Real-Time Violence Detection in Video Footage Using a Mobile-Friendly CNN-Based Model

Real-Time Violence Detection in Video Footage Using a Mobile-Friendly CNN-Based Model

1st Malvina Halilaj
Department of Computer Science
University of Tirana
Tirana, Albania
malvina.halilaj@fshn.edu.al

2nd Valeria Cannone
Department of Information Engineering
Polytechnic University of Marche
Ancona, Italy
s1125525@studenti.univpm.it

3rd Paolo Sernani
Department of Law
University of Macerata
Macerata, Italy
paolo.sernani@unimc.it

4th Aldo Franco Dragoni
Department of Information Engineering
Polytechnic University of Marche
Ancona, Italy
a.f.dragoni@univpm.it

***Abstract*—Detecting violence in video content, particularly within domestic environments, presents an ongoing challenge in both social and technological contexts. This paper proposes a lightweight deep learning framework for real-time violence detection, optimized for mobile and edge deployment. The approach is based on MoViNet-A0, evaluated in both Base and Stream configurations, and is complemented by a custom Conv2D-based baseline designed for ultra-low-latency inference. All models were trained and validated on the AIRTLab dataset, which includes 350 annotated videos representing violent and non-violent scenes. The MoViNet-A0 Base model achieved a validation accuracy of 92.8%, while the Conv2D-based model reached 89.6% validation accuracy, along with a precision and F1-score close to 90%. Performance benchmarks conducted on Android devices and desktop platforms show that real-time inference is feasible, with latencies as low as 0.9 seconds per 10-frame sequence on mid-range smartphones.**

The entire pipeline has been designed for mobile deployment, and integration into a functional prototype application is currently in progress, aiming to enable real-time violence detection directly on mobile devices.

***Index Terms*—deep learning, violence detection, mobile inference, MoViNet-A0, Conv2D, Real Time Distributed, edge AI.**

I. Introduction

Violence, particularly domestic violence, remains a widespread societal issue affecting vulnerable populations such as women, children, the elderly, and marginalized communities. Many violent incidents go unreported due to fear, lack of evidence, or social barriers, making early detection and intervention extremely challenging. In recent years, advances in Artificial Intelligence (AI) and Deep Learning have enabled new methods for automated violence recognition, especially through video analysis. The ability to detect violent behavior in real-time holds great potential for improving public safety and enabling rapid intervention in critical scenarios.

The detection of violence in video streams, however, remains a complex task for modern surveillance and safety systems. Traditional approaches often rely on computationally intensive architectures or handcrafted features, which are poorly suited to real-time applications on mobile or embedded platforms. As the demand for efficient and portable AI solutions increases, lightweight video classification models have become an attractive area of research.

In this work, we investigate the effectiveness of MoViNet-A0, a mobile-optimized convolutional video network, for real-time violence detection. We also evaluate an alternative baseline architecture based on a combination of two-dimensional convolutional layers (Conv2D) and real-time distributed inference principles, designed for low-power environments with strict latency requirements. Both models are tested on the AIRTLab dataset, a curated collection of violent and non-violent scenes suitable for training and evaluating violence detection systems.

Research question: *Can the MoViNet-A0 model provide reliable real-time violence detection on mobile devices, achieving accuracy comparable to larger models while reducing latency and computational overhead?*

To address this question, we conduct a series of experiments aimed at measuring the classification performance, resource usage, and inference latency of each model. The results are analyzed to understand the trade-offs involved in deploying deep learning systems for violence detection on mobile and embedded devices.

II. MoViNet: Mobile Video Network

MoViNet (Mobile Video Network) is a family of computationally efficient deep learning models designed for real-time video classification [1]. It supports inference on streaming video and utilizes 3D convolutional networks (Fig. 1.); however, in our application, we use it with 2D convolutional networks to process single frames.

