Project Report

Anomaly-Driven Video Summarization for Real-Time Surveillance Systems

1. Project Information:

Title: Anomaly-Driven Video Summarization for Real-Time Surveillance Systems

Team:

- Vishnu Priyan Sellam Shanmugavel - A20561323
- Akash Thirumuruganantham - A20539883

1.1. Main Paper:

Name: Real-world Anomaly Detection in Surveillance Videos [2019]
Authors: Waqas Sultani , Chen Chen , Mubarak Shah

1.2. Additional Paper:

Name: A Three-Stage Anomaly Detection Framework for Traffic Videos [2022]
Authors: Junzhou Chen, Jiancheng Wang, Jiajun Pu, and Ronghui Zhang

1.3. Option Chosen

This project follows Option 2: Improving the Paper by modifying the loss function to incorporate Temporal Smoothness Loss and Ranking Loss, utilizing C3D and 3D ResNet architectures for spatial and temporal feature extraction. Additionally, a video summarization approach was implemented, involving the extraction of keyframes and segment scoring.

1.4. Substantial Code Modifications

The following significant changes were made to the original implementation to address the challenges of anomaly detection in video data:

1.4.1. Loss Function

The MIL ranking loss function was enhanced by integrating Temporal Smoothness Loss and Ranking Loss components. These modifications improved anomaly localization by better capturing the temporal dynamics within video sequences.

1.4.2. Feature Extraction

The feature extraction pipeline was upgraded by incorporating both C3D and 3D ResNet architectures. This dual approach ensures comprehensive extraction of both spatial and temporal features, critical for effective anomaly detection.

1.4.3. Video Summarization

A scoring mechanism was introduced to rank video segments based on anomaly likelihood. Additionally, a method for generating keyframes was implemented to provide a summarized visual output, aiding in better interpretability of detected anomalies.

2. Problem Statement:

Surveillance systems generate large volumes of video data, which are typically collected and stored passively, leading to inefficiencies in monitoring and retrieval. The manual review of such video data is labor-intensive and prone to errors. The key challenge is to automatically detect anomalies within the videos and summarize the most relevant footage for efficient storage and quick threat identification. This project addresses how to optimize video processing to detect and summarize anomalous activities in surveillance footage without requiring real-time analysis.

2.1. Critical Challenges

The following challenges were addressed in this study to improve anomaly detection in video data:

1. Real-Time Anomaly Detection: Identifying unusual or suspicious activities quickly to enable prompt action.
2. Efficient Data Management: Summarizing lengthy video footage into concise, relevant segments for simplified storage and retrieval.
3. Scalability: Managing the increasing volume and complexity of video data without relying heavily on human resources to manually review footage.

2.2. Importance of the Problem

2.2.1. Enhanced Security

Automating anomaly detection minimizes errors arising from human oversight and ensures faster responses to security threats.

2.2.2. Cost Efficiency

Summarizing video data significantly reduces storage requirements and operational costs associated with monitoring personnel.

2.2.3. Practical Applications

The ability to pinpoint and extract meaningful events from surveillance footage is vital for applications in law enforcement, traffic monitoring, and public safety management.

3. Proposed Solution and Implementation Details

To address the challenges in anomaly detection and video summarization for surveillance systems, we developed a pipeline that segments video data into two classes: normal and anomaly. This approach ensures significant, anomalous events are effectively highlighted and summarized, while normal footage is deprioritized.

3.1. Shot Segmentation

The video is divided into temporal segments to isolate specific scenes for detailed analysis. Each segment is classified into one of two classes: **normal** or **anomalous**. This segmentation facilitates focused detection of events, ensuring efficient processing and accurate classification.

To optimize storage and computation, the segmented data is stored in .npy format, allowing for faster loading and processing during feature extraction and classification stages.

3.2. Feature Extraction

The proposed method integrates two advanced feature extraction techniques to ensure robust representation of both spatial and temporal features:
C3D (Convolutional 3D Network): Extracts spatial and temporal features to capture motion and scene context within each segment.
3D ResNet: Enhances feature representation for robust classification.
Features are extracted using both C3D and 3D ResNet for each video segment, combining their strengths to improve anomaly detection accuracy.

