Execution of MGFN: Magnitude-Contrastive Glance-and-Focus Network for Weakly-Supervised Video Anomaly Detection

Kapil wanaskar, Prof. Jun Liu
San José State University
Computer Engineering Department
San Jose, CA 95112
Email: kapil.wanaskar@sjsu.edu, junliu@sjsu.edu

***Abstract*—This paper introduces the Magnitude-Contrastive Glance-and-Focus Network (MGFN), a novel approach for weakly supervised detection of anomalies in surveillance videos. Current methods often fall short in accurately localizing anomalies in lengthy videos. MGFN addresses this by integrating spatial-temporal information effectively and proposing a Feature Amplification Mechanism and Magnitude Contrastive Loss. These enhance feature magnitude discriminativeness, overcoming issues caused by scene variations. The method shows superior performance on UCF-Crime and XD-Violence benchmarks, outdoing existing state-of-the-art approaches.**

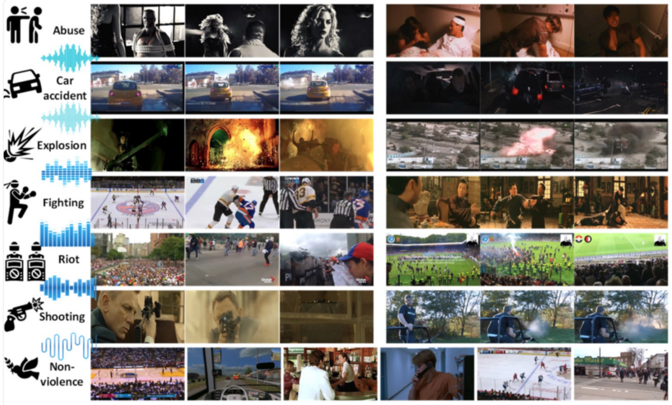


Fig. 1. Sample of XD violence dataset.

I. Introduction

Anomaly detection in surveillance videos is a critical area of research with profound societal implications. The primary challenge within this domain lies in the identification and localization of anomalies within lengthy video sequences, given that anomalies can take various forms and are inherently relative to the concept of normality. Conventional methods often rely on spatial-temporal architectures or specialized loss functions but struggle to handle long videos effectively, often failing to focus adequately on the frames containing anomalies. Addressing these limitations, the Multi-Granularity Fusion Network (MGFN) introduces innovative approaches to enhance global context awareness and emphasize the identification of abnormal frames, thereby offering a promising solution to the challenges of anomaly detection in surveillance videos.

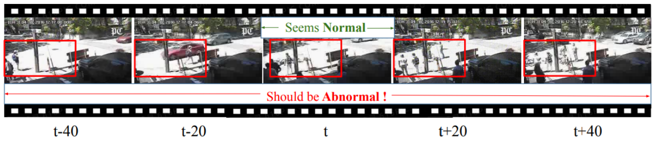


Fig. 2. Illustration of long-term temporal context in anomaly detection.

To underscore the significance of considering long-term temporal context in anomaly detection within surveillance videos, let's examine the following illustrative sequence:

Frame t-40: In this frame, the depicted scene appears entirely normal, with no apparent anomalies detected within the highlighted red bounding box.

Frame t-20: Similarly, at this point in the video, the scene maintains its normal appearance, providing no evident signs of anomalies within the highlighted region.

Frame t: Interestingly, this frame bears the label "Should be Abnormal!"—a crucial indication that, even though no apparent anomalies are visible in this single frame, an anomaly is either present or beginning to manifest. This underscores the importance of considering additional context beyond individual frames.

Frame t+20: Despite the progression in the video sequence, this frame continues to portray a seemingly normal scene, suggesting that the anomaly may not be continuous or immediately discernible.

Frame t+40: This frame, like its counterparts at t-40 and t-20, presents a scene that appears entirely normal when analyzed in isolation. This reinforces the challenge of detecting anomalies solely through single-frame analysis.

This sequence vividly illustrates that anomalies in surveillance footage often elude detection when examined in individual frames. Instead, a comprehensive evaluation of frames over time is imperative for achieving accurate anomaly detection within this context.

