Robust Anomaly Detection in CCTV Surveillance

Thomas Scholtz
21681147@sun.ac.za
Supervisor: Dr Mkhuseli Ngxande
Division of Computer Science
University Of Stellenbosch

Abstract

Given the vast amount of publicly available CCTV surveillance and the capabilities of modern computer vision algorithms, the task of automatic anomaly detection is due to be solved in the near future. A solution that is competent over the large problem domain requires a certain level of sophistication such that it can replicate the contextual understanding of a human monitor. It is hypothesised that a single approach to anomaly detection can not be expected to perform both low-level and high-level monitoring of video frames which is required for robust anomaly detection. This paper proposes a solution to the anomaly detection problem in the form of a consensus framework that combines inputs from three sources to provide a final verdict on the perceived degree of anomaly contained in a video. The first approach, later introduced as the base model, is an implementation of previous work in anomaly detection that is specifically chosen for its emphasis on the learning of high-level context. The second and third are novel anomaly detection heuristics that operate on a per-frame basis i.e., with no regard for high-level context. The paper concludes with an evaluation and analysis of the three approaches and a discussion of the merit of a consensus framework. A final AUC of 0.7156 is achieved on the UCF Crime dataset; however, this result is not attributable to the consensus framework.

Keywords: anomaly detection, multiple instance learning, 3D convolutional neural networks, optical flow

1 Introduction

In 2012, only 0.5% of all data was analysed [38]. The quantity of data accumulated in the years 2015 and 2016 exceeded that of the entire previous history [38]. Currently, the ratio between unique and replicated data is projected to be between 1:10 by 2024 [38]. In the context of CCTV surveillance, this information suggests that the use of human monitors for anomaly detection is no longer realistic, nor is it necessary. This is for two reasons in particular: first, there is a glaring deficiency in the use of surveillance cameras due to an unworkable ratio of surveillance data to human monitoring capabilities; second, recent progress in the fields of computer vision and deep learning implies that artificial agents are capable of performing anomaly detection with competitive accuracy and full coverage of footage. These agents are frameworks that detect anomalies through various techniques which aim to learn the appearance, motion and context which characterise anomalies. In unseen footage, the unpredictability of appearance and motion, or replication of learned anomalous context, is flagged by a competent framework.

The task of real-world anomaly detection spans a variety of diverse and complex situations, each with different versions of normal and anomalous activity. In this work, the anomaly detection problem is somewhat reduced to thirteen classes of anomalous activities contained within the UCF Crime dataset [6]. The dataset consists of real-world videos captured by CCTV surveillance cameras. The setting is not constrained in any way, although the majority of instances depict roads, public walkways, residential areas, and shops. The problem statement, at the highest level, is to detect anomalies within footage by mapping sequences of frames to anomaly scores. Frames that correspond to scores that exceed a discriminative threshold are considered to be anomalous. Hypothetically, a trained model of the proposed solution is applicable to on-line anomaly detection on a vast majority of unseen CCTV footage by way of transfer learning.

The lack of a general definition for 'normal' and 'anomalous' poses a difficult challenge when attempting to develop a robust framework i.e., one that performs across the full spectrum of anomalies. Additionally, anomalous events occur infrequently in comparison to normal activities and therefore solutions should have a certain degree of sophistication to deter false positives.

Given the pressing need for artificial surveillance monitoring and the complexity that comes with developing such a system, robust real-world anomaly detection is currently an important field of research that receives a considerable amount of attention [34] [39] [25] [26] [12] [24].

This paper presents a solution in the form of a consensus between three models. Where the majority of solutions in the literature rely on a single measure of anomaly in all scenarios (provided by a single approach) [34] [25] [26] [24], the proposed solution receives input from three models/heuristics with contrasting approaches. That way, the strengths of multiple solutions can be leveraged to assist in covering the spectrum of potential anomalies. The benefits of this decision are expected to be two-fold: multiple detection techniques are at the disposal of the framework, and false positives are required to be redundant if they are to prevail.

Of the three models implemented, the most substantial approach is adopted from Sultani et al. [34] - an architecture that implements a 3D Convolutional Network for feature extraction (C3D), Artificial Neural Network (ANN), and Multiple Instance Learning (MIL) ranking loss. This approach uniquely formulates anomaly detection as a weakly supervised regression problem and achieved state-of-the-art results in 2019. The remainder of this paper will refer to this model as the base model.

Furthermore, two novel unsupervised approaches were developed during this work, namely CRAFT (Consecutive frame construction with RAFT optical flow estimation) and LKKM (Lukas Kanade K-Means pattern deviation). These approaches, which are better described as anomaly detection heuristics, serve the purpose of detecting the locality of anomalies on a fine-grained level by performing computations on a per-frame basis. CRAFT, in particular, only has a single frame look-ahead. CRAFT translates recent research in optical flow estimation into a frame construction heuristic that quantifies anomaly by reconstruction error. LKKM applies cumulative clustering to sparse optical flow vectors to provide a measure of deviation from regular patterns.

The rationale behind the structure of the consensus framework is as follows: the base model learns the context surrounding anomalies and how that differs from normal, yet anomalous seeming, footage. The expectation is that it detects anomalies on a coarse level - not necessarily with pinpoint accuracy. Thereafter, CRAFT and LKKM will be applied to suspicious sections of footage (identified by the base model) to detect

the locality of a potential anomaly. If the base model is sufficient in deterring false positives, the result is a filtered version of anomalous flags (supplied by CRAFT and LKKM) which mostly correspond to true positives. In summary, this paper makes the following contributions:

- The consensus framework - a combination of three approaches to form a robust anomaly detection strategy
- A modified version of the framework proposed by Sultani et al. [34] was implemented in Tensorflow [1] & Keras [7]. The modifications are listed:
  - modifications to the training method such that contribution from the full training set is ensured
  - cross-validation of additional video-level evaluation metrics to measure performance during training
  - an early stopping callback function for prevention of overfitting
  - experimentation with a deeper architecture and an additional loss constraint to provide improved training incentive
- CRAFT - a novel unsupervised heuristic for anomaly detection
- LKKM - a second novel unsupervised heuristic for anomaly detection
- A web application to demonstrate the performance of the anomaly detection framework on the UCF Crime dataset [6]

The results provide evidence to support the argument for the use of deep learning in anomaly detection, specifically the approach suggested by Sultani et al. [34]. Additionally, CRAFT and LKKM prove to be effective at quantifying anomalies on a fine-grained level but have the expected drawback of being over-sensitive owing to the lack of higher-level context on footage. The merit of a consensus framework is questioned as a result of the difficulties introduced during the combination of the scores of the base model, CRAFT, and LKKM.

2 Related Work

2.1 Anomaly Detection

Anomaly detection aims to quantify the degree of abnormality by assigning anomaly scores along the temporal axis of a video such that peaks in anomaly scores correspond to true anomalies. This problem statement is inherently vague because there is no prior information provided on the contents of videos - the only assumption is that the camera position is fixed.

Consider two scenarios: a rare bicycle passing through a sidewalk of pedestrians; and a road accident on a highway. It seems obvious that the second example is more anomalous than the first, given the context that humans possess. Technically, both are anomalous in the absence of this context. This comparison highlights that anomalies are difficult to describe to computers. It is not the appearance and motion aspect, but rather the contextual aspect of anomalous activity that is the source of difficulty. With that being said, deep learning approaches, such as the one implemented by the base model, allow the training data to loosely dictate the general type of anomaly to be detected by formulating the appropriate loss function.

This research focuses on anomaly detection in its most useful application to society i.e. as an alerting system to unwanted activity (accidents, crime, malice). Note that anomalies are not classified, but rather recognised as anomalous as opposed to normal.

2.2 Existing Approaches to Anomaly Detection

A common approach in the literature is to learn an idea of normal activity and judge the degree of anomaly in unseen footage on whether or not it conforms to the learned idea of normal. This approach is prevalent in anomaly detection frameworks which fall under the category of *encoder-based methods*, which train both the feature encoder and classifier simultaneously [12]. The approach is implemented with various methodologies [25] [26] [24]. All of them operate on the assumption that part of the definition of an anomaly is that it is rare and therefore not learned as part of the framework's idea of normal activity. Frame reconstruction is a popular method used in determining the similarity between an unseen frame and the idea of normal. The reconstruction error is inversely proportional to the similarity to the learned representation of normal activity and thus proportional to the contained anomaly in a video.

Specific implementations are elaborated on: Appearance-Motion Correspondence [25] learns to reconstruct deconstructed normal frames, using a Convolutional Auto-encoder, in terms of plain appearance and motion (motion represented as optical flow). In testing, anomalous frames yield high reconstruction error because the reconstruction of anomalous frames is not learned. Memory guided normality [26] takes a similar approach to Appearance-Motion Correspondence except a memory module is introduced. The memory module records many prototypical features of commonly seen items, forming a dictionary of normal features from different viewpoints. A robust collection of normal activity is learned and queried with the features of new frames to retrieve similar features which aid in reconstruction. Anomaly is quantified by reconstruction error and the distance between query features and the nearest items in the memory module.

Conceptually, there are some obvious problems with the general approach of normal frame reconstruction:

- It is required that a new concept of normal is learned for each situation because it is not anomalies that are learned but rather the absence of normal activity.
- It is difficult to account for all normal events.
- A dictionary of normal events does not adjust well to environmental changes (for example, day to night) and, as a result, a high false-positive rate manifests.

