Video object and action recognition for criminal activity using YOLO and Hypergraph transformer

Giorgia Bertacchini
274372@studenti.unimore.it
University of Modena and Reggio Emilia
Modena, Italy

Matteo Bulgarelli
276511@studenti.unimore.it
University of Modena and Reggio Emilia
Modena, Italy

Edoardo Pottocar
272638@studenti.unimore.it
University of Modena and Reggio Emilia
Modena, Italy

ACM Reference Format:
Giorgia Bertacchini, Matteo Bulgarelli, and Edoardo Pottocar. 2025. Video object and action recognition for criminal activity using YOLO and Hypergraph transformer. In . ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Abstract

The detection of harmful objects and recognition of criminal acts are critical tasks in ensuring public safety and security. This paper introduces a novel hybrid model that leverages the real-time object detection capabilities of YOLO (You Only Look Once) with the contextual reasoning power of a novel Transformer architecture to address these challenges. The proposed system is designed to identify harmful objects, such as weapons, and analyze temporal sequences to recognize potential criminal behaviors in video footage.

YOLO focuses on detecting harmful objects, such as knives and firearms, within individual frames, ensuring high spatial accuracy and real-time performance, while also extracting people in parallel using a separate YOLO instance. The Transformer then uses the extracted skeletons combined with the tracked objects information to predict their actions by analyzing temporal sequences. This parallel architecture enables the system to associate detected objects with the actions and identities of individuals, facilitating a holistic interpretation of the scene.

We fine-tuned YOLO on a custom dataset of harmful objects to extend its class detection capabilities, ensuring accurate identification of specific threats. For the Transformer, we utilized a pre-trained model that was already trained on a dataset encompassing multiple actions, and fine-tuned it with CCTV violence datasets to fit our specific needs.

The goal is to ensure timely detection and analysis of harmful objects and malicious actions as they occur, allowing for immediate intervention during ongoing events captured in the frames.

2 Introduction

In an era marked by increasing criminality, particularly in densely populated urban areas, the importance of advanced surveillance systems has never been greater. Traditional surveillance often relies on post-event analysis, where authorities intervene only after crimes have occurred, resulting in delayed responses and limited opportunities to prevent further harm. To enhance public safety, it is crucial to develop systems capable of identifying potential threats and malicious actions. Such systems are paramount in enabling law enforcement to act effectively. Beyond facilitating timely interventions, these advanced capabilities also have the potential to act as a deterrent for criminals, who may be discouraged from engaging in illegal activities if they know that sophisticated surveillance is in place.

3 Related work

3.1 Object and people tracking

For Object tracking the state of the art, and one of the most efficient methods, is the "You Look Only Once (YOLO)" [8]. We actually employed two separate instances of the YOLO model, each configured for a distinct task. The first instance was fine-tuned to detect knives, firearms, optimizing its performance for weapon recognition in diverse scenarios. The second instance retained its default configuration but was adapted to exclusively track and identify individuals by suppressing the detection of all other object classes. This dual-model setup enabled precise weapon detection and reliable human tracking, providing a comprehensive framework for addressing the challenges of both tasks within the same system.

3.2 Action and criminal act recognition

For what concerns the general action recognition task in particular, while the state of the art a few years ago heavily relied on RNN, and in particular LSTM [5], new approaches make use instead of graphs and Graph Neural Networks [11].
Our work focuses on human action recognition for surveillance applications, aiming to identify and classify human activities in real-world scenarios.

Traditional approaches to surveillance often rely on anomaly detection methods. However, these methods typically exhibit several limitations:

- Poor Performance in Noisy Environments: Anomaly detection algorithms often struggle in environments with high levels of background noise, such as crowded public spaces. These methods tend to flag unusual events, but may not accurately differentiate between truly anomalous behavior

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

(e.g., criminal activity) and simply unexpected or unusual behavior (e.g., someone running to catch a bus).