A. 3D CNN or 2D?

To optimize real-time processing, MoViNet employs a stream buffer mechanism that separates memory usage from

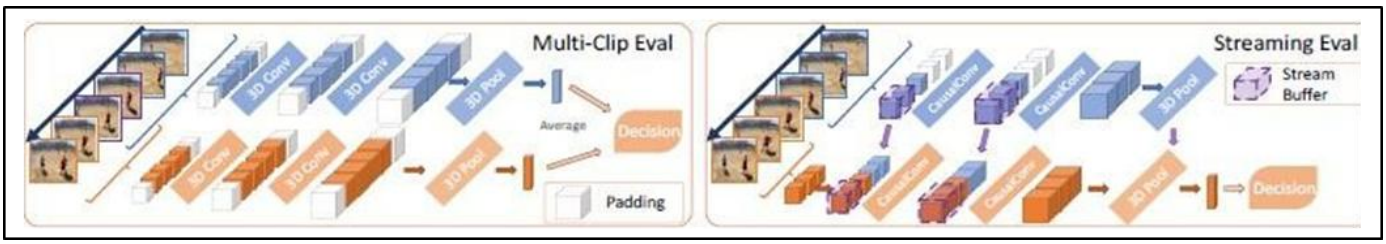

Fig. 1. MoViNet Structure

video clip duration. This allows the network to handle streaming video sequences of arbitrary length while maintaining a constant memory footprint, enabling efficient online inference without requiring frame batching into fixed-length clips. MoViNet can be implemented using either 3D convolutional neural networks (CNNs) or Conv2Ds. While 3D CNNs [2] provide higher accuracy by capturing temporal dependencies between frames, they require significantly more computational resources and are not well-suited for real-time streaming applications due to their reliance on future frames[1]. Conversely, Conv2Ds process spatial dimensions in each frame independently and aggregate temporal features over time. Although they may not capture temporal dependencies as effectively as 3D CNNs, studies have demonstrated that Conv2Ds, when combined with techniques such as rolling averages, can achieve comparable performance while being considerably more computationally efficient. This makes Conv2Ds a more practical choice for environments with limited computational resources. By integrating MoViNet into our system, we aim to provide accurate and efficient real-time detection of violent scenes, even on resource-constrained devices. This approach holds significant potential for enhancing safety measures and enabling timely interventions in domestic settings.

B. MoViNet-A0: A Lightweight Model for Mobile Deployment

The MoViNet family includes several variants (A0–A6), each designed to balance accuracy and computational cost. Higher variants such as A4–A6 achieve state-of-the-art performance on large-scale datasets like Kinetics-600 (e.g., 83.5% top-1 accuracy for A6 [3]), but require substantial compute resources. In contrast, MoViNet-A0 is the smallest and most resource-efficient variant, using a 172×172 input resolution and depthwise separable convolutions to enable real-time inference on mobile and embedded devices.

Due to these characteristics, MoViNet-A0 was selected for this study as the optimal choice for edge deployment. Despite its compact design, it achieves strong performance (72.28% top-1 accuracy and 90.92% top-5 accuracy on Kinetics-600 [3]), offering an excellent trade-off between efficiency and classification quality.

III. MoViNet-A0: Base version VS Stream Version

Both variants of MoViNet-A0 are based on a 3D convolutional neural network (3D CNN) designed to extract spatio-temporal features from video sequences. The structure is optimized for computational efficiency, balancing depth and width of the model, in order to maintain low memory requirements and high inference speed. The model is mainly composed of:

- Separable 3D convolutional blocks: allow to decompose the volumetric convolution into separate spatial and temporal operations, reducing the computational cost.
- Temporal pooling: progressively reduces the temporal dimension, maintaining the relevant features.
- Nonlinear normalization and activations: batch normalization and ReLU or Swish to stabilize training and improve learning.
- Final classifier: fully connected layer that produces the prediction on the video or frame.

The MoViNet-A0 Stream and MoViNet-A0 Base models are two variants of the MoViNet family, designed for efficient video processing in low-latency environments, such as mobile or edge devices.

