and Lv et al [7].