3.3. Loss Function Modification

To improve anomaly localization and capture temporal dependencies, the following loss

3.3.1. Ranking Loss

Minimizes intra-class variance and maximizes inter-class differences, ensuring anomalous segments are assigned higher scores than normal ones.

$$loss = y_{true} \,.\, (y_{true} - y_{pred})^2 + (1 - y_{true}) \,.\, max(0,\, m - (y_{true} - y_{pred})^2)$$

3.3.2. Temporal Smoothness Loss

Enforces consistency across consecutive frames, accounting for temporal structure in the video data.

$$loss = mean(|y_{pred}[i + 1] - y_{pred}[i]|)$$

3.4. Classification

The video segments are classified into two categories: normal and anomaly. This classification is based on the anomaly scores generated by the model, which are derived from the fused features extracted from both C3D and 3D ResNet (as shown in **Figure 1**). The anomaly scores reflect the significance of each segment, determining whether it exhibits abnormal patterns or behaviors.
The model utilizes a **combined loss function** that balances **Binary Cross-Entropy (BCE)**, **Ranking Loss**, and **Temporal Smoothness Loss** to optimize the classification process. The combined loss is defined as:

$$total\_loss = BCE(y_{true,} y_{pred}) + w_r \,.\, ranking\ loss - w_s \,.\, temporal\ loss$$

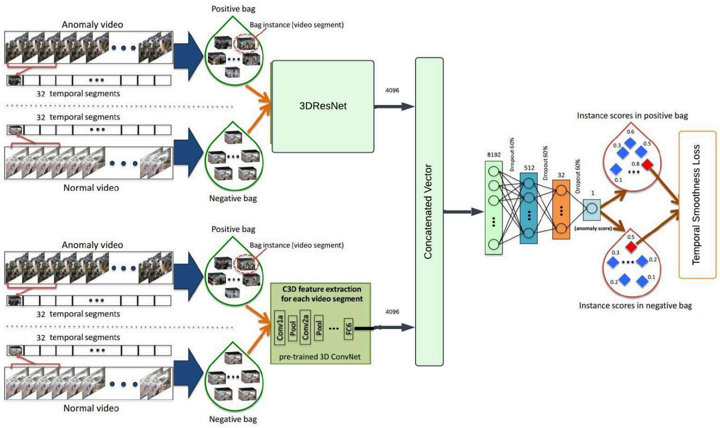


Figure 1. *Proposed Anomaly detection approach (Feature Extraction using both C3D and 3D ResNet)*

Ranking Loss is added, while Temporal Smoothness Loss is subtracted for the following reasons:

3.4.1. Ranking Loss (Added)

Ranking Loss is added to the total loss to optimize the ranking of video segments based on their anomaly scores. The objective of Ranking Loss is to minimize the distance between positive pairs (anomalous segments) and maximize the margin for negative pairs (normal segments). This encourages the model to prioritize anomalous segments by pushing them further apart from normal segments in the feature space. This addition helps in enhancing the model's ability to distinguish anomalous segments from normal ones, improving classification performance by ensuring that anomalous segments are given higher anomaly scores and ranked accordingly.

3.4.2. Temporal Smoothness Loss (Subtracted)

Temporal Smoothness Loss is subtracted to enforce consistency across consecutive frames in the video. The purpose of this loss is to ensure that predictions for neighboring frames remain smooth and coherent, preventing large fluctuations in the model's output from one frame to the next. This is particularly important in video data, where anomalies often evolve gradually over time.

By subtracting this loss, the model is penalized for sharp, disjointed changes in the predictions between consecutive frames, encouraging it to detect anomalies in a more stable and temporally consistent manner. This penalty helps the model capture continuous patterns of anomalous behavior over time, which is critical for anomaly detection in dynamic video scenarios.

3.4.3. Overall Effect

The inclusion of Ranking Loss ensures that anomalous segments are correctly prioritized and ranked higher in the feature space, while the subtraction of Temporal Smoothness Loss ensures that the model's predictions remain consistent over time. Together, these two losses help the model improve anomaly detection performance by striking a balance between distinguishing anomalies and maintaining temporal coherence in the predictions.

3.5 Model Comparison and Evaluation

In this section, the performance of three different model configurations, C3D, 3D ResNet, and C3D + 3D ResNet, was evaluated both with and without the custom loss function. The purpose of this comparison is to assess the impact of incorporating the custom loss in anomaly detection tasks, with the goal of determining the optimal model and feature extraction method. The custom loss function combines Binary Cross-Entropy (BCE), Ranking Loss, and Temporal Smoothness Loss, aiming to enhance anomaly localization and overall classification performance.

