



# EVENT DETECTION SYSTEM FOR THE PENITENTIARY INSTITUTIONS USING MULTIMODAL DATA AND DEEP NETWORKS

Piotr Bilski <sup>1</sup>, Marcin Lewandowski <sup>1</sup>, Adrian Bilski <sup>2,\*</sup>, Andrzej Buchowicz <sup>1</sup>,  
Jacek Olejnik <sup>3</sup>, Paweł Mazurek <sup>1</sup> and Konrad Jędrzejewski <sup>4</sup>

<sup>1</sup>*Institute of Radioelectronics and Multimedia Technology, Warsaw University of Technology, Warsaw, Poland*

<sup>2</sup>*Institute of Information Technology, Warsaw University of Life Sciences – SGGW, Warsaw, Poland*

<sup>3</sup>*JAS Technologie Sp. z o.o., Warsaw, Poland*

<sup>4</sup>*Institute of Electronic Systems, Warsaw University of Technology, Warsaw, Poland*

\*Corresponding author: [adrian.bilski@sggw.pl](mailto:adrian.bilski@sggw.pl)

**Abstract** The aim of the paper is to present the distributed system for the unwanted event detection regarding inmates in the closed penitentiary facilities. The system processes large number of data streams from IP cameras (up to 180) and performs the event detection using Deep Learning neural networks. Both audio and video streams are processed to produce the classification outcome. The application-specific data set has been prepared for training the neural models. For the particular event types 3DCNN and YOLO architectures have been used. The system was thoroughly tested both in the laboratory conditions and in the actual facility. Accuracy of the particular event detection is on the satisfactory level, though problems with the particular events have been reported and will be dealt with in the future.

**Keywords:** deep learning, posture-based event detection, multimodal analysis

## 1. Introduction

The contemporary society faces multiple challenges related to the internal and external security. The efficiency of the police and internal affairs agencies influences the national security level, as perceived by the citizens. Modern technologies come with help regarding tasks of monitoring and detecting dangerous or unwanted events (such as robbery, vandalism, or civil unrest) based on the input from the surveillance cameras. The most advanced cities are already supervised by the multiple types of sensors and systems (including the infrared or thermal imaging devices). They are supported at the increasing rate by the Artificial Intelligence (AI), which assists human operators in the proper situation assessment. Image processing coupled with Deep Learning (DL) are able to automatically detect and isolate objects, persons or their specific behavioural traits.

Though such monitoring systems belong to the state-of-the art and are introduced commercially, there are many specific applications where the closed-architecture systems fail. For instance, in the case of the closed penitentiary institutions (such as prisons) the set of the detectable events is very different from the typical scenarios encountered in the outside world. For instance, besides the typical aggressive behaviour (with fights being the most obvious), there are also multiple location-specific events, including the

suicide or escape attempts. In such a location the already existing systems are relatively simple, with most of the responsibility put on the human operator. Introduction of the AI-based modules faces multiple challenges, making the whole endeavour complex and difficult to accomplish.

The aim of the paper is to present the AI-based system for the unwanted (anomalous) event detection based on many multimedia streams delivered to the computing platform. Characteristic features and parameters of the system (considering the predefined requirements) are presented, followed by the detailed description of the solution, both from the hardware and software perspective. The AI part is based on the well-known Artificial Neural Network (ANN) architectures, but manages them in the specific way, applicable to the location it is implemented in. Experimental results (both in the laboratory conditions and in the field) show the system can operate in the real-world conditions and ensures accuracy high enough to support the human operator.

The content of the paper is as follows. In Section 2 the current knowledge about the event-detection technologies is presented. Section 3 contains specific requirements for such a system working in the closed facilities (such as prisons). In Section 4 the architecture of the system and the used DL algorithms are discussed. Experiments and their results are in Section 5. Conclusions about the system's performance and future prospects are in Section 6.

## 2. State-of-the-Art

Human action recognition in CCTV surveillance systems is a critical aspect for security and monitoring applications [5, 19, 31]. Techniques based on Deep Learning have gained particular importance for video action recognition in recent years [38]. Progress in this field has been significant, although less spectacular than in 2D image analysis. Currently DL are the leading solution due to their high accuracy and ability to mimic visual cortex [6]. Three-dimensional Convolutional Neural Networks (3D CNN), being the extension of the classic 2D CNNs into the third dimension was introduced in [21]. This extension is a basis on which the Convolutional 3D (C3D) neural network [35] was created. Another important implementations of the 3D CNN concept are the Inception 3D (I3D) network [11] and the Squeeze-and-Excitation Layer C3D (SELayer-C3D) model [22]. Various structures of Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM) Networks have been proposed to enhance the accuracy of identifying human actions in surveillance footage [13, 27]. Hybrid Deep Neural Networks (DNN) like Convolutional LSTM (ConvLSTM) Networks and Long-term Recurrent Convolutional Networks (LRCN) have been developed to classify actions efficiently, with models utilizing different architectures for spatial-temporal feature extraction. Additionally, the use of Self-Organizing Maps (SOM) based on time-series inference skeletons extracted from the CCTV footage has shown promise in behaviour recognition, especially when

implemented on edge AI processors, demonstrating the potential for real-time action recognition in surveillance systems [27].

Another widely used mechanisms for human action recognition in CCTV surveillance systems are the two-stream networks. They are composed of two processing blocks: the first one is for analyzing the stream of video frames (2D images) while the second one is for analyzing the optical flow [17,20,25] calculated for each video frame. The video action is recognized by combining the texture and motion analysis results [18,33,37]. Recurrent Neural Networks (RNN) are used for temporal modeling of video sequences and determining the dependencies between consecutive video frames [15,28], while Transformer-Based Networks (TBNs) were developed to solve the problem of sequence transduction (any task that transforms an input sequence to an output sequence) [8,16].

The range of events to be detected mainly covers the violent behaviour. In [14] the crowd analysis framework has been discussed, including the large volume of data processing. Here Histogram of Oriented Gradients was used to improve performance of the detection close to the Real-Time conditions. In [4] the anomaly detection scheme is applied for the combination of the ANN and Gaussian distribution. Similarly, in [3] the anomalous behaviour is detected through the YOLO v2 network. In the presented cases the binary classification scheme is employed to distinguish the *nominal* behaviour of monitored persons from anything else. In most cases, the image or videos sequence analysis is based on the complete scene processing in search for anomalies. Also, skeletal structures are used to extract pose estimation, which is characteristic for some events.