In contrast, *encoder-agnostic* methods use task-agnostic features of videos extracted from a vanilla feature-encoder [12] (e.g. C3D) to estimate anomaly scores. Most recently, a new stance on the formulation of the anomaly detection problem, falling under the encoder-agnostic category, has been developed by Sultani et al. [34]. The approach attempts to associate anomalous features with higher anomaly scores by learning a mapping between features and scores. Anomaly detection is converted into a regression problem where the weights of an ANN are the predictors and the anomaly score is the response. The complexity of this approach comes with providing appropriate incentive, in the form of a custom loss function, during training of the ANN. The work of Sultani et al. marks significant progress in encoder-agnostic anomaly detection, reflected by state-of-the-art results and further extensions contributed by Zhang et al (inner-bag score gap regularisation) [43], and Wan et al. (dynamic MIL loss and center-guided regularisation) [40]. The components introduced in the original paper [34] are elaborated on in 2.4 and 2.5.

2.3 Recurring Shortfalls

At the highest level, there are two recurring challenges faced by many anomaly detection frameworks:

- Frameworks struggle to cover the wide spectrum of potential anomalies. Each framework has its strengths and weaknesses and

therefore lacks the robustness to generalise across polar examples of anomaly. For example, obvious explosion versus subtle petty theft.
- False positives are difficult to identify. If predicted scores become saturated with false positives, a framework becomes useless.

2.4 3D Convolutional Neural Networks

Convolutional neural networks (CNNs) enable generic image description by reducing images into vectors which provide a compact representation of an image's defining features. This is achieved through a series of convolution and pooling layers which ultimately reduce the resolution of an image by extracting the most significant pixels within a receptive field [36]. These compact representations enable the application of deep learning techniques to computer vision problems by reducing model complexity and the computational expense required to process visual data.

C3D [36] refers to a specific architecture of a 3D Convolutional Neural Network (3D ConvNet) that is a simple yet effective approach for spatiotemporal feature learning which selectively attends to both motion and appearance. CNNs are typically applied in a 2-Dimensional setting i.e., each frame is processed independently. 3D ConvNets differ in the sense that frames are processed in volumes (a stack of consecutive frames). In 3D ConvNets, convolution and pooling operations are performed spatiotemporally while in 2D ConvNets they are performed only spatially. Therefore, the use of C3D is necessary to provide a general video descriptor that expresses both an image's appearance and salient motion. C3D uses a homogeneous architecture with small $3 \times 3 \times 3$ convolution kernels in all layers - this kernel size is empirically selected by the authors. With a linear SVM classifier, the video descriptor outperforms state-of-the-art methods [41] [15] on three different activity classification/scene recognition benchmarks [32] [8] [31] and is comparable on a fourth [17]. Finally, C3D's learned features are generic, compact and relatively efficient to compute.

2.5 MIL

Multiple Instance Learning (MIL) is a weakly supervised learning approach. MIL is applied in the case where instances are collected into bags and labels are only known at bag-level i.e., bags are labeled based on whether they contain a certain type of instance or not; however, no information is provided on individual instances. By repeatedly observing the features of instances in labeled bags, MIL learns to associate labels with instances and thus features. In the context of anomaly detection, bags are labeled as either normal or anomalous. Normal bags only contain normal instances whereas anomalous bags contain at least one anomalous instance. The training process learns certain features to be characteristic of anomaly by way of determining recurring features in all anomalous scenarios. The approach can also be used to learn false positives - which is impressive given the complexity of a false positive in anomaly detection and the simplicity of the concept behind MIL. Sultani et al. [34] propose this idea by employing MIL ranking loss to provide training incentive to an ANN which converts a bag of video segments into a bag of corresponding anomaly scores. Bags are processed in pairs (one anomalous and one normal) and the maximum score of each bag is used in computing loss. The loss equation incentivises the maximum distance between scores that are associated with true positive features and false-positive features.

MIL relaxes the assumption of having accurate temporal annotations of anomalies while still enabling the precise features of anomalies to be learned. This is vital in anomaly detection, where a fully supervised approach is unrealistic due to the variety of situations to be considered and the laborious task of obtaining temporal annotations on necessarily large datasets.

2.6 Optical Flow

Optical flow quantifies the relative motion of objects in video by estimating the displacement of pixels between two consecutive frames. In the case of dense optical flow, estimations are computed for each pixel in a frame. This differs from sparse optical flow where estimations are computed for key features in the frames. There have been a significant amount of recent applications of optical flow to higher-level problems in computer vision i.e., anomaly detection [23], view synthesis [44] and video prediction [20]. Traditional optical flow estimation systems formulate hand-crafted optimisation problems over the space of dense displacement fields between a pair of images [28]. In contrast, modern approaches apply deep learning techniques in the form of recurrent neural networks to iteratively learn and refine the quality of estimations [35] [27] [10]. Deep learning approaches are trained once-off to learn to produce estimations that generalise to unseen footage without the need to train the model for that specific context.

RAFT [35] is a state-of-the-art deep learning architecture for dense optical flow estimation. RAFT produces optical flow as a result of three phases of computation: feature extraction, correlation volume construction, and iterative update of predicted flow through correlation look-up. At the time of publishing, the framework achieved the best results in the field, beating the previous best by a thirty percent error reduction. RAFT also exhibits considerable computational efficiency and strong generalisation, which are of high relevance in the anomaly detection task.

The Lukas Kanade method [4] is a long-standing traditional approach for sparse optical flow. Under the assumption that optical flow is uniform over a *nxn* pixel window, optical flow is computed for that window by constructing a system of linear equations containing n^2 rows. Each equation expresses optical flow in terms of the partial derivative of pixel intensity in both the x direction and y direction, as well as the derivative on the temporal axis. The system is solved via the least-squares method for the optical flow vector which is subsequently applied to all pixels in the window. Commonly, these windows are formed around significant features of a frame (determined by Shi-Tomasi corner detection [30]), rather than the full frame - that way the Lukas Kanade method is applied to textured regions where spatial gradients are significant. This results in a large variance of values in the system of equations and linear independence between equations which ultimately realises well-defined optical flow after least squares regression.

2.7 K-Means Clustering

K-Means clustering [22] is a method of vector quantisation that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centroid), serving as a prototype of the cluster. K-Means finds optimal centroids by alternating between assigning data points to clusters based on the current centroids and choosing centroids based on the current assignment of data points to clusters. This process is an iterative minimisation of within-cluster variance (measured by Euclidean distance), with respect to centroid positions.

3 Implementation

This section details the implementation of the consensus framework's components, each of which converts videos into score profiles. The evaluation process of the framework, along with an explanation of performance metrics, is contained in 4.

3.1 Base Model

The implementation of the base model is divided into two parts: extraction of descriptive features from videos and training of an ANN to map said features to anomaly scores.

3.1.1 C3D: Feature Extraction. The C3D architecture is used for feature extraction. A version of Du Tran's [36] architecture is implemented in Keras & Tensorflow. The architecture is comprised of eight convolution layers, five pooling layers, two fully connected layers, and a softmax output layer. For the purpose of this work, only the output of the first fully connected layer, `fc6`, is relevant - the final fully connected layer, together with the softmax output, is for classification purposes. l_2 normalisation is applied to the output of `fc6` to form a final descriptor. Figure 1 contains a diagram of the implemented C3D architecture.

C3D lends itself well to transfer learning which allows for the use of pre-trained weights from the Sports-1M dataset [17]. Feature extraction is executed once-off for all videos as the features yielded are deterministic and the extraction process is computationally expensive and time-consuming. The result of the feature extraction phase is, for each video in the relevant dataset, a mapping per 16 frames to a 4096D column vector which can be used as input to the ANN.

3.1.2 ANN: Mapping Features to Scores. The ANN maps feature vectors to anomaly scores. An anomaly score is a floating-point number in the range $[0, 1]$, with higher scores correlating to a higher degree of contained anomaly within features. The ANN architecture is comprised of 3 layers. The input layer has 4096 units followed by two hidden layers, 512 and 32 units, respectively. The output layer has one unit. ReLU and Sigmoid activation functions are used for the first and last fully connected layers, respectively. Dropout regularisation is applied after each hidden layer to prevent overfitting [33].

3.1.3 MIL: Provision of Loss. The loss function dictates that training is conducted according to the following structure: videos are divided into $\mathcal{S}$ segments and all 4096D feature vectors within a segment are averaged to form a single 4096D feature vector to represent a segment. A video is now described by a $\mathcal{S} \times 4096$ array. Videos, represented by respective $\mathcal{S}$ feature vectors, feed through the ANN in batches of $2 \times \mathcal{B}$ ($\mathcal{B}$ normal, $\mathcal{B}$ anomalous). Each of the $2 \times \mathcal{B}$ videos are now represented by $\mathcal{S}$ scalars (anomaly scores) corresponding to $\mathcal{S}$ feature vectors, this $\mathcal{S}$-score representation of a video is referred to as a bag, B. Each of the $\mathcal{B}$ normal bags is paired with one of the $\mathcal{B}$ anomalous bags. The maximum score is extracted from the $\mathcal{S}$ instances in each bag of the pairing - ideally the maximum score of the anomalous bag is a true positive and that of the normal bag is a false positive. To obtain the loss, the maximum scores from each of the $\mathcal{B}$ pairings are inputted into the following loss function:

$$l(B_a, B_n) = \max(0, 1 - \max_{i \in B_a} f(S_a^i) + \max_{i \in B_n} f(S_n^i)) \tag{1}$$

where:

B_a, B_n are anomalous and normal bags, respectively.
f is the mapping from $4096D$ feature vector to anomaly score scalar.
S^i is the 4096D feature vector representing the ith segment of the relevant video.