- Lack of Granularity: Anomaly detection and anomaly classification are often trained using weakly supervised learning or bag-level classification techniques, which leads to a lack of granularity. Labeling an entire video with an action that can be observed only in a few frames of it doesn't allow to precisely identify the actors and the exact instant the action happens.
- Difficulty in Discriminating Between Good and Bad Actors: In a surveillance context, the primary goal is not simply to identify unusual behavior, but to accurately discriminate between innocent individuals and those engaging in criminal activity. Anomaly detection methods may flag a wide range of behaviors as abnormal, leading to a high rate of false positives and potentially compromising privacy.

To address these limitations, this project explores the use of skeleton-based action recognition techniques, leveraging on the use of a new Transformer architecture. By analyzing the movement patterns of human skeletons extracted from video footage with a self-attention approach, we aim to:

- Achieve Higher Accuracy: Leverage the rich information contained in human skeletal data to improve the accuracy of action classification compared to generic anomaly detection methods.
- Enable Fine-Grained Analysis: Classify a wider range of human actions with greater precision, allowing for the identification of specific behaviors of interest, such as punching, kicking and shooting a gun.
- Improve Discrimination of Criminal Activity: Develop a more nuanced understanding of human behavior, enabling the system to better differentiate between innocent and potentially harmful actions.

This approach allows for a more targeted and effective surveillance system by focusing on the specific actions and behaviors that are most relevant to security concerns.

A project that tries to achieve our same result with a slightly different approach is the one proposed by Rendón-Segador et al.[9], which makes use of a normal ViT, which works with a similarity graph feature map extracted from the optical flow of videos. Even if this approach makes use of the ViT, which is proven to be a good choice for processing sequences, and it's the reason we also make use of it, we assumed that the use of skeleton data instead of optical flow more accurate for the action recognition task.

The model on which we based our work finally, is the one made by Yuxuan et al. [13] called, for short, "Hyperformer", which relies on a hypergraph to analyze and correlate joints positions and skeleton poses to create representative feature vectors, which are then passed to a particular novel Transofmer architecture made by the authors, that makes use of a combination of Hypergraph self-attention layers, and temporal convolutional layers, to make the prediction on the action class.

4 Data

For the instance of YOLO used for object tracking, we utilized a combination of two datasets: the Linksprite Gun Detection [7] dataset, which consists exclusively of firearm images, and the OD-WeaponDetection dataset, from which we selectively extracted only the knife images. The OD-WeaponDetection dataset, available at https://github.com/ari-dasci/OD-WeaponDetection and developed by the Andalusian Research Institute in Data Science and Computational Intelligence, is specifically designed for weapon detection tasks. It comprises a diverse collection of images annotated with bounding boxes and labels identifying various types of weapons, such as guns and knives, along with other common handheld objects. However, since YOLO is already trained on a large dataset containing many everyday objects, we focused only on the firearm images from Linksprite and the knife images from OD-WeaponDetection to enhance the model's knowledge of these weapon types. The data was then pre-processed by applying a stretch and resize operation to each image to standardize them to 640 × 640 pixels, followed by augmentation techniques including blur (up to 2 pixels), salt & pepper noise (1.5% of pixels), horizontal and vertical flips, cropping (2% min zoom and 25% max zoom), and shearing (±13° horizontally and vertically).

On the other hand for the action recognition task we used a combination of 2 different datasets:

- The first one called "A Dataset for -Automatic Violence Detection in Videos master" [2], made by a group of Italian researchers, which is specifically designed to mimic the perspective of a CCTV camera, while recording physical violence actions like punching, kicking, restraining and so on.
- The second one called "Firearm-related action recognition and object detection dataset for video surveillance systems" [10], which is instead a collection of videos made to simulate the occupation of armed people of a building, consisting of people walking around rooms with firearms in various stances and at different walking paces.

While we used the first dataset to reinforce the model comprehension of the violence actions, the second one was mostly utilized for the classes of action that involved the presence of a weapon in the scene, so to be used together with the YOLO-Weapon extracted information. We tried to make a dataset as balanced as possible with a 20% as a test split, and a stratification made to have in each split the number of samples of the "null" class, equal to the sum of samples of elements of all the other classes.

We also acknowledge the dataset on which the Hyperformer model was originally trained: the NTU RGB+D 120 dataset [10], a large-scale benchmark for 3D human activity recognition. While a smaller version of the dataset exists, containing only half of the action classes, we opted to use the full 120-class version. This choice was motivated by the fact that several violent and criminal actions are included exclusively in the complete set, leading us to assume that Hyperformer would already possess some prior knowledge of the types of actions we are interested in detecting.

