Context-Aware Zero-Shot Anomaly Detection in Surveillance Using Contrastive and Predictive Spatiotemporal Modeling

Md. Abrar Hasan[1] Md. Rashid Shahriar Khan[1] Mohammod Tareq Aziz Justice[1]
Department of Computer Science Brac University[1] Dhaka, Bangladesh
{md.abrar.hasan, rashid.shahriar.khan, mohammod.tareq.aziz@g.bracu.ac.bd}@g.bracu.ac.bd

Abstract

Detecting anomalies in surveillance footage is inherently challenging due to their unpredictable and context-dependent nature. This work introduces a novel context-aware zero-shot anomaly detection framework that identifies abnormal events without exposure to anomaly examples during training. The proposed hybrid architecture combines TimeSformer, DPC, and CLIP to model spatiotemporal dynamics and semantic context. TimeSformer serves as the vision backbone to extract rich spatial-temporal features, while DPC forecasts future representations to identify temporal deviations. Simultaneously, a CLIP-based semantic stream enables concept-level anomaly detection through context-specific text prompts. These components are jointly trained using InfoNCE and CPC losses, aligning visual inputs with their temporal and semantic representations. A context-gating mechanism further enhances decision-making by modulating predictions with scene-aware cues or global video features. By integrating predictive modeling with vision-language understanding, the system can generalize to previously unseen behaviors in complex environments. This framework bridges the gap between temporal reasoning and semantic context in zero-shot anomaly detection for surveillance.

1. Introduction

Anomaly detection is a fundamental aspect of various domains, including finance, healthcare, cybersecurity, etc., where identifying irregular patterns is crucial for ensuring security and operational efficiency. In the realm of surveillance in crowd areas, anomaly detection is particularly important for identifying potential threats. Since surveillance activity is rapidly growing in the world day by day, it has become a crucial demand to analyze videos through automated systems. As a result, video anomaly detection (VAD) has emerged as an important research area that leverages computational models to automate the identification of abnormal events in security videos. Ensuring public safety in environments such as streets, campuses, airports, and shopping malls heavily relies on effective VAD systems that can alert authorities to crimes, accidents, or other irregular behaviors as they occur. In the early stage, when the video anomaly detection system was newly introduced to us, it used to focus on manually crafted features and statistical models. Traditional computer vision algorithms used motion trajectories, object speed, or pixel-level changes to characterize normal patterns, raising alerts when deviations occur. However, these old systems had limitations as they were hand-crafted. For example, capturing complex spatiotemporal patterns in crowded or dynamic scenes was a major challenge where these approaches were not suitable. After that, a revolutionized anomaly detection technique called deep learning emerged, with the capability of learning from rich feature representations directly from data. These deep learning models automatically learn what defines normal behavior in a video, making them suitable for detecting subtle or complex anomalies that traditional algorithms often miss.

Despite advances, video anomaly detection systems still face significant challenges in efficiency and real-world applicability. A key limitation is the lack of context awareness and zero-shot generalization. Many existing methods treat all environments uniformly, ignoring that normal behavior varies with context like time or events. This leads to missed detections or false alarms in dynamic, long-term surveillance—for example, a crowd might be normal at one time but anomalous at another. Additionally, most deep learning models struggle to detect novel anomalies unseen during training, as they rely on patterns implicit in the normal data. This causes poor performance with subtle or context-dependent anomalies, such as weather-related changes. Given the rarity and diversity of abnormal events, predefining all anomalies in training is infeasible, making anomaly detection essentially a zero-shot learning problem where models must identify unknown anomalies without prior examples.

The primary objective of this research is to design a novel

deep learning model for surveillance videos that incorporates context-conditioning. The model should learn representations of normal behavior not in isolation, but in relation to contextual factors. This involves creating a dual-stream architecture where one stream models the video's spatiotemporal content and another stream provides contextual input, enabling the system to adjust its understanding of normality based on context. Secondly, leveraging contrastive learning and external semantic knowledge to allow the detection of anomalies that were never explicitly observed during training. Specifically, integrate a vision-language model like CLIP to imbue the system with semantic understanding, so that if an anomalous event corresponds to a known concept, the model can recognize it via textual descriptions even without direct training examples. Additionally, use self-supervised predictive modeling to learn generalizable temporal patterns that can flag novel deviations. Thirdly, utilizing a TimeSformer (Time-based Transformer) to capture spatial and temporal features of video sequences in a unified framework. Also, employing a contrastive predictive coding loss to train the model to predict future frame representations, reinforcing the learning of normal spatiotemporal dynamics. By combining contrastive learning (for context and semantic alignment) with predictive modeling (for temporal consistency), to ensure that the learned representation is robust to variations and sensitive to meaningful deviations. Finally, evaluating the developed model on both public benchmark datasets and a context-rich surveillance dataset rigorously. Key performance indicators include frame-level and video-level anomaly detection accuracy (e.g., measured by area under the ROC curve), as well as the false alarm rate in various contexts. So, our objective is to compare the performance against state-of-the-art anomaly detection methods (both context-agnostic and context-aware variants) to demonstrate improvements. Performing statistical analyses to verify the significance of performance gains and to quantify the contributions of each component (context conditioning, contrastive loss, predictive coding, etc.) to the overall system. Achieving these objectives will result in a context-aware, zero-shot anomaly detection system that advances the field of intelligent surveillance and has practical implications for deploying artificial intelligence in real-world security environments.