A. MoViNet-A0 Base

The Base variant of MoViNet-A0 is characterized by an architecture that processes the entire video segment as input in a single inference. This approach allows the model to exploit the entire available temporal information, without limitations imposed by temporal causality constraints. The input consists of a sequence of video frames, on which the model performs optimized three-dimensional convolutional operations to extract spatio-temporal features. The non-causal inference mode allows to achieve high performance in terms of accuracy, being particularly suitable for offline applications or where latency is not a stringent constraint. However, this model requires a larger memory capacity and a longer processing time, since the entire video must be acquired before the prediction can be performed.

B. MoViNet-A0 Stream

The Stream version of MoViNet-A0 was developed with a specific focus on real-time video processing. Unlike the Base variant, MoViNet-A0 Stream operates in a causal mode, processing video frames sequentially. The internal streaming component incorporates internal memory mechanisms (hidden states) to retain key information from previous frames, thus allowing it to operate in real time without having to wait for the entire video. This mode enables low-latency processing with limited computational requirements, making it ideal for edge or mobile applications where fast response is crucial, such as live event recognition or real-time surveillance. While

maintaining similar size and complexity to the Base model, the streaming mode may imply a slight trade-off in terms of accuracy, due to the limited availability of future information during prediction.

IV. Alternative Baseline: Conv2D with Real-Time Considerations

To evaluate the trade-off between computational complexity and detection performance, we implemented a lightweight baseline model based on 2D convolutional layers. Unlike architectures such as ResNet, LSTM, or MoViNet, this model processes short sequences of video frames using spatial-only convolutions, avoiding the overhead of temporal or recurrent modules.

The proposed Conv2D-based architecture was designed with real-time constraints in mind. It simulates streaming input by processing frames sequentially without pre-buffering, enabling efficient inference on resource-constrained devices such as smartphones. This setup aligns with mobile deployment scenarios, where low latency and minimal power consumption are essential.

This model serves as a comparative baseline to assess the effectiveness of MoViNet-A0. While deep or recurrent architectures often provide higher accuracy, they typically require greater computational resources and are less suitable for real-time execution on edge devices. Our goal was to determine whether a simpler, modular solution could deliver acceptable performance while significantly reducing inference time and system complexity.

As shown in Section VI, the Conv2D baseline achieved competitive results and sub-second latency on mid-range mobile hardware, confirming its viability for real-time detection.

Recent work in video-based violence detection combines spatial-temporal features with attention to improve classification. One notable example is the method by Fu et al. [10], which integrates a pruned EfficientNet-B0 with a self-attentive Separable ConvLSTM and background suppression. The model achieved 90.75% accuracy on the RWF-2000 dataset with only 2.4 million parameters, outperforming heavier architectures like I3D and ConvLSTM. However, the system was trained on a high-end GPU and evaluated offline, limiting its suitability for real-time or mobile applications.

In contrast, our work focuses explicitly on real-time, low-latency deployment. We evaluate MoViNet-A0, a mobile-optimized architecture based on causal separable 3D convolutions, and a lightweight Conv2D-based model designed for efficient streaming inference. Both models are tested directly on commercial smartphones, demonstrating competitive accuracy and sub-second latency without GPU support.

V. Methodology

In this study, we developed a deep learning model for automatic violence detection in videos using MoViNet-A0 from TensorFlow Hub [2]. The methodology consisted of the following steps:

- *Dataset Preparation*: We used the AIRTLab dataset [4], which consists of labeled videos categorized into violent and non-violent scenes, captured from two cameras (cam1 and cam2).
- *Frame Extraction*: Single frames were extracted at a resolution of 172×172 and organized into labeled directories using OpenCV [6], an open-source computer vision library widely used for video and image processing tasks.
- *Preprocessing*: The extracted frames were normalized and preprocessed to improve computational efficiency.
- *Model Integration*: MoViNet-A0 was integrated as a feature extractor [6]-[8] within a custom TensorFlow model, with additional fully connected layers for classification.
- *Training and Optimization*: The model was trained using the Adam optimizer with sparse categorical cross-entropy following the approach outlined in the official TensorFlow Hub tutorial [4][7][8]. We tested also binary cross-entropy loss, ensuring optimized learning performance in both cases.
- *Deployment*: After training, the model was converted to TensorFlow Lite [8] format for efficient deployment in an Android application.