3.5.1. C3D Model: With and Without Custom Loss

Without Custom Loss: The C3D model captures both spatial and temporal features by processing video data in 3D convolutions. While it is effective in recognizing motion patterns, its performance in distinguishing between normal and anomalous segments is limited without the custom loss. Relying solely on BCE, the model struggles to accurately localize anomalies, resulting in higher false positives and negatives.

With Custom Loss: Incorporating the custom loss significantly improves the C3D model's performance. The Ranking Loss helps to prioritize anomalous segments, while the Temporal Smoothness Loss enforces consistency in the predictions across consecutive frames. This combination of losses enables the model to more effectively differentiate between normal and anomalous events, reducing classification errors and improving overall accuracy.

5.5.2. 3D ResNet Model: With and Without Custom Loss

Without Custom Loss: The 3D ResNet model focuses primarily on spatial feature extraction, leveraging residual networks to capture complex spatial patterns. While this makes the model effective for recognizing intricate features within each frame, it lacks sensitivity to temporal changes. As a result, it struggles with detecting anomalies that span multiple frames or involve temporal dependencies, which limits its effectiveness in dynamic video sequences.

With Custom Loss: When the custom loss is applied, the 3D ResNet model's performance is significantly enhanced. The Ranking Loss assists in identifying and prioritizing anomalous segments, while the Temporal Smoothness Loss ensures that predictions remain consistent across adjacent frames. This improved temporal modeling allows the 3D ResNet to better detect anomalies in time-series video data, making it more robust in handling dynamic scenarios.

3.5.3. C3D + 3D ResNet Model: With and Without Custom Loss

Without Custom Loss: Combining C3D and 3D ResNet provides a comprehensive feature extraction mechanism that captures both spatial and temporal information. However, without the custom loss, the model does not effectively reconcile the spatial features from 3D ResNet and the temporal features from C3D. This can lead to conflicts in feature extraction, resulting in a model that underperforms in terms of anomaly detection, with inconsistent predictions and higher error rates.

With Custom Loss: The integration of the custom loss into the C3D + 3D ResNet model improves performance by addressing feature fusion issues. The Ranking Loss helps prioritize anomalous segments, and the Temporal Smoothness Loss ensures consistent predictions across frames. This combination helps the model effectively balance spatial and temporal feature extraction, leading to improved anomaly localization and detection accuracy.

3.5. Video Summarization

Anomaly-classified segments are prioritized, and keyframes are extracted to create a concise video summarization, highlighting only the most relevant footage. Anomaly scores are computed using the ranking mechanism, evaluating segments on memorability, entropy, and temporal dynamics.

4. Dataset

4.1. UCF-Crime Dataset

The UCF-Crime dataset is a benchmark dataset widely used for real-world anomaly detection in surveillance videos. It consists of videos capturing both normal and anomalous activities, representing real-world scenarios such as theft, accidents, and other crime-related events. The dataset's scale and diversity make it a robust foundation for training and evaluating anomaly detection models.

This table provides a clear and structured overview of the dataset composition, showing the distribution of normal and anomaly videos and segments across the training, validation, and testing sets.

Video Type	Total Clips (.mp4)	Training Set (.npy)	Validation Set (.npy)	Test Set (.npy)
Normal Videos	950	798	191	156
Anomaly Videos	650	633	167	150

Table 1. *Dataset Composition*

4.2. Preprocessing

In the preprocessing phase, videos are divided into temporal segments for detailed analysis. Each segment is labeled as either normal or anomalous, enabling the model to learn the distinct patterns associated with abnormal events. This segmentation process aligns with the temporal structure of the videos and supports the incorporation of Temporal Smoothness Loss during training.

For preprocessing, 950 normal videos were selected, from which frames were extracted and saved as 1,145 segments in .npy format, each containing 32 frames. Similarly, 650 anomaly videos were used to generate 950 segments (32 frames each) after frame extraction as Shown in *Table 1*.

```
"video.mp4": [
   {
      "start": 3,
      "end": 13
   }
]
```

Pseudocode 1. Annotations JSON Format

Additionally, each video is processed with annotations, which are stored in a .json format. The annotations define the time intervals for anomaly segments within the video. For example, an annotation for the video "video.mp4" could look like the following:

The start and end times indicate the temporal range of the anomalous segment, which is used during the frame extraction process to ensure that only relevant anomaly clips are included for training and testing.