2. Related Works

Early approaches to surveillance anomaly detection were often based on **manually crafted features** and assumptions about typical motion patterns. For example, techniques like **trajectory analysis** and **optical flow** were widely used to model normal movement within a scene, with any deviations from these patterns considered anomalies [9]. Traditional methods often relied on statistical models that analyzed features such as object speed, direction, and distances between objects.

Addressing practical implementation, Sharma et al. [14] proposed a broader solution featuring not only a CNN-LSTM model but also a well-defined **system architecture**. This approach yielded the highest accuracy of 98.87% and innovatively introduced a **mobile application** using scalable cloud services to notify responsible personnel. A significant drawback, however, was the dataset, which consisted only of data from movies, potentially limiting its future real-world performance [14]. Other works focused on improving model effectiveness in complex scenarios; for instance, Nasaruddin et al. [10] introduced a hybrid of **Background Subtraction** with a **3D CNN** model, incorporating Visual Attention algorithms to enhance efficiency in diverse environments, addressing issues where previous models failed in highly crowded places.

Another line of research explored **generative models** for anomaly detection. Mahdyar Ravanbakhsh et al. [13] developed a model using a double conditional **GAN** trained exclusively on normal data. This model identifies anomalies by its inability to reconstruct unseen, abnormal regions in data frames. The approach, however, struggles to capture small or obscure abnormal objects. Concurrently, the limitations of CNNs, specifically their limited receptive field, led researchers to explore **Transformer-based architectures**. Transformers can extract features across different time steps on a wide range of inputs [11][1][15]. While a Transformer-based GAN achieved satisfactory results, it was not suitable for **real-time application** and faced challenges with computational overhead and occluded objects [18][12].

The high computational costs and limitations of earlier models spurred the development of **Zero-Shot Learning (ZSL)** techniques. The **ALFA** framework introduced by Zhu et al. [21] utilizes **Run-time Prompts (RTP)** to generate context-aware prompts, achieving a 93.2% AUROC. Another method synthesized pseudo-outliers from inlier data, eliminating the need for real anomaly samples during training and improving AUC by 15-20% [5]. Vision-language models like **CLIP** became pivotal. The **WinCLIP** model, proposed by Jeongheon et al. [5], leverages CLIP for scalable zero-shot and few-shot anomaly classification, achieving up to 95.2% image-level AUROC. Other models like **PromptAD** [7] and **AnomalyCLIP** [20] also integrated text prompts with CLIP to detect anomalies without requiring target-domain training data.

Further research has extended into **Generalized Zero-Shot Learning (GZSL)**, which tests on both seen and unseen classes, with one study achieving a harmonic mean of 65.3% by using a **CBAM** attention mechanism [8]. The most recent developments focus on **context-awareness**. Sun and Gong (2023) [16] proposed a scene-aware tech-

nique using hierarchical semantic contrast, and Yang and Radke (2024) [19] introduced the **Trinity** framework to model appearance, motion, and context. This thesis aims to synthesize these threads, focusing on the success of **contrastive learning** and **predictive modeling** to create a contextually informed, zero-shot anomaly detection model suitable for real-world requirements.

3. Methodology

Our proposed method offers a zero-shot approach to anomaly detection in surveillance videos, requiring no abnormal samples during training. The model is designed to be both context-aware and temporally predictive by leveraging a dual-stream architecture. One stream uses a transformer-based model, TimeSformer [11][18][2], to extract spatiotemporal features from video input. The other processes contextual metadata—such as time, day, or scene descriptions—using a text encoder like CLIP, generating semantic embeddings. These two streams are fused to create a joint representation that reflects the scene's behavior within its specific context.

The training employs two complementary loss functions: a contrastive loss that aligns matching video-context pairs while separating mismatches, and a predictive loss based on Contrastive Predictive Coding (CPC) [8], which encourages the model to forecast future video features. At inference, the system produces a context alignment score and a predictive score; mismatches in either signal potential anomalies. This dual-scoring mechanism enables the detection of both contextual anomalies (e.g., people in restricted areas at odd hours) and temporal anomalies (e.g., sudden or unusual movements). Using CLIP's vision-language capabilities further enhances generalization, allowing detection of novel behaviors not seen during training.