Note that loss is only computed as a function of the maximum scores of each bag such that the difference between true and false positives is maximised. It is expected that the ANN learns weights such that the network will learn a generalised model to predict high scores for anomalous segments in positive bags, and low scores for anomalous-seeming scores in negative bags.

Additional constraints are introduced to the loss function to provide sufficient incentive with respect to the regularisation of weights and sophistication of score profiles.

A temporal smoothness constraint penalises erratic score profiles, incentivising the minimisation of differences between temporally consecutive scores:

$$t(B_a) = \lambda_1 \sum_{i}^{m=\mathcal{S}-1} (f(S_a^i) - f(S_a^{i+1}))^2 \tag{2}$$

A sparsity constraint penalises the abundant allocation of high anomaly scores, incentivising the minimisation of anomaly scores:

$$s(B_a) = \lambda_2 \sum_{i}^{m=\mathcal{S}} f(S_a^i) \tag{3}$$

The final loss function combines the above components and appends a regularisation term, extending the regression problem to a ridge regression problem:

$$L(\mathcal{W}) = l(B_a, B_n) + s(B_a) + t(B_a) + ||\mathcal{W}|| \tag{4}$$

$\mathcal{W}$ refers to the weights of the network. A shrinkage penalty is applied to the values of the weights to constrain them towards zero, thus reducing the variance of the model in exchange for a small amount of bias, ultimately improving generalisation of the model [16]. A third constraint, which incentivises large differences between minimum and maximum scores in score profiles of anomalous videos, is introduced for the purpose of experimentation on the base model in 4. The third constraint is multiplied by coefficient λ_3.

The computed loss is back-propagated for the whole batch by way of mini-batch stochastic gradient descent. The above process describes a single training batch. Training batches are executed sequentially over the dataset such that all normal and anomalous data is used. The completion of training on the full dataset, in batches, represents one training epoch. The dataset is shuffled between batches to obtain different pairings between normal and anomalous videos.

An early-stop callback function is implemented to prevent overfitting of the training instances. This function computes validation loss after a fixed number of epochs and monitors the validation loss divided by the training loss. If the metric is monotonically increasing for a certain amount of consecutive epochs, execution is halted as a precaution against overfitting.

Finally, video-level evaluation (VLE) is conducted at fixed intervals in the training such that improved feedback on the ability of the model can be obtained throughout training. VLE involves evaluating the current state of the model by considering a normal video with a full score profile below a threshold as a true negative instance; and an anomalous video with at least one score above the same threshold as a true positive. The VLE metric is represented as area under curve (AUC) [3], thereby conducting this evaluation at multiple thresholds. VLE is a lenient metric

with respect to anomalous instances; however, it provides a good approximation of the model's quality as the only aspect it does not account for is the locality of a flagged anomaly in anomalous video.

Figure 1 displays a diagram that outlines the base model's training process.

3.2 CRAFT

The CRAFT anomaly detection heuristic is developed to reflect low-level abnormalities in activity in the consensus score, where the approach of the base model is too high-level to detect such a low-level anomaly. Du Tran states that C3D is not a good descriptor for low-level vision tasks; and that *C3D is designed to complete a high-level vision task* [37], therefore the heuristic is based around RAFT's optical flow estimations instead of the already computed C3D features. CRAFT uses RAFT [35] to construct future frames by applying optical flow to current frames i.e., realising predictions of motion by literally displacing pixels according to their predicted displacement. RAFT is trained on normal data with the expectation that it will output sub-standard optical flow estimations when presented with anomalous data. CRAFT is designed to exploit RAFT's poor performance on precarious sections of video such that a significant reconstruction error can be obtained, thereby quantifying the degree of anomaly within a video.

One problem with CRAFT's initial approach was that it depended on ground truth optical flow between frames such that optical flow predictions could be evaluated. The solution to this problem draws inspiration from the prevalent reconstruction error approach appearing in previous anomaly detection frameworks [25] [26]. CRAFT's implementation is adapted such that ground truth can be obtained as the consecutive frame, the frame which is to be reconstructed. In the case that perfect optical flow is computed, an application of the optical flow to the current frame results in the consecutive frame. Therefore, the quality of prediction of optical flow can be judged as proportional to the Euclidean distance between the predicted consecutive frame and the true consecutive frame. Pseudo-code for CRAFT is displayed in Algorithm 1. Furthermore, due to the absence of ground truth optical flow for common datasets and the challenging problem that optical flow evaluation presents [2], RAFT is trained on datasets that have ground truth optical flow available, namely, KITTI [13] and MPI-Sintel [5]. The quality of each dataset is judged qualitatively by plotting the profile of reconstruction errors produced by CRAFT when applied to normal and anomalous videos. By inspection, MPI-Sintel produces better score profiles (for a subset of UCF Crime) as RAFT outputs flow estimations of higher quality, therefore resulting in less frequent peaks in CRAFT scores. RAFT trained on MPI-Sintel is used in the application of CRAFT to the test set.

3.3 LKKM

LKKM is a second novel anomaly detection approach that shares a similar purpose to CRAFT in terms of its relevance to the consensus score. However, LKKM specialises in the deviation of the trajectory of objects' paths from typical paths learned throughout the processing of the video.

Given two consecutive frames in a video and a vector of 2D significant points for which flow estimation is required, a vector of 2D points containing the calculated new positions of input features in the second frame is obtained via the Lukas-Kanade method. A data vector is formed with the initial points being the initial coordinates and the returned points being the terminal coordinates. These coordinates are normalised by dividing them by the applicable dimension of the frame (width or height).

To initialise LKKM, the data vectors of the first ten frames contribute to a partial fit of a K-Means model operating in $4D$ feature space. For each additional frame after initialisation is complete, the K-Means clustering of previous data vectors is queried to obtain the nearest centroid distance for each data vector extracted from the frame. A frame's score is computed as an average of the largest distances of data vectors to nearest cluster. Thereafter, the data vectors are accumulated into the training set for the K-Means instance and new centroids are computed. The optimal number of centroids to be used is updated throughout execution via the elbow method [18].

As a video progresses, typical trajectories of paths are repeated and stronger clusters are formed, therefore the quality of anomaly scores improves (consider the scene of a highway or walkway). The incremental improvement of the model was verified by applying LKKM to a video that repeats the same clip multiple times - the results show that the same score profile is repeated except with a significant drop in the average level of scores for each repetition.

This issue is addressed by applying a time series decomposition of the score profile of a video to separate any cyclical/seasonal trend which occurs as the quality of the model improves. The score profile is selected as the residual component of the time series decomposition i.e., changes that are not cyclical/seasonal but originate as a result of inherent noise (anomaly) in a time series. A concise description of the LKKM algorithm is provided in the pseudo-code of Algorithm 2.

4 Experiments

This section provides a detailed description of the experiments conducted to apply the proposed methodology (3) to the UCF Crime dataset.

4.1 UCF Crime Dataset

Recall that this work focuses on anomaly detection in its most useful application to society i.e. as an alerting system to unwanted activity. The UCF Crime dataset [6] is of significant relevance to public safety. The dataset contains 950 anomalous videos and 950 normal videos. The cumulative length of footage is 128 hours, with the average number of frames per video equal to 7247 (approximately 4 minutes at 30 fps). Only footage sourced from CCTV surveillance cameras is included - edited video or staged anomaly is excluded. The anomalous videos contain anomalies classified into 13 classes of dangerous/malicious/alarming activity: Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. UCF Crime provides a worthy challenge to the anomaly detection problem for the following reasons:

- The dataset contains long, untrimmed surveillance videos which truly replicate the scenario of real-world anomaly detection.
- The problem-domain of unwanted activity still leaves the anomaly detection problem under-specified, differentiating anomaly detection (regression) from activity recognition (classification), and implying that the solution should be highly generalisable.
- State-of-the-art anomaly detection frameworks, many of which are subjected to unrealistic or artificial datasets, achieve poor results on UCF Crime [21] [14].