5. Results and Discussion:

5.1 Performance Metrics Analysis

In this section, we present the performance metrics of the anomaly detection models under different configurations: C3D, 3D ResNet, C3D + 3D ResNet, and their variations with the custom loss function. The models are evaluated across several metrics: Accuracy, Loss, Precision, Recall, F1-Score, AUC, mAP, IoU, and Best IoU. The configurations and performance metrics are summarized in **Table 3**, and the parameters used for reproducibility are presented in **Table 2**.

Parameter	Value
Learning Rate	0.001
Optimizer	Adam
Batch Size	32
Number of Epochs	20
Loss Function	BCE (with and without custom Loss)

Table 2. *Parameters for Reproducibility*

5.2. Performance Overview

The **C3D** model performs well across training, validation, and test sets, achieving accuracy values of 0.904, 0.785, and 0.830, respectively. This model exhibits high precision (0.99), strong recall (0.83), and an excellent AUC score (0.91). The high performance on the training set, paired with reasonable results on the validation and test sets, indicates that C3D captures temporal and spatial features well, though its performance could be further improved with the custom loss function.

In contrast, **3D ResNet** performs adequately on the training set with an accuracy of 0.676, but it struggles on the validation and test sets (accuracy values of 0.687 and 0.650, respectively). The precision and recall for 3D ResNet are significantly lower compared to C3D. This model excels at spatial feature extraction but does not adequately capture temporal dynamics, which is crucial for effective anomaly detection in video data. This limitation results in its lower performance in anomaly detection tasks.

The **C3D + 3D ResNet** combined model demonstrates better performance than the individual models, with an accuracy of 0.838 during training, 0.768 during validation, and 0.775 on the test set. However, it still lags behind C3D with custom loss in terms of AUC and F1-Score. This can be attributed to feature redundancy and overfitting. While 3D ResNet enhances spatial feature extraction, it introduces conflicts with C3D's temporal feature extraction, leading to diminished overall performance.

Model	Dataset	Accuracy	Loss	Precision	Recall	F1-Score	AUC	mAP	IoU	Best IoU
C3D	Training	0.904263	0.222517	0.992614	0.835821	0.907495	0.913923	0.992212	0.345592	0.3456
	Validation	0.784916	0.45763	0.850649	0.708108	0.772861	0.78758	0.900692	0.306766	
	Test	0.830065	0.425666	0.933333	0.717949	0.811594	0.832308	0.924393	0.284775	
3D Resnet	Training	0.67645	0.597061	0.700119	0.734336	0.71682	0.668906	0.808579	0.399535	0.3995
	Validation	0.687151	0.593824	0.704663	0.712042	0.708333	0.685362	0.810057	0.366399	
	Test	0.650327	0.613895	0.638418	0.724359	0.678679	0.648846	0.765505	0.371699	
C3D + 3D Resnet	Training	0.837876	0.330532	0.998239	0.710526	0.830161	0.854473	0.995612	0.301878	0.3019
	Validation	0.768156	0.611334	0.990909	0.570681	0.724252	0.782346	0.950285	0.242196	
	Test	0.77451	0.56789	0.948454	0.589744	0.727273	0.778205	0.931075	0.242944	
C3D with Custom loss	Training	**0.98812**	0.469773	0.986301	0.992481	0.989382	0.987552	0.999659	0.388304	0.3883
	Validation	**0.837989**	0.816955	0.863388	0.827225	0.84492	0.838762	0.940092	0.353278	
	Test	**0.866013**	0.783398	0.885906	0.846154	0.865574	0.86641	0.94793	0.33165	
3D Resnet with Custom loss	Training	0.620545	0.899452	0.598304	0.972431	0.740811	0.574683	0.871989	0.527274	0.5273
	Validation	0.631285	0.915808	0.594249	0.973822	0.738095	0.606671	0.883627	0.495512	
	Test	0.539216	0.975607	0.526132	0.967949	0.681716	0.530641	0.833888	0.49316	
C3D + 3D Resnet with custom loss	Training	**0.856744**	0.626417	0.795613	1	0.886174	0.838073	0.995533	0.45046	0.4505
	Validation	**0.790503**	0.878001	0.732	0.958115	0.829932	0.778459	0.922001	0.433586	
	Test	**0.72549**	0.918901	0.656522	0.967949	0.782383	0.720641	0.94572	0.436305	