3.1. Proposed Model Architecture

The proposed Context-Aware Zero-Shot Anomaly Detection framework is a unified model that jointly learns spatiotemporal representations and semantic context for surveillance videos. The architecture consists of four main components, a global scene encoder for capturing holistic video context, a predictive modeling module for anticipating future scene dynamics, a context-conditioning network to modulate predictions based on scene context, and a CLIP-based text encoder to inject high-level semantic knowledge. These components are trained together in an end-to-end manner to learn a shared embedding space for video and textual context, enabling zero-shot inference. Figure 1 illustrates the architecture. Input video frames are processed by the TimeSformer-based encoder, whose outputs feed into the DPC-RNN predictive module. A context-conditioning subnetwork takes global scene features or associated text descriptions to produce modulation parameters that inform the predictive module. In parallel, a text encoding branch using CLIP's language encoder provides a semantic context vector. All feature streams are projected into a common embedding space and optimized with a contrastive InfoNCE loss and a hybrid Contrastive Predictive Coding (CPC) loss [17]. This design allows the model to learn what constitutes "normal" patterns in a scene and detect deviations as anomalies without explicit anomaly examples, in a zero-shot way. Next, we detail each module and the training methodology.

3.1.1. Zero-Shot Learning (ZSL)

What if a machine learning model could learn and work without ever needing labeled data? This is precisely where Zero-Shot Learning comes into play. It is a machine learning algorithm that enables a model to identify or classify new instances of data or concepts that it had not previously seen or was not explicitly trained on such labeled data as well as having the ability handle data for which it wasn't particularly trained. It allows the model to recognize things that it had never encountered. It utilizes the information it already knows and connects it to the new situation, which was presented in front of it. Let me explain this with an example: Suppose a model is trained to recognize animals but not particular ones like Zebra in its training phase. In case of other machine learning approaches, the model will fail to classify an animal being Zebra because it was never explicitly shown any labeled data of Zebra during its training period. But ZSL will be able to figure it out by using description even though it was also not shown any examples of it. The ZSL model knows about the information of animals from its training phase like 'they have four legs,' 'has stripes' or 'has a tail'. The ZSL model is given a new description that reads as, 'A horse is an animal with black and white stripes and a tail that lives in Africa.' Using its knowledge on animal attributes and characteristics along with the given description it will successfully deduce that the animal is in fact a Zebra even though it had no prior knowledge about it during its training. This is the power of Zero-Shot Learning. It uses these auxiliary information to its advantage.

3.1.2. Contrastive Predictive Coding (CPC)

This is an unsupervised learning approach that extracts useful representations from high-dimensional data. The key idea behind this is to learn such representation by predicting the future latent feature representation through the use of autoregressive models. The key insight here is that using past contexts, CPC predicts future parts of a sequence but not by directly generating the data but by distinguishing the true future from the false ones, a learning procedure known as contrastive learning. In this case, a probabilistic contrastive loss is used alongside negative sampling. It learns rich temporal data without labels, which is ideal for tempo-

ral modeling. **Equation 3**, $\mathcal{L}_{align+pred}/\mathcal{L}_{total}$, shows how we combined these two losses for our particular task.

3.1.3. TimeSformer for Spatiotemporal Features

To capture the overall appearance and movement patterns in surveillance videos, we use TimeSformer, a transformer-based model designed specifically for video understanding[1]. TimeSformer treats a video not as a single image sequence but as a series of smaller visual patches. Each frame in the video is split into patches, which are then turned into vectors through linear projection[3]. To help the model understand the position and order of these patches, we add spatial and temporal position information to each one.

These processed patches form a long sequence of tokens that are passed through multiple transformer layers. Through self-attention mechanisms, the model learns how different patches relate to each other across both space and time. This helps the model capture long-range dependencies and understand the broader context of the video clip. We use a specific variant of TimeSformer known as "divided space-time attention," which separates the spatial and temporal attention computations to make the model more efficient, especially for longer video clips. The result of this processing is a rich sequence of feature vectors that describe the video at each moment. In our implementation, we use these outputs in two main ways. First, the feature from each frame is passed to a predictive model that learns how the scene evolves over time. Second, these features are mapped into a shared embedding space so that they can be compared directly with semantic context information from text descriptions.

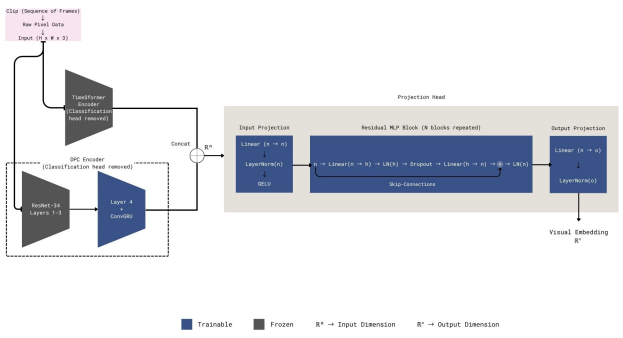

Figure 1. Video Architecture Pipeline

By using TimeSformer, our model can pay attention to all parts of a scene at once. This is especially important in anomaly detection, where unusual events often involve complex interactions or context-sensitive behavior. Unlike traditional convolutional networks which focus on local patterns, TimeSformer's global self-attention enables it to understand the full picture. This makes it ideal for modeling what is "normal" in different surveillance scenarios.

