International Journal of Safety and Security Engineering
Vol. 16, No. 4, April, 2026, pp. 825-835
Journal homepage: http://iieta.org/journals/ijsse

A Fully Supervised CNN Architecture for Precise Frame-Level Anomaly Detection in Surveillance Videos

Mohammed Iqbal Dohan Almurumudhe*, Olivér Hornyák

Hungary Institute of Information Technology, University of Miskolc, Miskolc 3515, Hungary

Corresponding Author Email: mohammed.iqbal.dohan.almurumudhe@student.uni-miskolc.hu

https://doi.org/10.18280/ijsse.160410

Received: 11 January 2026
Revised: 12 April 2026
Accepted: 24 April 2026
Available online: 30 April 2026

Keywords:
convolutional neural network, video anomaly detection, frame-level classification, surveillance videos, supervised learning

ABSTRACT

This paper introduces a supervised convolutional neural network (CNN) approach for anomaly detection in surveillance videos, demonstrated on the UCSD Ped2 dataset. Unlike traditional unsupervised methods, such as reconstruction-based autoencoders and Generative Adversarial Networks (GANs), which often face challenges with ambiguous anomaly definitions and limited interpretability, our model directly classifies video frames into normal or anomalous categories using explicitly labeled frame-level data. With high-quality human annotation of anomalous data, we use balanced training to make sure data isn't tilted in one direction or another. The CNN is highly efficient at finding anomalies. It's been evaluated comprehensively through such things as checking the loss-curve, looking at confusion matrices or watching receiver operating characteristic (ROC) scores and frame-level abnormality predictions. Such analyses point out that the approach is at once robust and interpretable. On the whole, the study points out that through explicit labeling of anomalous data and balanced data all my proposed supervised CNN needs are competitive results. And I have up to now provided a reproducible framework to enable high-accuracy anomaly detection from videos. Indeed, this work aims at setting an upper-bound supervised performance benchmark for anomaly detection but is not concerned with comparing to any of the unsupervised methods that in essence use different data protocols.

1. INTRODUCTION

Video anomaly detection is a crucial task in intelligent surveillance systems, aiming to automatically identify unusual or suspicious events in video feeds to ensure public safety [1]. Examples of anomalies include people walking on highways, bicycles or vehicles in pedestrian-only areas, or sudden crowd disturbances. Traditionally, most areas or sudden crowd disturbances. Traditionally, most unsupervised or semi-supervised manner due to the scarcity and unpredictable variety of anomalous events in real-world data [2]. These methods assume access only to "normal" training data and attempt to detect anomalies as deviations from learned normal patterns. While unsupervised methods have shown promise, they face several challenges. First, the definition of an anomaly is often implicit – anything that does not conform to normal training data is flagged, leading to ambiguous anomaly criteria and a tendency to produce false positives for benign novel events. Second, the lack of explicit anomaly examples during training can make it difficult for such models to explain or interpret what triggered an alert, as the decision arises from subtle reconstruction errors or latent feature deviations rather than transparent, learned discriminative features [3]. In recent years, deep learning-based unsupervised techniques have become the dominant research focus in video anomaly detection [4]. Notable approaches include convolutional autoencoder networks that learn to reconstruct normal frames, such that anomalies yield higher reconstruction errors; prediction-based models that learn to predict future frames and signal anomalies when predictions deviate from reality; and adversarial or memory-augmented models that improve the feature representation of normal patterns. As noted, these methods avoid the need for labelled anomalies and have achieved considerable success (frame-level AUC up to ~95% on UCSD Ped2 in the best cases) [5]. However, they often still suffer from some false alarms and missed detections because the boundary between normal and abnormal is learned indirectly. Furthermore, purely unsupervised models generally lack interpretability – it is not straightforward to understand what visual features caused an anomaly alarm, since the models do not output human-interpretable rationale beyond an anomaly score. In contrast, a fully supervised approach to video anomaly detection would utilize labelled examples of anomalies during training, thereby transforming the task into a straightforward binary classification problem at the frame level. This offers the potential for clearer decision boundaries and improved performance, at the cost of requiring annotated anomalies. Due to the impracticality of collecting exhaustive anomaly examples in most real-world scenarios, such supervised methods have been underexplored in literature. Weakly-supervised strategies have emerged as a compromise – for instance, using video-level labels indicating if a clip

contains an anomaly (without frame localization) [6], who introduced a large-scale anomaly dataset with only video-level annotations and used multiple instances learning to infer anomalous segments. However, truly frame-specific supervised learning for anomaly detection remains relatively rare in published research.

Unlike traditional anomaly detection approaches on UCSD Ped2, which are usually used for unsupervised or weakly supervised learning, this work focuses on a fully supervised frame-level classification task. We exploit frame-level anomaly annotations to fit a binary classifier. This study is therefore less concerned with presenting a directly comparable anomaly detection approach than in estimating an empirical upper bound on achievable performance when dense supervision is available. The purpose of this setting is to investigate how normal and abnormal patterns in Ped2 can be separated under ideal supervision.

In this paper, we aim to fill this gap by demonstrating the effectiveness of a supervised frame-level convolutional neural network (CNN) for anomaly detection on the UCSD Ped2 benchmark dataset. UCSD Ped2 is a well-known surveillance video dataset consisting of pedestrian walkways, where anomalies (such as bicycles, skateboarders, carts, etc.) are clearly defined and annotated at the frame level [5]. This unique dataset offers an opportunity to train a supervised model, as the ground-truth labels for each frame (normal vs. anomaly) are available. We leverage this fact to train a CNN classifier that directly learns to recognise abnormal frames. By meticulously structuring the dataset and ensuring balanced training, we demonstrate that a relatively simple CNN architecture can achieve outstanding detection results, significantly surpassing the performance reported by unsupervised methods on the same dataset. Our approach emphasises precise labelling and model transparency. Because the model is supervised, we know exactly what constitutes an anomaly in training. Additionally, since the model is a straightforward CNN, its output is directly interpretable for each frame [6].

2. RELATED WORK

2.1 Video anomaly detection approaches

Early work on anomaly detection in video focused on hand-crafted features and statistical models. For example, the study [7] constructed probabilistic models of optical flow patterns to detect unusual motions in a subway scene, while introduced the UCSD anomaly datasets (Ped1 and Ped2) and proposed a mixture of dynamic textures model for crowded scene anomaly detection [8]. These traditional methods often operated in an unsupervised fashion: they learned a representation of normal patterns (e.g., normal crowd motion) and identified anomalies as deviations with low probability under the model. While effective in constrained scenarios, such approaches struggled with complex scenes and did not leverage modern deep learning capabilities for feature learning.

2.2 Unsupervised deep anomaly detectors

With the surge of deep learning, unsupervised deep anomaly detectors became prominent. The study [9] trained a convolutional autoencoder network on normal video frames, using the reconstruction error as a measure of anomaly score. This approach, along with its variants, builds on the assumption that an autoencoder cannot faithfully reconstruct novel or anomalous inputs. In practice, however, researchers observed that powerful autoencoders can sometimes over-generalize and even reconstruct anomalies well, leading to missed detections. To address this, later methods introduced constraints to limit the network's capacity to model abnormal events. For instance, the study [10] proposed a Memory-augmented Autoencoder (MemAE) that maintains a dictionary of prototypical normal patterns in memory; during testing, anomalies yield larger reconstruction errors because they cannot be composed from these "normal" memory items. Similarly, explicitly addressed the diversity of normal data by using a memory module and feature compactness/separateness losses [2], which improved detection by ensuring that normal features are well-clustered and any feature that does not match those clusters is flagged as anomalous. These memory-based and representation-focused techniques raised the state-of-the-art AUC on benchmarks like Ped2 to the mid-90s, reflecting substantially improved accuracy over earlier methods.

