Enhancing Security of Smart Cities with “Signal for Help” Recognition System

Enhancing Security of Smart Cities with “Signal for Help” Recognition System

Federico Buccellato, Corrado De Sio,
Eleonora Vacca, and Sarah Azimi
Dipartimento di Automatica e Informatica
Politecnico di Torino
Turin, Italy
{name.surname}@polito.it

***Abstract*— The COVID-19 pandemic brought an alarming surge in violence against women and children, referred to as The Shadow Pandemic. In response, a Canadian foundation introduced the "Signal for Help" gesture, a discreet way for individuals in danger to alert others. However, the success of this gesture hinges on its recognition and appropriate reaction by bystanders. This paper introduces an innovative real-time system designed to detect these silent pleas for help within surveillance footage. The system integrates three key components: a person tracking mechanism utilizing YOLOv7 and Deep SORT to identify and follow individuals in videos; a hand feature extraction module based on MediaPipe to capture hand-related data; and a machine learning classification model to discern the presence of a help request. Our proposed model and pipeline architecture deliver real-time inference speeds without compromising on prediction accuracy, offering a potent tool to enhance safety in smart cities.**

Keywords: Violence Detection, Smart Cities, Artificial Intelligence.

I. INTRODUCTION

Despite technological progress, one of humanity's most complex issues remains unresolved: gender violence. Although violence and abuse are not new phenomena, global stressors like the COVID-19 pandemic have worsened gender-based violence, a situation called "The Shadow Pandemic." Victims often struggle to report their experiences due to fear, leading them to remain silent and hope the situation improves on its own. The Canadian Women’s Foundation introduced the "Signal for Help," a non-verbal method for victims to discreetly indicate they are in danger, as shown in Figure 1. This hand gesture involves placing the palm outward, tucking the thumb into the palm, and folding the fingers over the thumb. Widely recognized and used in public places to signal danger, the gesture's effectiveness depends on people recognizing and understanding it. Despite extensive awareness campaigns, some instances where the gesture was used went unnoticed, highlighting the need for broader recognition and understanding.

State-of-the-art solutions addressing violence detection are based on Convolutional Neural Networks (CNNs) [1][2], which require visible items to recognize violence (e.g., weapons, blood, etc.). Still, only a few approaches [3][4] have extended detection capabilities to situations without explicit violence indicators. Previous works on the detection of Signal for Help [5][7] mainly rely on 3D CNN to detect hand patterns within clips recorded by surveillance cameras. However, domain adaptation issues often affect such solutions, especially when the training dataset is limited.

The work presented in this paper aims to propose a system that leverages smart city technology to identify emergency signals and notify the relevant authorities, by combining surveillance cameras with advanced AI algorithms. This technology can be deployed in public areas such as restaurants, shopping malls, and parks, offering individuals a discreet means to call for help and enhancing their likelihood of receiving timely assistance.

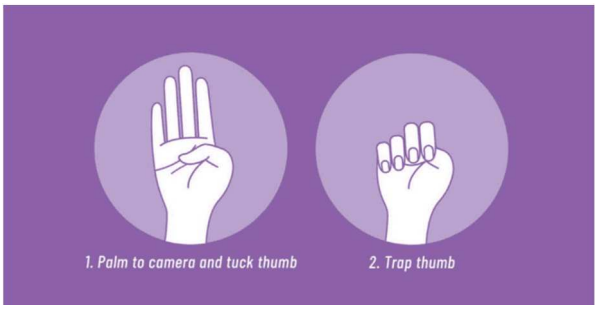

Figure 1. The Signal for help hand gesture.

A. Main Contribution

We propose a highly efficient real-time system designed to detect the "Signal for Help" hand gestures using video feeds from surveillance cameras while operating on low computational devices. Our methodology is based on a three-step pipeline:

- The system processes the input video from surveillance cameras, outputting frames cropped to the bounding boxes of each person.
- These cropped frames are analyzed for the presence of hands; if hands are detected, their features are extracted using the MediaPipe framework.
- The extracted features are classified by a machine learning (ML) model to determine the presence of a "Signal for Help" gesture.