In most presented solutions the number of events to detect is strictly limited to single type events. Also, the presented works omit the problem of processing large number of streams. Assuming that the presented architectures are effective enough to be used in practice, this paper shifts towards the ensemble of models to cooperate during the detection of multiple events at the same time. Also, multiple types of media are assumed (visible range video streams, infrared streams, audio analysis).

### 3. Problem statement

The task of the on-line monitoring of large number of inmates located in the penitentiary institution (or similar location, where, potentially dangerous, individuals are being held captive) creates multiple problems, not critical for the typical visual event-detection framework (like the ones proposed for stadiums). These are issued by the prison facility officers, being the main beneficiaries of the implemented system. They see it as the useful tool supporting officers responsible for monitoring the specific location, by drawing their attention to the particular camera image in the control room. The most important requirements include the following.

- The predefined set of 18 event types for detection, which are difficult or impossible to observe in the outside world. These specifically include fights between inmates, arson,

assault on the officer, riots, escape or suicide attempts, death of the inmate, illegal verbal or non-verbal communication between inmates, or with the outside world, or passing the forbidden object (such as a weapon) into the cell.

- The ability to operate with the predefined time-delays (near real-time mode), allowing for the fast reaction to the dangerous events. Based on the discussion with the officers and criminologists the threshold was set to 5 s.
- The ability to process large number of streams from up to 150 cameras, which is the significant challenge regarding the assignment of computing resources.
- Architecture of the system must apply to the technical infrastructure of the location, regarding the computer network topology, types of wiring and separation from the Internet (which eliminates the usage of the cloud services).
- Specificity of the location making usage of the distributed intelligence (with the video sequences processing next to the cameras) impossible to use (as the inmates often attack and damage the sensors).

These requirements make the construction of the system a difficult task, from both the research and technical perspective (for example, verification of the hardware capable of processing the mentioned number of streams).

The focus of the developed solution was at the detection of undesirable behaviour based on the extraction of features from data streams provided by multiple sensors in the on-line mode (Fig. 2). Each data source is a separate stream of information analyzed at the software level that models the AI-based classifier. The typical camera is capable of working in the visible light and infrared, making possible detection of events during the night (but requires separate training data for AI models). Also, some of them are equipped with microphones, which facilitates events related with sound. In specific locations two cameras are installed. Especially inside the cells, they are located opposite to each other to maximize the chance of detecting the event and eliminate blind spots. This leads to the multimodal analysis of the single scene, as presented in Fig. 1.

For such a task DNN are usually applied to obtain critical features from the image and classify them simultaneously, combined with a specific algorithm for describing the observed scene.

The effectiveness of the AI-based module depends on the number and diversity of available learning patterns (multiple-instance learning). In particular, DNN require significant amounts of versatile information (patterns) to correctly operate on the available data. The system analyzes each event as a set of sequences of numerical values representing one of the assumed anomalous behaviours. In the presented case the input video streams are processed by the classifier, which makes a decision about raising the alarm. Each sequence is a set of image frames provided by the camera with the constant speed, so the architecture of the particular network must be able to operate on the time series patterns. To make the DNN useful, it must be trained specifically to the solved tasks, which requires the individual data set preparation. All video sequences must then

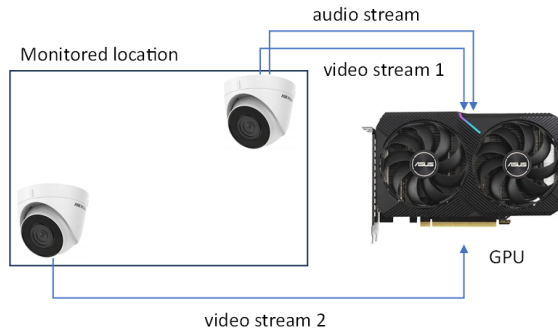


Fig. 1. Typical single scene analysis module, including video and audio streams (where applicable).

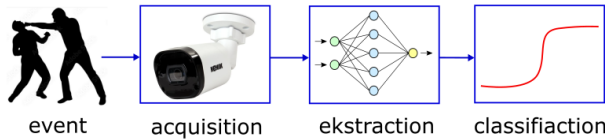


Fig. 2. Event detection by an artificial intelligence algorithm based on images obtained from a video camera.

be divided into those showing the unwanted events and the ones representing normal behaviour. Labeling of the sequences is therefore the crucial step in the proper development of the classification software. Processing schemes for traditional image processing based on feature analysis and processing using DL are presented in Fig. 3.

The presented problem may be treated as the binary or multi-category detection task. The latter is the more complex and from the practical point of would not be relevant,

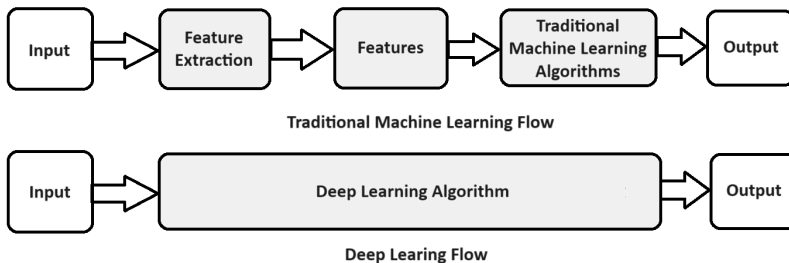


Fig. 3. The difference between the traditional machine learning-based input classification model and the deep learning version.

therefore the former was selected. In this case all data are labeled with two categories: positive (unwanted event) or negative (nominal situation). The actual nature or cause of the event are not relevant here, as the main task would be to focus the officers' attention to the particular location (while the subsequent code of conduct will depend on the nature of the actual event). This way the classifier would be only responsible for detecting any of the predefined events with the maximum possible accuracy. Consequences of the incorrect classification are represented by the specificity and sensitivity. In this project these have the special meaning: the false positive ratio will determine the number of false alarms, which should be suppressed (to avoid tiring the officers); the false negative ratio should be brought down to zero, as missing the suicide attempt or the aggressive behaviour may lead to the death of the inmate.