A. Dataset description

The dataset used for training is the AIRTLab dataset, consisting of 350 videos labeled as violent or non-violent. Actions in the videos were simulated by non-professional actors. Violent clips show common aggressive behaviors like kicking and punching, while non-violent clips specifically include behaviors such as hugging, clapping, and cheering, which can cause false positives in violence detection due to their rapid movements and similarity to violent actions. The dataset is organized into two main subdirectories, “cam1” and “cam2,” each corresponding to a different camera angle:

- Non-violent/cam1 & cam2: 60 clips each;
- Violent/cam1 & cam2: 115 clips each.

All videos are in MP4 format (H.264 codec), with a resolution of 1920×1080 pixels and a frame rate of 30 fps. The clips have an average duration of 5.63 seconds, ranging from 2 to 14 seconds.

VI. Experimental Setup

In this section, we describe the configuration used to train and evaluate the proposed models.

A. Data Processing and Input Format

The AIRTLab dataset was preprocessed by extracting frames from each video at a resolution of 172×172 pixels using OpenCV. Each sample consists of a sequence of 10 frames with 3 color channels, resulting in an input shape of [1, 10, 172, 172, 3] (Table I), where:

Normalization was applied to map pixel values to the [0, 1] range. Data augmentation included horizontal flipping and brightness variation to improve model generalization.

TABLE I
INPUT PARAMETERS FOR TRAINING SAMPLES

Parameter	Value
Batch size	1
Number of frames	10
Resolution	172×172
Channels	3

The dataset of 350 videos was divided as follows:

- Training set: 60%;
- Validation set: 20%;
- Test set: 20%.

Class imbalance was addressed using augmentation primarily on the minority class (non-violent). In the MoViNet-A0 Base model, data can be processed in batches larger than one; however, we used the configuration represented in Table I. Summarizing in Table II below the configuration used for all the models tested:

TABLE II
TRAINING CONFIGURATION FOR ALL MODELS

Parameter	Value
Input shape	[1, 10, 172, 172, 3]
Batch size	1
Epochs	10
Optimizer	Adam
Learning rate	0.001
Loss function	Sparse Categorical Crossentropy / Binary Crossentropy
Augmentation	Horizontal flip, brightness variation

B. Hardware and Inference Environment

The training phase was conducted on a local desktop machine equipped with an Intel CPU (detailed specifications to be provided). No dedicated GPU was used during training, which emphasizes the lightweight nature of the proposed models.

To evaluate the deployment performance in real-world scenarios, inference tests were conducted across various platforms, including both emulated and physical Android devices, as well as a desktop PC environment. Specifically, the models were benchmarked using the TensorFlow Lite (TFLite) framework, focusing on the MoViNet-A0 Stream and the Conv2D-based architectures.

The platforms used for inference are listed below:

- *Samsung Galaxy A50*: 8-core processor, 4 GB RAM.
- *Samsung Galaxy A12*: Entry-level device with 3 GB RAM.
- Redmi Note 8 Pro: Mid-range device with 6 GB RAM.
- *Desktop PC*: Python environment running TFLite interpreter for latency benchmarking under optimal conditions.

Inference latency was measured for each platform using a 10-frame video sequence with input shape [1, 1, 172, 172, 3]. This setup simulates real-time streaming inference by passing one frame at a time. As for the combination with Real Time Distributed and Conv2D, a single inference was performed with the entire image sequence, but with a smaller size, i.e. from 172 to 128 pixels. In detail, the following input was used: [1,10,128,128,3].