Please note that not only does our approach detect the occurrence of Signal for Help, but it is also effective in the presence of more than one person and consecutively more than two hands in the scene, which has not been elaborated within previous works, but it is mandatory to make the system suitable to operate in a real-world scenario.

Our approach achieves 91% accuracy in detecting the "Signal for Help" gesture and effectively addresses the domain adaptation challenges commonly associated with 3D CNNs encountered in previous works.

979-8-3503-5229-0/24/$31.00 ©2024 IEEE

The rest of the paper is organized as follows. Section 2 introduces related works on violence detection methodologies. Section 3 presents the background necessary to understand the proposed model. Section 4 deeply details the proposed solution, while experimental results are provided in Section 5. Finally, Section 6 addresses the conclusion and paves the way for future work.

II. Related Works

Given the multifaceted nature of violence, researchers have developed various methodologies to address these complex issues. For instance, Clarin et al. [1] introduced a system that categorizes extreme actions by identifying visual clues such as skin and blood. Their method uses self-organizing maps and assesses pixel motion intensity to classify scenes involving violence. Similarly, Zajdel et al. [3] developed the CASSANDRA system, which detects video motion patterns and identifies auditory cues resembling screams. By leveraging both video and audio data, they employ dynamic Bayesian networks to identify aggressive behavior in public spaces. Chen et al. [2] proposed another approach focused on motion detection of facial expressions and the presence of blood, aiming to detect and analyze violent behavior through these specific visual indicators. Furthermore, an innovative technique proposed by Deniz [4] was adopted as a primary feature for violence detection of people's sudden movements, offering a novel approach to this challenging problem.

The Signal for Help gesture presents unique challenges because of its nonverbal and secretive nature, which distinguishes it from typical, more overt, violent actions. Consequently, this covert request for assistance poses difficulties that conventional violence detection mechanisms cannot effectively address. Additionally, since it predominantly involves hand gestures, it's crucial to prevent false positives triggered by similar everyday gestures. As a result, Signal for Help falls under gesture recognition challenges.

Recent research has devoted significant attention to hand gesture recognition within the fields of gesture control systems and the interpretation of sign language interpretations [8][9][10]. These efforts have achieved notable results through the use of CNNs [11][12][13]. A notable contribution in this field was made by Dominio [14], who proposed a methodological framework that combines various depth-based descriptors for hand gesture recognition. In a separate endeavor, Liang [15] developed an innovative methodology that leverages depth-based features for gesture detection, complemented by fingertip tracking using a particle filter. This combination of advanced techniques has improved the precision and reliability of gesture recognition. Additionally, Hong [16] introduced a new technique for gesture recognition based on the analysis of convexity defect histograms, providing an additional level of sophistication and accuracy to this field of research. Based on previous studies, initial models for recognizing the Signal for Help have been explored, primarily using CNNs [5][7]. These models face significant challenges due to the nature of 3D convolution filters, which operate at the pixel level and extract features from both the background and foreground of images or videos, incurring domain adaptation [17]. To differentiate between important and irrelevant features for prediction tasks, these models require training with extensive datasets that exhibit high variability in background pixels, lighting conditions, and other factors. Specifically, authors in [7] proposed a solution relying on a small dataset of 112 images, which the authors increased to 2352 images through data augmentation techniques. However, the limited size of the dataset may lead to insufficient variability both without and with augmentation. As a result, the reported accuracy results of 87.5% without augmentation and 100% with augmentation might rise some doubts. Indeed, when only a small amount of data is available, characterized by limited variability in recording setup, lighting conditions, camera position, and other configurations, achieving a high level of generalization with 3D-CNNs is very challenging. On the other hand, the proposed solution in [5] is based on transfer learning with the Jester dataset for the pre-training phase. However, when pre-training a model on a dataset like Jester and then fine-tuning it for the Signal for Help recognition, there is an implicit assumption. Specifically, this approach assumes that the features extracted by the pre-trained model are transferable and can effectively discriminate between the Signal for Help and non-Signal for Help classes.