5 Methods

The project comprises two primary components: a harmful object tracking system and an action recognition pipeline. Each component leverages state-of-the-art deep learning techniques and frameworks to address the problem comprehensively.

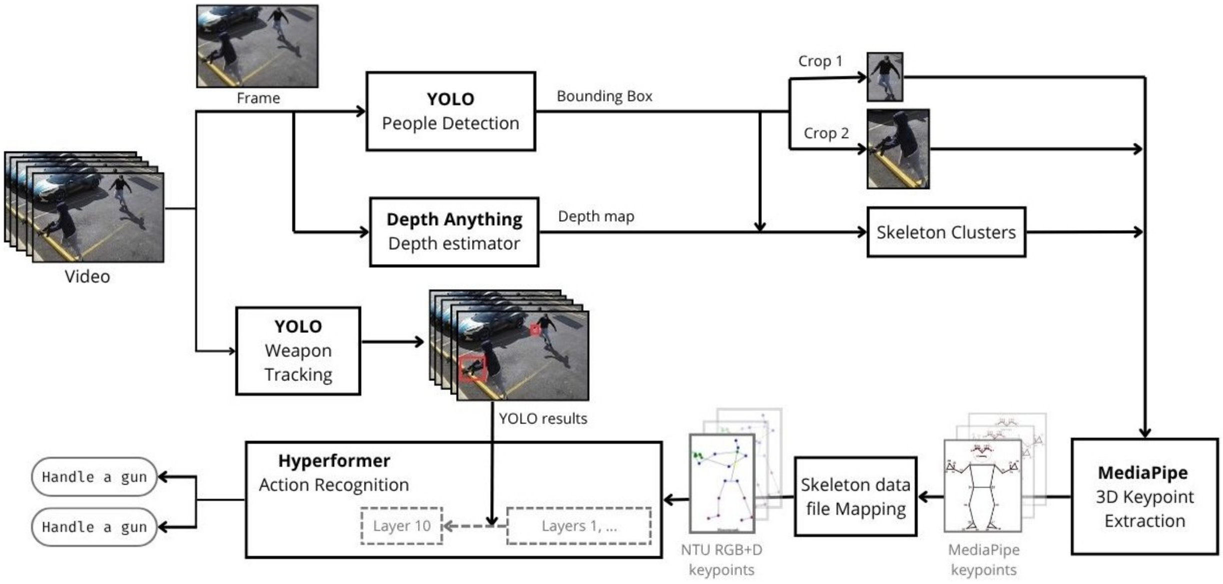

Figure 1: Full architecture pipeline

5.1 Harmful Object Tracking

The first component employs the YOLO (You Only Look Once) object detection framework to identify and track harmful objects, such as firearms and knives, within video frames. YOLO is chosen for its high speed and accuracy in real-time object detection tasks. The model is fine-tuned on a dataset containing annotated images of harmful objects to optimize its detection performance in the target domain. Once detected, the bounding boxes and associated confidence scores are used to track these objects across consecutive frames, enabling consistent monitoring of potentially dangerous items. This model is used in particular just until the layer before the linear one for the object class prediction, to generate the highest resolution activation maps, which are then processed by an average pooling layer to make them 1-dimensional vectors of 216 elements.

5.2 Action Recognition Pipeline

The second component of the project focuses on human action recognition, consisting of multiple interconnected stages:

(1) Person Detection: A separate YOLO instance is utilized to detect and track individuals within video frames. The bounding boxes of the detected persons are extracted and serve to crop the frames in various regions, to obtain small images, each including only one specific person. This crops are then passed as input to the next block.

(2) Depth estimation: The video is processed frame-by-frame in parallel with YOLO, also by an instance of the model "DepthAnything" [12], which extract the depth map of each single frame and uses the YOLO resulting bounding boxes of the frame to cluster people that are near to each other: this is done to create inside each frame a data that expresses if people have to be considered together or close enough, to be able to perform, possibly, a mutual action. This step is crucial because the Hyperformer model is pre-trained on a dataset of skeleton files that almost always contain a pair of human skeletons. Specifically, for certain mutually dependent classes, the model requires the skeleton file of the person performing the action to also include the skeleton of the person undergoing that action. This process generates intermediate files which are then later used by the production of the skeleton files themselves.