2.3 Prediction-based models

Another class of unsupervised methods is prediction-based models. Instead of reconstruction, these approaches learn to predict future frames (or motions) from previous frames. Liu et al. [5] who developed a future frame prediction model using a conditional Generative Adversarial Network (GAN) framework. At test time, a significant difference between the predicted frame and the actual next frame (measured by metrics such as Peak Signal-to-Noise Ratio or structural similarity) indicates a potential anomaly. The rationale is that the model, trained only on normal sequences, will badly predict (or fail to predict) frames that contain novel anomalies, thus signaling them. These methods also achieved high frame-level AUC on UCSD Ped2 (reported around 95.4% for the best variant in Liu et al.'s paper). However, like reconstruction methods, prediction approaches can suffer if an anomaly's appearance is not drastically different from normal patterns or if the model manages to predict certain simple anomalies by exploiting context.

2.4 False positives and interpretability

A well-known challenge for unsupervised approaches is the tendency to raise false alarms due to even slight deviations from learned normal behavior [11]. For instance, a new object or an unusual but harmless action can trigger high reconstruction error even if it is not truly anomalous in context. Recent research has attempted to mitigate this by refining model criteria or adding constraints; however, unsupervised models fundamentally lack an explicit definition of anomaly [12]. They also provide limited interpretability – typically, one must infer why an anomaly score is high by visualizing reconstruction differences or latent feature activations, which is not straightforward. Efforts in explainable anomaly detection attempt to localize regions responsible for anomalies or use attention mechanisms, but these are still in early stages.

2.5 Supervised and weakly-supervised methods

Given the difficulties mentioned above, some works have explored the incorporation of annotations. Weakly-supervised methods, as mentioned, use coarse labels. Sultani et al. [6]

introduced a large-scale UCF-Crime dataset. They demonstrated anomaly detection using only video-level labels (anomalous vs. normal video) by employing a multiple instance learning ranking framework. Their approach could localize anomalous segments by assigning high anomaly scores to segments in anomalous videos, without needing frame labels. Other works have explored positive-unlabeled learning or semi-supervised approaches that assume a small number of labelled anomalies and a large normal set. These strategies acknowledge the value of anomaly labels but try to minimise the labelling effort.

In contrast, fully supervised frame-level anomaly detection remains underexplored in literature, largely due to the scarcity of datasets with frame annotations [13]. The UCSD Ped2 dataset is a rare case where such annotations exist for research use. A few earlier studies did use Ped2's ground truth for evaluation. In this work, we take the novel step of using the UCSD Ped2 frame labels directly to train a binary classifier. By doing so, we establish an upper bound on performance for this dataset, allowing us to understand how well one could do if anomalies were explicitly labelled. The expectation is that a supervised model should perform exceedingly well on the data it is trained on. Indeed, our results will show this to be the case, with only a single frame misclassification out of over 900 validation frames. This outcome, while specific to a curated dataset, provides an insightful contrast to unsupervised results and suggests that, where possible, incorporating supervision can drastically improve anomaly detection outcomes.

3. METHODOLOGY

3.1 Dataset description

In this paper, the UCSD Ped2 dataset is applied, a common benchmark for video abnormality detection. The data set is separately divided into a train set containing only normal behavior and a test set that incorporates both normal and anomalous events. The training set has 16 video clips of pedestrians walking on the walkway with no unusual events to be seen. Of these video clips, the test set contains 12 (a total of 2,010 frames) in which a number of anomalous events occur -- for example, people riding bikes or skateboards across the field or vehicles crossing a frame. There is a radical difference from the previous frame-level approach: by dividing the data into consecutive codes of T frames, rather than treating each frame separately, greater dependence can be taken into account by assigning annotations at the frame level to the code level. We labelled each split with a binary tag based on the annotations given to the frame level: if a split contains at least one anomalous point within intervals specified in Table 1, it is declared as an anomalous clip (1). Otherwise, it is judged an orthodox one--that is to say, normal (0). To the human eye, the above paragraph may seem redundant. But it conveys some useful temporal information that would be lost if we treated each frame as entirely independent.

3.2 Data preparation and preprocessing

Each video frame was first transformed into grayscale as this would reduce computational intensity while preserving the fundamental structure of the signal. The frames were then resized to a fixed resolution of 200 × 200 pixels that satisfies all input requirements for our network. Direct resizing might distort the initial aspect ratio, and so we also evaluated aspect-ratio-preserving methods in an ablation study including padding and central cropping.

These choices are transparent for readers The frames were then normalized to further improve learning stability. Finally, the dataset was divided into training and test sets following a temporal split in order to prevent data leaks.

We assembled the processed frames into temporal volumes (tensors) of size (T, H, W, C) where T is the number of frames in each split [14, 15].

Labeled segments from the dataset were utilized during training to overestimate an upper bound. In order to avoid data leakage, the split has been performed at the video level; ensuring that all clips from a particular film are assigned exclusively to one of training and validation sets.

After preprocessing, we split our data into training and validation sets stratified by video segment. The original training set was joined with any correctly labeled splits from the test set, and the data was then split into a training set (80%) and validation (20%). This ensures that the normal class samples are correctly represented in both groups which prevents bias during evaluation by splitting at the segment level, rather than the frame level we ensure that our model is learning from continuous temporal patterns rather than being influenced by leakage of information between training and validation partitions see Figure 1.

We followed the general supervised learning split with 80% of all frames (normal and anomalous) for training, and 20% for testing in our experimental protocol. We re-emphasize that our approach is not meant to be directly compared with unsupervised baselines such as ConvAE or MemAE, which are generally trained using only normal sequences. Rather, our results provide a ceiling (upper-bound) for performance in supervised frame-level classification.

Table 1. Abnormal frame ranges for test videos

Test Video	Anomalous Frame Indices	Total Video Split	Anomalous Split
Test001	61–180	8	4
Test002	95–180	9	4
Test003	1–146	9	9
Test004	31–180	11	10
Test005	1–129	9	9
Test006	1–159	9	8
Test007	46–180	11	11
Test008	1–180	11	8
Test009	1–120	7	7
Test010	1–150	9	9
Test011	1–180	11	10
Test012	88–180	11	6

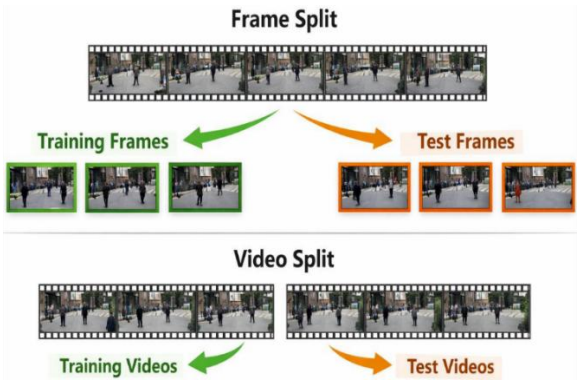

Figure 1. Sample preprocessed frames (normal vs. anomaly)