The special scope of the project is also driven by the strictly limited training material that available to train the DNN. There are video databases suitable for training models analyzing typical human behaviours, such as practicing sports, work activities, etc. The most popular ones are HMDB51 [24], UCF101 [34], YouTube8M [2], and Kinetics [23]. The number of databases representing violent actions (such as fights) is much smaller. The largest database is RWF-2000 [12], which contains 2,000 video sequences recorded by video surveillance cameras. Other databases, such as MoviesFight and HockeyFight [7], AIRTLab [9] contain fewer recordings. So far, no databases containing recordings regarding specific events related to penitentiary institutions are publicly available.

This problem would have to be solved by employing public databases to train the implemented networks with the additional support from application-specific dataset created in parallel with other activities in the project. This specifically refers to the events that are not present in the public sets.

#### **4. Proposed solution**

Approaching all events to be detected required the analysis of the specific scenarios of the sequences captured by the sensors. After the discussion with the experts in the psychology and criminology it became clear that it is not possible to use the single event detector for all of them. They are represented by the large variety of behavioural patterns expressed by the human beings. Therefore the events were divided into group[s] and for each the separate approach was designed. The latter was also dependent on the type of the detected event (video vs. audio stream).

This section contains the description of the technological solution of the presented problem. Because it refers to the specific implementation of the DNN, details of this unique implementation are given. First, the general architecture of the system is presented. Next, the particular algorithms applied for the event detection are described. Finally, preparation of the data set is outlined. These steps make the data processing framework complete and ready for testing.

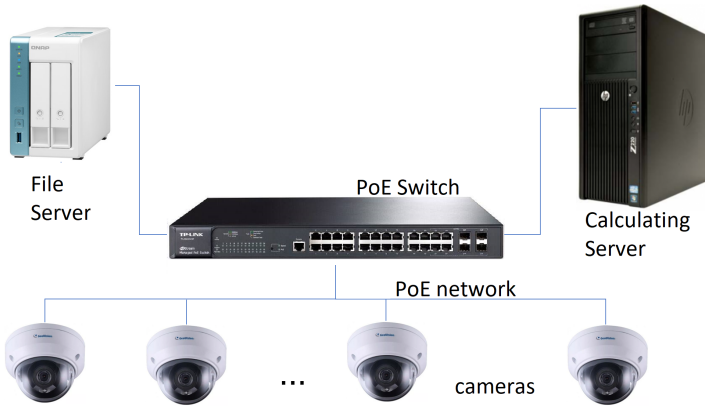


Fig. 4. Diagram of connecting IP cameras to a PoE switch and a calculating graphics server.

#### 4.1. System architecture

The proposed system architecture is presented in Fig. 4. The most important element of the hardware solution is the central processing server. The DNN architectures were implemented there, using NVidia CUDA-compliant accelerators [29]. Its main characteristics covers the number of cores, determining capabilities of processing multiple video streams (as stated in the documentation). The open question was if this parameter holds after adding to the card the DNN functionality. Each camera generates a separate video pipeline, which is then processed by the DNN (one network per pipeline). The computing server operates based on the signal from cameras connected to the system via the POE switches. Cameras that are connected to the system should be properly configured, as each generates two streams (primary and secondary). The former is directed to the recorder so it can archive video materials in full resolution and at full transmission speed (bitrate – Kb/s). The secondary stream is provided to the computing server. Due to the need to save memory, the additional stream should have its bit rate reduced by half, e.g. if the camera transmits the image at a default rate of 2048 Kb/s, then in the additional stream, it should be modified to 1024 Kb/s.

Due to the requirements of the project, the distributed system has the graphical user interface in the form of the web application, written in JavaScript and allowing the for configuration of the cameras and the detection thresholds. Also, this interface is intended to raise alarms in case of event detection. The web server is the separate node, communicating with the processing server through the asynchronous WebSocket interface. The detection module was written in Python and prepare to work on the selected graphic cards.

## 4.2. Selected DNN architectures

The networks used for the presented tasks have a specific, top-down structure, which makes them useful for event recognition under certain limitations (including, for example, the number of different events and, therefore, sequences that should be remembered). If the network capacity is exceeded, it will not be able to learn subsequent sequences, which may require the use of a larger and more complicated architecture characterized by greater computational requirements. The important assumption was that the functionality of the event detection module relies on the location of cameras delivering the data. For instance, if the device is located outside the building, it does not have to detect suicide or arson, aiming mainly at the escape attempt. Similarly, the most important events detected inside the cell are violent behaviour or suicide attempts. This allows for connecting the particular DNN model for the selected streams, suppressing the amount of computations.

All the network architectures have been implemented in Python language, which is currently the reasonable choice for data science applications. Due to the usage of the DL, the proper library (with the proper network models) had to be selected. In this case, TensorFlow and PyTorch [1, 30], which use hardware acceleration of calculations in a graphics processor with the CUDA architecture [29], were selected to implement the neural networks. The PyTorch and TensorFlow libraries include classes and data structures enabling implementation of all stages of DL, in particular, defining the architecture, reading training data, optimizing the network structure, and evaluating it after the training is complete.

### 4.2.1. Convolutional Neural Networks - CNN

The 3D CNN are widely used for violence detection in surveillance videos [32]. They process the batch of frames, defined by the height, width (single frame dimension), and depth (color depth). In the project the SELayer-C3D (Squeeze-and-Excitation) [22] mechanism was employed. It is an extension of the C3D model [35], i.e., a deep three-dimensional CNN with a homogeneous architecture based on convolution through a  $3 \times 3 \times 3$  kernel function. Here it is used to provide weights to each frame of the video. The SELayer has been used in relation to the attention mechanism. It extracts the importance of different parts of the analyzed image and combines weights with the original image. Thus the attention mechanism grasps the importance of each image frame and weights it. This way, only the most significant frames of the video are processed.

The network consists of 8 convolutional layers (Conv1a-Conv5b), 5 pooling layers (Pool1-Pool5), 2 fully connected layers (fc6 and fc7), and one softmax layer, which transforms outputs into a probability distribution over the input classes (Fig. 5).