While transfer-learning techniques can be effective for similar tasks, this assumption can pose another significant problem for the 3D-CNN solution. Given these limitations, our proposed method bypasses these issues by using a new approach: a three-step pipeline based on YOLOv7, DeepSort, and Mediapipe [18][19][20]. Specifically, by integrating MediaPipe into the detection system, we can automatically filter out any information unrelated to the hand gesture. This guarantees that each model of the proposed pipeline focuses exclusively on pertinent features to the Signal for Help gesture. Moreover, the classifiers employed in this framework boast a low parameter count, facilitating effective training even with a limited number of samples without incurring overfitting on the training dataset.

III. Background

This section details the general characteristics of the three key models exploited in our proposed approach to detect the Signal for Help gesture. Specifically, YOLOv7, DeepSORT, and MediaPipe Hand. These models play a crucial role in various applications, ranging from object detection to hand tracking and gesture recognition. Finally, we will introduce the "Signal For Help" dataset, which has been used as the starting point to train the proposed model.

A. YOLOv7

YOLOv7 [18], the acronym for You Only Look Once version 7, is a state-of-the-art object detection model based on a deep CNN architecture. It excels in detecting objects in both images and videos with remarkable accuracy and real-time performance. YOLOv7's hallmark feature lies in its ability to rapidly process input data without compromising detection precision, making it ideal for applications requiring swift and reliable object detection.

B. DeepSORT

Real-Time Deep Learning-based Object Tracking combines deep learning techniques with traditional tracking algorithms to provide robust object-tracking capabilities in video sequences. By leveraging deep neural networks,

DeepSORT [19] enables accurate association of objects across frames, enhancing tracking precision in challenging scenarios such as occlusions and complex motion patterns.

C. MediaPipe Hand

Developed by Google, it is a state-of-the-art hand-tracking model designed to accurately detect and track human hands in images and video streams. Unlike traditional object detection models, MediaPipe [20] associates each hand with 21 key points, as shown in Figure 2, enabling precise localization and tracking of hand movements. This fine-grained level of detail facilitates advanced hand gesture recognition and analysis tasks, making it suitable for applications such as sign language interpretation, hand gesture-based user interfaces, and augmented reality interactions. Additionally, MediaPipe's robustness and efficiency enable deployment in various computing environments, ensuring smooth and responsive hand tracking even in resource-constrained settings.

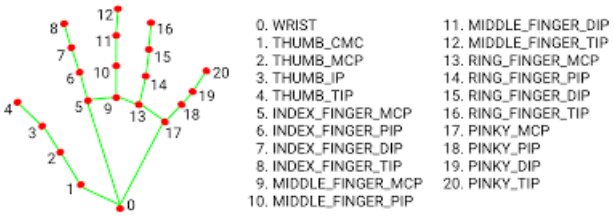

Figure 2. Hand landmarks extracted by the MediaPipe [21].

D. Signal for help Dataset

This medium-to-large dataset, developed in collaboration with Alta Scuola Politecnica (ASP), with students from Politecnico di Milano and Politecnico di Torino, contains approximately 4000 videos. These videos, recorded using a variety of devices, angles, and quality levels, present a wide array of scenarios, covering both indoor and outdoor settings with diverse lighting conditions and backgrounds. We took this dataset as our starting point. To further enhance our model's learning capability, especially in recognizing ambiguous gestures that the initial dataset struggled with, we added additional videos. These supplementary videos were specifically curated to include scenarios where the gestures are less clear, ensuring that our model could learn to distinguish between subtle differences and improve its accuracy. This diversity has proven invaluable for training our model, ensuring it can generalize well across different conditions and enhancing its robustness and effectiveness in real-world applications.

IV. The "Signal for Help" Detection Model

The proposed methodology for efficiently detecting the "Signal for Help" hand gesture utilizes a three-stage pipelined flow that can be summarized as the Person Tracking stage, the Feature Extraction, and the Real-time detection stages. Initially, the model processes video input is captured by a camera. This video feed is then transmitted to the first stage of the pipeline, which tracks the people present in the video. For each person, batches of 30 frames at a time are extracted and sent to the next step for hand feature extraction. Finally, all the extracted features are passed to a classifier, determining whether the Signal for Help gesture is present. An overview of the proposed methodology is provided in Fig. 3.