3.1.4. Temporal Predictive Modeling with DPC-RNN

Figure 1 illustrates how the video pipeline is structured in its entirety. Layer4 & ConvGRU of DPC-RNN [4] is fine-tuned for domain adaption. Two separate inputs enter both the feature extractor backbones, varying in terms of the number of frames that the backbones anticipate. Their initial output embeddings are concatenated and is then passed to the residual MLP projection block, where it will be projected to the CLIP's space. We used multiple residual blocks in the projection head, which will help make the training stable and the transmission of information and gradients easier. Furthermore, the Projection Head will learn to project its output into CLIP's space in such a way that visual embeddings will land closer to its corresponding textual embedding and further away from negatives. Firstly, inside the input projection, the concatenated visual embedding passes through a Linear layer, which is a FCNN or a Dense layer, and the layer will learn a Linear transformation of the input features. The output from this is then normalized using Layer Normalization, which helps to stabilize the training process. Also ensures that the network's layer receives data information with a consistent distribution with a mean of 0 and standard deviation of 1. A Gaussian Error Linear Unit (GELU) is used to introduce non-linearity into the model. The Residual MLP Block is the core processing unit of the Projection Head. As mentioned above, N residual blocks are used, which helps the model to learn progressively more abstract and refined features of the input as a result of stacking of these blocks. The arrow from the input to the final output of this block that bypasses the inner layers, LN, activations is a 'skip' or 'residual' connection. This is the essence of a Residual Network (ResNet) which solves the vanishing gradient problem. If we trace the path inside this block, the output from the Input Projection block is passed through a Linear layer which projects it to a higher dimension h allowing the model to learn more richer representation. This a h-dimensional hidden vector. LN is applied to it. Also a regularization method is used here, allowing the model to generalize better to unseen, new data. After that the h-dimensional hidden vector is projected back to the original dimension. A skip-connection allows for the original input to be added to the output of this final linear layer. The entire sequence of this block is repeated N times. It is then normalized and passed to the Output Projection block as input. The input is passed to through another linear layer where its projected to the specific dimension and normalized and thus produces the visual embedding, where $\mathbf{v} \in \mathbb{R}^d$.

The training strategy follows a contrastive approach which is using the infoNCE loss. The model learns to make its predicted future feature similar to the actual future feature while making it dissimilar from unrelated ones. These unrelated or negative examples are typically taken from other parts of the same video or from different videos in the same batch. This way, the model learns meaningful patterns that consistently appear in normal videos, while ignoring small random changes like noise or slight movement that don't indicate anomalies. By summing the predictive loss over multiple future steps and across time, the DPC-RNN becomes skilled at learning how scenes usually unfold [6]. This is useful for detecting anomalies because during the training phase the model only sees normal video clips. If something unusual happens later it won't be able to predict it accurately, resulting in a large difference between the predicted and actual features. This prediction error becomes a clear signal that something unexpected is happening in the video.

3.1.5. Context-Conditioning Network for Scene-Aware Modulation

A key innovation in our model is its ability to adjust its behavior based on the specific scene it is observing. In real world surveillance, what is considered "normal" varies greatly from one location to another [19]. For example, heavy pedestrian activity might be typical in a shopping mall but highly unusual in a restricted area. To account for this, we use a Context-Conditioning Network that adjusts the model's predictions according to scene-specific information.

The context used to guide the model can come from two sources, visual features and textual descriptions. Visually, the TimeSformer encoder captures static aspects of a scene, such as its layout or common background elements. Textually, we can provide a natural language description of the scene which is encoded using the CLIP text encoder. These inputs are processed to form a context vector that is used to adjust how the prediction model behaves. In our case, raw pixel data from a single mid-frame of a clip is provided to the Context-Conditioning Network as its input. It outputs a context vector embedding and is passed through a ContextGate Block which is FCNN or MLP. We introduced a learnable parameter here, β, as illustrate in **Figure 2**, which is used to control how much information will be passed to the textual embedding so that not all its own information is washed out by the context vector. The ContextGate learns over time as to how much information is needed by the textual embedding to understand the surrounding context or whether it is capable of determining the context by itself or not. It is then fused with the text embedding through residual addition. Alternatively, the context vector can be directly added to the visual features before transferring it to the RNN. These adjustments help the model make scene-specific predictions and reduce false alarms.