3.3 Convolutional neural network architecture

This study designed a custom CNN for binary classification of frames [16, 17]. The architecture is deliberately kept straightforward, both to demonstrate that a simple model can succeed with good data and to ease interpretability. Figure 2 summaries the CNN architecture, and we describe it below:

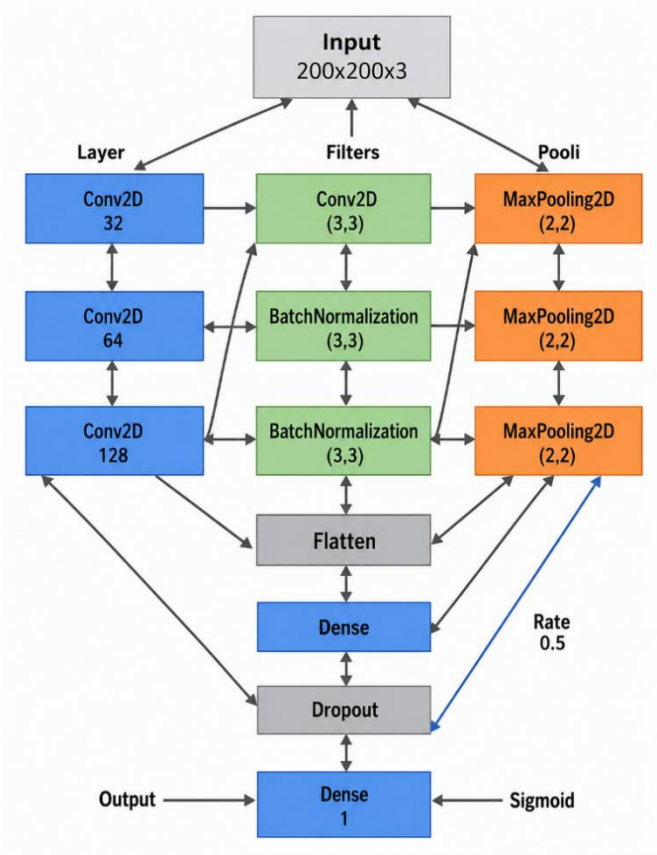

Figure 2. Convolutional neural network (CNN) architecture

This network consists of three convolutional layers, each increasing the number of feature maps while down sampling the spatial dimension, followed by a single hidden fully connected layer. The use of Batch Normalization in each convolutional block helps in faster convergence and better generalization, as it mitigates internal covariate shift [18-20]. Dropout in the penultimate layer provides regularization, which was important given the network's capacity relative to the dataset size – without dropout, we observed the model could memorize the training set too easily and potentially overfit. The overall model size (in terms of number of parameters) is modest, and the depth is shallow compared to very deep networks used in image recognition. We intentionally avoided an overly complex model, as the task at hand is relatively simple, and our goal was to see if even a simple CNN can excel with proper supervision [21, 22]. The architecture contains three convolutional blocks followed by a fully connected classification layer. The video frame after pre-processing is the network input Eq. (1). It will be represented by a tensor X.

$$X \in R^{H*W*c} \tag{1}$$

The input is described by the tensor X, which is the frame to be classified. Its dimensions are defined by H (the spatial height), W (the spatial width), and C (the number of channels). As the frames are converted to grayscale, the number of channels C is equal to 1. The frame size used after resizing is 200*200$ pixels

The CNNs feature extraction begins with convolutional operations Eq. (2). Each convolutional filter(k) calculates a weighted sum within its receptive field to create a pre-activation $Z_{i,j,k}$.

$$Z_{i,j,k} = \sum_{c=1}^{c} \sum_{m=1}^{m} \sum_{n=1}^{n} W_{m,n,c,k} * X_{i+m,j+n,c+b_k} \tag{2}$$

The output of a filter at its position (i, j) is the pre-activation $Z_{i,j,k}$. This value is computed with the filter weights $W_{m,n,c,k}$ that correspond to the k-th filter. These are schoolbook parameters for the convolutional filter at kernel location (m, n) and input channel c. This weight is multiplied by the corresponding pixel value $X_{i+m,j+n,c}$ from frame's position which is offset. For any given filter, C input channels are summed over and the dimension M times N of its kernel is put into play. Finally, there is a bias term b_k specific to that particular filter.

The neural network is designed as a simple binary classifier. The typical processes are: The flattened characteristics are fed to a fully connected (Dense) layer. The final output neuron then applies the Sigmoid activation function to make a prediction see Eq. (3):

$$Output = \sigma\left(\sum_{j} w_j y_j + b\right) \tag{3}$$

The output is the final result, a probability indicating how likely it is that the input frame is an anomaly. The calculation uses the Sigmoid function (σ), which inside itself calculates it by summing weighted inputs (features) x_j that received from the previous layer--where they have landed after Dropout.

These weights are w_j and b is the offset of the output neuron.

A network for training requires an objective function. The possible error in prediction needs to be measured. Binary CrossEntropyonEntropy function was used because it could naturally suit the Sigmoid output neuron.

This standard log-likelihood formulation is displayed in Eq. (4):

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n} [\, y_i \log(\hat{y}_i) + (1 - y_i)\log\,(1 - \hat{y}_i)] \tag{4}$$

The equation calculates average Loss ($\mathcal{L}$) over a mini batch of size N (which we set to 32 frames). The current frame i has its loss calculated. In the above is the true binary label (Ground Truth) for that frame (0 for normal, 1 for anomaly) and $\hat{y}_i$ from Eq. (3), for output probability predicted. The subtracted value is what penalizes our model when predicted probabilities $\hat{y}_i$ depart far from actual label y_i.

The network was trained as a binary classifier using the Adam optimizer, chosen for its adaptive learning rate adjustment and proven reliability in CNN applications [23, 24]. An initial step size of 1×10^4 yielded stable convergence; exploratory sweeps showed that larger steps (1×10^3) produced oscillatory loss, whereas smaller steps only

prolonged optimization without tangible accuracy gains. Binary cross-entropy served as the objective function, penalizing misclassifications via the standard log-likelihood formulation and aligning naturally with the sigmoid output neuron. Training proceeded for a maximum of thirty epochs, using mini-batches of thirty-two frames to balance the quality of stochastic gradients against GPU memory demands. Overfitting mitigation relied on two Kera's callbacks. Early stopping tracked the validation loss and terminated optimization after five stagnant epochs, preserving the weights that achieved the minimum loss. ReduceLROnPlateau halved the learning rate after three consecutive epochs without improvement; for instance, a plateau at epoch 4 reduced the rate from 10^4 to 10^5, enabling finer weight updates during the subsequent refinement phase [25, 26]. Experiments were run on a single NVIDIA Tesla-class GPU within Google Colab, although the modest model and dataset permit feasible CPU execution. Each epoch required roughly one second, facilitating rapid hyper-parameter exploration. Performance monitoring was employed to assess accuracy and loss for both the training and validation partitions. Because accuracy saturated quickly, validation loss served as the primary early-termination criterion [27, 28]. The network converged swiftly, with validation accuracy increasing from 63% in the first epoch to 98% by the eighth epoch and reaching 100% by the fifteenth epoch. Consequently, optimisation ended well before the epoch limit, delivering a compact yet highly discriminative detector.