A. Person Tracking

In the Signal for Help detection mechanism, the first step involves tracking and verifying the presence of persons in the input video. This functionality is accomplished by integrating the YOLOv7 model and the DeepSORT algorithm. Specifically, YOLOv7 is responsible for person detection, while DeepSORT handles person tracking. The entry point of our model is a version of YOLOv7 pre-trained on the MS COCO dataset [22]. The person tracking stage is crucial for two main reasons. Firstly, it aims to prevent unnecessary data processing in subsequent stages if no individuals are detected. Since the videos are processed in real time and continuously sourced from security cameras, this verification step helps avoid computational waste. Secondly, it allows handling scenarios where multiple persons are present in the video. Indeed, for each detected person, this step is tasked with tracking each individual, cropping their bounding box, and storing the data for each person. To maximize the probability of capturing the sequence of the "Signal for help" gesture using the minimum number of frames possible, a batch of 30 frames is used. Once a batch of 30 frames available, it is sent to the next step. In cases where multiple people are present within the frame, for each individual, when enough frames are stored, they are sent in parallel to step two for hand feature extraction.

For each person, the batch of frames is obtained through a sliding window, as shown in Fig. 4. Specifically, each batchi overlaps with the batchi-1 by 15 frames, being 15 the optimal overlapping value, allowing the capture of comprehensive temporal information. This is particularly important since Signal for Help is a sequence of hand gestures.

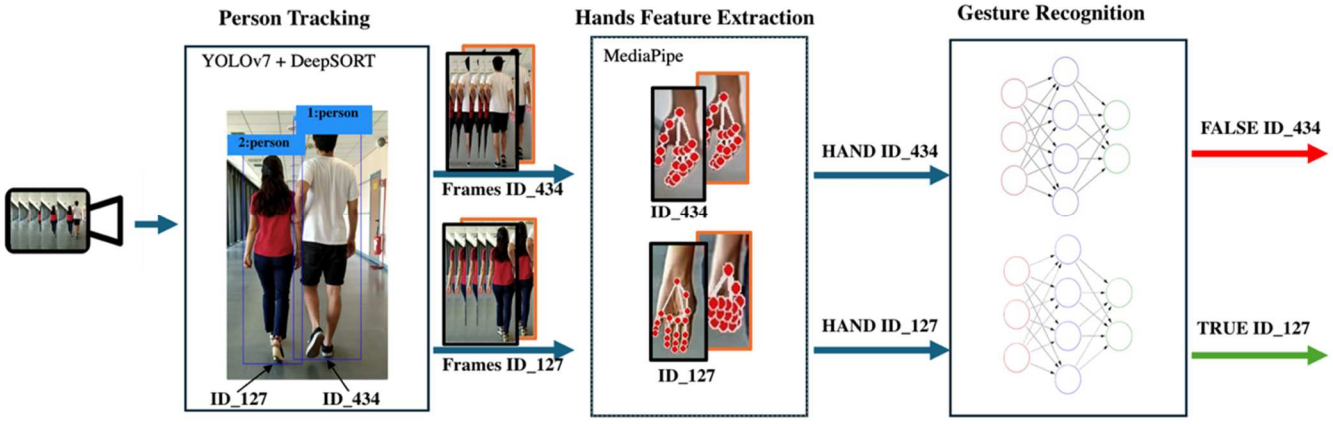

Figure 3. An overview of our proposed three-step pipeline model.

B. *Feature Extraction*

For each person detected and tracked in the previous step, one batch of 30 frames at a time is feeding the Feature Extraction module. The feature extraction is implemented by exploiting the Google MediaPipe framework. This step aims to extract hand landmarks from the input frames using the MediaPipe Hand Landmarker task. Specifically, given the input image, the task returns a list of hands, with each hand represented by a set of 21 three-dimensional points in the (x,y,z) coordinates. Processing these points for all 30 frames, we extract spatial and temporal hand information, creating a single feature corresponding to hand movement in a specific batch. The resulting time series dimension, related to each hand, is calculated by multiplying the number of landmarks by the three dimensions and the number of frames, as shown in Eq. 1.

$$\text{Timeseries} = \text{Landmarks} \times \text{Spatial Dimensions} \times \text{Frames} = 21 \times 3 \times 30 = 1890 \quad (1)$$

After the features are normalized, they are passed to the next step for classification.