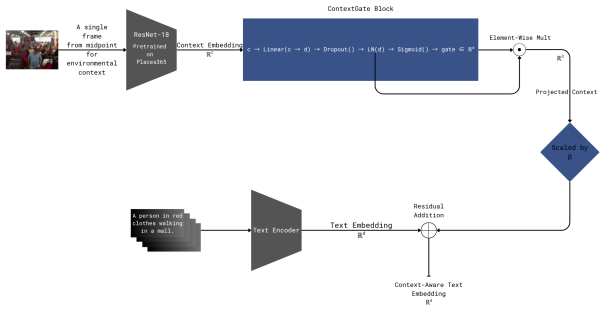

Figure 2. Text Pipeline

This design enables the model to adapt to new environments without retraining, which is essential for zero-shot anomaly detection. It allows the system to understand what kind of behavior is expected in a given context and to detect deviations effectively.

3.1.6. CLIP-Based Text Encoder for Semantic Context

In our system, we use the CLIP text encoder as shown in **Figure 2** is used to convert descriptive sentences about a scene into feature vectors. These descriptions define what is typically expected in a given setting. Each sentence is transformed into a semantic embedding that represents the normal behavior or appearance of a specific scene. Rather than relying on CLIP's image encoder, we train our TimeSformer-based video encoder to produce embeddings that align with these text representations. During training, we use contrastive learning to bring the video embedding closer to the matching context text and push it away from unrelated ones. This teaches the model to understand and align the video content with the semantic expectations.

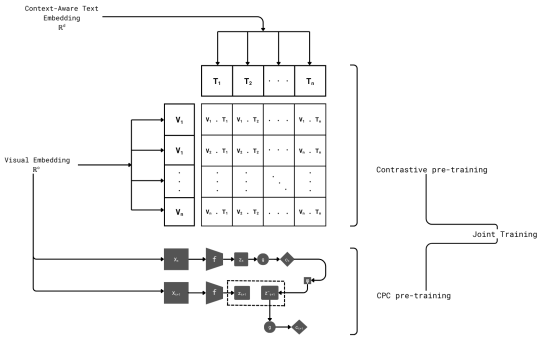

Figure 3. Joint training on prediction loss & infoNCE

3.1.7. Joint Feature Fusion and Training Objective

Among the major components of our system, the TimeSformer feature extractor backbone is kept frozen, only the last two layers of DPC-RNN predictor are fine-tuned for domain adaption and the rest are kept frozen, the context-conditioning module is learnable, and the CLIP text encoder is kept frozen, the residual MLP projection is trainable and together using a unified training loss we effectively train our anomaly detection system. Feature fusion happens at several points in this architecture to ensure that the model can leverage both visual and contextual information.

Firstly, the input raw pixel data from varying frames are sent to both TimeSformer and DPC and later their feature output is then concatenated and projected into the CLIP's embedding space where the textual embeddings already resides. The projection is done using residual MLP which ensures that the model can capture complex patterns. Also, the residual addition provide the MLP a shortcut which is that if the layer by layer processing is not useful the FCNN can reatin its original input without little to no changes through the use of slip-connections as illustrated in Projection Head section in the **Figure 1** while in other cases it adds a small change instead of complex transformation at every layer. This small tweaks to the input makes the learning faster and easier. This determines whether a layer is useful or not and whether it will learn useful changes or not, acting like a memory lane for gradients and information, helping them flow easily.

Later, we project the outputs of both the video and text branches into a shared embedding space, ensuring they are directly comparable. We use learnable linear layers to adjust the video features so they align with the fixed dimension of CLIP's text embeddings. The predictive loss teaches the RNN to forecast future visual representations of normal events, while the contrastive loss ensures that the video features align with the correct text description which is depicted in the **Figure 3** where the selected layers and models are trained on both training objectives and the weighted sum of the two backpropagated infoNCE-style objectives. **Equation 1** $\mathcal{L}_{\text{align}}$ is the classic contrastive term between visual embedding and correct textual embedding.

$$\mathcal{L}_{\text{align}}(i) = -\log \frac{\exp\left(\text{sim}\left(v_i, t_i\right)/\tau\right)}{\sum_k \exp\left(\text{sim}\left(v_i, t_k\right)/\tau\right)}. \tag{1}$$

Here, with each sample i, consisting of the visual embedding v_i and the positive textual embedding t_i along with a set of negatives t_k. The training objective here is to bring the positive visual-textual pair closer to each other while pushing away the negatives as much as possible.

The Contrastive Predictive Coding (CPC) on the other hand, also implements the core infoNCE loss but instead of video-text pairs, they predict future latent representation z_{i+1} from a given context embedding c_i at time-step i and

$$\mathcal{L}_{\text{pred}}(i) = -\log \frac{\exp\left(f\left(c_i\right)^\top z_{i+1}\right)}{\sum_k \exp\left(f\left(c_i\right)^\top z_k\right)}, \tag{2}$$

pulls positive pairs, that being, the prediction and the ground truth at next-step $i+1$ while simultaneously pushing the negative set z_k further away. **Equation 2** $\mathcal{L}_{\text{pred}}$ illustrates the prediction loss. $f(.)$ is the small MLP used here for prediction.