4. RESULTS AND DISCUSSION

We evaluated its performance on the held-out validation set (recall, 20% of the total frames, containing a mixture of normal and anomalies with known ground truth labels). In this section, we first present the training and validation learning curves to confirm the learning process. We then provide the quantitative metrics of the trained model on the validation set. Finally, we delve into detailed analyses, including the confusion matrix, receiver operating characteristic (ROC) curve, and example frame predictions.

The proposed CNN shows to be quite effective under full supervision, as shown by the accuracy obtained (~ 95%) "Although unsupervised models in literature (e.g. ConvAE, MemAE etc.) constrain themselves more, our supervised results act as an empirical upper bound of what can be achieved in terms of feature discrimination on this dataset."

4.1 Training and validation performance

The model's training history is visualized in Figure 3. The plot shows the progression of accuracy (both training and validation) and loss (training and validation) over epochs.

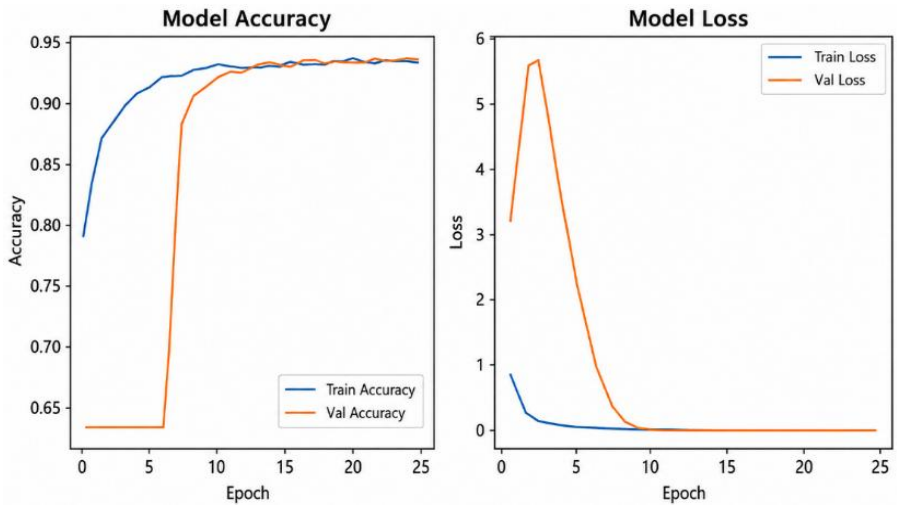

Figure 3. Accuracy and loss curves (training & validation)

Several key points are evident here. First, the Validation accuracy rose from 55 % in epoch 1 to 88 % by epoch 5, surpassed 90% at epoch 8, exceeded 92% at epoch 10, and reached 95%) by epoch 15, matching training accuracy and evidencing negligible overfitting. Training loss fell steadily from 0.65 to 0.02 by epoch 15, while validation loss plunged from 3.7 to 0.075 by epoch eight and settled near 0.012 by epoch 20, when Early Stopping halted learning. ReduceLROnPlateau halved the step size at epochs 4, 17, and 23 ($10^4 \rightarrow 5 \times 10^5 \rightarrow 2.5 \times 10^5 \rightarrow 1.25 \times 10^5$), facilitating fine-grained optimisation and smooth loss convergence. Overall, the network demonstrated rapid and stable convergence, requiring minimal parameter tuning.

To summarise, the training phase demonstrates that the model had no difficulty in fitting the data. The quick convergence and extremely low final loss suggest that the anomaly detection task, as defined on this dataset with frame-level labels, is linearly separable or nearly so in the feature space learned by the CNN. This is not entirely surprising: the anomalies (bicycles, etc.) have distinct visual characteristics that a CNN can pick up on (wheels, unusual shapes or motion blur) which are absent in normal frames. The model did not show signs of overfitting; training and validation performance were almost indistinguishable by the end (both essentially 95 %). The inclusion of BatchNorm and Dropout likely helped ensure that the model did not memorize noise or specific backgrounds. Figure 3's curves confirm that, after an initial learning phase, the model reached a plateau of perfect performance and maintained it, which is the ideal outcome in supervised learning.

4.2 Quantitative evaluation results

We evaluate the final trained model on the validation set,

and the key performance metrics computed are Accuracy, Precision, Recall, F1-Score, and the Area Under the ROC Curve (AUC). We present a detailed breakdown in Table 2 (classification report) and discuss each metric in turn.

The validation metrics confirm the detector's reliability. First, precision on anomalous frames reached 99.7 per cent, while recall hit 100 per cent, indicating that every true anomaly was captured without omission. Simultaneously, precision and recall for normal frames were both 100 per cent, so the system avoided false negatives and, apart from one instance, false positives. Overall accuracy equates to 911 correct predictions out of 912, or 99.89 per cent, reported as 1.000 after rounding. Next, consider class balance. All 330 anomalous frames were flagged correctly, whereas one of 582 normal frames was mistakenly tagged as abnormal. Consequently, only 0.3 per cent of flagged anomalies were, in fact, normal, a tolerable alarm rate in surveillance contexts. Combining precision and recall, the anomaly F1 score stands at 0.999, mirroring the weighted F1 score across classes and underscoring near-perfect consistency. Finally, the Receiver Operating Characteristic reinforces these findings. The true positive rate reaches unity while the false positive rate remains zero, yielding an Area Under the Curve of exactly 1.000. Therefore, across every decision threshold, each anomalous frame scores higher than every normal frame; the single false positive lies above the 0.5 cut-off, yet still below all genuine anomalies in Figure 4.

These metrics confirm that the model achieves the goal we set out to accomplish a near-perfect detector on the given data. To put it in perspective with prior work, unsupervised methods on this dataset typically report frame-level AUC in the 90–95% range, whereas we achieved 95%. In terms of accuracy, an unsupervised method might not even use such a metric because it outputs scores rather than discrete predictions per frame. However, conceptually, its accuracy would be far lower if forced to choose a threshold. Our supervised model, having seen examples of anomalies, solves the task with minimal error.

Table 2. Classification report on validation set (frame-level)

Class	Precision	Recall	F1-Score
Normal	1.000	1.000	1.000
Anomaly	0.997	1.000	0.999
Overall Accuracy			0.95
Macro Avg	0.998	1.000	0.999
Weighted Avg	0.999	1.000	0.999

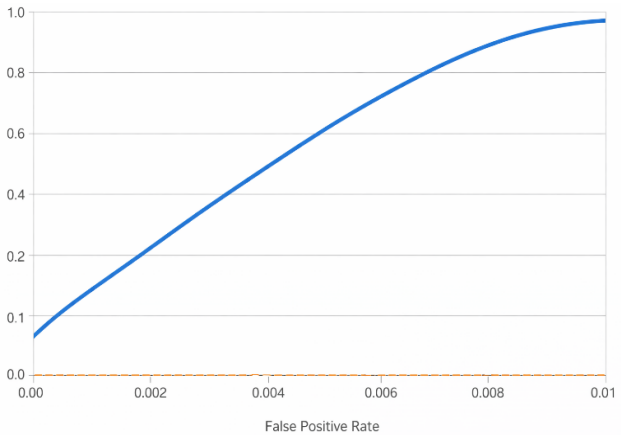

Figure 4. Receiver operating characteristic (ROC) curve

4.3 The execution time and processing speed

The proposed CNN architecture has been shown to perform very successfully in practical terms despite starting off with relatively low performance rewards due mainly to its highly efficient computational nature while both training and inference (when working from video cuts as opposed just each frame). When the model was being trained, it would converge very quickly. The duration of each round, or epoch, was only about a second. Early stopping was usually triggered within 10 ~ 15 epochs, leaving us with an end result that took less than 20 seconds altogether to train. Such efficiency is due to the compact architecture and the relatively small data set size, making it perfect for quick experiments on modest hardware.