For both anomalous and normal videos, 850 instances form the training set and 140 instances are reserved for the independent test set. Videos within the dataset are vastly different from one another and, for this reason, an evaluation on the test set is a good approximation of a test of the mapping learned on the full problem domain even though the test set is only a finite sample of the problem domain. That is, the evaluation

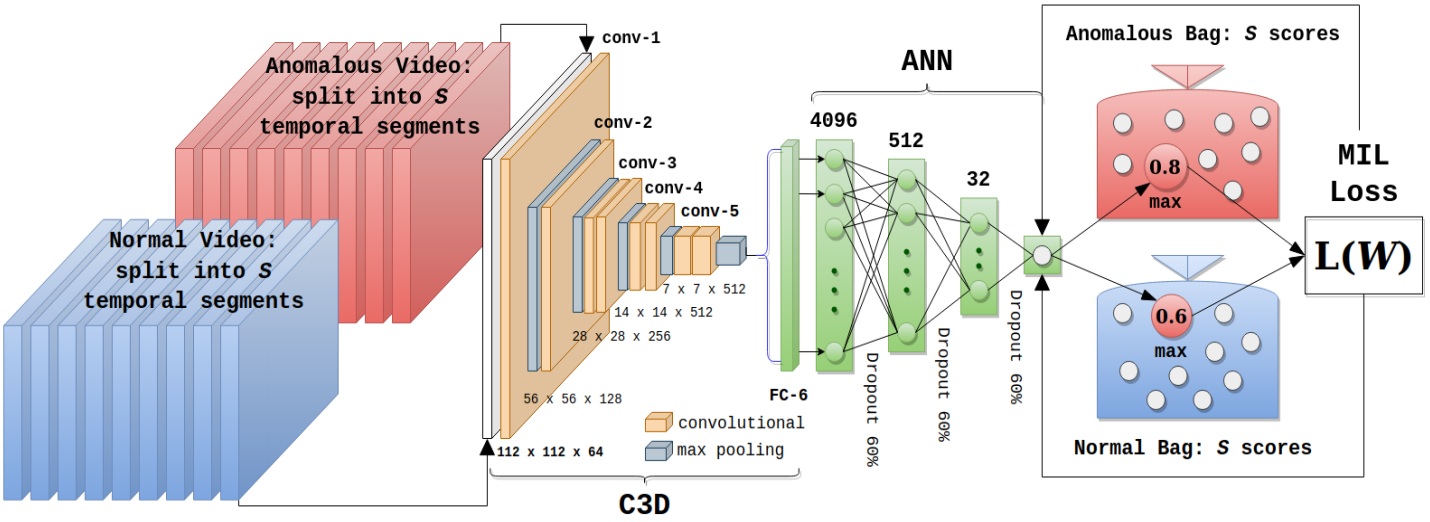

Figure 1. A flow diagram of the base model's training procedure. The pair of anomalous and normal videos represent a single training instance in a batch of $\mathcal{B}$ pairs. Note that the $L(\mathcal{W})$ refers to equation 4.

is better described as an evaluation of the ability for the framework to perform transfer learning as unseen footage is likely to be dissimilar to the training set and overfitting on the training set will result in a severe penalty to performance.

4.2 Base Model Experiments

The first phase of experimentation is concerned with training an optimal base model. Five *versions* of the base model are trained. The standard base model, discussed in previous sections, is the first version and represents a base case. Additional versions are characterised by: modification to ANN architecture, the introduction of an additional constraint in the objective function, a combination of the previous two modifications, and transfer learning with weights provided by Sultani et al. (followed by a short fine-tuning stage). Each version is trained for a number of epochs determined by the patience threshold. That is, the number of consecutive epochs for which $\rho = \frac{\mathcal{L}_V}{\mathcal{L}_T}$ is monotonically increasing before training is terminated (where $\mathcal{L}_V$ and $\mathcal{L}_T$ are VLE validation loss and VLE training loss, respectively). Three patience thresholds are tested such that performance of the model can be measured at multiple balances of bias and variance - training schedules are dictated by patience rather than epochs as it provides a better indication of the model's expected ability to generalise to unseen data. For comparison, an additional experiment is run at an over-estimated fixed number of epochs. A *version* is therefore represented by four experiments, each of which reports VLE-AUC that is cross-validated over ten folds for a certain patience threshold.

The best version, with the optimal training schedule, is identified from a table of VLE-AUC and a final experiment is run with identical settings, except that no cross-validation is performed, therefore allowing training to execute on the totality of the training set. The resulting weights represent the base model in 4.3.

4.3 Inference & Score Combination

The consensus framework is introduced during the evaluation of the three models/heuristics (base model, CRAFT, LKKM). Each approach is applied to the independent test set to yield respective score profiles. The CRAFT and LKKM heuristics are applied, as explained, to each frame in a test instance, whereas the base model obtains scores by performing inference on each video, represented by $\mathcal{S} \times 4096D$ vectors, and subsequently obtaining a score profile consisting of $\mathcal{S}$ temporal scalars.

During score combination, just as the base model scores are divided into $\mathcal{S}$ segments and averaged within segments, so too are the CRAFT and LKKM score profiles. CRAFT and LKKM scores are then standardised and filtered such that all scores below two standard deviations from the mean are set to 0. This is necessary as CRAFT and LKKM are sensitive heuristics that produce erratic score profiles. Thereafter, three score profiles, each of length $\mathcal{S}$, are combined according to the pseudo-code in Algorithm 3. Note the intention to leverage the higher-level context contained in the scores of the base model with the expectation that false positives are deterred.

4.4 Implementation details

4.4.1 Feature Extraction. Videos contain 8-bit unsigned integer pixel values in range $[0, 255]$. In line with C3D's training procedure, frames are not normalised/standardised/centered in any way. The UCF Crime dataset does not lend well to standard image/video preprocessing techniques as videos originate from vastly different settings and computing the mean frame of the dataset for standardisation does not seem to be a sensible approach. Standardising per frame or per video was considered but decided against so as to preserve temporal continuity in 16-frame video sequences which otherwise may have been disrupted by sudden shifts in means. Before feature extraction, videos are converted to RGB in three channels and resized to $112 \times 112 \times 3$ (by intercubic interpolation). Features resulting from normalised C3D are stored in 32-bit floating-point representation with scale of 4 decimals. A video's features are averaged within $\mathcal{S} = 32$ segments. Feature extraction is performed on a Tesla V100-PCIE-32GB GPU with 80 cores and 1.38GHz core clock. The total elapsed time for feature extraction on this architecture is 25 hours 41 minutes.

Version Key	ANN Architecture	Weight Initialisation	Loss Constraints
`base`	4096-512-32-1	glorot-uniform	$\lambda_1 = 8 \times 10^{-5}$, $\lambda_2 = 8 \times 10^{-5}$, $\lambda_3 = 0$
`base-arch`	4096-512-32-8-1	glorot-uniform	$\lambda_1 = 8 \times 10^{-5}$, $\lambda_2 = 8 \times 10^{-5}$, $\lambda_3 = 0$
`base-constrain`	4096-512-32-1	glorot-uniform	$\lambda_1 = 8 \times 10^{-5}$, $\lambda_2 = 8 \times 10^{-5}$, $\lambda_3 = 8 \times 10^{-4}$
`base-arch-cons`	4096-512-32-8-1	glorot-uniform	$\lambda_1 = 8 \times 10^{-5}$, $\lambda_2 = 8 \times 10^{-5}$, $\lambda_3 = 8 \times 10^{-4}$
`base-transfer`	4096-512-32-8-1	Sultani et al. [34]	$\lambda_1 = 8 \times 10^{-5}$, $\lambda_2 = 8 \times 10^{-5}$, $\lambda_3 = 8 \times 10^{-4}$

Table 1. Base Model: Experiment design.

4.4.2 Base Model. The experimental design for obtaining an optimal base model is detailed in Table 1. Each version is run at patience thresholds of 25, 50 and 100 training epochs and the additional fixed-epoch experiment is run for 8000 training epochs.

The batch size is held constant at $\mathcal{B} = 30$ i.e., 60 videos forming 30 anomalous-normal pairs. l_2 regularisation is applied to the ANN weights at penalty coefficient of ($l_2 = 0.001$). Adagrad optimiser [9] is employed with an initial learning rate of 1×10^{-3}. The dropout rate is set to 0.6 - note that this is the probability of *retaining* a unit p within a hidden layer on a given training pass i.e., units are randomly deactivated with a probability of $1 - 0.6 = 0.4$. Temporal and sparsity constraints are set to $\lambda_1 = 8 \times 10^{-5}$ and $\lambda_2 = 8 \times 10^{-5}$, respectively. On the same hardware used for feature extraction, it takes 14 hours and 43 minutes to execute the maximum length training simulation (8000 epochs). For clarity, `base-transfer`, listed in Table 1, describes a version of the base model where weights resulting from the training of Sultani et al. [34] are loaded into the standard base model. Thereafter the top layer is removed and replaced by an 8-unit fully connected layer, `fc-8`, which is followed by a single unit output, `fc-1`. During training, all layers' weights except for `fc-8` and `fc-1` are frozen i.e., not trainable. After the frozen model has been trained to exceed the patience threshold, all weights are set to trainable and 100 epochs of fine-tune training is conducted with Adadelta optimiser [42] at a learning rate of 1×10^{-5}.

4.4.3 CRAFT and LKKM. Videos are converted to the format detailed in 4.4.1, except for the case of LKKM where videos are represented in single-channel format i.e., $112 \times 112 \times 1$. Before computation of optical flow via RAFT, Gaussian blur with a kernel size of 5×5 is applied to each frame to remove excessive detail, thereby reducing the general difficulty of the optical flow problem. In line with original RAFT settings [35], CRAFT produces optical flow estimations after 24 prediction iterations. For LKKM, K-Means is initialised with 5 centroids and k-means++ centroid initialisation. The optimal number of centroids is computed every 5 iterations. A maximum of 300 significant features is extracted per frame by Shi-Tomasi corner detection [30]. The window size set for the Lukas-Kanade flow is 15×15 pixels in dimension. The number of centroids is recomputed with the elbow method every 5 frames. The maximum distance of data vector to cluster center is computed as an average over that of the largest 50 distances yielded by LKKM. Time series decomposition is performed with a period equal to 10.

4.5 Performance Metrics

The consensus score profiles of videos in the independent test set are evaluated according to the following criteria: the test set contains annotations correlating to the anomalous frames of anomalous videos. A true positive (TP) occurs in the case that at least one score within the annotated window exceeds the threshold. A false negative (FN) occurs if no score in the annotated window exceeds the threshold. In the case that there are two anomalies in one video, each is treated as an individual instance according to the same criteria. In the case that a score outside of the annotated window exceeds the threshold, provided it is an anomalous video, the score is not penalised. This is because anomalies tend to disrupt the environment, often eliciting further anomalous activity which can not be objectively defined as anomalous or not. Instead, the deterrence of false positives is tested on the normal videos of the test set.