Both losses are combined into a single-joint loss which is scaled by α, deciding how much weight each loss will contribute during training and testing.

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{align}} + (1-\alpha)\mathcal{L}_{\text{pred}} \tag{3}$$

The typical value for the weighting factor as illustrated in **Equation 3** α is 0.5.

The CLIP text encoder remains fixed to preserve its pre-learned semantic knowledge, while the rest of the model is fine tuned.By the end of training, the model has learned to group normal behavior patterns properly around their corresponding context descriptions which allows it to flag any deviation as an anomaly without ever having seen abnormal events during training while also being able to use the prediction loss for a surprise during the testing flagging behaviors early that strays away from the normal line of projection.

3.1.8. Adaptive Inference and Scene-Conditioned Anomaly Detection

During inference, the model adaptively processes video frames to compute anomaly scores at each time step. Although trained on short clips, the DPC-RNN retains temporal context, enabling analysis of longer sequences during testing for improved stability and reduced false positives. The model incorporates scene-specific context by using text descriptions corresponding to each camera view, allowing it to compare predicted vs. actual frames and assess alignment with expected behavior. Anomalies are flagged when either prediction error or context mismatch is high. Detection thresholds can be scene-specific, and using CLIP's shared embedding space allows for potential explanations of anomalies through comparison with descriptive texts.

4. Our Contributions

We have contributed in three key areas of our proposed methodology. The first key contribution is that our entire end-end pipeline is trained purely in a Zero-Shot manner. That is, we have achieved pure Zero-Shot Learning through our own modus operandi. We did not expose our model to any other information beyond normal behavior, which is what it was trained entirely on. This is what we call true Zero-Shot Anomaly Detection (ZSAD). We were able to achieve this because of a joint training objective that we implemented for our model. This is the second key contribution of our work, which is adapting the alignment loss and prediction loss from contrastive loss. The former being the core to CLIP's ZSL approach, and the latter from Contrastive Predictive Coding (CPC). Using this combined training objective is what led us to attain a pure Zero-Shot scenario. Our last contribution lies in adapting the textual embeddings to the surroundings of the events described, effectively making the textual embeddings context-aware and in turn rendering our whole pipeline the same. This key addition makes the model capable of differentiating place, time, and scenarios with respect to the events that occurred and in turn helped reduce false positives/alarms.

5. Experiments

To evaluate the result of our proposed model, we performed a comparative analysis against several methods that have already worked on the UCF-Crime dataset, a benchmark that is widely used for anomaly detection in surveillance scenarios. We focused on key evaluation metrics ROC-AUC and PR-AUC, which are essential for measuring a model's ability to distinguish anomalies, especially in the context of unbalanced surveillance data.

Table 1. Comparison of ROC-AUC & PR-AUC scores for anomaly detection models with zero-shot learning on the UCF-Crime dataset.

Model (Approach)	Pure Zero-Shot?	Text?	ROC-AUC ↑	PR-AUC ↑
Flashback (ViT-L)	No	No	87.3 %	75.1 %
Our Model (CLIP + DPC + TSF)	**Yes**	**Yes**	**84.5 %**	**72.3 %**
AnomalyCLIP (ViT-B/16 + CLIP)	No	Yes	82.4 %	68.7 %
ViT-I3D	No	No	72.1 %	57.2 %
Inflated 3D-CAE (I3D)	No	No	68.0 %	51.4 %

Table 1 presents a performance comparison on the UCF Crime Dataset using ROC-AUC and PR-AUC metrics. While Flashback (ViT-L) achieves the highest scores (87.3,% ROC-AUC, 75.1,% PR-AUC), it relies solely on visual features without leveraging semantic context. Our hybrid model—combining TimeSformer, DPC-RNN, and CLIP—achieves a competitive 84.5,% ROC-AUC and 72.3,% PR-AUC, making it the top performer among zero-shot and vision-language-based methods. It outperforms AnomalyCLIP (82.4,%, 68.7,%), which lacks predictive and temporal modeling. Traditional vision-only models like ViT-I3D and Inflated 3D-CAE perform significantly worse, highlighting our model's strong balance of precision, generalization, and context-aware anomaly detection.

Table 2. Comparison of mAP (%) & Detection Delay (s) for anomaly detection models with and without zero-shot learning.

Model (Approach)	ZSAD?	mAP (%)	Detection Delay (s)
Our Model (CLIP + DPC + TSF)	**Yes**	**62.5 %**	**0.45**
Multimodal Asynchronous Hybrid Net	No	54.2 %	0.05
Rethinking VAD (Continual Learning)	No	48.7 %	0.60
VADA / RTFM-style (Weak Supervision)	No	52.1 %	0.80
Flashback (Memory-Driven ZSAD)	Yes	45.5 %	1.20

As shown in **Table 2**, our hybrid model—integrating TimeSformer, DPC-RNN, and CLIP—achieves superior performance with a high mean Average Precision (mAP) of 62.5,%, outperforming Flashback (45.5,%) and Multimodal Asynchronous Hybrid Net (54.2,%). It also offers low detection latency at 0.45 seconds, faster than Flashback (1.2s) and competitive with models like Rethinking VAD (0.60s) and VADA-style methods (0.80s). Although Multimodal Asynchronous Hybrid Net reports lower latency (0.05s), it relies on specialized hardware, limiting its general applicability.