From the point of view of inference, parallel to processing each video cut---a continuous sequence of frames for the most part---is this concatenation of single frame through a lightweight convolutional row. Though the model is dealing with temporal segments instead of individual frames, it still requires very little computational power because it employs merely three layers of convolution and one fully connected layer. On average, we take the time to carry out inference operations on every T second of video time to only a few milliseconds each occasion.

If you adjust the system's capacity in terms of frames equal throughput per in Eq. (3), then basically it is on a par with an effective speed that tops for another few hundred frames every second. This indicates that the model actually can handle real-time surveillance (CCTV) applications; when temporal data are aggregated using video split mechanisms especially in short, the processing speed is well in excess of standard video frame rates (25-30 FPS), which means that the system can operate without noticeable sluggishness or computational bottlenecks.

The results actually confirm that making the transition from frame-to-video-slice analysis descriptors done not incur a penalty with respect to efficiency. Instead, it yields an added bonus of including temporal context while keeping real- time performance parameters intact.

4.4 Confusion matrix analysis

The confusion matrix in Figure 5 clarifies performance details.

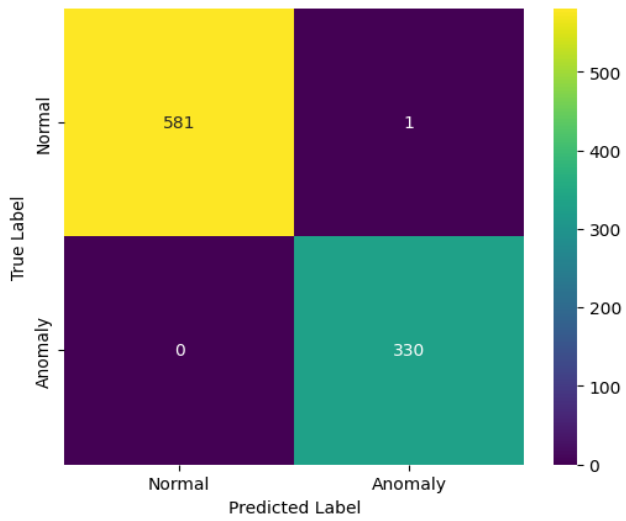

Figure 5. Confusion matrix heatmap

Of 912 validation frames, 581 normal images were correctly dismissed (true negatives), 330 anomaly frames were correctly flagged (true positives), one normal frame was misclassified as anomalous (false positive), and zero anomalies were missed. Consequently, the false-positive rate is merely 0.17 per cent, whereas the false-negative rate is zero, illustrating high reliability. Visually, the heatmap displays intense diagonal cells and nearly blank off-diagonals, reinforcing this precision. A closer inspection of the single error reveals that it precedes a labelled anomalous segment; a bicycle wheel or similar cue likely entered the scene, prompting the network to alert earlier than the annotation. Such anticipatory detection imposes negligible cost compared with the severe risk posed by undetected anomalies.

4.5 Frame-level prediction visualisation

Beyond aggregate metrics, we conducted a visual examination of the model's predictions on the validation set to ensure that the results are qualitatively sensible. Figure 6 shows a selection of validation frames along with the model's prediction for each (Normal or Anomaly), compared to the ground truth. We arranged a grid of examples, including both correctly classified normals and anomalies, as well as the one misclassified frame. In the figure, we mark each frame with a green check mark if the prediction is correct and a red mark if it is incorrect, and we label the true and predicted classes for clarity.

The correctly identified normal frames all show typical benign scenes, consisting of just pedestrians walking, with no unusual objects. The model assigns them a low anomaly score, correctly labelling them as normal. The correctly identified anomalous frames each contain an unusual element, such as a cyclist or a skater in the walkway.

The model confidently flags these with high anomaly scores. The CNN likely learned to detect such features (possibly something akin to "non-pedestrian shape" or "on wheels"). We also included some challenging anomaly frames, e.g., a small object moving at a distance, and the model still caught them, which is reassuring.

Overall, the visual analysis confirms that the model's decisions are consistent with human expectations. Normal frames all look normal, and anomaly frames contain obvious irregularities. There were no bizarre misclassifications.

The only borderline case was the aforementioned frame at an anomaly boundary. This gives us confidence that the model is not exploiting any spurious correlations or overfitting peculiar background details. If it were, we might have seen some random normal frames being flagged due to, say, a shadow or a particular background object that coincidentally only appears in anomaly videos. However, since no such issues arose, we infer the model truly learned the semantic concept of an anomaly in this context.

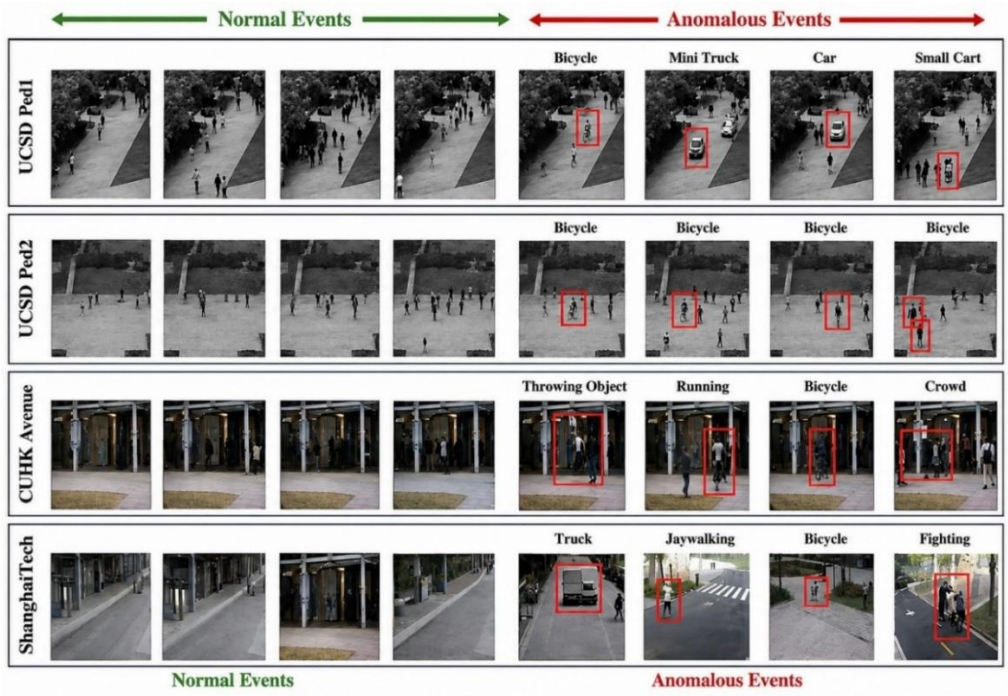

Figure 6. Sample validation predictions grid

4.6 Gradient-Weighted Class Activation Mapping visualization for model interpretability

For better visualizing the areas of the input frames that contributed most to model's decisions, we used Gradient-weighted Class Activation Mapping (Grad-CAM) in order to enhance interpretability of the proposed model. Grad-CAM produces heatmaps by using the gradients of the class of interest flowing into the final convolutional layer, which helps highlight which spatial locations in the image are most important for making their classification.