TABLE III
AVERAGE INFERENCE TIME FOR EACH DEVICE

Device	Input	Avg Time
Pixel 4 (Emu)	[1,1,172,172,3]	2.75 s
Galaxy A50	[1,1,172,172,3]	15.2 s
Local PC	[1,1,172,172,3]	25 ms
Galaxy A12	[1,10,128,128,3]	2.78 s
Redmi Note 8 Pro	[1,10,128,128,3]	0.89 s

TABLE IV
MEASURED INFERENCE TIMES PER DEVICE (MS)

Device	Times
Pixel 4 (Emu)	2756, 2749, 2756, 2747
Galaxy A50	15428, 15438, 15295, 15186
Local PC	29.11, 24.58, 25.10, 25.15, 26.54
Galaxy A12	2730, 2864, 2866, 2712
Redmi Note 8 Pro	868, 900, 855, 930

VII. RESULTS

This section presents the performance evaluation of the proposed models, including MoViNet-A0 (Base and Stream variants) and a Conv2D-based baseline. We begin by analyzing the performance trade-offs between the Base and Stream variants of the MoViNet-A0 architecture (Table V).

TABLE V
PERFORMANCE COMPARISON OF MOVINET-A0 BASE AND STREAM

Metric	Base	Stream
Accuracy	0.8969	0.67
Loss	0.2104	0.55
Val accuracy	0.9280	0.65
Val loss	0.1626	0.63

MoViNet-A0 Base achieved high accuracy but required more processing time. The Stream version offered real-time inference but showed a moderate drop in accuracy.

Additional experiments were conducted using alternative models compatible with mobile inference. One of these is represented by a combined architecture between Real Time Distributed and Conv2D. The model was always trained with input [1,10,172,172,3]. It is inspired by the original architecture of MoViNet. These are the results obtained from the training (Table VI- Table VII):

TABLE VI
TRAINING METRICS FOR REAL-TIME DISTRIBUTED + CONV2D

Metric	Value
Accuracy	0.9960
Loss	0.0584
Validation Accuracy	0.8958
Validation Loss	0.3524

Regarding the classification report on 48 samples:

Graphically we can observe in the images below these values from the confusion matrix (Fig. 2) and the training

TABLE VII
DETAILED METRICS FOR REAL-TIME DISTRIBUTED + CONV2D MODEL

Class	Precision	Recall	F1-Score	Support
Non-violent	88%	92%	90%	24
Violent	91%	88%	89%	24
Accuracy	—	—	90%	48
Macro avg	90%	90%	90%	48
Weighted avg	90%	90%	90%	48

history with accuracy (Fig. 3) and loss (Fig. 4). The orange segment indicates the training history of the validation phase, while the blue one indicates the training history in the training phase.

The confusion matrix in Fig. 2. summarizes the classification performance of the MoViNet-A0 model on the validation set. The model correctly identified 22 non- violent and 21 violent video instances, while misclassifying 2 non-violent cases as violent and 3 violent cases as non-violent. This demonstrates a balanced performance with high precision and recall for both classes, confirming the model's robustness in differentiating between violent and non-violent actions.

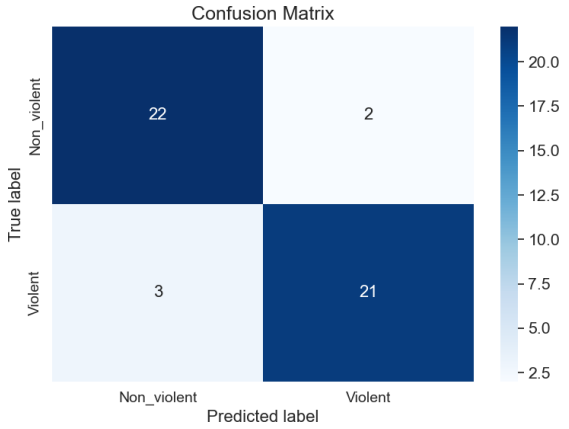

Fig. 2. Confusion Matrix

Fig. 3. illustrates the model's accuracy over 10 training epochs. The training accuracy steadily improves, reaching close to 98%, while the validation accuracy stabilizes around 92%. Despite minor fluctuations, the validation curve closely follows the training trend, indicating good generalization and minimal overfitting. These results confirm the effectiveness of the MoViNet-A0 architecture for this classification task.