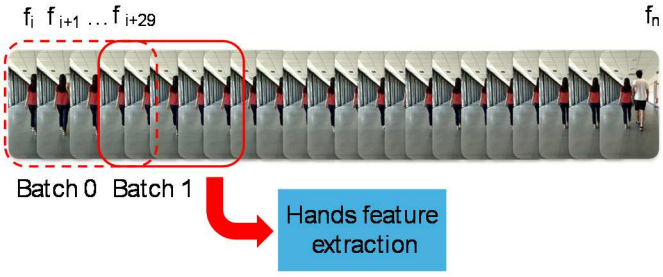

Figure 4. Representation of the sliding window process on a video.

C. *Real-time Signal for Help Detection*

The final step of the proposed pipeline involves a machine learning (ML) model trained using features extracted from the developed Signal For Help Database. After extensive evaluation of different ML approaches testing, the Random Forest (RF) model was selected for its superior accuracy and ability to handle the diverse feature set effectively. The RF model, an ensemble learning method, constructs multiple decision trees during training and combines their outputs to make a final prediction. This approach reduces overfitting and improves the model's generalization ability from the training data to unseen examples. The trained RF random Forest model processes the hands' features to identify potential instances of the Signal for Help. Given the critical nature of accurate detection and the potential consequences of false positives, we have integrated a double-check mechanism. This mechanism incorporates temporal redundancy. Specifically, if the model encounters the signal only once, it takes no action. However, if it detects the signal more than once within the specified time frame, it triggers the alert. By incorporating this double-check step, we aim to significantly reduce the likelihood of false positives, thereby increasing the precision and trustworthiness of our detection system. This approach ensures higher accuracy and minimizes unnecessary alerts, optimizing the overall efficiency and reliability of the surveillance system.

V. Experimental Results

The proposed methodology was tested on multiple devices. Initially, it was tested on a machine equipped with an Apple M3 chip, featuring a 3.2 GHz CPU, 8 GB RAM, and an integrated Apple M3 GPU. Subsequently, it was tested on an NVIDIA Jetson Orin Nano to evaluate its performance on an embedded system. As mentioned in the previous section, we evaluated the performance of different ML classification models to select the solution offering the optimal tradeoff between latency and accuracy. Specifically, we evaluated Random Forest (RF) [23], Support Vector Machine (SVM) [24], Logistic Regression (LR) [25], K-Nearest Neighbors (KNN) [26], Multilayer Perceptron (MLP) [27], AdaBoost (Ada) [28], and Gaussian Naive Bayes (GNB) [29] models to conduct classification. These models have been compared concerning the following performance metrics.

Accuracy

This metric indicates the model's ability to correctly classify instances across all classes [30][31]. It is calculated as the ratio of the sum of true positive (TP) and true negative (TN) instances to the total number of instances, including TP, TN, false positive (FP), and false negative (FN) .

$$\text{Accuracy}=(TP+TN)/(TP+TN+FP+FN) \quad (2)$$

Class-specific Precision

Precision measures the accuracy of positive predictions for a specific class [32]. It is calculated as the ratio of TP to the sum of TP and FP. A high precision indicates a low false positive rate.

$$\text{Precision}=TP/(TP+FP) \quad (3)$$

Class-specific Recall

Recall, also known as sensitivity, quantifies the ability of the model to correctly identify positive instances of a specific class [32]. It is calculated as the ratio of TP to the sum of TP and FN. A high recall indicates a low false negative rate.

$$\text{Recall}=TP/(TP+FN) \quad (4)$$

Class-specific F1-Score

The F1-Score combines precision and recall into a single value. It is the harmonic mean of precision and recall and provides a balanced classifier performance assessment. It ranges from 0 to 1, where a higher value indicates better performance [33].

$$\text{F1-score}=2 \times (\text{Precision} \times \text{Recall})/(\text{Precision}+\text{Recall}) \quad (5)$$

Area Under the Curve

The Area Under Curve (AUC) is a fundamental measure used to evaluate the performance of a binary classifier. It represents the area under the Receiver Operating Characteristic (ROC) curve, which illustrates the model's ability to discriminate between positive and negative classes across different decision thresholds. The AUC provides a comprehensive assessment of the model's discriminative ability, regardless of the decision threshold used. It is a value ranging from 0 to 1, where a higher AUC indicates better discriminative ability. In practice, an area under the ROC curve closer to 1 corresponds to a classifier better performing separating positive and negative classes [30][31].