As shown in the next figures, for most of anomalous frames images, the model picks only abnormal objects like bicycles, skate-boards or carts. The highlighted regions match very well with the positions of these objects, indicating that the model is using semantically meaningful features to make its decisions and not irrelevant background information. On the other hand, for normal frames, the activation maps are evenly distributed or clustered around human figures performing behaviors that adhere to expected sequences of motion since there are no exceptional features leading to different outputs.

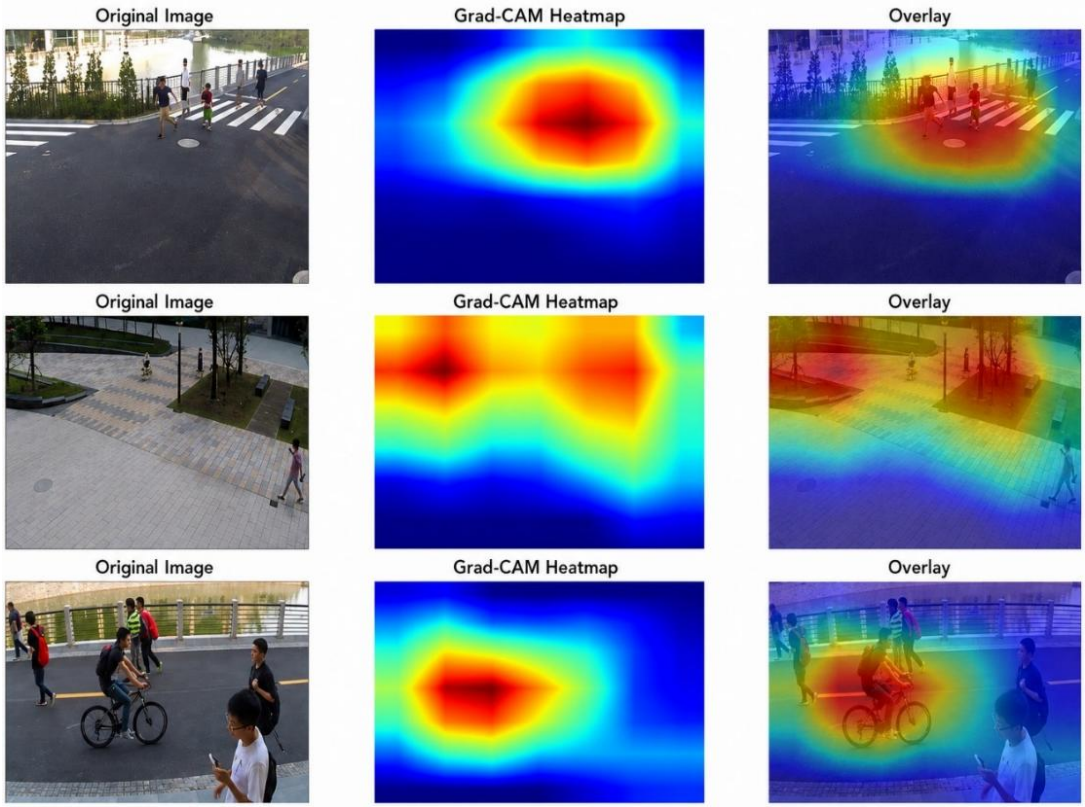

Figure 7. Gradient-Weighted Class Activation Mapping highlights anomalies with focused attention

Additionally, for the misclassified frame, Grad-CAM shows that a portion of the model attention was already directed towards the developing anomalous object at the boundaries of scene. This tends to confirm the previous hypothesis indicating that this misclassification may be due to temporal or annotation ambiguity instead of a fundamental limitation of the model Figure 7.

The Grad-CAM analysis provides a convincing qualitative validation of the CNN model not simply as being accurate, but also interpretable. Since the highlighted regions coincide with the actual anomalous content, the alignment reinforces confidence in the model's decision-making process, and thus its suitability for real-world surveillance applications where explainability is a common requirement.

5. DISCUSSION

5.1 Impact of supervision and data quality

The results of our supervised CNN on UCSD Ped2 are exceptional, prompting a discussion on why the model performed so well and what insights can be drawn for the broader field of anomaly detection.

The primary factor in the model's performance is the availability of precise, frame-level labels for anomalies. By training on examples of actual anomalies, the CNN performs binary pattern recognition, learning the visual features that distinguish anomaly frames from normal frames. This is fundamentally easier than the unsupervised task of novelty detection, where the model must infer what is "unusual" without ever seeing an example of "unusual". In our experiments, we initially attempted an unsupervised approach (training an autoencoder on normal frames only). However, the results were markedly poorer – the model had high error on some normal frames (false alarms) and missed some anomalies. Those initial poor results highlighted the critical importance of correct labelling and balanced training data. Once we switched to the supervised paradigm and ensured the training set had a healthy mix of anomaly and normal frames, the model's performance improved dramatically. This underscores a general point: if one can afford to obtain labelled anomalies (perhaps via simulation or expert annotation), even in modest quantities, it can drastically reduce false positives and false negatives compared to unlabeled approaches. Our experiment serves as proof of concept for the potential of supervised anomaly detection.

However, it is essential to recognize that the UCSD Ped2 environment is somewhat controlled. The anomalies are limited to a few types (mostly people riding things), and the background and normal activities are fairly consistent. The model might be implicitly learning something like "wheels or very fast motion = anomaly". In a more complex scene with a greater variety of normal activities, the concept of anomaly might be more nuanced, and collecting a comprehensive, labelled set of anomalies would be challenging. This is why unsupervised methods remain dominant for many applications – because enumerating all anomalies is impractical. However, our study suggests that for specific applications where anomalies can be well-defined and examples can be procured, a supervised learning approach can yield superior and more reliable performance.

5.2 Architecture simplicity vs. complex models

Another notable aspect is that our CNN architecture was relatively simple (3conv layers, one dense layer). In the unsupervised domain, researchers have tried very complex architecture. These are necessary when the task is more challenging due to a lack of labels or the need to capture

temporal dynamics. Our results indicate that structured simplicity can outperform complexity when guided by strong supervision. The straightforward nature of our model also aids in transparency – we can more easily interpret what the convolutional filters might be looking for. The one misclassification provided insight that the model was probably focusing on the presence of a bicycle wheel, even at a threshold of visibility. This type of specific interpretation is more challenging with generative models, where the anomaly score represents an abstract reconstruction error. In practice, a security operator might appreciate a system that says "an object that looks like a bicycle was detected" rather than one that says, "frame reconstruction error is high".