For normal videos, the requirement for a true negative (TN) is that not one score exceeds the threshold. For a false positive (FP) to occur, at least one score must exceed the threshold. A full normal video is considered to be one normal instance.

The decision boundary of the mapping between features and scores is an arbitrary constant in the range $[0, 1]$ and, provided a discriminative threshold is fixed for a complete evaluation process, a specific discriminative threshold (such as 0.5) does not need to be manually chosen. The threshold referred to in the criteria above is a variable in the range $[0.1, 0.9]$.

At each threshold, the confusion matrix is computed and the true positive rate (TPR) and false-positive rate (FPR) are subsequently computed as:

$$TPR = \frac{TP}{TP + FN} = \frac{\text{\#true-positive instances}}{\text{\#anomalous instances}} \tag{5}$$

$$FPR = \frac{FP}{FP + TN} = \frac{\text{\#false-positive instances}}{\text{\#normal instances}} \tag{6}$$

Thereafter, a frame-based receiver operating characteristic (ROC) curve is employed. Area under curve (AUC) [3] is used to evaluate the performance of the framework, as in earlier research in anomaly detection [19]. Equal error rate (EER) [19] is disregarded because it does not accurately measure anomaly, especially if only a small fraction of a long video contains anomalous behavior.

5 Results

	AUC			
Version Key	$P = 25$	$P = 50$	$P = 100$	$E = 8000$
`base`	0.7251	0.7235	0.7356	0.7808
`base-arch`	0.7825	0.8233	0.8154	0.7176
`base-constrain`	0.7525	0.8012	0.8242	0.7793
`base-arch-cons`	0.8031	0.8267	0.8148	0.7413
`base-transfer`	0.7535	0.7756	0.7693	0.6982
with fine-tuning	0.7767	0.7945	0.8015	0.7424

Table 2. Base Model: Cross-validated AUC at patience threshold P and number of training epochs E.

Table 2 reports VLE-AUC for each version of the base model, trained at the specified patience threshold (or number of epochs in case of the last column). The general quality of the base model's predictions, irrespective of version, is relatively high in comparison to AUC quoted

in previous works [1]. This is largely owing to the fact that VLE ignores precise locality of an anomaly; however, that is the only factor that VLE does not take into account and any penalty that the base model incurred during strict evaluation is as a direct result of inability to perform accurate locality of anomaly. The highest VLE-AUC is achieved by `base-arch-cons` at $P = 50$, suggesting that the base model benefits from a deeper architecture and/or further specification of training incentive. However, note that there is a clear tendency for deeper architectures to overfit given an extended number of training epochs (`base-arch` and `base-arch-cons` at $E = 8000$). Interestingly, the standard `base` version reports better AUC for $E = 8000$ than for training simulations conducted with patience thresholds. It is plausible that the significant levels of dropout result in a convoluted profile of ρ ratios, causing training to converge to local minimas which require an extended amount of training epochs to escape.

Strict evaluation of the base model with respect to annotations yields the blue ROC curve depicted in Figure 2, corresponding to AUC of 0.7156. This is a quality result from the base model considering the complexity of the problem, which is highlighted by the AUC of 0.5 that results from application of a binary SVM classifier to the anomaly detection problem. Sultani et al. [34] suggests the use of the binary SVM classifier as a generic representative of traditional classification methods [2].

With that being said, the decline from VLE-AUC to strict evaluation AUC is approximately 10%. This result confirms the expectation that the base model may struggle to provide fine-grained anomaly detection over a large problem domain, thereby reinforcing the case for combining the base model with additional CRAFT and LKKM heuristics. The

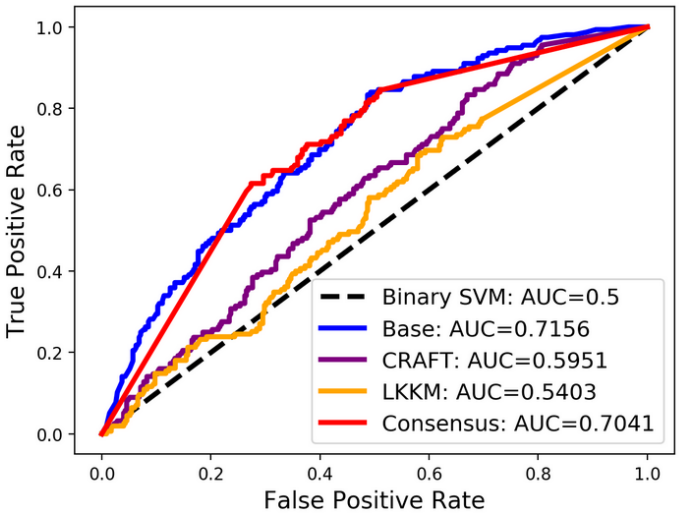

Figure 2. Receiver operating characteristic (ROC) comparison of binary SVM classifier, base model, CRAFT, LKKM, and consensus framework.

CRAFT and LKKM heuristics perform as intended i.e., with an excessive amount of flaggings, although often including true anomalous flags. The purple and orange curves in Figure 2 correspond to ROC for CRAFT and LKKM, respectively. These curves represent CRAFT and LKKM as stand-alone anomaly detection solutions. Both CRAFT (AUC=0.5951) and LKKM (AUC=0.5403) out-perform the binary SVM classifier, thereby indicating that there is merit to the methods. Figure 3 depicts an analysis of score profiles provided by CRAFT and LKKM on anomalous video `Explosion011`. Notice how CRAFT, denoted by yellow scores, yields a peak in reconstruction error by exploiting poor optical flow prediction from RAFT in the case that a white cloud of smoke suddenly appears against a bus in the frame. Thereafter, a tuk-tuk swerves left to avoid the commotion and LKKM, denoted by red scores, yields a peak in its scores due to a significant deviation from relatively repetitive patterns of motion (road traffic). Both CRAFT and LKKM provide monitoring of frames at a level of detail that can not be expected from the base model; however, the heuristics suffer from the type of scores seen earlier on the temporal axis of Figure 3 where, in this case, CRAFT produced a peak in scores due to a vehicle suddenly coming into view at the bottom of the frame. Given the high FPR and high TPR which characterise the CRAFT and LKKM curves, it is evident that the competency of the heuristics is attributable to their anomaly detection ability and not at all to their ability to deter false positives. This behavior prevents the use of CRAFT or LKKM as a stand-alone anomaly detection solution as an excessive amount of false positives is guaranteed.

Finally, the base model is combined with the CRAFT and LKKM heuristics to form the consensus framework. Ideally, the consensus framework is able to combine the scores in such a way that the merit of all three approaches is reflected in the score profile. The red curve in Figure 2 depicts the performance of the consensus framework. By inspection, the consensus framework does not provide a clear improvement in anomaly detection in comparison to the original base model. A comparison of AUC (consensus: 0.7041, base: 0.7156) indicates that the combination of approaches results in an adverse effect on overall anomaly detection ability. The combination of the three score profiles is the point of failure in the consensus framework. Without prior information on whether a particular video may contain anomaly or not, the task of optimal combination of score profiles is surprisingly non-trivial. The score combination technique applied in this case (detailed in Algorithm 3) was chosen via an empirical comparison between various alternatives. The selected technique prioritises low scores provided by the base model for normal videos as this feature of the base model is paramount to robust anomaly detection. Under this score combination technique, the CRAFT and LKKM heuristics play a supporting role where the base model's scores are only amplified by the heuristics'; however, in the case that the base model produces scores below a minimum threshold, successful anomaly detection from CRAFT and LKKM is discarded (refer to `Shooting022` in Figure 13). With a more elaborate score combination process, the consensus framework may be able to better utilise the heuristics for an improvement in overall anomaly detection ability.

The optimal anomaly detection performance amongst all experiments is displayed in the confusion matrix of Figure 4 - this evaluation is performed at a discriminative threshold of 0.4. The source of the result, `base-arch-cons`, confirms that there is in fact utility to be gained from an additional layer of abstraction and the addition of a constraint to loss specification such that increased 'peakedness' is incentivised by the training process of the base model. Furthermore, the number of true positive and true negative cases is well balanced, indicating a certain level of sophistication to the base model's approach in that sufficient context is learned to deter false positives.

Figures 5 to 14 present a selection of qualitative results from the consensus framework. The score profiles (base model, CRAFT & LKKM, consensus framework) depicted for each instance are those used in evaluation of the particular instance to arrive at the ROC curves and confusion matrix. In particular, note in Figure 5 how base model scores

[1] 0.5060 [14], 0.6551 [21]

[2] AUC of 0.5 in binary classification implies a model which is equivalent to random class selection at $P(ANOMALOUS) = 0.5$ and $P(NORMAL) = 0.5$

are amplified by CRAFT scores such that a clear indication of anomaly is provided by the consensus framework. Also, note in 5 and 6 that base model scores which are below a threshold are mapped to zero scores in the consensus score profile.