Table 3. Performance Metrics Table

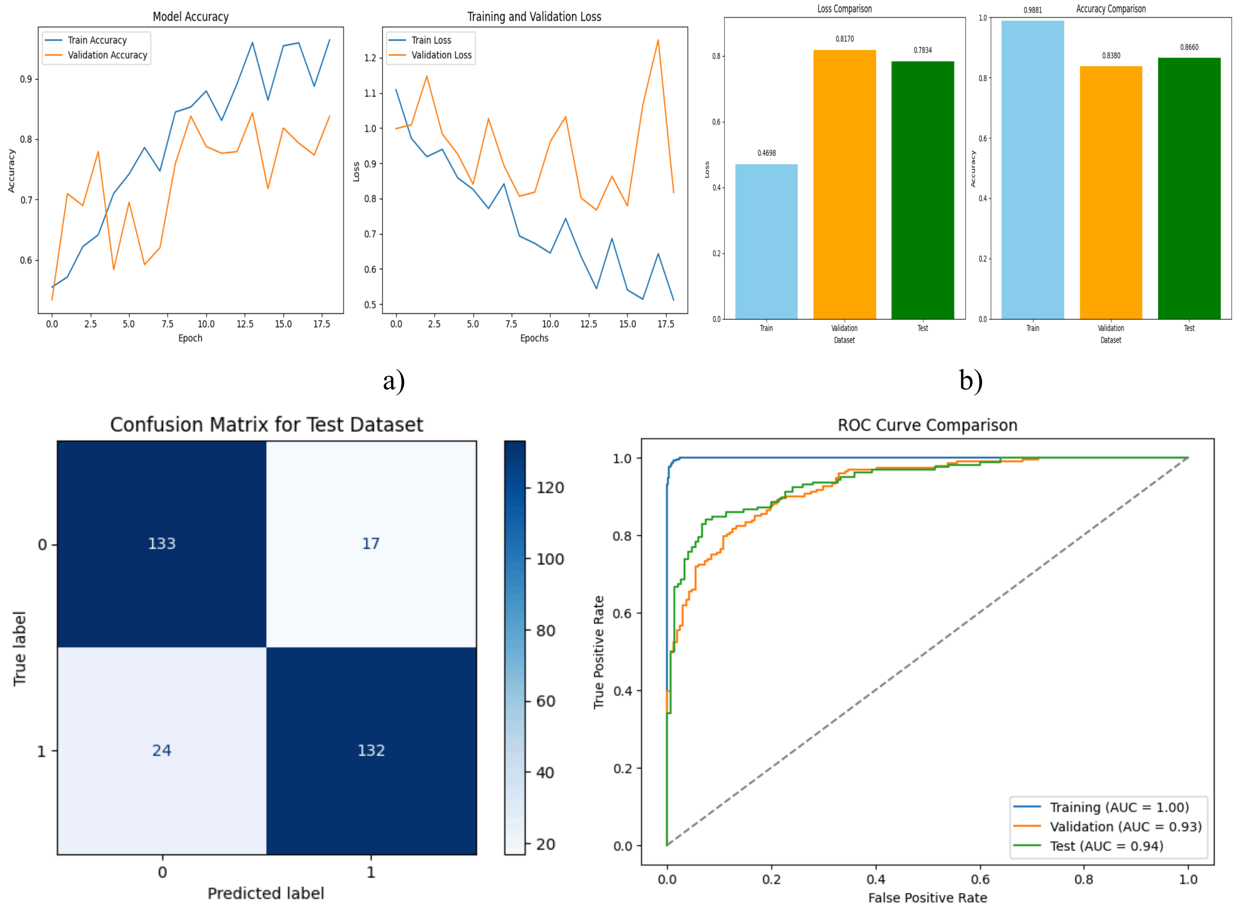
The **C3D** with **Custom Loss** model shows significant improvement, particularly in precision, recall, and F1-Score. The training accuracy reaches 0.988, while validation and test accuracies are 0.838 and 0.866, respectively. The custom loss function, which balances Binary Cross-Entropy (BCE), Ranking Loss, and Temporal Smoothness Loss, plays a crucial role in improving precision and recall, reducing false positives, and enhancing generalization. This improvement is particularly evident in the test set results, where the model achieves an F1-Score of 0.866 and AUC of 0.866, indicating more balanced and accurate anomaly detection.