5.1. Ablation study

Ablation study is done to dissect our entire architecture and evaluate each module separately to understand how much each module contributed to the entire pipeline. This is used to determine whether the modules used provide any useful gains or just an overhead to the overall architecture. The **table 3** below shows the progressive run in which we made meaningful changes to the overall pipeline and how much each addition contributed to the final result.

Table 3. Ablation on **UCF-Crime**. Each row toggles a single factor; training budget and optimizer settings are otherwise identical.

Variant	ROC-AUC ↑	PR-AUC ↑
B1 CLIP+TSF, concat, γ=1 (no DPC)	46.2 %	22.3 %
B2 + DPC predictive loss (γ=0.5, α=0.8)	51.8 %	39.6 %
B3 B2 + re-balance α=0.2	57.9 %	47.5 %
B4 B3 + Residual MLP Projection (1024→512)	70.4 %	61.1 %
B5 B4 + LN-Gate (β residual) **(our final)**	**84.5 %**	**72.3 %**

- **B1: CLIP + TSF, concat, $\gamma = 1$ (no DPC)** This variant serves as the baseline. It utilizes CLIP (Contrastive Language-Image Pre-training, a common base for multimodal tasks) and TSF (Temporal Self-Attention/Features). Here, the term concat refers to combining the features through concatenation, which are extracted from CLIP and TimeSformer. The notation "$\gamma = 1$ (no DPC)" implies a specific configuration of a

parameter where DPC (a variant of Contrastive Predictive Coding extended to the video domain) remains inactive. Its performance, with ROC-AUC at 46.2% and PR-AUC at 22.3%, establishes a starting point for optimization.

- **B2: + DPC predictive loss** ($\gamma = 0.5, \alpha = 0.8$) Building based on `B1`, this variant introduces or activates the DPC predictive loss, making the inclusion of a predictive component aimed at enhancing temporal understanding or anomaly prediction. Specific hyperparameters for this loss are provided: "$\gamma = 0.5$" and "$\alpha = 0.8$". The main advantage of DPC predictive loss is improving performance. After activating this, ROC-AUC has risen to 51.8 % and PR-AUC has risen to 39.6 %. The substantial increase in PR-AUC is particularly noteworthy, as this metric is often more sensitive to the imbalanced datasets typical of anomaly detection tasks.

- **B3: B2 + re-balance** $\alpha = 0.2$ This variant enhances `B2` by introducing a re-balance mechanism with a parameter "$\alpha = 0.2$". It involves an adjustment to mitigate the effects of class imbalance, a critical challenge in anomaly detection where anomalous events are rare. The re-balancing could be implemented through adjusted loss weights or specific sampling strategies. This modification further improves both metrics, with ROC-AUC that reach to 57.9% and PR-AUC to 47.5%, indicating better handling of rare anomalous events.

- **B4: B3 + Residual MLP Projection (1024→512)** This variant advances from `B3` by including a residual MLP Projection. This denotes the addition of a Multi-Layer Perceptron (MLP) with a residual connection, designed to project features from a dimension of 1024 down to 512. This step serves for dimensionality reduction, feature refinement, and increasing the capacity of the model to learn intricate relationships within the features. The inclusion of residual connections typically aids in the stable training of deeper networks. This addition results in a very significant performance boost, with ROC-AUC jumping to 70.4% and PR-AUC to 61.1%, suggesting effective refinement of the learned representations.

- **B5: B4 + LN-Gate (β residual) (our final)** This is the last phase that is considered as our most advanced variant, built on `B4`. It integrates two key components, where one is the LN-Gate (Layer Normalization Gate). It refers to a mechanism that controls information flow through a layer, potentially enhancing model stability and learning efficiency. Another is β Residual, which is a special type of residual connection integrated with the LN-Gate. It used to allow better gradient flow or feature refinement This variant is explicitly highlighted as our final model. It achieves the highest performance among all evaluated variants, with a strong ROC-AUC of **84.5%** and PR-AUC of **72.3%**, demonstrating the effectiveness of the LN-Gate in providing a substantial final boost to anomaly detection capabilities.

Key observations: The ablation study highlights the architectural enhancements of our model, showing how the inclusion of `Residual MLP Projection` and `LN-Gate` progressively improves anomaly detection performance. Each model variant from `B1` to `B5` exhibits consistent gains in both ROC-AUC and PR-AUC metrics. The final model `B5` achieves notable improvements of **+38.3% in ROC-AUC (from 46.2% to 84.5%)** and **+50.0% in PR-AUC (from 22.3% to 72.3%)** over the baseline `B1`, emphasizing the effectiveness of these tailored components in enhancing feature refinement and information flow for complex anomaly detection tasks.