Figures 5, 6, 7, 8, 9 provide a demonstration of CRAFT and LKKM's tendency to produce an excessive amount of false positives, although the frequency of true positives should also be noted. Referring back to

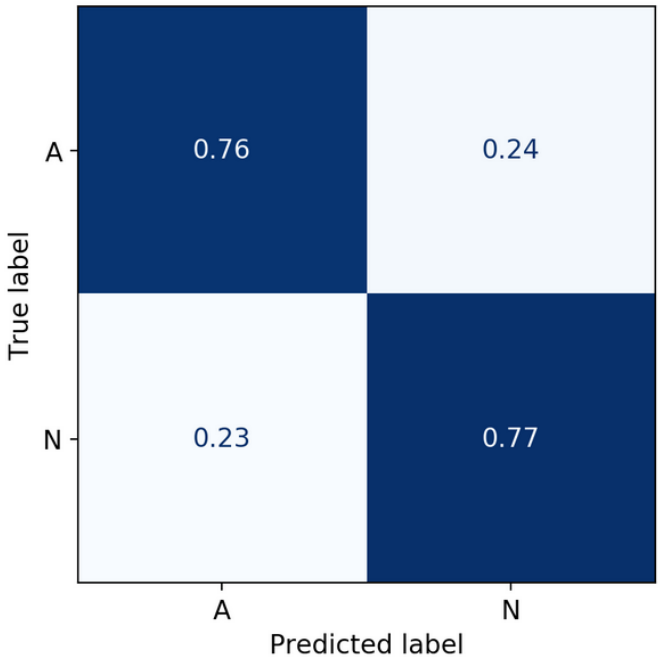

Figure 3. Optimal confusion matrix produced by anomaly detection framework at the discriminative threshold of 0.4. 'A' corresponds to anomalous instances and 'N' to normal.

frequent shortfalls of anomaly detection frameworks, discussed in 2.3, recall that a prevalent problem of anomaly detection is that frameworks cannot generalise across polar examples of anomaly. The consensus scores in Figure 6 and Figure 9 demonstrate instances where the consensus framework successfully detects anomaly in opposing classes, namely explosion and fighting. Additionally, in the fighting instance, notice that the contribution by CRAFT and LKKM reinforce the detection of the specific anomaly. Figures 10, 11, 12 demonstrate the value of the base model in deterring false positives which may arise in normal instances. In particular, the base model is able to produce a score profile of zeros in Figure 10 which corresponds to a score profile for nighttime CCTV surveillance. The fact that the base model has not associated anomalous activity with nighttime activities is a positive reflection on the depth of context learned by the base model.

Figures 13 and 14 show failure cases of the consensus framework. In particular, Figure 13 shows a case where CRAFT and LKKM accurately detect anomaly however, the base model produces an inaccurate score profile. The score combination process discards the scores from CRAFT and LKKM to prioritise the zero-scores from the base model and the heuristics are unable to contribute to successful anomaly detection. This demonstrates the heavy dependency placed on the base model by a score combination process which may seem somewhat rudimentary in some instances. Figure 14 demonstrates a false positive produced by the base model which may be attributable to the unusual camera angle (birds-eye) of `Normal041`. The base model would not have been trained on footage from this perspective and therefore it may misinterpret appearance and motion to be similar to that of anomalous activity.

The respective score profiles for videos in the test set are available at: https://share.streamlit.io/tomschdev/cctvanomalydetection/demo/src/pred_evaluation.py

6 Conclusion

This paper investigates automated anomaly detection from three perspectives. First, a modified version of a prevalent deep learning approach, devised by Sultani et al. [34], is implemented and evaluated with thorough experimentation on a challenging dataset. The results indicate that this approach presents a realistic solution to anomaly detection, confirming that deep learning approaches are indispensable to the development of state-of-the-art anomaly detection frameworks.

Second, the usefulness of low-level anomaly detection heuristics, CRAFT and LKKM, is investigated. Through qualitative analysis of a selection of score profiles and quantitative analysis by AUC of respective ROC curves, CRAFT and LKKM are proven to be of use in an anomaly detection setting; however, the heuristics are sensitive and erratic which implies that the underlying concepts are not necessarily useful in isolation but need to be combined with a more sophisticated solution.

Finally, the consensus framework attempts to combine the base model, CRAFT, and LKKM into a single, robust solution. Qualitative analysis of score profiles demonstrates instances where this concept is useful however, the decline in AUC from the base model to the consensus framework shows that the consensus framework ultimately fails to add value to the overall anomaly detection ability of the base model.

7 Future Work

The UCF Crime dataset contains videos that include impurities such as cut-scenes, branding, and black-screen introductions that contain text. Technically, the base model is equipped to deal with these impurities by recognising such frames as false positive instances that are naturally assigned lower scores by the loss equation. However, this only holds under the assumption that the impurities are well represented in the set of normal videos. Nonetheless, it would undoubtedly benefit the base model if such impurities were removed so that the learned idea of anomalous activity is more accurate.

The performance of the consensus framework suggests that a single source of anomaly scores is preferred over a combination of scores. Additionally, CRAFT and LKKM do not receive enough incentive to correspond peaks in scores with true anomalies. These two observations suggest that the utility provided by CRAFT and LKKM can only be accessed if their methodologies can be integrated with the base model such that heuristics can be refined/trained based on the quality of scores produced, and the score combination problem can be avoided. Furthermore, the base model is not trained end-to-end as pre-trained weights are substituted into the C3D implementation. An end-to-end version of the anomaly detection framework should be a priority in future work as it would enable the learning of specific low-level features which correspond to anomaly. By improving the low-level monitoring abilities of the base model, the need for low-level heuristics such as CRAFT and LKKM may be invalidated altogether, therefore end-to-end training of

the base model should be investigated before considering integration between CRAFT/LKKM and the base model.

The base model can be extended to perform activity classification, rather than just event detection, by removing the single-unit top layer of the ANN and attaching a 13-unit output layer. Each of the 13 units will correspond to one of the 13 anomaly classes of UCF Crime and, by employing a softmax activation function, the output activations can be interpreted as the probability of activity belonging to the corresponding class. Perhaps the output layer includes 14 units, with the extra unit corresponding to normal activity.

Finally, a lower priority enhancement is the implementation of active learning i.e., uncertainty sampling [29] or sensitivity analysis for selective learning [11], for the ANN of the base model. This enhancement allows for more informative patterns, which lie closer to decision boundaries, to be prioritised in training - a useful function given a large, dissimilar dataset. Note that the implementation would require predictions of the base model to be strictly evaluated at a discriminative threshold during training as active learning applies to classification problems only.

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. Tensorflow: A system for large-scale machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283). TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J. and Szeliski, R., 2011. A database and evaluation methodology for optical flow. International journal of computer vision, 92(1), pp.1-31.

[3] Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition, 30(7), pp.1145-1159.

[4] Bouguet, J.Y., 2001. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel corporation, 5(1-10), p.4.

[5] Butler, D., Wulff, J., Stanley, G. and Black, M., 2012. MPI-Sintel optical flow benchmark: Supplemental material. In MPI-IS-TR-006, MPI for Intelligent Systems (2012. Citeseer.

[6] Chen, C. and others, 2018. UCF. Available at: https://www.crcv.ucf.edu/chenchen

[7] Chollet, F. and others, 2015. Keras. Available at: https://github.com/fchollet/keras.

[8] Derpanis, K.G., Lecce, M., Daniilidis, K. and Wildes, R.P., 2012, June. Dynamic scene understanding: The role of orientation features in space and time in scene classification. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1306-1313). IEEE.

[9] Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic optimisation. Journal of Machine Learning Research, 12, 2121–2159. Retrieved from http://jmlr.org/papers/v12/duchi11a.html

[10] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D. and Brox, T., 2015. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 2758-2766).

[11] Engelbrecht, A.P., 2001. Sensitivity analysis for selective learning by feedforward neural networks. Fundamenta Informaticae, 46(3), pp.219-252.

[12] Feng, J.C., Hong, F.T. and Zheng, W.S., 2021. Mist: Multiple instance self-training framework for video anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14009-14018).

[13] Geiger, A., Lenz, P., Stiller, C. and Urtasun, R., 2013. Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11), pp.1231-1237.

[14] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K. and Davis, L.S., 2016. Learning temporal regularity in video sequences. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 733-742).

[15] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014, November. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678).

[16] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

[17] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).

[18] Kodinariya, T.M. and Makwana, P.R., 2013. Review on determining number of Cluster in K-Means Clustering. International Journal, 1(6), pp.90-95.

[19] Li, W., Mahadevan, V. and Vasconcelos, N., 2013. Anomaly detection and localization in crowded scenes. IEEE transactions on pattern analysis and machine intelligence, 36(1), pp.18-32.

[20] Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X. and Yang, M.H., 2018. Flow-grounded spatial-temporal video prediction from still images. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 600-615).

[21] Lu, C., Shi, J. and Jia, J., 2013. Abnormal event detection at 150 fps in matlab. In Proceedings of the IEEE international conference on computer vision (pp. 2720-2727).

[22] Likas, A., Vlassis, N. and Verbeek, J.J., 2003. The global k-means clustering algorithm. Pattern recognition, 36(2), pp.451-461.

[23] Liu, W., Luo, W., Lian, D. and Gao, S., 2018. Future frame prediction for anomaly detection–a new baseline. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6536-6545).

[24] Liu, W., Luo, W., Lian, D. and Gao, S., 2018. Future frame prediction for anomaly detection–a new baseline. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6536-6545).

[25] Nguyen, T.N. and Meunier, J., 2019. Anomaly detection in video sequence with appearance-motion correspondence. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1273-1283).

[26] Park, H., Noh, J. and Ham, B., 2020. Learning memory-guided normality for anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14372-14381).