The **3D ResNet** with **Custom Loss** model shows that, while the custom loss function improves performance, it still struggles when compared to C3D with custom loss. The training accuracy is 0.621, but the performance degrades on the validation and test sets, with test accuracy dropping to 0.539. Although the custom loss function reduces overfitting, the model's limited temporal sensitivity hinders its ability to effectively detect anomalies in video data.

The **C3D + 3D ResNet** with **Custom Loss** model results in better performance than without the custom loss, but it still underperforms when compared to C3D with custom loss alone. The accuracy for the training, validation, and test sets are 0.857, 0.791, and 0.725, respectively. Despite the custom loss's contribution, the combined model's performance is diminished due to overfitting and redundant features from both C3D and 3D ResNet. The custom loss helps improve precision and recall slightly, but it fails to resolve the underlying conflict between the two models.

The performance of the models is further evaluated using visualizations, including accuracy plots, loss comparisons, confusion matrices, and ROC curves. ***Figure 2*** shows the performance of **C3D with Custom Loss**. ***Figure 2(a)*** plots accuracy against the number of epochs for the C3D with Custom Loss model. The plot indicates a steady improvement in accuracy over the epochs, with a notable increase during training, followed by a gradual stabilization during validation and testing. ***Figure 2(b)*** presents a comparison of accuracy and loss across the test, validation, and training sets. It highlights the model's