The pooling mechanism takes an average of each frame. It is higher for frames considered important by the algorithm and lower for not so useful ones.

The model was trained to detect the following unwanted, anomalous events:



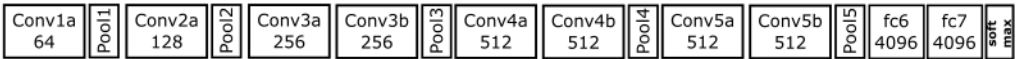


Fig. 5. Architecture of the C3D network.

- events involving violent behaviour of inmates (fight, attack on an officer, rebellion, rape, abuse of power by an officer);
- suicide and suicide attempt;
- death of an inmate;
- unauthorized verbal contact;
- escape attempts (finding an inmate in an area where no one should be);
- attempts to transfer an unauthorized item.

The network is supposed to process the batch of the colour images (RGB scale), extracted from the 5-second sequence, further called a *scene* (every fifth frame is extracted for the single batch). For such a file the softmax layer provides the real number representing the confidence about the detection of the specific event. Now, the crucial element is setting the threshold, above which the processed event is considered anomalous. This is actually the main challenge during application of this network to the described project. The problem is the same event will be seen differently in variety of environmental conditions (day or night, sunny or rainy aura). Also, positions of the cameras vary (the inmates’ cell, the corridor, or in the outside perimeter). These conditions enforce the adaptive threshold adjustment, which at this point is a separate task, requiring first finding the optimal values for particular conditions. Due to the potential instability of the system after changing the environment (for example implementing it in another prison), it is recommended to retrain the system on the location-specific data (recorded on-site).

The best predictions were achieved for events involving violent behaviour of inmates and suicide attempts. A sample of detecting violent behaviour is shown in Fig. 6 where a low probability value of detecting abnormal behaviour is shown on the left side of the image (prediction value of 0.39) and a high probability value of detecting abnormal behaviour is shown on the right (prediction value of 0.86).

Experiments performed with the C3D model in laboratory conditions showed that some of the anomalies were not properly detected and algorithms needed to be improved. Unauthorized physical contact, escape attempts, and transfer of an unauthorized item are detected based on C3D-trained network predictions combined with contour detection and its intersection with specific lines and bounding boxes (regions of interest, ROIs) for each group of cameras placed at specific areas in buildings. These shapes are different for the interior of the building and outside the building and are fitted to each camera’s video resolution [10]. In Fig. 7, there is an example of ROI for supporting the detection of escape attempts.



Fig. 6. Example of abnormal behaviour's detection and corresponding prediction value. *Nominalnie* – nominal; *ramka* – frame, *Wysoki* – high.



Fig. 7. Example of ROI line for detecting escape attempts outdoors and alarm notification.

#### 4.2.2. YOLO architecture

A separate problem is detection of an inmate's death. In this case, the pose estimation and specific relationship between the head, hips, and feet, which are key points in the human body, were the most important parameters to analyze and support the main

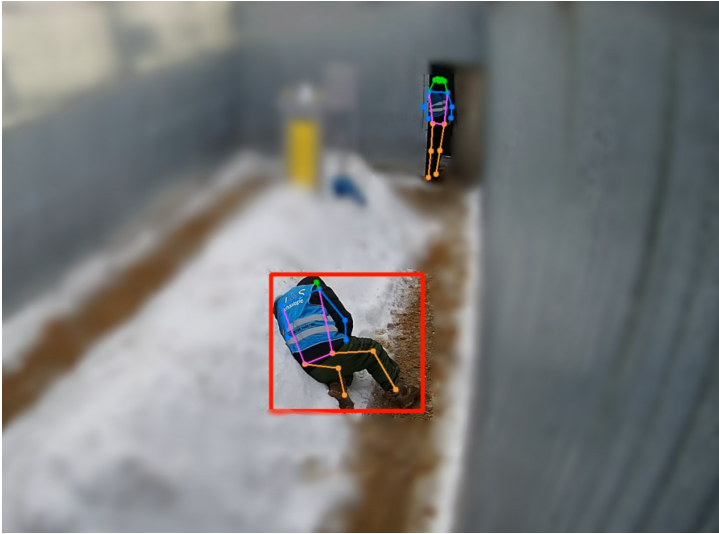


Fig. 8. Example of YOLOv7 pose estimation and prediction of inmate's death.

C3D network predictions. Pose estimation and key point detection were implemented using YOLOv7 (Your Only Look Once, version 7) Fully Connected Neural Network model [36] and in Fig. 8 there is an example of two detected persons, their body key points and alarm notification (red box around a person falling on the ground). The YOLO framework consists of three components:

- Backbone, which extracts essential features (characteristic points) of an image and feeds them to the Head of the network through Neck;
- Neck, which collects feature maps and creates so-called feature pyramids;
- Head, which consists of output layers that predict the locations and classes of objects around which bounding boxes should be drawn.

YOLOv7 is a multi-head framework (Fig. 9). The head responsible for final output is called the Lead Head, while the one used to assist training in the intermediate layers is the Auxiliary Head. The weights of the latter are updated with the help of an assistant loss. These auxiliary classifiers provide direct supervision on the hidden layers in addition to the overall network output. They attach a specific loss function to the intermediate layers, enabling the gradient to be directly propagated back to earlier layers in the network. This helps with gradient flow and facilitates training deep networks.