(3) Keypoint Extraction: At first, for the keypoints extraction task, we thought of using a 2D keypoint extractor followed by a Stereo Vision method for the extraction of the depth. For the keypoints extractor we firstly went towards OpenPose [3], but it's too old and no longer updated so it was impossible to implement it. Even after finding a better 2D keypoints extractor, the problem was then in the depth estimation method, that didn't achieved good results, so we finally found the MediaPipe framework [6], which makes use of the BlazePose model [1] for both the 2D keypoints extraction, and the z dimension estimation. MediaPipe receives the crops of the individuals and then outputs a set of 33 tri-dimensional keypoints for each person in each frame, capturing detailed body posture and movement information.

(4) Skeleton Transformation: To ensure compatibility with the NTU-RGB+D dataset format used for training the action recognition model, the extracted 33 keypoints are mapped to a reduced set of 25 keypoints. This transformation aligns the data structure with the standardized format of the NTU-RGB+D dataset, allowing seamless integration with the downstream model.

(5) Clustering: At this stage, data extracted from MediaPipe and DepthAnything are integrated to accurately reconstruct the individual skeleton files. This integration ensures that when multiple people are visible in a scene, a person's skeleton

is matched with that of the closest individual. The closest person is then considered the most probable candidate for engaging in a mutual action with them.

(6) Temporal Skeleton File Generation: For each detected person, a skeleton file is created, encapsulating their motion over a variable number of consecutive frames. These files provide a temporal representation of human movement, which is critical for recognizing actions.
(7) Action Prediction with Hyperformer: The generated skeleton files are input into the Hyperformer model, a transformer-based architecture designed for spatio temporal action recognition. The Hyperformer processes the sequential skeleton data and predicts the action being performed by each individual in the video. This model leverages the temporal dependencies and spatial relationships inherent in human movement to deliver accurate action classification.

We used Hyperformer as-is as a base benchmark, which we then tried to improve by not only fine-tuning it with specific CCTV data, but also injecting during fine-tuning the data extracted by the YOLO-Weapon instance. Hyperformer was further modified in order to return 10 target classes chosen by us, instead of the original 120, because we wanted to focus only on violent and harmful actions, while ignoring any other one that has been classified as 'other'. The classes that our models return are:

(1) Throwing punch
(2) Kicking
(3) Pushing
(4) Slapping
(5) Handling gun
(6) Handling rifle
(7) Handling knife
(8) Swinging blunt object
(9) Grabbing/restraining
(10) Other/Nothing

Specifically we created and fine-tuned 3 different versions of the model, from the simplest to the most complex:

- In the first and simplest model, we modified the architecture by concatenating the tensors produced by the last layer of Hyperformer and the vector from YOLO-Weapon, resulting in a feature vector that is then passed to the linear layer. We also had to adapt the linear layer to perform classification over the 10 target classes of interest.
- In the second model, we decided to perform fine-tuning also on the last layer of Hyperformer other than the linear layer (modified as in the previous model). In this case we obviously had to adapt also the last layer of the architecture to accommodate the concatenated data from the YOLO pipeline related to weapons and the skeleton tensors as input.
- In the third model finally, we decided to fine-tune the last three layers and the linear layer (modifying the latter as in the previous models). We had to adapt the input of the third-to-last layer to accommodate the concatenated data from the YOLO pipeline related to weapons and the skeleton tensors. The reason for leaving these additional layers unfrozen was to give the model even more flexibility and understanding capabilities since the original labels differed in a certain measure from the ones we chose.

6 Experiments

Our experimental methodology involved investigating three distinct model configurations, primarily differentiated by the point at which YOLO-Weapon information is integrated into the architecture. This strategic integration allows the model to leverage external knowledge at various depths.