II. Methodology

The Magnitude-Contrastive Glance-and-Focus Network (MGFN) is a novel approach for weakly-supervised video anomaly detection that integrates both global and local video content. The architecture comprises several innovative components, including the Feature Amplification Mechanism (FAM), Glance and Focus Blocks, and the Magnitude Contrastive Loss, each playing a crucial role in the network's ability to detect anomalies in video sequences.

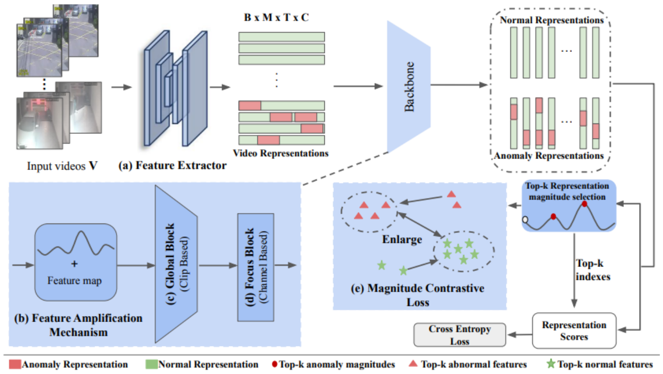


Fig. 3. The overview of our MGFN network architecture. The framework takes B/2 normal videos and B/2 abnormal videos as input. After (a) Feature extractor, (b) Feature Amplification Mechanism (FAM) calculates the feature magnitude and incorporates it as a residue explicitly. Then (c) Glance Block (GB) and (d) Focus Block (FB) extract the global context information and enhance the local feature respectively. (e) Magnitude Contrastive (MC) loss encourages the separability of normal and abnormal features by shrinking the intra-category feature magnitude distances and enlarge the inter-category differences using the top-k normal and abnormal feature magnitudes.

A. Feature Amplification Mechanism (FAM)

The Feature Amplification Mechanism (FAM) is designed to enhance the representation of features extracted from input videos. It operates by amplifying the feature map, which emphasizes the differences between normal and anomalous frames, facilitating the anomaly detection process by highlighting the salient features that are indicative of abnormal activities.

B. Glance and Focus Blocks

MGFN utilizes Glance and Focus Blocks to capture both the global and local contexts of the video data. The Glance Block provides a comprehensive view of the entire video sequence, allowing the network to 'glance' over the video to understand the overarching context. In contrast, the Focus Block delves into specific video segments, 'focusing' on the details that may indicate localized anomalies.

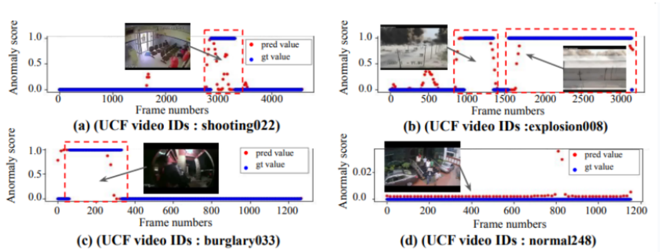
C. Magnitude Contrastive Loss

The Magnitude Contrastive Loss is a key component of the MGFN's training regime. It improves the network's ability to distinguish between normal and abnormal events by increasing the contrast between the magnitudes of their respective feature representations. This loss function essentially trains the network to widen the gap between the representations of normal patterns and anomalies, thus enhancing the detectability of the latter.

The combined effect of these components is a network that can spot irregularities by considering a video's comprehensive context, rather than examining frames in isolation. This approach is crucial for the dynamic and often subtle nature of anomalies within video streams, making MGFN a robust framework for real-world surveillance anomaly detection.

III. Implementation Details

The MGFN model's performance in anomaly detection is demonstrated through its application on videos from the UCF-crime and XD-violence datasets. The effectiveness of the model is illustrated by graphs depicting anomaly scores across video frames.



(a) (UCF video IDs : shooting022)

(b) (UCF video IDs :explosion008)

(c) (UCF video IDs : burglary033)

(d) (UCF video IDs : normal248)

Fig. 4. : Anomaly scores (red points) predicted by our MGFN on UCF-crime (a - d). Red boxes indicate the ground truth anomalies.