The final layer aggregation is done by the Extended Efficient Layer Aggregation Network (E-ELAN), which enables the YOLOv7 framework to improve training process. To increase the performance of a model without increasing the training cost, YOLOv7 utilizes so-called Planned Re-parameterized Convolution. Two types of re-parameterization

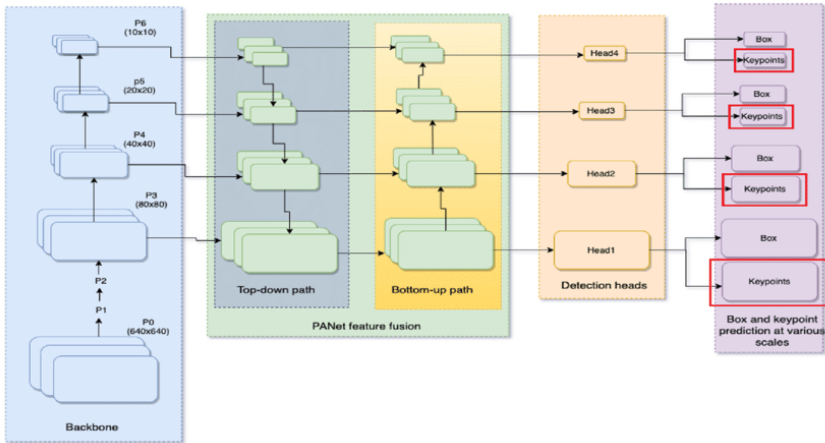


Fig. 9. Architecture of the YOLO network [26] (license: CC BY 4.0).

are used to finalize models: model level and module level ensemble. The first method uses different training data and identical settings to train multiple models. Then averaged weights are used to obtain the final model. The second model training is split into multiple modules. The outputs are combined to obtain the final model.

Joint detection and 2D multi-person detection in an image is conducted by the YOLO-Pose. It is a modified version of YOLOv5 model that learns to jointly detect bounding boxes for multiple persons and their corresponding 2D poses by associating all keypoints of a person (so-called anchors). They are matched with the ground truth box stores forming the 2D pose along with the bounding box location. Keypoints associated with an anchor are already grouped. In case of human pose estimation, each person has 17 associated keypoints, identified by a particular location on a 2D figure with a certain measure of confidence. The keypoint head predicts 51 elements, while the box head predicts six elements. The output of this pose estimation model is a set of points that represent the keypoints on an object in the image, usually along with the confidence scores for each point.

YOLO assigns confidence scores to each predicted bounding box. These scores represent the model's confidence in the accuracy of the prediction. High confidence scores indicate a high probability of the bounding box containing a valid object. If the keypoint is either visible or occluded, the ground truth confidence score is set to 1. Otherwise, if it is outside the field of view, confidence is set to zero. Keypoints outside of the field of view are disregarded.

### 4.2.3. Line crossing

This method is used to detect events related with objects crossing the line defined inside the field visible by the camera. The user must set the particular coordinates to define where the exactly the crossing should be detected. This approach is applicable to areas with no nominal movement at all, like in the perimeter around the prison block buildings, or on the roof with the view on the windows of cells. The events detected this way include the escape attempts and passing forbidden objects through the outside environment. This process is based on extracting detected moving objects' contours in consecutive video frames and then finding the intersections with ROIs (Region of Interest) defined as lines or specific bounding boxes customized to each camera. All processing is performed with OpenCV library functions [10].

### 4.2.4. Sound-based events detection

This task was relatively simple, as the number of events related with uttering sound was strictly limited to detecting the unwanted verbal contact between inmates or the verbal contact with the outside world. Also, the number of cameras equipped with the microphones is relatively small (up to 5 units). Because the task was only to identify the contact and not its details (like spoken/shouted words), the simple batch processing of the audio stream was used, consisting in computing energy of the 250 ms time frames. The alarm was risen for the energy above 18 dB.

## 4.3. Training data preparation and preprocessing

The fundamental importance for the effectiveness of the DL-based system is to provide it with the appropriate amount of data in the form of recorded scenarios of various situations containing undesirable behaviour. The network should obtain a large number of scenes in which anomalous behaviours appear in as many environments as possible (knowing that the network must learn how to ignore the background and focus on the scene). Metadata that may help in describing the particular scenario (the number of participants, recognized behaviour, and place of the event) are also added to each sample. Multiplication of such scenes (by creating different versions of scenarios) helps in reaching the generalization, essential while implementing the system in the Real-World environment.

First, the proper data sets had to be prepared. The process of acquiring information important to train the network was divided into three phases (Fig. 10). The first one used the preliminary data  $D_1$  from the publicly available sets. This did not allow for preparing the system to detect all events, but was enough to test the algorithms and implement them in the first version of the framework. In the second phase (also executed in the laboratory conditions) the original data set  $D_2$  was prepared to supplement and optimize already existing system. It contained scenarios of unique events that could not be extracted from the set  $D_1$  (for instance, suicide attempt). Also, the more general

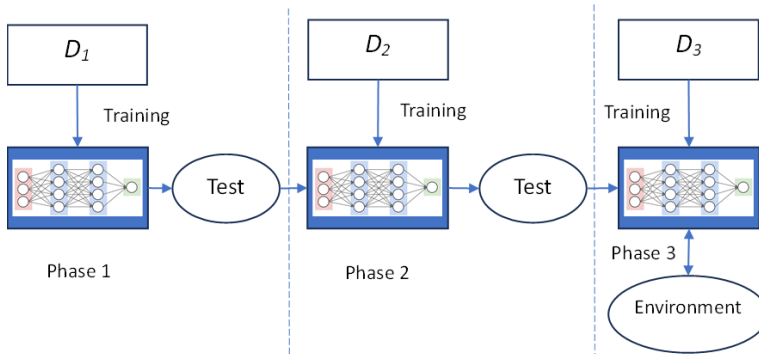


Fig. 10. Training framework for the constructed system.

events (like fights) were added to consider specific locations (corridors, narrow passages, small rooms) in hope of increasing the accuracy of their identification. The final phase was executed in the premises of the facility, where additional scenes have been recorded to form the set  $D_3$ . After each phase the testing was conducted to verify the overall accuracy of the system. From this step all incorrectly detected scenes were also added to the next version of the set. After stabilizing the system on-site, it was run in the on-line mode, with the intention of stabilizing its behaviour in the long term (especially after detecting flaws in the real-world conditions).

The process of preparing the algorithm for identifying individual patterns of undesirable behaviours begins with extracting them from available training scenes. An automatic frame extraction algorithm was developed to create training examples. Each data source is a separate stream of information analyzed at the software level, based on open-source libraries prepared in Python. Each recording was divided into 32 segments, constituting a separate set of patterns during the training process. The C3D features were extracted from each example, forming a training set.