5.3 Visualization and Performance Evaluation

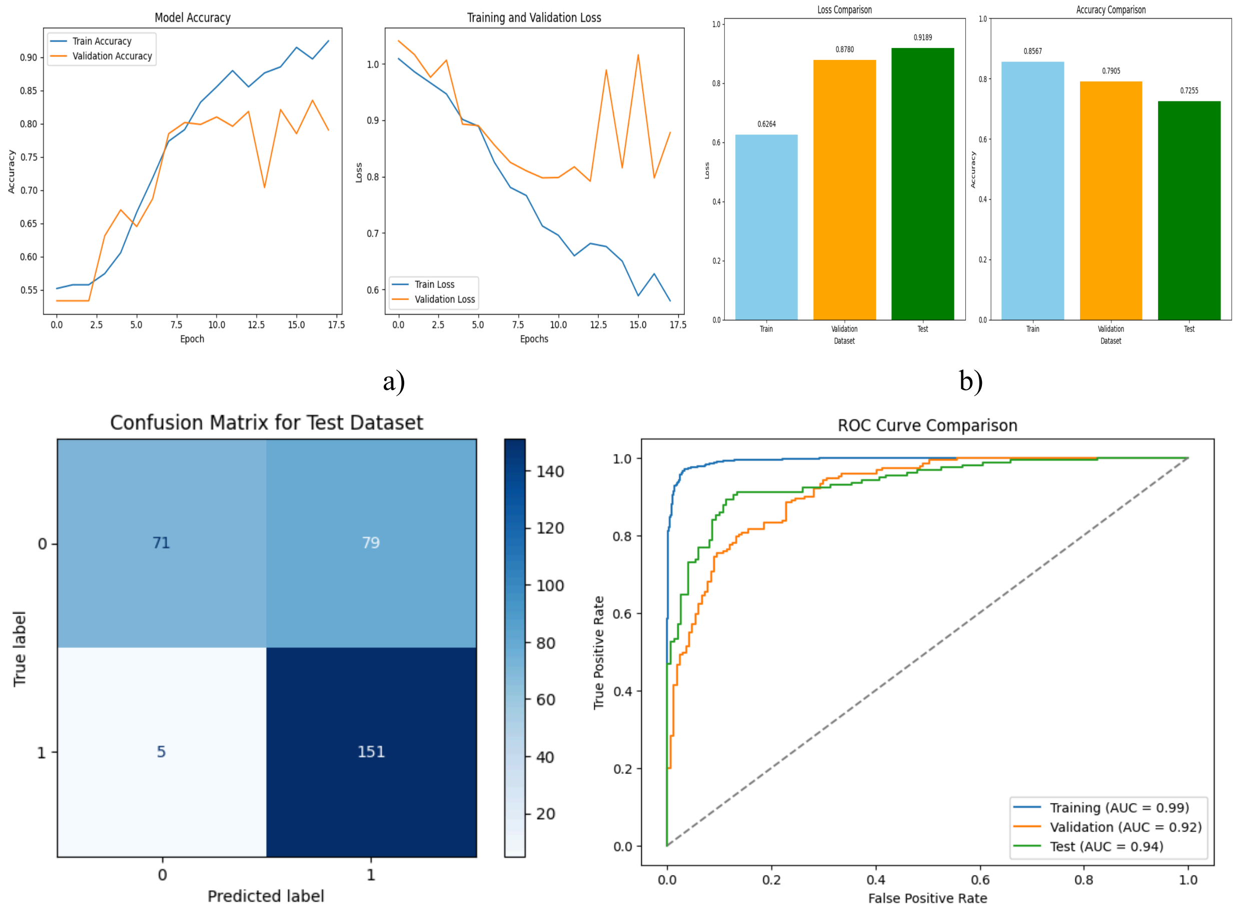


a) b)

c) d)

Figure 2. a) Accuracy vs Epochs for C3D with Custom Loss. b) Accuracy and Loss Comparisons (Test, Validation, Test), c) Confusion Matrix for Test Set, d) ROC Curve for Train, Test and Validation.

strong performance on the training and validation sets, as well as reasonable performance on the test set. ***Figure 2(c)*** displays the confusion matrix for the test set, showing a high number of true positives, indicating that the model effectively detects anomalies. The confusion matrix confirms the high precision and recall observed in the previous metrics. ***Figure 2(d)*** depicts the ROC curve for the train, test, and validation sets, demonstrating that the C3D with Custom Loss model provides high true positive rates with relatively low false positive rates, as evidenced by the curve's position near the upper left corner.



a)

b)

c)

d)

Figure 3. a) Accuracy vs Epochs for C3D + 3D Resnet with Custom Loss. b) Accuracy and Loss Comparisons (Test, Validation, Test), c) Confusion Matrix for Test Set, d) ROC Curve for Train, Test and Validation.

Figure 3 illustrates the performance of the proposed **C3D + 3D ResNet with Custom Loss** model. ***Figure 3(a)*** shows accuracy versus epochs, where the model exhibits slower convergence and lower final accuracy compared to C3D with Custom Loss. The accuracy increases over time but does not reach the high levels seen in C3D with Custom Loss. ***Figure 3(b)*** compares the accuracy and loss for the test, validation, and training sets. The model's performance is lower than that of C3D with Custom Loss, indicating some conflict between the spatial features of 3D ResNet and the temporal features of C3D, which impacts generalization to the test set. ***Figure 3(c)*** provides the confusion matrix for the test set. It shows that while the model has a reasonable number of true positives, the false positives are slightly higher compared to the C3D with Custom Loss model, pointing to some imbalances in the model's predictions. ***Figure 3(d)*** illustrates the ROC curve for the train, test, and validation sets. The ROC curve shows that, although the model performs reasonably well, it does not achieve the optimal performance of C3D with Custom Loss, which is reflected in its less favorable positioning on the curve.

These visualizations further support the findings from the performance metrics, confirming that **C3D with Custom Loss** outperforms **C3D + 3D ResNet with Custom Loss**, primarily due to its superior handling of temporal features and improved generalization with the custom loss function.

5.4. Conclusion of Visualization Analysis

From the visualizations, we can conclude that the C3D with Custom Loss model achieves a better trade-off between precision and recall, as shown by its ROC curve and confusion matrix. The inclusion of the custom loss function plays a significant role in enhancing temporal consistency, which is crucial for video anomaly detection. On the other hand, **C3D + 3D ResNet with Custom Loss** suffers from feature redundancy and conflicts between the models, which limits its ability to effectively generalize, as indicated by the less favorable accuracy, loss, and ROC results.

5.5. Conclusion on Video Summarization

In the video summarization process, anomaly-classified segments are prioritized to create a concise summary of the most relevant footage. To enhance the anomaly detection process, the calculated anomaly scores, based on memorability, entropy, and temporal dynamics, help in evaluating and ranking each segment.