- Hyperf_1: This represents the simplest configuration, where YOLO-Weapon data is injected directly before the final linear layer. Consequently, only this terminal layer is updated during fine-tuning.
- Hyperf_2: In this variant, the YOLO-Weapon data is introduced prior to the last transformer layer. This enables the fine-tuning of both the final transformer layer and the subsequent linear layer, allowing for broader adaptation to the new information.
- Hyperf_3: This is the most comprehensive model, with weapon data introduced before the third-to-last layer (Layer 8). This design grants three transformer layers the opportunity to process and integrate the new data, aiming for the most accurate and robust action predictions.

These experiments were designed to demonstrate the efficacy of our approach and to illustrate how progressively unfreezing more transformer layers and introducing YOLO-Weapon data at earlier stages significantly enhances model performance and action prediction capabilities.

We conducted extensive experimentation with various hyperparameter values to achieve optimal results, tailoring a unique set of parameters for each model variant. Our observations generally aligned with expected behaviors when adjusting commonly recognized hyperparameters:

- Initial manual tuning of the base learning rate, decay rate, and step size yielded suboptimal results, particularly for the Hyperf_1 model, which achieved only 55% accuracy. To address this, we implemented a learning rate scheduler (ReduceLROnPlateau). The integration of this component proved highly effective, leading to a nearly 5% improvement in the Hyperf_1 model's accuracy.
- Batch Size: We observed that excessively large batch sizes, while potentially improving initial performance, concurrently increased the model's susceptibility to overfitting. Consequently, we selected different batch sizes based on model complexity: 128 for Hyperf_1 and Hyperf_2, and 256 for Hyperf_3.
- Number of Epochs: Given the comparatively higher complexity and learning capacity of Hyperf_2 and Hyperf_3, we adjusted the number of training epochs accordingly. Hyperf_1 was trained for 150 epochs, while Hyperf_2 and Hyperf_3 were trained for 100 epochs, allowing us to achieve comparable behavioral outcomes across models.
- Dropout: To regularize the output of Hyperf_2 and Hyperf_3, we fine-tuned the dropout value. Through various trials, a relatively low dropout rate of 0.1 consistently yielded the best prediction performance.

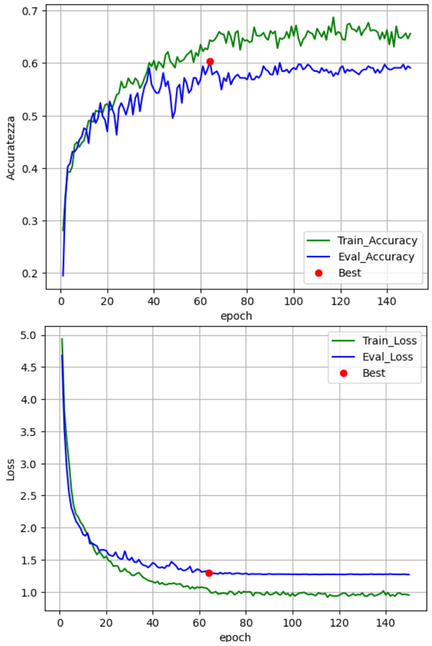

Figure 2: First model results

- Other Parameters: We also explored the impact of other parameters, such as weight decay, which did not produce notable changes in the results. Furthermore, we experimented with different optimizers, including Stochastic Gradient Descent (SGD), Adam, AdamW, and NAdam. SGD consistently led to significantly inferior results, whereas Adam, AdamW, and NAdam, belonging to the same family, exhibited comparable performance. Ultimately, AdamW was selected due to its observed stability during training.

During training we obtained the results that can be seen in images (2), (3), (4). The red dot expresses what is theoretically the best epoch in terms of weights, which are the weights to take in order to avoid overfitting even further.