Fig. 4. shows the model's loss progression over 10 epochs. The training loss consistently decreases, reflecting improved model learning. Although the validation loss initially spikes, it quickly stabilizes and follows a downward trend, reaching a low value around epoch 9. This behavior indicates effective convergence and good model generalization, with minimal risk of overfitting.

Subsequently, the MoViNet-A0 Stream model was slightly improved, yielding the performance metrics reported in Table VII.

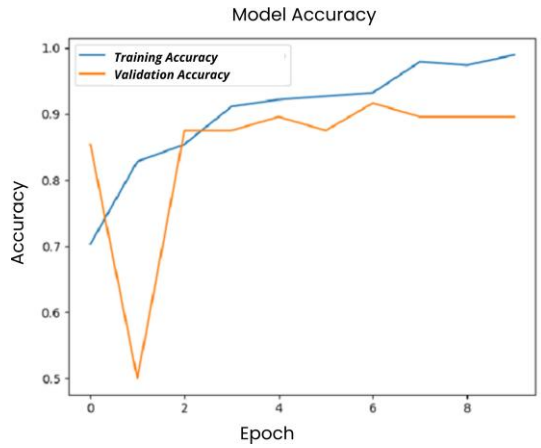

Fig. 3. Accuracy training history

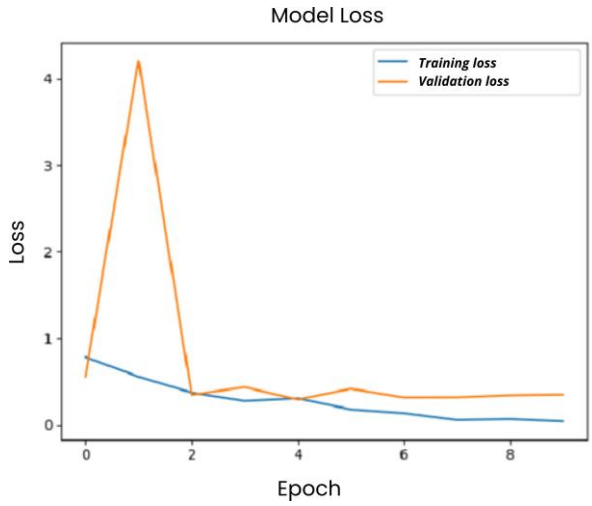

Fig. 4. Loss training history

The model was trained with input shape `[1, 10, 172, 172, 3]` for 70 epochs in 10 hours with CPU. Evaluation results are illustrated through the confusion matrix in Fig. 5, and the training history of accuracy (Fig. 6) and loss (Fig. 7).

The confusion matrix shows that most violent and non-violent samples were correctly classified, though some confusion remains between classes, particularly false negatives. For this reason, the model is still being actively refined to improve performance and stability in real-time applications.

VIII. DISCUSSION AND CONCLUSION

Our experimental results reveal a clear trade-off between classification performance and real-time feasibility in mobile environments. While the MoViNet-A0 Base model achieved the highest validation accuracy (92.8%), its latency on mobile devices remains a limiting factor for real-time use without acceleration techniques. In contrast, the Stream variant offers significantly reduced latency and lower memory usage, albeit

TABLE VIII
EVALUATION METRICS FOR THE IMPROVED MOVINET-A0 STREAM MODEL

Class	Precision	Recall	F1-Score	Support
Non-violent	67.93%	93.04%	78.52%	843
Violent	95.14%	75.62%	84.26%	843

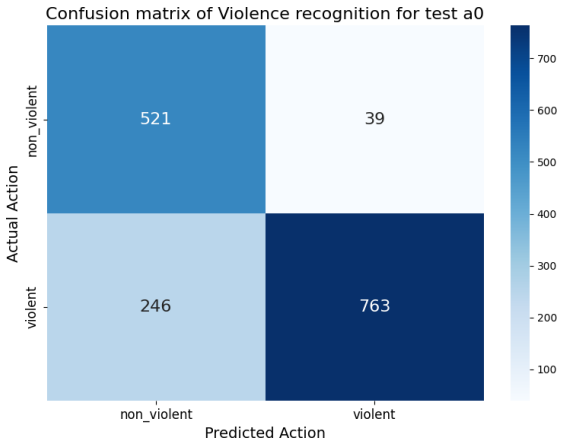

Fig. 5. Confusion matrix showing performance of the improved MoViNet-A0 Stream model