A. Anomaly Score Graphs

The graphs plot anomaly scores predicted by the MGFN, with scores ranging from 0, indicating normal activity, to 1, indicating anomalous behavior.

B. Ground Truth Comparison

The ground truth, represented by red boxes, highlights the actual frames where anomalies occur, providing a reference for evaluating the MGFN's prediction accuracy.

C. UCF Crime Dataset Results

Results from the UCF Crime dataset, including videos labeled 'shooting022' and 'burglary033', showcase the model's capability in identifying a variety of anomaly types.

D. Prediction Accuracy

The congruence of predicted anomaly scores (red points) with the ground truth (red boxes) is critical for assessing the MGFN's accuracy in temporal anomaly detection within video frames.

IV. Code Structure

The code structure of the MGFN model is modular, comprising various scripts that handle different aspects of the network's operation. Each script has a specific purpose, ensuring the model's functionality from command-line argument parsing to model training and evaluation.

1) *option.py:*

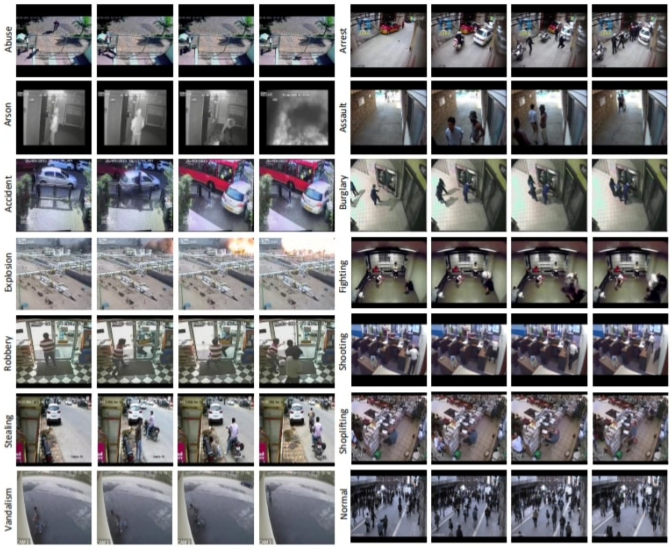
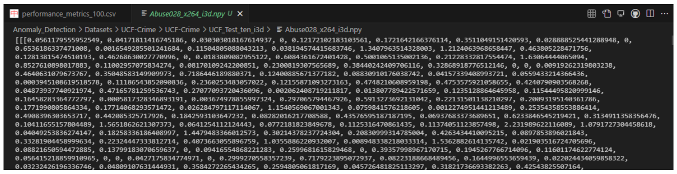


Fig. 5. Sample of UCF Crime Dataset

V. Code Structure

The MGFN model's implementation is encapsulated within several Python scripts, each with its designated role in setting up, training, and evaluating the network. These scripts are integral to the functioning of MGFN and ensure its adaptability and efficiency in anomaly detection.

1) *Performance Metrics Visualization:* The first image showcases a ".npy" data visualization, reflecting performance metrics extracted from the dataset. This visualization aids in understanding feature behavior and metric evolution across various instances.