Recordings of anomalous behaviour (intended to trigger an alarm) and neutral situations (not triggering an alarm) must be in MP4 format with  $\text{fps} = 25$  frames/second. Each recording of anomalous behaviour should include 1 second before the event and 1 second after the event (approximately 5-6 seconds in total). Recordings of neutral behaviours may be longer (up to 30 seconds). Recordings are recommended to be in a resolution of at least  $640 \times 480$  pixels (which facilitates scaling down the original image to the input of the network, which is a map of the size of  $224 \times 224$ ). The AI models utilized here were trained on the basis of approximately 2500 recordings of anomalous sequences intended to trigger an alarm and 4000 recordings of nominal sequences (not causing an alarm).

## 5. Results and discussion

This section covers the verification details of the systems efficiency in the binary classification task (though driven by different events). First, efficiency measures are presented. Then, they are used to verify the DNNs performance. Finally, the time efficiency analysis is performed on the system operating in the on-line and off-line mode.

### 5.1. Accuracy measure

The basic measure of the system's effectiveness is accuracy, which is measured as the percentage of correctly detected events. Since calculating such a coefficient for an online system is difficult to implement (among others because, most of the time, video streams provide data that does not contain any abnormal behaviour), the system's accuracy is determined on the basis of the sampling error  $e_s$ . It is calculated for individual events  $x_i$  separately (each has different details that constitute individual level of difficulty for the system) as a percentage of correctly detected events (i.e. having the response  $h(x_i)$  identical to the actual category  $c_i$ ) based on a selected set of video sequences  $T$  presented to the system.

$$\text{acc} = 1 - e_s = \frac{|x_i \in T : h(x_i) == c_i|}{|T|} \cdot 100\% \quad (1)$$

To check the accuracy in the laboratory conditions (off-line mode), the specific testing sets were created. They contain sequences of the same type as for the training set, but with different actual scenes. Due to the high time consumption of the testing process, each event was represented by 20 scenes of unwanted events and the same number of nominal sequences. This way the accuracy for detecting the particular type of event is between 0 and 100 percent (with the step of 2.5 percent). It is possible to determine accuracy of the detecting individual events and the average accuracy of the system (for all events).

### 5.2. Off-line accuracy evaluation

In laboratory conditions (locations of the Warsaw University of Technology), the accuracy was on average 90%. Relatively the easiest to detect were violent behaviours such as fights, assaults, rebellion, etc. A very high level (close to 100%) was achieved for events such as fights, escape, and non-verbal contact (passing an object through the windows of the prison pavilion). The transfer of prohibited items is relatively the most difficult to detect because cameras make have problems with capturing the particular behaviour of inmates.

The selected detection threshold values significantly influence the accuracy of the detection system. This is one of the critical parameters requiring individual tuning

Tab. 1. Confusion matrix for the violent behaviour of the inmates.

	$A_{\text{actual}}$	$N_{\text{actual}}$
$A_{\text{predicted}}$	19	0
$N_{\text{predicted}}$	1	20

Tab. 2. Confusion matrix for the suicide attempt.

	$A_{\text{actual}}$	$N_{\text{actual}}$
$A_{\text{predicted}}$	15	0
$N_{\text{predicted}}$	5	20

when installing the system in a new location. The detection threshold (provided by the DNN) is a coefficient with values in the range (0, 1) and should be carefully adjusted to environmental conditions. For values close to 1, false alarms are rarely detected, but the same applies to the detection of undesirable events. For values close to 0, all events will be detected, but with a large number of false alarms. In practice, the threshold should be set in the range of 0.8-0.98, depending on the specific location.

The accuracy described in this way does not include the system's sensitivity to false alarms. These can be deduced from the confusion matrices, such as in Tab. 1. Here  $A$  stands for the anomalous event, while  $N$  is the nominal event. The violent behaviour of the inmates is relatively easy to detect, as all abrupt changes in the scene were detectable in various locations. On the other hand, suicide attempts were more challenging, and detection is successful only if the actor performs the scenario to the point (Tab. 2). In almost all situations the main problem was missing the event, not the false alarm.

Based on the available results it is assumed the number of false alarms is reciprocal to the amount of training data. Therefore it is expected that the system's robustness will increase with the duration of its operation in the specific facility. To obtain the required immunity to false alarms, additional video sequences must be provided for re-training, especially with scenes that have incorrectly classified as alerts (e.g., meals being delivered or an inmate entering the toilet, as a result of which the light turns on and the camera switches from night to day mode).

On the other hand, the ability to detect all positive cases depends on the nature of such an event. In some cases (for instance, suicide) the event is very difficult to detect, especially if it is not commenced exactly as it was assumed during the scenario creation. Therefore the system is able to detect only the event presented in the specific form. To extend the capabilities of the system, probably additional sensors would be required (such as thermal imaging).

The approach taken to reach the maturity in the presented technology depends on



the constant availability of the new training data (provided that deep networks were selected correctly and are able to extract all required features). Though initially widely available sets are enough to verify the desired detection accuracy, in the repeated cycles the main focus is on the suppression of the false alarms with the zero omissions of the actual events. Unfortunately, only the application-oriented data may help to improve the accuracy. In the presented project the data must come from the penitentiary institutions, as no other source allows for extracting any relevant information regarding the events of interest.

### 5.3. On-line accuracy evaluation

The on-line accuracy evaluation is significantly different from estimating the sample error in the laboratory conditions. These do not reflect the actual operation of the system in the near Real-Time mode. In the practical application, the system works uninterrupted with multiple data streams incoming all the time. Therefore the evaluation of its efficiency should follow the formula (2), with  $T_1$  being the number of actual events that should be detected (from all streams) and  $T_0$  – the number of normal situations. It reflects the continuous flow of data requiring the assistance from the central system. The processing module makes decision in discrete steps, after collecting the batch of frames from the particular stream. During each time interval the constant number of batches are generated, so based on the duration of the system operation, it is possible to estimate the number of analyzed events (most of them being nominal).

$$\text{acc} = \frac{|\{x \in |T_0| \cup |T_1| : h(x) == c(x)\}|}{|T_0| + |T_1|} \cdot 100\% \quad (2)$$

As a result of the experiments and the validation procedure, a detection accuracy of 84.7% was obtained. The experiments were performed on the premises of the actual penitentiary institution “ZK Chełm”, also with the participation of officers (staff of this facility). The tests consisted of extracting selected scenes in specific locations: a cell, a corridor, a walking area and observing whether the system reacted to the event, and the user interface on the computer in the monitoring center informed appropriately about the event. This accuracy was achieved after the system’s initial configuration, which enabled the determination of optimal thresholds that ensure the detection of typical events without generating false alarms. The “ZK Chełm” employees confirmed the operation of the tool and its effectiveness. Results presented in Tab. 3 were obtained after running the system for 30 minutes, during which 6 cameras remained operational and the processing module was aiming at the violent behaviour detection. This allowed for producing 1872 events evaluated by the system, as presented in the log files. After the event was detected in the particular stream, it was switched off to avoid detecting the same event twice,

Tab. 3. Confusion matrix for the violent behaviour of the inmates in the on-site experiments.