Fig. 6. Accuracy training history

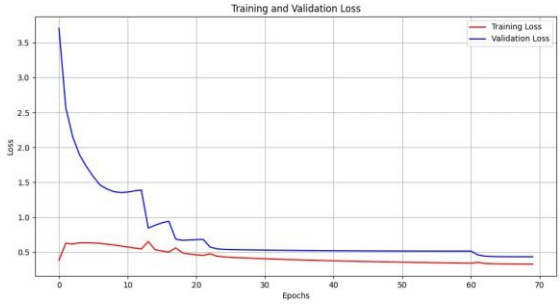

Fig. 7. Loss training history

with moderately lower accuracy (up to 84.3% F1-score for violent class after improvements).

The Conv2D + Real-Time Distributed baseline demonstrated that simpler architectures can still deliver competitive results—achieving 90% accuracy and inference times below one second on mid-range smartphones—making it a promising option for embedded applications where responsiveness and efficiency are critical.

This qualitative analysis confirms that deploying deep learning systems for on-device violence detection involves balancing accuracy, latency, and resource consumption. Our findings suggest that lightweight and modular solutions can enable practical real-time violence detection, even on low-power mobile platforms.

Future developments will focus on enhancing the MoViNet-A0 Stream variant, integrating pose estimation and audio-based cues, and expanding the dataset to further improve generalization. The complete integration into a mobile app is ongoing, with the goal of enabling reliable real-time inference for home security use cases.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Prossima srl for their support during this project, which was co-financed by the Marche Region under the Regional Development Fund (PR) ERDF 2021/2027 – Specific Objective 1.1 – Action 1.1.1 – Call for Proposals 2023. This work was carried out as part of the project “AI Assistant for Home Security” (CUP: B79J24000910007).

DECLARATION ON GENERATIVE AI

During the preparation of this work, the author(s) used X- GPT-4 and Gramby for: Grammar and spell checking. After using these tools/services, the authors have reviewed and edited the content as necessary and take full responsibility for the content of the publication.

REFERENCES

[1] D. Kondratyuk, L. Yuan, Y. Li, L. Zhang, M. Tan, M. Brown, and B. Gong, ”MoViNets: Mobile Video Networks for Efficient Video Recognition,” *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

[2] S. Ji, W. Xu, M. Yang, and K. Yu, ”3D Convolutional Neural Networks for Human Action Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, 2012.

[3] J. Carreira et al., “A short note about Kinetics-600,” arXiv preprint arXiv:1808.01340, 2018.

[4] AIRTLab, “A Dataset for Automatic Violence Detection in Videos,” GitHub Repository, Available: https://github.com/airtlab/A-Dataset-for-Automatic-Violence-Detection-in-Videos

[5] OpenCV Developers, “Open Source Computer Vision Library,” [Online]. Available: https://opencv.org/.

[6] TensorFlow Hub, ”MoViNet: Mobile Video Networks for Efficient Video Recognition,” Available: https://www.tensorflow.org/hub.

[7] TensorFlow Developers, “Transfer Learning with MoViNet,” Available: https://www.tensorflow.org/tutorials/video/transfer_learning_with_movinet.

[8] TensorFlow Developers, ”TensorFlow: An End-to-End Open Source Machine Learning Platform,” Available: https://www.tensorflow.org/.

[9] TensorFlow Colab Tutorial, “MoViNet Demo,” Available: https://colab.research.google.com/github/tensorflow/models/blob/master/official/projects/movinet/movinet_tutorial.ipynb

[10] Z. Fu, J. Huang, and L. Zhang, “Violence Detection Using EfficientNet-B0 and SA-Separable Convolutional LSTM,” in *Proc. Int. Conf. on Neuromorphic Computing (ICNC)*, 2024, pp. 1–6.