A. Model Results

The complete Signal for Help detection mechanism was comprehensively evaluated across different classification model solutions, using the performance metrics specified earlier. The results of this evaluation are presented in Table I. The RF model stands out for its superior performance across most metrics, achieving the highest accuracy (91.7%), precision (94.9%), and F1-score (90.3%). Only in the AUC metric does the MLP (Multilayer Perceptron) neural network hold a slight advantage, with a score of 0.9668 compared to 0.9641 for the RF.

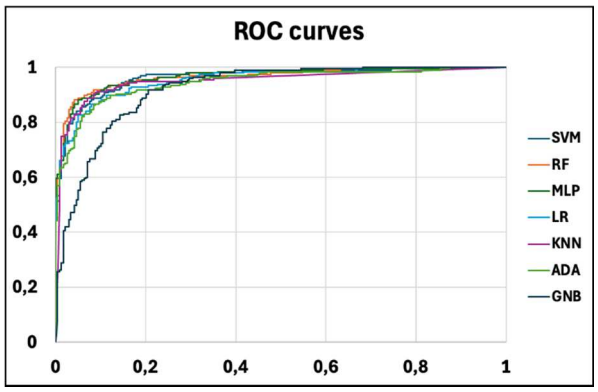

Figure 5. ROC curves, comparison of all classifiers under consideration.

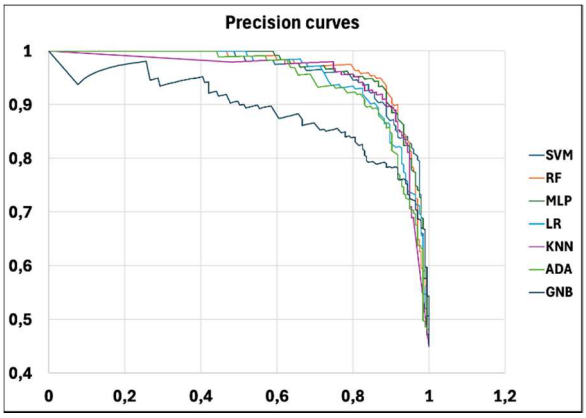

Figure 6. Precision-Recall curves, comparison of all classifiers under consideration.

However, this difference in AUC is minimal. RF's strong performance can be attributed to its robust learning approach, which allows it to handle smaller or noisier datasets more effectively than neural networks or traditional classifiers [23]. Its ability to maintain high accuracy, even with diverse and imbalanced data, combined with its resistance to outliers and noise, makes it particularly well-suited for real-world applications. For further comparison, Fig. 5 shows the ROC curves for all models, while Fig. 6 displays the Precision-Recall curves. It's important to note that all these curves, except for the Gaussian Naive Bayes (GNB), look very similar. Apart from GNB, the plots point out that no model truly outperforms the others across all metrics. However, considering the overall balance of metrics such as accuracy, recall, and F1-score, RF was ultimately selected as the most reliable model.

The graphical representation of the Random Forest's confusion matric in Fig. 7 further validates its efficacy, demonstrating its comprehensive recognition of the "Signal for help" gesture while maintaining a low probability of misclassification.

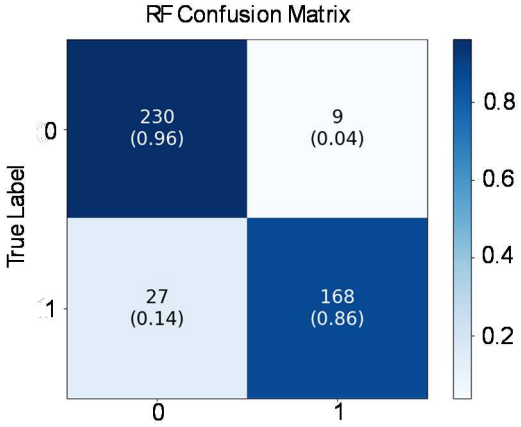

Figure 7 Confusion matrix of Random Forest model.