	$A_{\text{actual}}$	$N_{\text{actual}}$
$A_{\text{predicted}}$	17	4
$N_{\text{predicted}}$	3	1848

The main difference between the laboratory and on-site testing is the increasing number of false alarms, generated by the events not considered during the DNN training. This is because in the facility many activities are performed by the inmates the design team was not aware of. Some specific locations caused additional problems. For instance, the cameras installed outside could record the birds, which would cause false alarms of crossing the predefined line. Attempts to suppress these were made during the maintenance stage of the system's life cycle.

#### 5.4. Time efficiency evaluation

This section presents three time aspects of the system's time efficiency. The first one refers to the delays related with the video streams propagation inside the network. The second is the DNN training duration (performed in the offline mode), while the last one is the reasoning duration for the single video stream.

The speed of the system's reaction to the occurrence of an event consists of two components. The first one is the speed of streaming transmission inside the computer network (with the introduced delay is of order of 50-100 ms – in some cases, it may be slower and therefore noticeable for the human operator). the delay associated with calculating the prediction of anomalous situation. In the latter case, the delays are an average of 4 s (calculated based on 180 trials), which is the time needed to acquire the required number of frames, preprocess them, and send them to the detection algorithms. Such a delay may be extended depending on the current load inside the computer network and the number of simultaneously supported camera streams. This is optimized by spreading the prediction callbacks over several cameras at once using proper fork mechanisms for managing multiple processes run in a Linux system.

Duration of the learning process is of secondary importance for the practical use of the system. It determines the operations carried out besides the nominal operation of the system (probably performed on the separate machine). This time depends on the size of the training data set and the speed of available hardware (CPU and graphics cards clocking). Because checking the latter would require using multiple different models, the card configuration was assumed constant (two nVidia Telsa M10 cards with 24MiB of VRAM each) and only the influence of the size of the data set on the training duration was checked. Results of time measurements are in Tab. 4.

Tab. 4. Algorithms learning speed depending on the size of the data set.

Learning set [GB]	Validating set [GB]	Training time for 50 epochs [h]
1	0.08	6
2	0.16	15

The additional delay is related with acquiring images from IP cameras for the prediction algorithms. Experiments carried out in the laboratory conditions (with small LAN) have shown that the times associated with transmitting signals from sensors is up to 1 second. During the field tests (with much larger network inside the facility) this delay increased and is up to 3 s.

## 6. Conclusions

The system presented in the paper is capable of detecting the selected set of unwanted events recorded by the IP cameras inside the penitentiary institution. It has the distributed architecture with the separate computing server and the www server presenting the user interface for the human operator. The processing part is based on the nVidia CUDA platforms, where the range of DNN implemented.

Performance of the system verified both in the laboratory and on-site conditions shows its usefulness to the defined task. Accuracy of the system tested in the facility is acceptable from both the research and practical perspectives. The main problem with the implementation of the system in the actual conditions. The significant number of events is classified incorrectly as the false alarms. This is caused by the events present on location, but not considered during the training. They should be suppressed in the future during stabilizing the detection modules.

The evolution of the project is focused on the increase of the accuracy with the suppression of the false alarms, which requires the constant iterative training of the system on the newly delivered data. This, however, can be done only with the video sequences extracted on-site, i.e., from the actual events (which, for instance, in case of the suicide attempts, are rare and difficult to collect). Currently works on the data set extensions in cooperation with penitentiary institutions are carried out.

The possible extensions of the project would include the further development of the classification module. As there are multiple DNNs developed every year, the potential accuracy and sensitivity may be increased by the classifier with better discrimination abilities. As the system has the open architecture, it can be also applied to different locations.

## Acknowledgment

This work was supported by the Polish National Centre for Research and Development, grant no. DOB-BIO10/16/02/2019, “Intelligent decision support system based on the algorithmic image analysis in the operations of the justice services”, project value: 6 311 438 PLN.

## References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. <https://www.tensorflow.org/>, Software available from tensorflow.org.
- [2] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, et al. YouTube-8M: A large-scale video classification benchmark. *arXiv*, 2016. ArXiv.1609.08675. doi:10.48550/arXiv.1609.08675.
- [3] A. Al Ibrahim, G. Abosamra, and M. Dahab. Deep convolutional framework for abnormal behaviour detection in a smart surveillance system. *Engineering Applications of Artificial Intelligence*, 67:226–234, 2018. doi:10.1016/j.engappai.2017.10.001.
- [4] A. Al Ibrahim, G. Abosamra, and M. Dahab. Real-time anomalous behavior detection of students in examination rooms using neural networks and Gaussian distribution. *International Journal of Scientific and Engineering Research*, 9(10):1716–1724, 2018. doi:10.14299/ijser.2018.10.15.
- [5] A. S. Alturki, A. H. Ibrahim, and F. H. Shaik. Real time action recognition in surveillance video using machine learning. *International Journal of Engineering Research and Technology*, 13(8):1874–1879, 2020. doi:10.37624/IJERT/13.8.2020.1874-1879.
- [6] C. Amrutha, C. Jyotsna, and J. Amudha. Deep learning approach for suspicious activity detection from surveillance video. In: *Proc. 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 335–339, 2020. doi:10.1109/ICIMIA48430.2020.9074920.
- [7] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar. Violence detection in video using computer vision techniques. In: P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano, and W. Kropatsch, eds., *Proc. Conf. Computer Analysis of Images and Patterns (CAIP)*, vol. 6855 of *Lecture Notes in Computer Science*, pp. 332–339. Springer Berlin Heidelberg, 2011.
- [8] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In: M. Meila and T. Zhang, eds., *Proc. 38th International Conference on Machine Learning*, vol. 139 of *Proceedings of Machine Learning Research*, pp. 813–824. PMLR, 2021. <https://proceedings.mlr.press/v139/bertasius21a.html>.
- [9] M. Bianculli, N. Falcionelli, P. Sernani, S. Tomassini, P. Contardo, et al. A dataset for automatic violence detection in videos. *Data in Brief*, 33:106587, 2020. doi:10.1016/j.dib.2020.106587.
- [10] G. Bradski. The OpenCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.
- [11] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In: *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017. doi:10.1109/CVPR.2017.502.
- [12] M. Cheng, K. Cai, and M. Li. RWF-2000: An open large scale video database for violence detection. In: *Proc. 2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4183–4190, 2021. doi:10.1109/ICPR48806.2021.9412502.