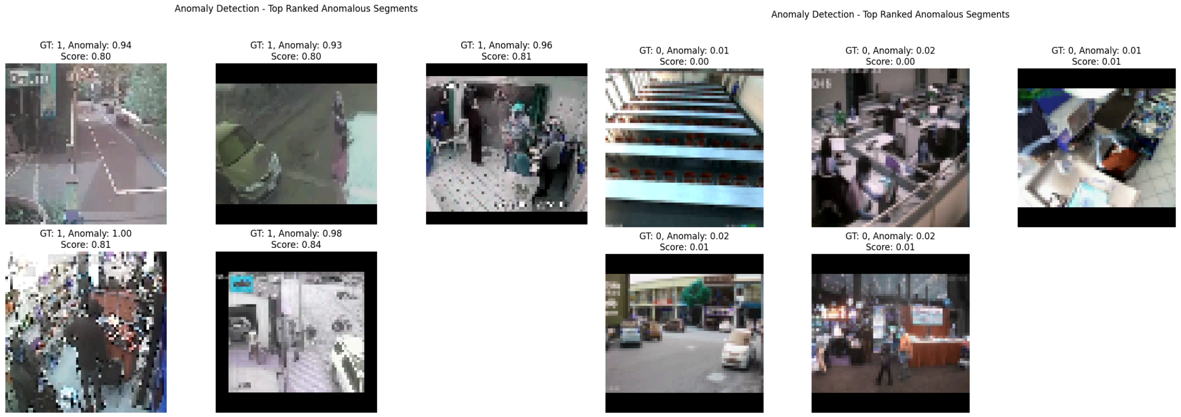


Figure 2. (a) Ground Truth (GT): The first frame of each segment, representing the true label. (b) Prediction (Probability): The predicted probability for each segment, indicating the likelihood of it being anomalous. (c) Calculated Anomaly Score: The score computed for each segment, reflecting the strength of the anomaly detection.

Figure 4 and ***Figure 5*** demonstrate the performance of the model by showing 5 segments for both anomalous and normal video clips. For each set of segments, the first image represents the ground truth (GT) label, followed by the predicted probability and the calculated anomaly score. These visualizations help in understanding how well the model differentiates between normal and anomalous behavior in the video, providing a clearer insight into the overall performance of the anomaly detection and video summarization system.

6. Instructions for Using the Program

6.1 Installation Requirements

To execute the program, ensure the following dependencies and modules are installed. These dependencies are listed in requirements_kaggle.txt:

For local execution, install these dependencies using the following command:

```
Kaggle Version Used
pip install -r requirements_kaggle.txt

or

Local Versions Used
pip install -r requirements.txt
```

6.2 Setup and Execution

Access the program through the provided Kaggle Notebook at:
https://www.kaggle.com/code/vishnupriyanss/cv-demo

Ensure input video files are in .mp4 format and pre-trained model files (e.g., .h5) are available in the environment or uploaded manually.

Execute the Python notebook sequentially to preprocess data, extract features, perform anomaly classification, and summarize the results.

Generated outputs, such as anomaly-classified video segments, will be saved in the output directory in .mp4 format.

6.3 Configuration Files or Input Format

Video Input:
The program accepts .mp4 files, which are divided into fixed-length segments for processing (default: 32 frames per segment).

Output Files:
The output includes anomaly-classified summarized videos in .mp4 format.

Pre-trained Model:
Ensure the path to the pre-trained model file is correctly configured in the notebook.

6.4 Parameter Modification

Segment Length:
The default segment length is set to 32 frames per segment. Modify the parameter in the extract_segments_from_video() function:

```
extract_segments_from_video(video_path, segment_length=desired_length)
```

Threshold for Anomaly Detection:
The threshold for anomaly classification is set to 0.5 by default. Modify the threshold parameter in the prediction section as required.

Frame Size:
Frames are resized to (64, 64) by default. Update the frame_size parameter in the extract_segments_from_video() function to adjust the resolution.

Loss Function:
The custom loss function can be replaced with another function during the training phase. Update this in the respective training section of the notebook.

6.5 Scoring Weights for Video Summarization:

Weights for memorability, entropy, and temporal dynamics can be adjusted in the scoring section of the notebook to customize anomaly prioritization.

Further details and execution can be found in the link:
https://www.kaggle.com/code/vishnupriyanss/cv-demo

7. References

[1] W. Sultani, C. Chen, M. Shah, "Real-world Anomaly Detection in Surveillance Videos," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6479-6488.

[2] J. Chen, J. Wang, J. Pu, R. Zhang, "A Three-Stage Anomaly Detection Framework for Traffic Videos," IEEE Transactions on Intelligent Transportation Systems, 2022.

[3] Halil İbrahim Öztürk and Ahmet Burak Can. 2021. ADNet: Temporal Anomaly Detection in Surveillance Videos. In Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV. Springer-Verlag, Berlin, Heidelberg, 88–101. https://doi.org/10.1007/978-3-030-68799-1_7