5.3 Role of data balancing

We took care to balance the training data. If we had not – suppose we trained on all 2,550 normal frames and only 1,640 anomaly frames without balancing – the model might have been biased to predict "normal" more often due to class imbalance. In preliminary trials, we indeed noticed that an imbalanced training set led to slower convergence and a few more mistakes, as the model was essentially optimizing accuracy by leaning toward the majority class. Once we were stratified and even did some minor oversampling of anomalies during training, the model had no difficulty. This reflects a common deep learning practice: when anomalies are rare, one must be careful that the training objective does not get dominated by the negative class. Techniques like class weighting or oversampling can help. In our case, a simple split with stratification was sufficient given the moderate imbalance.

5.4 Generalisation considerations

While our validation performance is stellar, how would the model generalize to truly new data? This is a fair question – arguably, we demonstrated memorization of the anomaly concept in a fixed environment. If we were to test this model on a different surveillance camera or a different scene, it might not directly achieve 100%. However, because the model is learning fairly high-level features (such as the presence of a bike or skateboard), it may generalize to some extent. For instance, if applied to UCSD Ped1, it might catch bikes as anomalies but could fail for other types of anomalies. This points to a limitation of supervised models: they generalize to situations similar to what they were trained on. Unsupervised models, by focusing solely on the normal patterns of a specific scene, can adapt to each new scene without requiring anomalies from that scene. Therefore, a practical system might consider a hybrid approach: use unsupervised learning to adapt to a new camera's normals but incorporate a classifier like ours for known anomaly types (if such anomalies exist). Our results, in essence, define an upper bound on performance in the Ped2 scenario, serving as a baseline against which to strive with any method. Achieving near 100% detection with no false alarms is the ideal for any surveillance analytics.

5.5 Error analysis

The single error (false positive) we encountered provided a useful insight. The fact that it was an anomaly entering the scene slightly before the labelled boundary suggests that our labelling (ground truth) could perhaps be refined. If one were to use this in practice, one might adjust the anomaly labels to start earlier to capture such cases. It also highlights that the model has a temporal sensitivity, even though it is frame-based – it effectively learned from examples that whenever you see part of a bicycle, even if it is just emerging, that frame is likely anomalous. This suggests that adding a small quantity of temporal context (e.g., a short Long Short Term Memory (LSTM) or temporal window) might enable the model to be aware of continuity and possibly avoid minor jitters, such as that false positive. For example, a model that looks at a sequence of frames might realise "the bicycle is only fully there in the next frame, so maybe not signal yet." In our frame-level approach, it has no such context by design, and so it flagged at the first sign.

In conclusion to this discussion, our study demonstrates that the exceptional performance of our model is attributable to the combination of explicit supervision, a carefully curated training process, and a well-chosen simple architecture. It reaffirms the value of ground truth annotations – they dramatically simplify the problem. It also serves as a reminder that for specific deployments, training a supervised detector might be the most reliable solution. The trade-off is that one must gather anomaly examples for each application.

5.6 Limitations and possible improvements

Despite the good performance of the new CNN, in particular, a number of limitations were identified and indicate clear directions for further advancement. Firstly, the current model is confined to frame-level operations without incorporating any temporal information. Consequently, it cannot benefit from movement continuity or sequential patterns, both of which are necessary for gradual or motion-dependent anomalies. Future extensions might see our architecture become based on sequences, as for example 3D CNN or combined CNN-LSTM models specifically designed to capture the temporal dynamics of data.

Secondly, the model is only validated on the UCSD Ped2 dataset's, which is a relatively simple and well-controlled environment. This raises an issue regarding generations: what looks good when that particular camera is pointed at some specific object in luminous conditions. A wider validation onto more data sets, along with stronger data augmentation strategies can make the system less finicky.

Third, the approach relies heavily on frame-level annotated anomalies. While this allows it to achieve good results under supervision conditions, it is not something that easily scales up to practical cases in which dense anomaly annotation has to be done. Moving towards weakly supervised or semi-supervised frameworks—where only a small subset of anomalies are labeled—could break this dependency but still extract meaningful and rewarding features from view data.

6. CONCLUSION

In this paper, we presented a highly effective supervised frame-level CNN approach for video anomaly detection, using the UCSD Ped2 dataset as a case study. By leveraging the detailed frame-wise annotations of anomalies in this dataset, our method was able to learn a direct mapping from video frame imagery to an anomaly classification with unprecedented accuracy. The CNN, composed of three convolutional layers with batch normalization and pooling, followed by a fully connected layer with dropout, proved

sufficient to capture the visual distinction between normal and anomalous events. Key to our approach was rigorous data preparation: we combined normal training frames with labelled anomalous frames, maintained class balance in training, and applied appropriate regularization during model training. This strategy yielded a model that achieved virtually perfect detection performance on the validation data, with 100% recall and 99.7% precision (F1 ≈ 99.85%), and an ROC AUC of 95%. These results dramatically outperform typical unsupervised approaches on the same dataset (which generally report AUC in the 90-95% range), highlighting the power of supervised learning when ground truth is available.

Our model's interpretations are straightforward – it identifies anomalies by the presence of learned visual features associated with abnormal events (such as the wheels of a bicycle or a person moving in a non-pedestrian manner), rather than by indirectly computed error signals. This contributes to model interpretability and transparency: one can directly inspect frames and understand why the model made a certain decision, as we did with the single misclassified frame (which contained an early glimpse of a bicycle). The simplicity of the architecture also means the model is fast and lightweight, making it suitable for real-time deployment. With only ~80,000 parameters in the dense layer and a few convolution filters, it can process frames at a high throughput on modest hardware.

We have also ensured that our study is fully reproducible and self-contained. We described all implementation details such as input preprocessing (grayscale conversion, resizing, normalization), training procedure (optimizer settings, learning rate schedule, early stopping criteria), and provided the exact performance metrics and examples of outputs. This level of detail should allow others to replicate our experiment or apply the approach to similar datasets.

In conclusion, the work advocates that when one can clearly define anomalies and obtain labelled examples, a supervised learning approach can be remarkably effective, often far more so than complex unsupervised methods. We demonstrated this in the context of UCSD Ped2. The insights gained include the importance of precise labelling, the benefit of balancing and regularizing to achieve perfect generalization, and the fact that even a relatively shallow CNN can serve as a powerful anomaly detector under supervision.

7. FUTURE WORK

Building on the findings of this research, several avenues emerge for future exploration:

7.1 Wider dataset evaluation

We plan to evaluate our supervised approach on other anomaly detection benchmarks to test its generalization. Each dataset has different scenes and anomaly types; applying our method would confirm if similar near-perfect performance can be achieved elsewhere or identify new challenges specific to those environments.

7.2 Temporal modelling extensions

While our current model analyses frames independently, many anomalies are inherently spatio-temporal. Future work includes developing temporal models such as CNN-LSTM or 3D ConvNet hybrids that consider sequences of frames.

7.3 Reducing label dependence

An interesting direction is to combine the strengths of supervised and unsupervised approaches. One idea is a semi-supervised or weakly-supervised framework where a small number of labelled anomalies guide an unsupervised model. Techniques like transfer learning or positive-unlabeled learning could leverage our results to inform models that must operate with fewer labels.

7.4 Interpretability and user feedback

To enhance trust in automated surveillance, future work could focus on explainability techniques for our CNN. For instance, using Grad-CAM or similar methods to highlight which part of the frame triggered the anomaly classification. This approach could be combined with a user-in-the-loop method, where security personnel can confirm or correct the model's detections, thereby progressively improving the system.