- [13] P. Dasari, L. Zhang, Y. Yu, H. Huang, and R. Gao. Human action recognition using hybrid deep evolving neural networks. In: *Proc. 2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2022. doi:10.1109/IJCNN55064.2022.9892025.
- [14] S. R. Dinesh Jackson, E. Fenil, M. Gunasekaran, G. Vivekananda, T. Thanjaivadivel, et al. Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM. *Computer Networks*, 151:191–200, 2019. doi:10.1016/j.comnet.2019.01.028.
- [15] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, et al. Long-term recurrent convolutional networks for visual recognition and description. In: *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634, 2015. doi:10.1109/CVPR.2015.7298878.
- [16] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, et al. Multiscale vision transformers. In: *Proc. 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6804–6815, 2021. doi:10.1109/ICCV48922.2021.00675.
- [17] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In: J. Bigun and T. Gustavsson, eds., *Image Analysis. Proc. 13th Scandinavian Conference (SCIA) 2003*, vol. 2749 of *Lecture Notes in Computer Science*, pp. 363–370. Springer Berlin Heidelberg, 2003. doi:10.1007/3-540-45103-X\_50.
- [18] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In: *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941, 2016. doi:10.1109/CVPR.2016.213.
- [19] S. Ganta, D. S. Desu, A. Golla, and M. A. Kumar. Human action recognition using computer vision and deep learning techniques. In: *Proc. 2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, pp. 1–5, 2023. doi:10.1109/ACCTHPA57160.2023.10083351.
- [20] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981. doi:10.1016/0004-3702(81)90024-2.
- [21] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. doi:10.1109/TPAMI.2012.59.
- [22] B. Jiang, F. Xu, W. Tu, and C. Yang. Channel-wise attention in 3D convolutional networks for violence detection. In: *Proc. 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*, pp. 59–64, 2019. doi:10.1109/ICEA.2019.8858306.
- [23] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, et al. The kinetics human action video dataset. *arXiv*, 2017. ArXiv.1705.06950. doi:10.48550/arXiv.1705.06950.
- [24] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In: *Proc. 2011 International Conference on Computer Vision (ICCV)*, pp. 2556–2563, 2011. doi:10.1109/ICCV.2011.6126543.
- [25] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In: *Proc. 7th Int. Joint Conf. Artificial Intelligence (IJCAI) 1981*, pp. 674–679, 24–28 Aug 1981. <https://hal.science/hal-03697340>.
- [26] D. Maji, S. Nagori, M. Mathew, and D. Poddar. YOLO-Pose: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. *arXiv*, 2022. ArXiv.2204.06806. doi:10.48550/arXiv.2204.06806.
- [27] A. Nakajima, Y. Hoshino, K. Motegi, and Y. Shiraishi. Human action recognition based on self-organizing map in surveillance cameras. In: *Proc. 2020 59th Annual Conference*

- of the Society of Instrument and Control Engineers of Japan (SICE), pp. 1610–1615, 2020. doi:10.23919/SICE48898.2020.9240260.
- [28] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, et al. Beyond short snippets: Deep networks for video classification. In: *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702, 2015. doi:10.1109/CVPR.2015.7299101.
- [29] NVIDIA, P. Vingelmann, and F. H. P. Fitzek. CUDA, release: 10.2.89, 2020. <https://developer.nvidia.com/cuda-toolkit>.
- [30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, et al. PyTorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32 – Proc. 33rd Conf. Neural Information Processing Systems (NeurIPS 2019)*, vol. 11, pp. 8024–8035. Vancouver, Canada, 8–14 Dec 2019. Accessible in arXiv. doi:10.48550/arXiv.1912.01703.
- [31] N. S. Rao, G. Shanmugapriya, S. Vinod, R. S, S. P. Mallick, et al. Detecting human behavior from a silhouette using convolutional neural networks. In: *Proc. 2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, pp. 943–948, 2023. doi:10.1109/ICEARS56392.2023.10085686.
- [32] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo, and A. F. Dragoni. Deep learning for automatic violence detection: Tests on the AIRTLab dataset. *IEEE Access*, 9:160580–160595, 2021. doi:10.1109/ACCESS.2021.3131315.
- [33] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In: *Proc. 27th International Conference on Neural Information Processing Systems*, vol. 27 of *NIPS Proceedings*, p. 568–576, 2014. <https://ora.ox.ac.uk/objects/uuid:1dd0bcd0-39ca-48a1-9c20-5341d6c49251>.
- [34] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, 2012. ArXiv.1212.0402. doi:10.48550/arXiv.1212.0402.
- [35] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In: *Proc. 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, 2015. doi:10.1109/ICCV.2015.510.
- [36] 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. In: *Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464–7475, 2023. doi:10.1109/CVPR52729.2023.00721.
- [37] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream ConvNets. *arXiv*, 2015. ArXiv.1507.02159. doi:10.48550/arXiv.1507.02159.
- [38] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, et al. A comprehensive study of deep video action recognition. *arXiv*, 2020. ArXiv.2012.06567. doi:10.48550/arXiv.2012.06567.