Anomaly Detection in Surveillance Videos Using CNN-LSTM Architecture

Author: Syed Muhammad Huzaifa , Anas and Zeshan

Institution: FAST NUCES Karachi

Date: 7th May 2025

1. Objective

The project aims to:

1. Develop an automated system for detecting anomalous activities in surveillance footage
2. Implement a hybrid deep learning model combining CNNs and LSTMs for spatio-temporal analysis
3. Achieve state-of-the-art classification performance on the UCF-Crime dataset
4. Optimize training efficiency using TPU acceleration

Key Technical Goals:

- ✓ Frame-level feature extraction using CNN
- ✓ Temporal sequence modeling with LSTM
- ✓ Attention mechanism for important frame selection
- ✓ Triplet loss for discriminative embeddings

2. Problem Statement

Current Challenges

Issue	Impact
Manual monitoring fatigue	High false alarm rates
Variable anomaly duration	Difficult to detect short events
Class imbalance	Bias toward frequent classes
Real-time processing needs	High computational requirements

Proposed Solution

Hybrid CNN-LSTM Model with:

- ResNet18 backbone for spatial features

- Bi-directional LSTM for temporal patterns

- Attention mechanism (frame importance weighting)

- Triplet loss (improved class separation)

3. Methodology

3.1 Data Pipeline

```
# Pseudocode for data loading
def load_videos():
   1. Sample 16 frames/video (uniform temporal sampling)
   2. Resize to 224×224 pixels
   3. Normalize using ImageNet stats
   4. Apply augmentations:
     - Random horizontal flip
     - Color jitter
```

3.2 Model Architecture

Key Components:

1. Feature Extraction

   - Pretrained ResNet18 (ImageNet weights)

   - Remove final FC layer → output 512-D features

2. Temporal Processing

```
   nn.LSTM(
      input_size=512,
      hidden_size=256,
      bidirectional=True,
      batch_first=True
   )
```

3. Attention Mechanism

   - Learns weights for each timestep

   - Context vector = weighted sum of LSTM outputs

4. Classification Head

   - 2-layer MLP (512 → 256 → 14 classes)

3.3 Training Protocol

Parameter	Value
Hardware	Google Cloud TPU v3-8
Batch Size	32
Optimizer	Adam (lr=1e-4)
Loss	Cross-Entropy + Triplet Loss (α=0.7)
Epochs	50

Triplet Mining Strategy:

- Online semi-hard negative mining

- Margin = 1.0

4. Results

4.1 Quantitative Analysis

Computational Efficiency:

Hardware	Time/Epoch
CPU	58 min
GPU	12 min
TPU	4 min

4.2 Qualitative Analysis

Confusion Matrix:

[Insert confusion_matrix.png here]

Key Observations:

- Highest confusion: Assault ↔ Fighting (similar visual patterns)

- Best performance: Explosion (distinct visual signature)

Embedding Visualization:

[Insert tsne.png here]

Triplet loss creates tighter clusters compared to baseline

5. References

5. Sultani, W. et al. (2018). Real-world Anomaly Detection in Surveillance Videos. CVPR.
6. He, K. et al. (2016). Deep Residual Learning for Image Recognition. CVPR.
7. Schroff, F. et al. (2015). FaceNet: A Unified Embedding for Face Recognition. CVPR.
8. Dataset: UCF-Crime (https://www.crcv.ucf.edu/projects/real-world/)

Appendix

A. Hardware Specifications

- TPU: v3-8 pod (128GB memory)

- CPU: 96 vCPUs, 360GB RAM

B. Software Stack

- Python

- PyTorch (+ torch-xla for TPU)

- OpenCV

C. Ethical Considerations

- Dataset contains violent content (used for research only)

Potential bias mitigation strategies:

- Class-balanced sampling

- Data augmentation for rare classes