[27] Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X. and Zha, H., 2017, February. Unsupervised deep learning for optical flow estimation. In Thirty-First AAAI Conference on Artificial Intelligence.

[28] Shah, S.T.H. and Xuezhi, X., 2021. Traditional and modern strategies for optical flow: an investigation. SN Applied Sciences, 3(3), pp.1-14

[29] Sharma, M. and Bilgic, M., 2017. Evidence-based uncertainty sampling for active learning. Data Mining and Knowledge Discovery, 31(1), pp.164-202.

[30] Shi, J., 1994, June. Good features to track. In 1994 Proceedings of IEEE conference on computer vision and pattern recognition (pp. 593-600). IEEE.

[31] Shroff, N., Turaga, P. and Chellappa, R., 2010, June. Moving vistas: Exploiting motion for describing scenes. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 1911-1918). IEEE.

[32] Soomro, K., Zamir, A.R. and Shah, M., 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.

[33] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp.1929-1958.

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

[35] Teed, Z. and Deng, J., 2020, August. Raft: Recurrent all-pairs field transforms for optical flow. In European conference on computer vision (pp. 402-419). Springer, Cham.

[36] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M., 2015. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

[37] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M., 2016. Deep end2end voxel2voxel prediction. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 17-24).

[38] Vermesan, O., EisenHauer, M., Serrano, M., Guillemin, P., Sundmaeker, H., Tragos, E.Z., Valino, J., Copigneaux, 2018. The next generation internet of things–hyperconnectivity and embedded intelligence at the edge. Next Generation Internet of Things. Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation.

[39] Wang, Y.K., Fan, C.T., Cheng, K.Y. and Deng, P.S., 2011, July. Real-time camera anomaly detection for real-world video surveillance. In 2011 International Conference on Machine Learning and Cybernetics (Vol. 4, pp. 1520-1525). IEEE.

[40] Wan, B., Fang, Y., Xia, X. and Mei, J., 2020, July. Weakly supervised video anomaly detection via center-guided discriminative learning. In 2020 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). IEEE

[41] Wang, H. and Schmid, C., 2013. Action recognition with improved trajectories. In Proceedings of the IEEE international conference on computer vision (pp. 3551-3558).

[42] Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. Retrieved from http://arxiv.org/abs/1212.5701

[43] Zhang, J., Qing, L. and Miao, J., 2019, September. Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection. In 2019 IEEE International Conference on Image Processing (ICIP) (pp. 4030-4034). IEEE.

[44] Zhou, T., Tulsiani, S., Sun, W., Malik, J. and Efros, A.A., 2016, October. View synthesis by appearance flow. In European conference on computer vision (pp. 286-301). Springer, Cham.

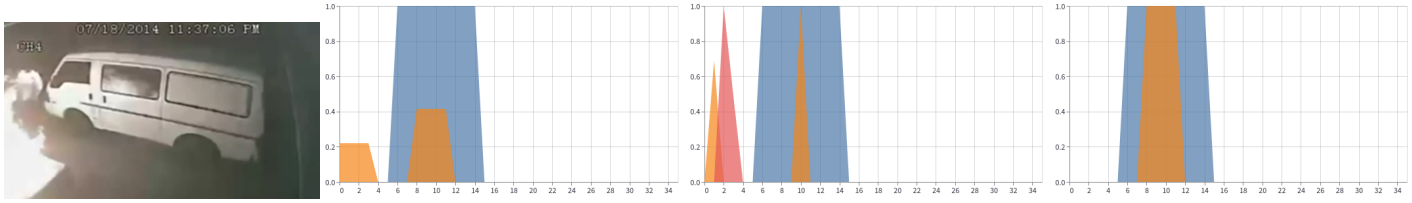

Figure 4. Arson009: Left: video instance. Second from left: base model score profile (orange) with annotation (blue). Second from right: CRAFT (orange) and LKKM (red) score profiles with annotation (blue). Right: consensus framework score profile (orange) with annotation (blue). The score profiles of instances displayed in Figures 6 to 14 follow identical order.