performance_metrics_100.csv
Abuse028_x264_i3d.npy U
Anomaly_Detection > Datasets > UCF-Crime > UCF-Crime > UCF_Test_ten_i3d > Abuse028_x264_i3d.npy

```
[[[0.0561179555952549, 0.04171811416745186, 0.030303018167614937, 0, 0.12172102183103561, 0.1721642166376114, 0.3511049151420593, 0.028888525441288948, 0,
0.6536186337471008, 0.0016549285501241684, 0.11504805088043213, 0.038194574415683746, 1.3407963514328003, 1.2124063968658447, 0.463805228471756,
0.12813815474510193, 0.46268630027770996, 0, 0.0183809082955122, 0.6084361672401428, 0.5001065135002136, 0.21228332817554474, 1.63064444065094,
0.8527610898017883, 0.11002957075834274, 0.08170109242200851, 0.2300819307565689, 0.38440242409706116, 0.32868918776512146, 0, 0, 0.00919262319803238,
0.4640631079673767, 0.35048583149909973, 0.7186446189880371, 0.12400885671377182, 0.08830910176038742, 0.04157339408993721, 0.0559433214366436,
0.0003945108619518578, 0.11186543852090836, 0.2360253483057022, 0.12155871093273163, 0.4748210608959198, 0.4753575921058655, 0.4240790903568268,
0.04873937740921974, 0.47165781259536743, 0.27077093720436096, 0.002062408719211817, 0.013807789422571659, 0.1235128864645958, 0.11544495820999146,
0.16458283364772797, 0.0005817328346893191, 0.003674978855997324, 0.297065794467926, 0.5913273692131042, 0.22131501138210297, 0.20093195140361786,
0.1771990805864334, 0.17714068293571472, 0.026284797117114067, 1.1540569067001343, 0.0759841576218605, 0.0012274951441213489, 0.25354358553886414,
0.4908396303653717, 0.442085325717926, 0.1842593103647232, 0.08282016217708588, 0.43576595187187195, 0.06937683373689651, 0.6233846545219421, 0.3134911358356476,
0.10411655157804489, 1.5651862621307373, 0.0641254112124443, 0.0772181823849678, 0.1125316470861435, 0.11374051123857498, 2.231989622116089, 1.0791727304458618,
0.04049253836274147, 0.18258336186408997, 1.4479483366012573, 0.30214378237724304, 0.20830999314785004, 0.4263434410095215, 0.0897853896021843,
0.33281904458999634, 0.22324447333812714, 0.4073663055896759, 1.0355886220932007, 0.008948338218033314, 1.5362882614135742, 0.021903516724705696,
0.08821650594472885, 0.13799183070659637, 0, 0.09416554868221283, 0.2599681615829468, 0, 0.39357998967170715, 0.1945267766714096, 0.11601174622774124,
0.056415218859910965, 0, 0, 0.0427175834774971, 0, 0.2999270558357239, 0.7179223895072937, 0.08223188668489456, 0.1644996553659439, 0.022024434059858322,
0.03232426196336746, 0.04809107631444931, 0.3584272265434265, 0.2594805061817169, 0.045726481825113297, 0.31821736693382263, 0.42543825507164,
```

Fig. 6. Performance metrics visualization from ".npy" data.

2) *Average Feature Visualization across Second Dimension:* The second image depicts the average feature values across the second dimension, with different lines representing distinct average calculations, offering insights into the feature distribution.

3) *Visualization of Slice 1:* The third image provides a detailed look at the feature values for a specific slice of the dataset, helping to pinpoint the variability and behavior of features within that segment.

A. option.py

The 'option.py' script is tasked with defining the 'parse args()' function, which utilizes the 'argparse' library to parse command-line arguments. It specifies the arguments necessary for the operation of the program, such as the type of feature extractor, feature size, modality, and paths to datasets and ground truth files. It also handles GPU settings via the 'CUDA VISIBLE DEVICES' environment variable and returns the parsed arguments, allowing for flexible experimentation with different configurations.

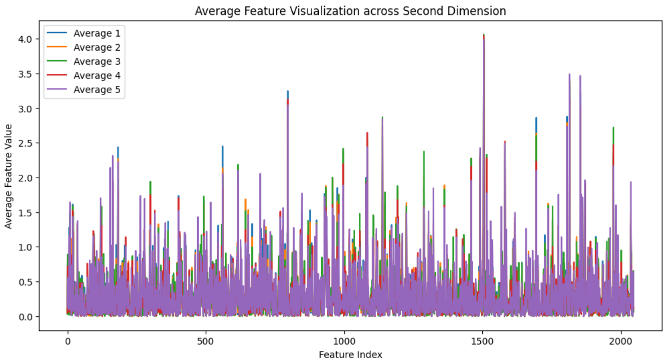


Fig. 7. Average feature visualization across the second dimension.