TABLE I. Performance metric comparison for various models

	Accuracy	AUC	Precision	Recall	F1-score
RF	0.9170	0.9641	0.9491	0.8615	0.9032
SVM	0.9009	0.9629	0.9222	0.8512	0.8853
LR	0.8894	0.9554	0.9298	0.8153	0.8688
KNN	0.9055	0.9505	0.9184	0.8606	0.8918
MLP	0.9101	0.9668	0.9431	0.8512	0.8948
Ada	0.8824	0.9460	0.8956	0.8358	0.8647
GNB	0.8041	0.9209	0.8618	0.6517	0.7550

B. Hardware Results

As mentioned earlier, the model was tested on multiple devices, specifically the Apple M3 and Jetson Orin Nano. Tables II provide additional insights into the inference speed and memory usage for both devices. The results show that the Jetson Orin Nano has a significantly faster inference speed compared to the Apple M3. However, the Apple M3 uses substantially less RAM memory during operation, likely due to the device's more intelligent memory usage [34].

TABLE II. Comparison of inference speed and memory usage across multiple devices.

Device	Inference time (s)	Memory (GB)
Mac M3	1.585	2.5
Jetson Orin Nano	0.715	4.5

Given that the final application will be deployed on an embedded system, these results are encouraging. The faster inference speed of the Jetson Orin Nano ensures that the system can respond quickly to real-time gestures, which is essential for the Signal for Help detection.

VI. Conclusions and Future Works

In this work, we proposed a three-step pipeline system able to perform real-time detection of the Signal for Help hand gesture. The developed model consists of two neural networks and a classifier. While the model addresses the state-of-the-art domain adaptation problem and it has proved to be suitable for real-time applications, it has a high

computational cost. One possible solution to reduce the computational overhead could be developing a hand-tracking model tailored specifically for this setup. Additionally, the developed Signal for Help dataset enables us to achieve satisfactory results, reaching 91.70%. of accuracy, we plan to increase the dataset further, realizing videos more similar to the final applications of the system to surveillance cameras to improve the model's training capabilities.

ACKNOWLEDGMENT

This research is a component of the PoC Instrument Linea 1- PoC Launchpad, Safe Smart City project, funded by Fondazione Compagnia di San Paolo. Additionally, we extend our gratitude to the members of the S2CITIES project at Alta Scuola Politecnica for their efforts in creating the "Signal for Help" dataset and their valuable input in developing the proposed methodology.

REFERENCES

[1] C. Clarin, J. Dionisio, M. Echavez, P. Naval, "DOVE: detection of movie violence using motion intensity analysis on skin and blood," PCSC, vol. 6, pp. 150–156, 2005.

[2] L.-H. Chen, C.-W. Su, H.-W. Hsu, "Violent scene detection in movies," Int. J. Pattern Recognit. Artif. Intell., vol. 25, pp. 1161–1172, 2011.

[3] W. Zajdel, J.D. Krijnders, T. Andringa, D.M. Gavrila, "CASSANDRA: Audio-Video Sensor Fusion for Aggression Detection," IEEE Conf. Advanced Video and Signal Based Surveillance, AVSS 2007 Proc., pp. 200–205. IEEE, 2007.

[4] O. Deniz, I. Serrano, G. Bueno, T.-K. Kim, "Fast violence detection in video," in Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP), vol. 2, pp. 478–485, 2014.

[5] S. Azimi, C. De Sio, F. Carlucci, and L. Sterpone, "Fighting for a future free from violence: a framework for real-time detection of 'signal for help'," ISSN 2667-3053, 2023.

[6] S. Azimi, C. De Sio and L. Sterpone, "Enhanced Video Surveillance Systems for "Signal for Help" Detection on Edge Devices," IEEE International Symposium on Technology and Society (ISTAS), 2023.