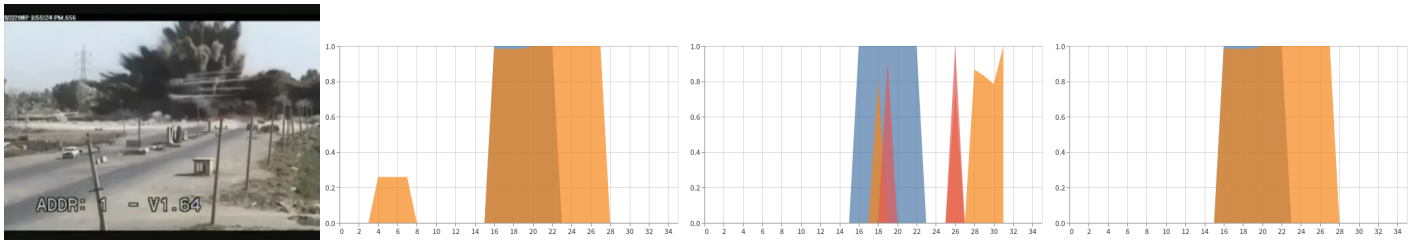

Figure 5. Explosion008

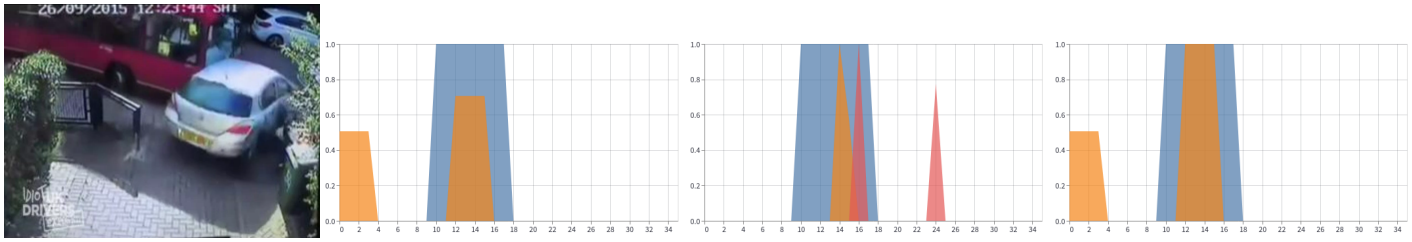

Figure 6. RoadAccidents010

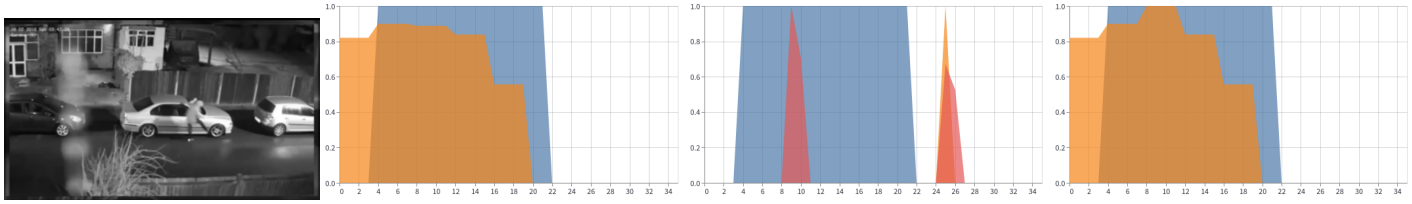

Figure 7. Vandalism007

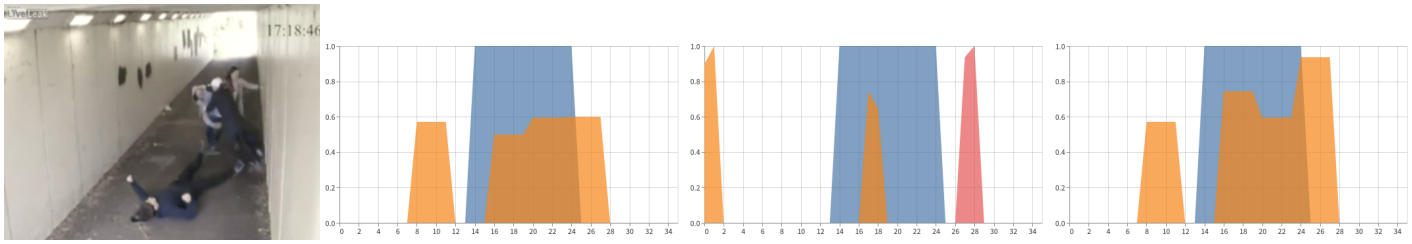

Figure 8. Fighting033

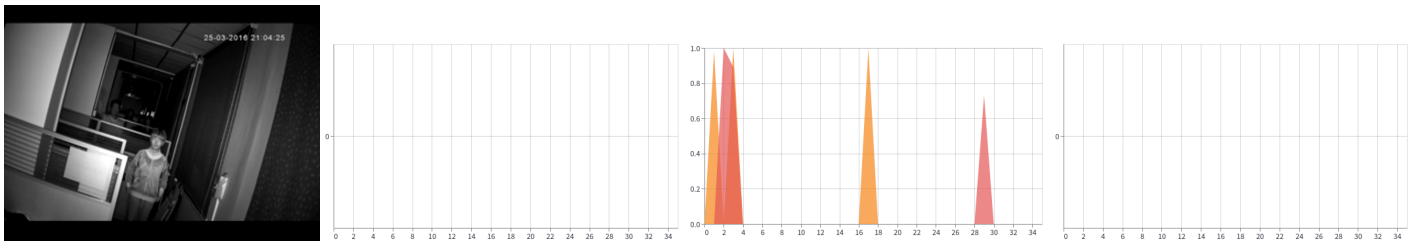

Figure 9. Normal010

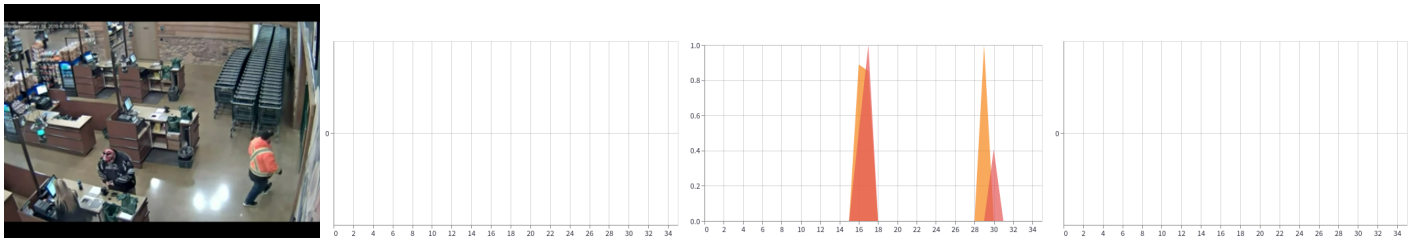

Figure 10. Normal019

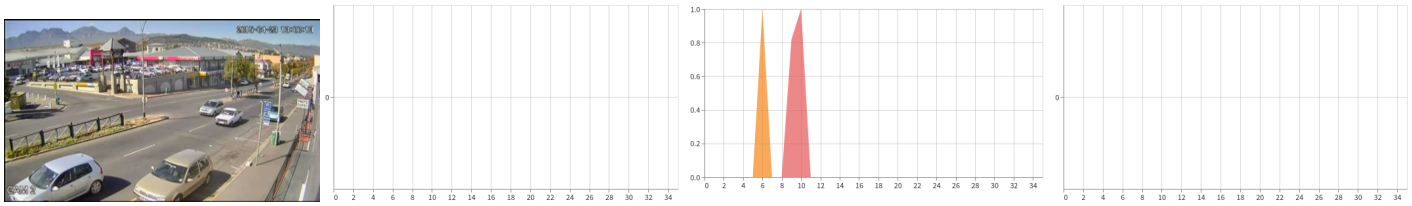

Figure 11. Normal006

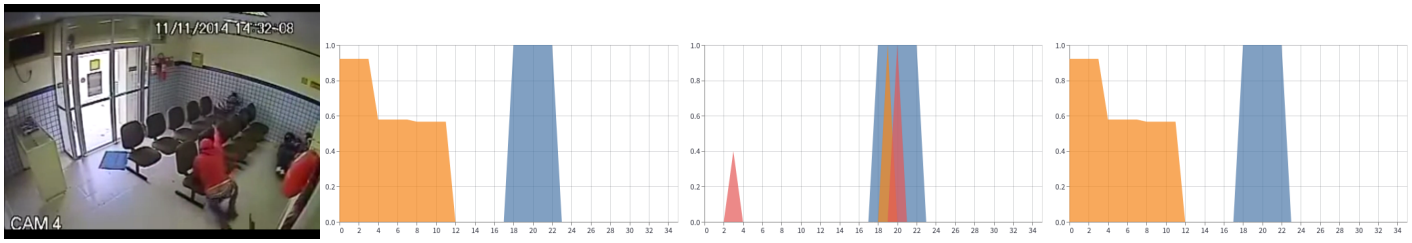

Figure 12. Failure Case: Shooting022

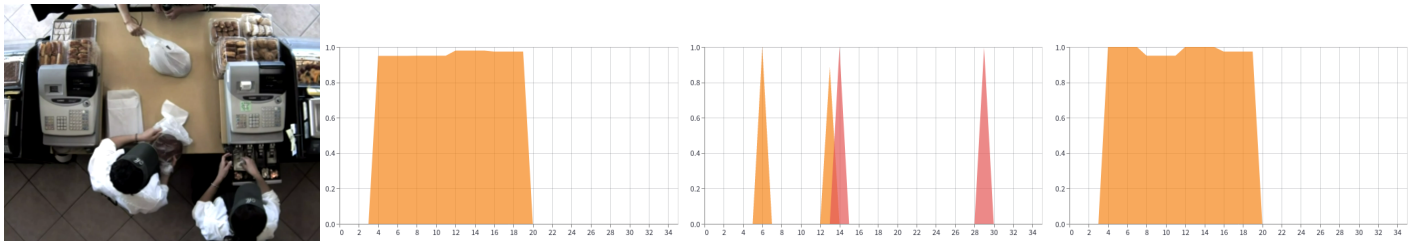

Figure 13. Failure Case: Normal041

Figure 14. Analysis of CRAFT and LKKM score profiles to demonstrate CRAFT's ability to detect anomaly characterised by unexpected change in appearance (sudden appearance of smoke cloud against bus), and LKKM's ability to detect deviation from repetitive patterns of motion (tuk-tuk swerves to avoid commotion).

Algorithm 1 Frame Construction with RAFT (CRAFT)

Require: $\mathcal{V}$, video instance
Require: $\mathcal{R}$, RAFT instance
$\mathcal{E} \leftarrow []$
$\mathcal{D} \leftarrow []$
$f_i \leftarrow \text{read}(\mathcal{V})$
$f_i' \leftarrow \text{gaussianBlur}(f_i)$
while hasNext($\mathcal{V}$) **do**
    $f_{i+1} \leftarrow read(\mathcal{V})$
    $f_{i+1}' \leftarrow \text{gaussianBlur}(f_i)$
    $h \leftarrow \text{height}(f_{i+1}')$
    $w \leftarrow \text{width}(f_{i+1}')$
    $p_{i+1} \leftarrow [\mathbf{0}]_{h \times w}$
    $o_i \leftarrow \text{estimateOpticalFlow}(\mathcal{R}, f_i', f_{i+1}')$
    $\hat{f_{i+1}} \leftarrow \text{applyFlowMap}(f_i', o_i, p_{i+1})$
    $e_i = ||(\hat{f_{i+1}} - f_{i+1}')||$
    $d_i = ||(f_i' - f_{i+1}')||$
    $\mathcal{E}$ appends e_i
    $\mathcal{D}$ appends d_i
    $f_i' \leftarrow f_{i+1}'$
end while
return $\mathcal{E}$, $\mathcal{D}$ ▷ $\mathcal{E}$ and $\mathcal{D}$ contain reconstruction error and similarity between consecutive frames, respectively, for all frames in a video.

Algorithm 2 Lukas-Kanade K-Means (LKKM) Algorithm

Require: $\mathcal{V}$, video instance.
Require: $\mathcal{K}$, initialised K-Means model.
Require: $\mathcal{L}$, frequency of update of optimal number of centroids.
Require: c, initial number of centroids.
$\mathcal{S} \leftarrow []$
$\mathcal{D} \leftarrow []$
$f_i \leftarrow \text{read}(\mathcal{V})$
$\vec{g}_i \leftarrow \text{extractFeatures}(f_i)$
$iter \leftarrow 0$
while hasNext($\mathcal{V}$) **do**
    increment $iter$
    $f_{i+1} \leftarrow \text{read}(\mathcal{V})$
    $\vec{g}_{i+1} \leftarrow \text{LukasKanade}(f_i, f_{i+1}, \vec{g}_i)$
    $d_{f_i} \leftarrow \text{computeDataVectors}(\vec{g}_i, \vec{g}_{i+1})$
    if $iter > 10$ **then**
        $\vec{p}_i \leftarrow \text{assignDataVectorsToCentroids}(\mathcal{K}, d_{f_i})$
        $\vec{q}_i \leftarrow \text{computeDistToCentroids}(\mathcal{K}, \vec{p}_i)$
        $s \leftarrow \frac{1}{m} \sum^{m} \max_m(\vec{q}_i)$
        $\mathcal{S}$ append s
        $\mathcal{D}$ append d_{f_i}
        if $iter \bmod \mathcal{L} == 0$ **then**
            $c = \text{computeOptimalCentroids}(\mathcal{K}, \mathcal{D})$
        end if
        $\text{partialFit}(\mathcal{K}, \mathcal{D}, c)$
    else
        $\mathcal{D}$ append d_{f_i}
        $\text{partialFit}(\mathcal{K}, \mathcal{D}, c)$
    end if
    $f_i \leftarrow f_{i+1}$
    $g_i \leftarrow g_{i+1}$
end while
return $\mathcal{S}$ ▷ $\mathcal{S}$ is a collection of anomaly scores per frame, each score quantifies the similarity of a frames' optical flow to that of previous frames.

Algorithm 3 Consensus Score Combination

Require: $\mathcal{B}^p$, base model score profile for video p.
Require: $\mathcal{R}^p$, CRAFT heuristic score profile for video p.
Require: $\mathcal{L}^p$, LKKM heuristic score profile for video p.
$C^p \leftarrow []$
for $i \leftarrow \mathcal{S}$ **do**
    if $\mathcal{B}_i^p < 0.1$ **then**
        $C_i^p \leftarrow 0$
    else
        $C_i^p \leftarrow \max(\mathcal{B}_i^p, \mathcal{R}_i^p, \mathcal{L}_i^p)$
    end if
end for
return C^p