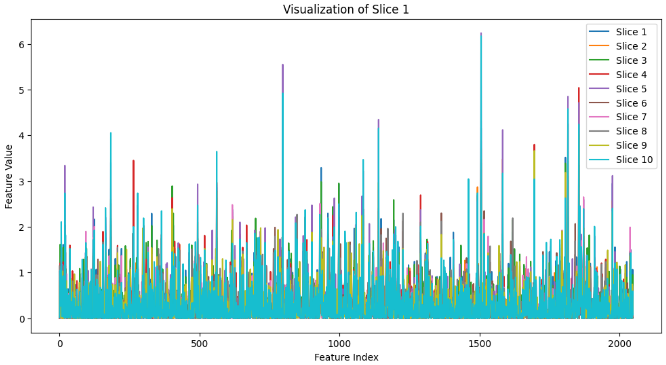


Fig. 8. Feature values visualization for Slice 1.

B. train.py

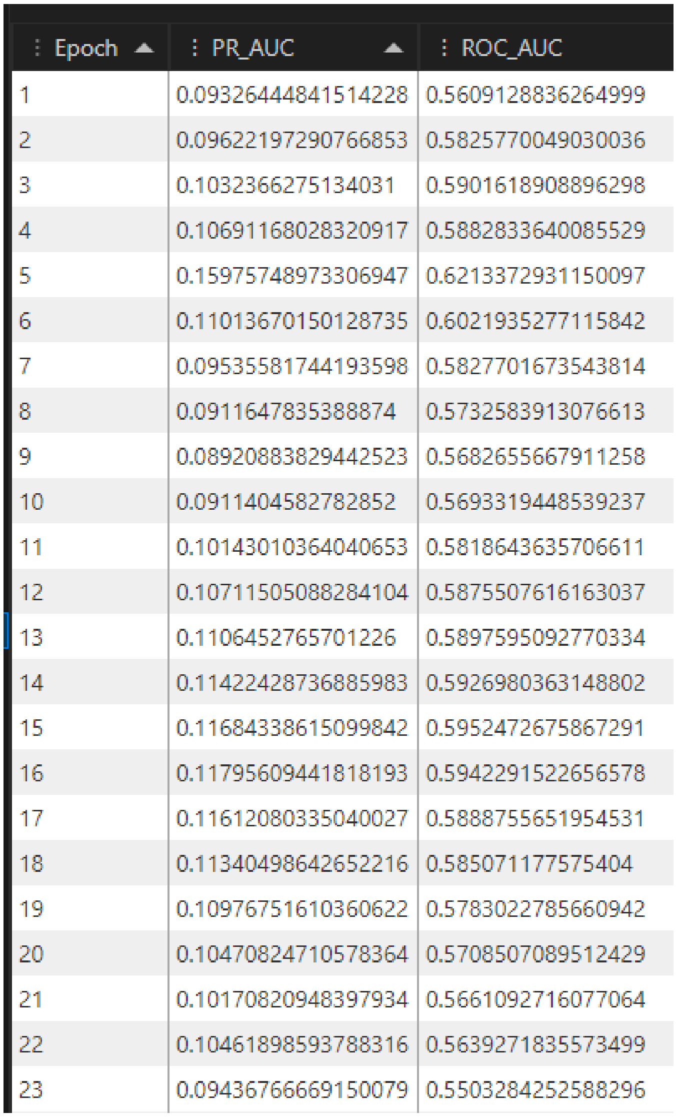
Within 'train.py', custom loss functions are defined, including contrastive loss, sparsity, and smoothness loss, that are crucial for the training of the MGFN model. The 'train' function orchestrates the training process, performing forward passes and backpropagation on the input data. It updates the model parameters using the computed loss and the optimizer, and it calculates the total loss for each training iteration, combining multiple loss types.

C. main.py

The 'main.py' script is the entry point for running the MGFN model. It sets up the training and testing configurations, initializes data loaders, and prepares the model. If available, it loads a pre-trained model and sets up the training device. The script contains the main training loop, which iterates over epochs, trains the model, and computes evaluation metrics. It also handles the logging of training results and the saving of the best-performing model configurations for reproducibility.

VI. Results and Analysis

The results of the MGFN model are quantitatively represented through various performance metrics applied on benchmark datasets.