6. Conclusion

In conclusion, this thesis introduced a context-aware zero-shot anomaly detection framework for surveillance system that integrates contrastive and predictive spatiotemporal modeling. It bridges the gap between low-level pattern anomaly detection and high-level situational awareness. Leveraging contextual information to tailor anomaly detection for scene-specific conditions, the proposed method integrates contrastive representation learning with predictive normalcy modeling to detect anomalies without the need for any training examples of abnormal data. Experimental results demonstrated that this method achieves high detection accuracy while maintaining a low false-positive rate. It effectively generalizes to previously unseen anomalous events across various scenarios. These findings highlight the potential of the proposed framework for real-world surveillance applications, as its modular, context-aware design allows for adaptation to new environments. While our approach has made notable progress, it is not without limitations. The dependency on context data means the system's performance can degrade if such data are missing or incorrect. Also, extremely subtle anomalies or those involving complex interactions remain challenging. Future work has been proposed to address several issues, including improving context auto-discovery, enriching context modalities, enabling the model to explain anomalies in natural language, and refining the model for better efficiency and adaptability

Bibliography

[1] N. Aslam and M. H. Kolekar. TransGANomaly: Transformer based generative adversarial network for video anomaly detection. *Journal of Visual Communication and Image Representation*, 100:104108, 2024. 2, 4

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 3

[3] K. Deshpande, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and S. Agarwal. Anomaly detection in surveillance videos using transformer based attention model. In *Communications in Computer and Information Science*, pages 199–211, 2023. 4

[4] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. *arXiv preprint arXiv:1604.04574*, 2016. 4

[5] J. Jeong, Y. Zou, T. Kim, D. Zhang, A. Ravichandran, and O. Dabeer. WinCLIP: Zero-/few-shot anomaly classification and segmentation. *arXiv preprint arXiv:2303.14814*, 2023. 2

[6] H. K. Joo, K. Vo, K. Yamazaki, and N. Le. CLIP-TSA: CLIP-Assisted temporal self-attention for weakly-supervised video anomaly detection. *arXiv preprint arXiv:2212.05136*, 2022. 5

[7] Y. Li, A. G. David, F. Liu, and C. Foo. PromptAD: Zero-shot anomaly detection using text prompts. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1082–1091, 2024. 2

[8] X. Liu, W. Luo, J. Du, X. Wang, Y. Dang, and Y. Liu. A robust generalized zero-shot learning method with attribute prototype and discriminative attention mechanism. *Electronics*, 13(18):3751, 2024. 2, 3

[9] W. Luo, W. Liu, and S. Gao. A revisit of sparse coding based anomaly detection in stacked RNN framework. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 341–349, 2017. 2

[10] N. Nasaruddin, K. Muchtar, A. Afdhal, and A. P. J. Dwiyantoro. Deep anomaly detection through visual attention in surveillance videos. *Journal of Big Data*, 7(1), 2020. 2

[11] Marwa Qaraqe, Y. D. Yang, E. B. Varghese, E. Basaran, and A. Elzein. Crowd behavior detection: leveraging video swin transformer for crowd size and violence level analysis. *Applied Intelligence*, 54(21):10709–10730, 2024. 2, 3

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2

[13] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo S. Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *International Conference on Image Processing*, 2017. 2

[14] S. Sharma, B. Sudharsan, S. Naraharisetti, V. Trehan, and K. Jayavel. A fully integrated violence detection system using CNN and LSTM. *International Journal of Electrical and Computer Engineering (IJECE)*, 11(4):3374, 2021. 2

[15] Wajid Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. *arXiv preprint arXiv:1801.04264*, 2018. 2

[16] S. Sun and X. Gong. Hierarchical semantic contrast for scene-aware video anomaly detection. *arXiv preprint arXiv:2303.13051*, 2023. 2

[17] Aaron Van Den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[18] C. Yang, S. Lan, W. Huang, W. Wang, G. Liu, H. Yang, W. Ma, and P. Li. A transformer-based GAN for anomaly detection. In *Lecture Notes in Computer Science*, pages 345–357, 2022. 2, 3

[19] Z. Yang and R. Radke. Context-aware video anomaly detection in long-term datasets. *arXiv preprint arXiv:2404.07887*, 2024. 3, 5

[20] Q. Zhou, G. Pang, Y. Tian, S. He, and J. Chen. AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2310.18961*, 2023. 2

[21] J. Zhu, S. Cai, F. Deng, and J. Wu. Do LLMs understand visual anomalies? Uncovering LLM capabilities in zero-shot anomaly detection. *arXiv preprint arXiv:2404.09654*, 2024. 2