We then trained the same models but without the incorporation of the YOLO-Weapon data in order to see if this information proved useful to the model accuracy and general performances, and we got the following results:

Table 1: Evaluation best values (red dot on images)

Model	Accuracy	Loss
Hyperf_1	60.38%	1.29
Hyperf_1_noYolo	54.31%	1.55
Hyperf_2	81.15%	1.37
Hyperf_2_noYolo	77.64%	1.51
Hyperf_3	84.03%	1.12
Hyperf_3_noYolo	75.4%	1.63

The major problems we encountered that are also somewhat visible from the results are:

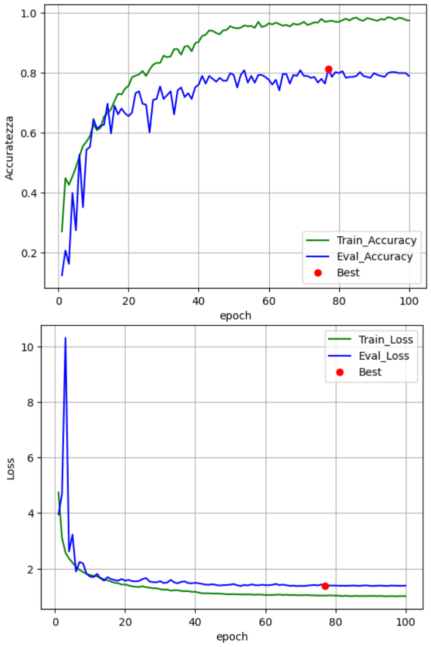

Figure 3: Second model results

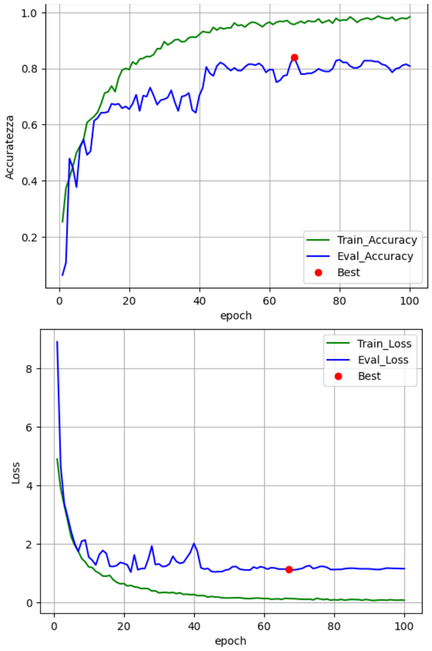

Figure 4: Third model results

- Few data available and of bad quality: mostly because of privacy reasons and old cameras. We actually found some data that we tried to use, but it was impossible because of the bad resolution and/or frame-rate. Most of the actual CCTV recordings had a really bad quality which did not allow to recognize the skeletons, and in some cases even the people, and so we had to settle for laboratory data, which still was very hard to find and in low quantity, over the fact that it was unnatural.

- Another major challenge we faced, primarily stemming from the first two issues, but also from our chosen approach, was the difficulty in isolating individuals within fight scenes using YOLO and then extracting their skeletal pose with MediaPipe from the resulting cropped images. The core problem lies in the high degree of occlusion prevalent in these scenes. This often led to YOLO mis-tracking the same person, mistaking them for another, or losing their track entirely. Additionally, rapid and abrupt movements frequently caused YOLO to discontinue tracking. Consequently, MediaPipe's performance suffered significantly. Because it relies on YOLO's output, any inaccuracies, missing frames, or other flaws in the initial bounding box extractions resulted in MediaPipe producing sparse skeletons (containing many zero values) or failing to extract any pose information whatsoever.
- Our final major challenge relates to the project's real-time applicability and its potential for immediate use by law enforcement.

7 Conclusion

We observed that the first model likely underfits the data, indicated by its poor accuracy and high loss values. Furthermore, the optimal performance point was reached very early in training, suggesting that the learning rate scheduling could be refined. Specifically, the initial learning rate might be too high, and its subsequent reduction after the first decay step might be too aggressive. A positive aspect of this model, however, is its relatively small generalization gap between training and evaluation performance. This indicates that while the model learns effectively, it may lack sufficient complexity to achieve higher accuracy.

In contrast, the second and third models clearly exhibit overfitting. This is evident from the substantial generalization gap, approximately 20 percentage points, observed between their training and testing plots. Additionally, the consistently high loss values on the testing set suggest that these models are memorizing the training data rather than truly learning the underlying task. Moreover, their evaluation plots show considerable fluctuation, further confirming unstable generalization.