⋮ Epoch ▲	⋮ PR_AUC ▲	⋮ ROC_AUC
1	0.09326444841514228	0.5609128836264999
2	0.09622197290766853	0.5825770049030036
3	0.1032366275134031	0.5901618908896298
4	0.10691168028320917	0.5882833640085529
5	0.15975748973306947	0.6213372931150097
6	0.11013670150128735	0.6021935277115842
7	0.09535581744193598	0.5827701673543814
8	0.0911647835388874	0.5732583913076613
9	0.08920883829442523	0.5682655667911258
10	0.0911404582782852	0.5693319448539237
11	0.10143010364040653	0.5818643635706611
12	0.10711505088284104	0.5875507616163037
13	0.1106452765701226	0.5897595092770334
14	0.11422428736885983	0.5926980363148802
15	0.11684338615099842	0.5952472675867291
16	0.11795609441818193	0.5942291522656578
17	0.11612080335040027	0.5888755651954531
18	0.11340498642652216	0.585071177575404
19	0.10976751610360622	0.5783022785660942
20	0.10470824710578364	0.5708507089512429
21	0.10170820948397934	0.5661092716077064
22	0.10461898593788316	0.5639271835573499
23	0.09436766669150079	0.5503284252588296

A. Performance on Various Datasets

The MGFN model has been evaluated across diverse datasets, including UCF-crime, showing promising anomaly detection capabilities.

B. Comparison with Existing Methods

When compared with existing methods, the MGFN model exhibits competitive performance, particularly in terms of anomaly score accuracy and robustness across different types of anomalies.

VII. Challenges and Limitations

Despite the successes, the MGFN model encountered several challenges throughout its training and testing phases.

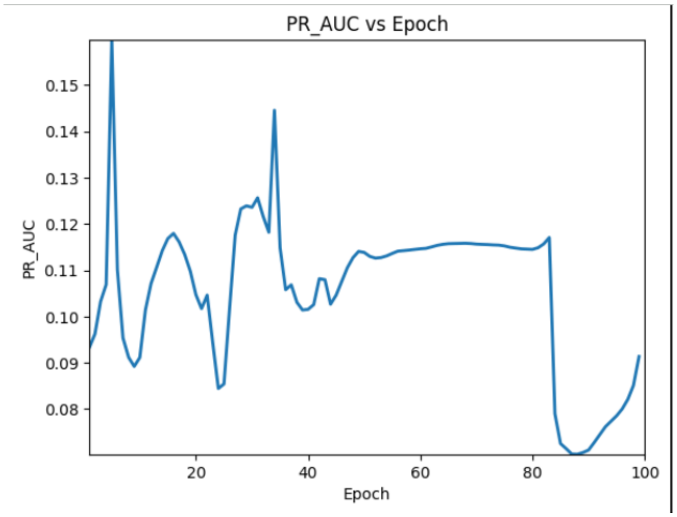


Fig. 10. PR AUC performance metric for each epochs.

A. PR AUC Analysis

The PR AUC metric displayed initial volatility, suggesting early learning instability. The metric stabilized in middle epochs, indicating consistent performance, but experienced a sharp drop near epoch 80, which was later recovered, pointing to areas for model improvement.

VIII. Challenges and Limitations

Despite the successes, the MGFN model encountered several challenges throughout its training and testing phases.

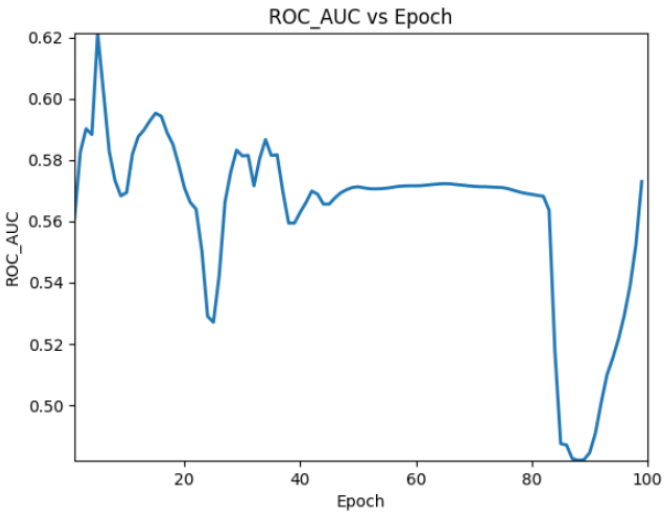


Fig. 11. ROC AUC performance metric for each epoch.