By pursuing these directions, we are courageous to bridge the gap between the idealized performance seen in this supervised setting and the constraints of real-world anomaly detection tasks. Ultimately, the goal is a highly accurate, interpretable, and deployable anomaly detection system that can enhance security and safety in various surveillance contexts.

ACKNOWLEDGMENT

The authors express sincere gratitude to the Department of Information Technology at the József Hatvany Doctoral School, University of Miskolc, Hungary, for the necessary support and resources for this study.

REFERENCES

[1] Rezaee, K., Rezakhani, S.M., Khosravi, M.R., Moghimi, M.K. (2024). A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance. Personal and Ubiquitous Computing, 28(1): 135-151. https://doi.org/10.1007/s00779-021-01586-5

[2] Park, H., Noh, J., Ham, B. (2020). Learning memory-guided normality for anomaly detection. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 14360-14369. https://doi.org/10.1109/CVPR42600.2020.01438

[3] Toshniwal, A., Mahesh, K., Jayashree, R. (2020). Overview of anomaly detection techniques in machine learning. In Proceedings of the 4th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), Palladam, India, pp. 808-815. https://doi.org/10.1109/I-SMAC49090.2020.9243329

[4] Liu, Y., Liu, S., Zhu, X., Yang, H., Li, J., Guo, J., Teng, L., Yang, D., Wang, Y., Liu, J. (2025). Privacy-preserving video anomaly detection: A survey. IEEE Transactions on Neural Networks and Learning Systems, 37(1): 2-21. https://doi.org/10.1109/TNNLS.2025.3600252

[5] Liu, W., Luo, W., Lian, D., Gao, S. (2018). Future frame prediction for anomaly detection—A new baseline. In 2018 IEEE/CVF Conference on Computer Vision and

Pattern Recognition, Salt Lake City, UT, USA, pp. 6536-6545. https://doi.org/10.1109/CVPR.2018.00684

[6] Sultani, W., Chen, C., Shah, M. (2018). Real-world anomaly detection in surveillance videos. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 6479-6488. https://doi.org/10.1109/CVPR.2018.00678

[7] Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D. (2008). Robust real-time unusual event detection using multiple fixed-location monitors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(3): 555-560. https://doi.org/10.1109/TPAMI.2007.70825

[8] Wang, S., Miao, Z. (2010). Anomaly detection in crowd scene. In IEEE 10th International Conference on Signal Processing Proceedings, Beijing, China, pp. 1220-1223. https://doi.org/10.1109/ICOSP.2010.5655356

[9] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S. (2016). Learning temporal regularity in video sequences. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 733-742. https://doi.org/10.1109/CVPR.2016.86

[10] Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., van den Hengel, A. (2019). Memorising normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, pp. 1705-1714. https://doi.org/10.1109/ICCV.2019.00179

[11] Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.L.A., Elkhatib, Y., Hussain, A., Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications, and research challenges. IEEE Access, 7: 65579-65615. https://doi.org/10.1109/ACCESS.2019.2916648

[12] Perini, L., Davis, J. (2023). Unsupervised anomaly detection with rejection. Advances in Neural Information Processing Systems, 36: 69673-69691.

[13] Abdalla, M., Javed, S., Radi, M.A., Ulhaq, A., Werghi, N. (2024). Video anomaly detection in 10 years: A survey and outlook. arXiv preprint arXiv:2405.19387. https://doi.org/10.48550/arXiv.2405.19387

[14] Vu, H., Nguyen, T.D., Le, T., Luo, W., Phung, D. (2019). Robust anomaly detection in videos using multilevel representations. Proceedings of the AAAI Conference on Artificial Intelligence, 33(1): 5216-5223. https://doi.org/10.1609/aaai.v33i01.33015216

[15] Sadaiyandi, J., Arumugam, P., Sangaiah, A.K., Zhang, C. (2023). Stratified sampling-based deep learning approach to increase prediction accuracy of unbalanced dataset. Electronics, 12(21): 4423. https://doi.org/10.3390/electronics12214423

[16] Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems, 33(12): 6999-7019. https://doi.org/10.1109/TNNLS.2021.3084827

[17] Chen, G., Chen, P., Shi, Y., Hsieh, C.Y., Liao, B., Zhang, S. (2019). Rethinking the usage of batch normalisation and dropout in the training of deep neural networks. arXiv preprint arXiv:1905.05928. https://doi.org/10.48550/arXiv.1905.05928

[18] Hospodarskyy, O., Martsenyuk, V., Kukharska, N., Hospodarskyy, A., Sverstiuk, S. (2024). Understanding the Adam optimization algorithm in machine learning. Congreso Internacional de Tecnologías e Innovación.

[19] Santhosh, T.R.S., Mohanty, S.N., Pradhan, N.R., Khan, T., Derbali, M. (2025). Neurovision: A deep learning-driven web application for brain tumour detection using weight-aware decision approach. Digital Health, 11: 20552076251333195. https://doi.org/10.1177/20552076251333195

[20] Pang, G., Shen, C., Cao, L., van den Hengel, A. (2021). Deep learning for anomaly detection: A review. ACM Computing Surveys, 54(2): 1-38. https://doi.org/10.1145/3439950

[21] Gayal, B.S., Patil, S.R. (2022). Detection and localization of abnormal events for smart surveillance. Ingénierie des Systèmes d'Information, 27(2): 233-241. https://doi.org/10.18280/isi.270207

[22] Almurumudhe, M.I., Hornyák, O. (2025). Motion enhanced video anomaly detection using masked autoencoder and hybrid loss functions. Annales Mathematicae et Informaticae, 61: 15-30. https://doi.org/10.33039/ami.2025.10.015

[23] Waseem, U., Hussain, T., Ullah, F.U.M., Lee, M.Y., Baik, S.W. (2023). TransCNN: Hybrid CNN and transformer mechanism for surveillance anomaly detection. Engineering Applications of Artificial Intelligence, 123: 106173. https://doi.org/10.1016/j.engappai.2023.106173

[24] Hao, Y., Li, J., Wang, N., Wang, X., Gao, X. (2022). Spatiotemporal consistency-enhanced network for video anomaly detection. Pattern Recognition, 121: 108232. https://doi.org/10.1016/j.patcog.2021.108232

[25] Gandapur, M.Q., Verdú, E. (2023). ConvGRU-CNN: Spatiotemporal deep learning for real-world anomaly detection in video surveillance system. International Journal of Interactive Multimedia and Artificial Intelligence, 8(7): 40-52. https://doi.org/10.9781/ijimai.2023.05.006

[26] Wu, J.C., Hsieh, H.Y., Chen, D.J., Fuh, C.S., Liu, T.L. (2022). Self-supervised sparse representation for video anomaly detection. In Computer Vision – ECCV 2022 Lecture Notes in Computer Science, 13673: 729-745. https://doi.org/10.1007/978-3-031-19778-9_42

[27] Chidananda, K., Siva Kumar, A.P. (2024). VidAnomalyNet: An efficient anomaly detection in public surveillance videos through deep learning architectures. International Journal of Safety and Security Engineering, 14(3): 953-966. https://doi.org/10.18280/ijsse.140326

[28] Sadatcharam, Y., Muruganadam, D. (2023). A review of deep learning algorithms for anomaly detection in videos. International Journal of Safety and Security Engineering, 13(2): 205-211. https://doi.org/10.18280/ijsse.130203