Given these results, we concluded that while hyperparameters could likely be fine-tuned further to achieve more desirable outcomes (especially for the first model, where a more advanced learning rate scheduler might push its "best" performance to a later epoch), the primary challenges stem from the scarcity and potential artificiality of the data.

Despite these limitations, our experiments allowed us to demonstrate a key hypothesis: the YOLO-Weapon data consistently enhanced the overall performance of each model. Specifically, the first and second models improved by a modest amount, around 6 percentage points, whereas the third model exhibited an average enhancement of almost 10

This approach, despite yielding suboptimal results in our specific case, holds promise for the identification of criminal actions, the tracking of potential weapons or harmful objects, and the integration of these two information streams to achieve more accurate classification of events within a violent scene.

For future improvements we thought of some possibilities:

- The first one is related to the problem of real time applicability. While YOLO is capable of real-time processing, the subsequent data processing pipeline, which takes the extracted people and weapons, demands significant time and computational resources. To achieve real-time functionality in the future, we could explore processing very small sets of frames. This approach would emulate a sliding window sampling of a video, allowing us to determine if the model can still accurately predict actions while drastically reducing processing time.
- The second and possible further improvement could be to try to generalize even more the approach of the hypergraphs to represent not only single skeletons but also the relationship between close people, in order to generate a feature vector already representing the possible interaction between the two skeletons.
- Another possible improvement could be done on data in order to get more of it, so the most basic approach could be to apply data-augmentation on the dataset original videos, but another way that we actually tried to explore, unsuccessfully, was the usage of a Variational Auto-Encoder (VAE) [4] to artificially generate more feature maps both for skeletons and YOLO-Weapon.

References

[1] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. Blazepose: On-device real-time body pose tracking. *CoRR*, abs/2006.10204, 2020. URL https://arxiv.org/abs/2006.10204.

[2] Miriana Bianculli, Nicola Falcionelli, Paolo Sernani, Selene Tomassini, Paolo Contardo, Mara Lombardi, and Aldo Franco Dragoni. A dataset for automatic violence detection in videos. *Data in Brief*, 33:106587, 12 2020. doi: 10.1016/j.dib.2020.106587.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018. URL http://arxiv.org/abs/1812.08008.

[4] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

[5] Jun Liu, Gang Wang, Ling-Yu Duan, Ping Hu, and Alex C. Kot. Skeleton based human action recognition with global context-aware attention LSTM networks. *CoRR*, abs/1707.05740, 2017. URL http://arxiv.org/abs/1707.05740.

[6] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172, 2019. URL http://arxiv.org/abs/1906.08172.

[7] Delong Qi, Weijun Tan, Zhifu Liu, Qi Yao, and Jingfeng Liu. A gun detection dataset and searching for embedded device solutions. *CoRR*, abs/2105.01058, 2021. URL https://arxiv.org/abs/2105.01058.

[8] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. URL http://arxiv.org/abs/1506.02640.

[9] Fernando J. Rendón-Segador, Juan A. Álvarez García, Jose L. Salazar-González, and Tatiana Tommasi. Crimenet: Neural structured learning using vision transformer for violence detection. *Neural Networks*, 161:318–329, 2023. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2023.01.048. URL https://www.sciencedirect.com/science/article/pii/S0893608023000606.

[10] Jesus Ruiz-Santaquiteria, Juan D. Muñoz, Francisco J. Maigler, Oscar Deniz, and Gloria Bueno. Firearm-related action recognition and object detection dataset for video surveillance systems. *Data in Brief*, 52:110030, 2024. ISSN 2352-3409. doi: https://doi.org/10.1016/j.dib.2024.110030. URL https://www.sciencedirect.com/science/article/pii/S2352340924000040.

[11] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *CoRR*, abs/1801.07455, 2018. URL http://arxiv.org/abs/1801.07455.

[12] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. pages 10371–10381, 2024.

[13] Yuxuan Zhou, Zhi-Qi Cheng, Chao Li, Yanwen Fang, Yifeng Geng, Xuansong Xie, and Margret Keuper. Hypergraph transformer for skeleton-based action recognition, 2023. URL https://arxiv.org/abs/2211.09590.