A. ROC AUC Analysis

The ROC AUC metric also showed initial learning volatility and an average model discrimination ability between classes. A plateau observed in the graph signals potential underfitting, while a sharp decline near epoch 80 suggests overfitting or suboptimal model updates. The subsequent increase in ROC AUC indicates a recovery in the model's performance.

IX. Future Work

A. Expansion to New Datasets

1) Mini-drone Video Dataset: The ”Mini-drone Video Dataset” presents a comprehensive collection of 38 high-definition videos, each lasting between 16 to 24 seconds, captured using a Phantom 2 Vision+ mini-drone within a parking lot setting. This dataset is meticulously curated into three distinct categories, depicting a wide range of human activities:

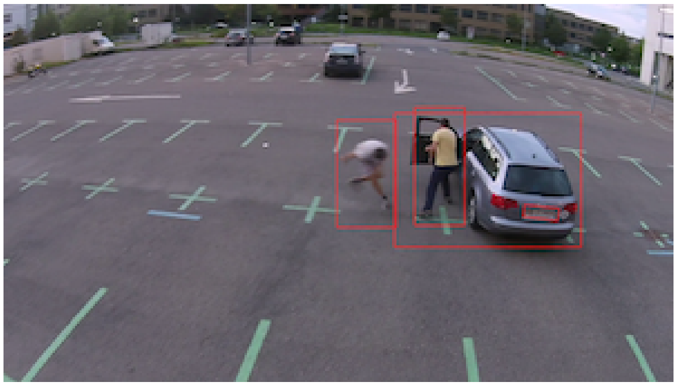


Fig. 12. Mini-drone Video Dataset.

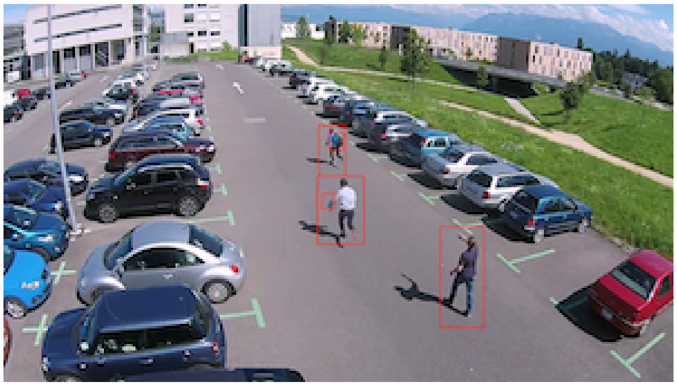


Fig. 13. Mini-drone Video Dataset.

- **Normal Behavior:** Features individuals engaging in everyday activities such as walking, parking vehicles, and getting into cars.
- **Suspicious Behavior:** Encompasses actions that, while not overtly wrong, could be perceived as questionable or out of the ordinary.
- **Illicit Behavior:** Captures various forms of misconduct, including improper parking, theft of items or vehicles, and physical altercations.

Participants in the dataset have given informed consent, and the content is structured to facilitate the evaluation of different privacy levels and definitions.

2) Annotations and Privacy Considerations: Annotations in the dataset include detailed markings of privacy-sensitive regions using the ViPER-GT tool, such as:

- **Body Silhouette:** Marked with additional attributes including gender, ethnicity, and age, alongside the main action and role of the individual.
- **Facial Region:** Due to privacy sensitivity, faces are annotated with a higher level of privacy.
- **Accessories and Personal Items:** Bags, backpacks, and other personal effects are marked for identification.
- **Vehicles and License Plates:** Vehicles are annotated with detail, and license plates have high privacy due to their traceability.