[7] G. Elliott, K. Meehan, J. Hyndman, ``Using CNN and Tensorflow to recognise 'Signal for Help' Hand Gestures," IEEE, 2021.

[8] R. Agrawal and N. Gupta, "Real time hand gesture recognition for human computer interaction," in Proceedings of the IEEE 6th international conference on advanced computing (IACC), pp. 470–475, 2016.

[9] R. Amutha Ashwini, R. Rajavel and D. Anusha, "Classification of daily human activities using wearable inertial sensor," in Proceedings of the international conference on wireless communications signal processing and networking (WiSPNET), pp. 1–6, 2020.

[10] Y. S. Tan, K. M. Lim and C. P. Lee, "Hand gesture recognition via enhanced densely connected convolutional neural network," Elsevier Expert Systems with Applications, 175, Article 114797, 2021.

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin and X. Tang, "Temporal segment networks: Towards good practices for deep action recognition," In Proceedings of the European conference on computer vision, pp. 20–36, Springer, 2016.

[12] Rajangam Athilakshmi, Ramadoss Rajavel and Shomona Gracia Jacob, "A survey on deep-learning architectures," Journal of Computational and Theoretical Nanoscience, 15(8), 2577–2579, 2018.

[13] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," In Proceedings of the advances in neural information processing systems, pp. 568–576, 2014.

[14] F. Dominio, M. Donadeo and P. Zanuttigh, "Combining multiple depth-based descriptors for hand gesture Recognition," Pattern Recognition Letters, vol. 50, pp. 101-111, ISSN 0167-8655, 2014.

[15] H. Liang, J. Yuan and D. Thalmann, "3D fingertip and palm tracking in depth image sequences," Proceedings of the 20th ACM international conference on Multimedi, ACM, New York, NY, USA, pp. 785-788, 2021.

[16] J. Hong, E.S. Kim and H.J. Lee, "Rotation-invariant hand posture classification with a convexity defect histogram," Circuits and Systems (ISCAS), 2012 IEEE International Symposium on, vol. X, no. X, pp. 774-777, 2012.

[17] G. Csurka, "Domain Adaptation for Visual Applications: A Comprehensive Survey," in *Domain Adaptation in Computer Vision Applications*, Springer Series: Advances in Computer Vision and Pattern Recognition, G. Csurka, Ed., Springer, 2017, pp. 1-1. DOI: 10.1007/978-3-319-58347-1_1.

[18] C.-Y. Wang, A. Bochkovskiy and H.-Y. M. Liao, ``Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv:2207.02696, 2022.

[19] N. Wojke, A. Bewley and D. Paulus, ``Simple online and realtime tracking with a deep association metric,' arXiv:1703.07402, 2017.

[20] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, M. Grundmann, ``Mediapipe hands: on-device real-time hand tracking," arXiv:2006.10214, 2020.

[21] Hand landmarks detection, MediaPipe Hands solution. https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker?hl=it

[22] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," arXiv:1405.0312 [cs.CV], 2014.

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[24] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of Machine Learning Research*, vol. 2, pp. 125-137, 2001.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[26] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175-185, 1992.

[27] G. Singh and M. Sachan, "Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition,"IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, pp. 1-5, 2014.

[28] M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, AdaBoost and Bregman distances," *Machine Learning*, vol. 48, pp. 253-285, 2002.

[29] I. Rish, "An empirical study of the naive Bayes classifier," in *IJCAI Workshop on Empirical Methods in Artificial Intelligence*, pp. 41-46, 2001.

[30] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, 2006.

[31] C. D. Brown and H. T. Davis, "Receiver operating characteristics curves and related decision measures: A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 80, no. 1, pp. 24-38, 2006.

[32] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Research*, vol. 2, no. 1, pp. 37–63, 2011.

[33] Y. Sasaki, "The truth of the F-measure," *Teach Tutor Mater*, vol. 1, no. 5, pp. 1-5, 2000.

[34] L. Sterpone, et al,"Analysis and Mitigation of Soft-Errors on High Performance Embedded GPUs," 2022 21st International Symposium on Parallel and Distributed Computing (ISPDC), Basel, Switzerland, 2022, pp. 91-98, doi: 10.1109/ISPDC55340.2022.00022.