The dataset prioritizes privacy concerns, rating regions of interest on a privacy scale from low to high. Sensitive regions like faces and license plates inherently carry a higher privacy level due to the potential for personal identification. Annotations are supplied in a versatile XML format, providing flexibility for analysis and research purposes.

B. Feature Extraction with I3D-LSTM

Future developments will incorporate the ”I3D-LSTM” model for feature extraction. This model, which combines 3D Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks, is particularly adept at capturing low-level spatial-temporal features and high-level temporal feature sequences. The I3D-LSTM model's use of a video action recognition dataset, such as Kinetics, for pretraining as opposed to still image datasets, presents a promising direction for achieving superior performance in human action recognition tasks.

X. Conclusion

In conclusion, the MGFN framework marks a significant advancement in the realm of weakly-supervised video anomaly detection. Through its innovative use of the Feature Amplification Mechanism, Glance and Focus Blocks, and the Magnitude Contrastive Loss, MGFN effectively addresses the challenges of anomaly localization in lengthy video sequences. Its superior performance on benchmark datasets like UCF-Crime and XD-Violence is indicative of its robustness and efficiency. Future work involving the Mini-drone Video Dataset and feature extraction through I3D-LSTM models holds promise for further elevating the capabilities of anomaly detection systems. The MGFN stands as a testament to the potential of integrating spatial-temporal information to enhance surveillance video analysis and safety measures in public spaces.

References

[1] B. Antic and B. Ommer, ”Video parsing for abnormality detection,” in *2011 International Conference on Computer Vision*, pp. 2415–2422, 2011.

[2] G. Bertasius, H. Wang, and L. Torresani, ”Is SpaceTime Attention All You Need for Video Understanding?” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[3] T. B. Brown et al., ”Language Models are Few-Shot Learners,” *CoRR*, vol. abs/2005.14165, 2020.

[4] Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In European Conference on Computer Vision (ECCV).

[5] Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[6] Chalapathy, R.; and Chawla, S. 2019. Deep Learning for Anomaly Detection: A Survey. CoRR, abs/1901.03407.

[7] Chang, Y.; Tu, Z.; Xie, W.; and Yuan, J. 2022. Clustering Driven Deep Autoencoder for Video Anomaly Detection. In Computer Vision – ECCV 2020, 16th European Conference, 329–345.

[8] Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; and Gao, W. 2021. Pre-Trained Image Processing Transformer. arXiv:2012.00364.

[9] Chi, C.; Wei, F.; and Hu, H. 2020. RelationNet++: Bridging Visual Representations for Object Detection via Transformer Decoder. CoRR, abs/2010.15831.

[10] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[11] Fayyaz, M.; and Gall, J. 2020. SCT: Set Constrained Temporal Transformer for Set Supervised Action Segmentation. CoRR, abs/2003.14266.

[12] Feng, J.; Hong, F.; and Zheng, W. 2021. MIST: Multiple Instance Self-Training Framework for Video Anomaly Detection. In 2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[13] Gong, D.; Liu, L.; Le, V.; Saha, B.; Mansour, M. R.; Venkatesh, S.; and Hengel, A. v. d. 2019. Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection. In IEEE International Conference on Computer Vision (ICCV).

[14] Gopalakrishnan, S. 2012. A public health perspective of road traffic accidents. Family medicine and primary care.

[15] Guansong, P.; Chunhua, S.; Longbing, C.; and Den, H. A. V. 2021. Deep Learning for Anomaly Detection: A Review. ACM Comput. Surv., 54(2).

[16] Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; Yang, Z.; Zhang, Y.; Tao, D. 2020. A Survey on Vision Transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–1.

[17] Hasan, M.; Choi, J.; Neumann, j.; Roy-Chowdhury, A. K.; and Davis, L. 2016. Learning Temporal Regularity in Video Sequences. In Proceedings of IEEE Computer Vision and Pattern Recognition.

[18] Ionescu, R. T.; Khan, F. S.; Georgescu, M.; and Shao, L. 2018. Object-centric Auto-encoders and Dummy Anomalies for Abnormal Event Detection in Video. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[19] Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. the 3rd International Conference for Learning Representations.

[20] Kratz, L.; and Nishino, K. 2009. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, 1446–1453.