Negative Filtering of CCTV Content -Forensic Video Analysis Framework

Franck Jeveme Panta, André Péninou, Florence Sèdes

Negative Filtering of CCTV Content -Forensic Video Analysis Framework

Franck Jeveme Panta
IRIT
Université Paul Sabatier
Toulouse, France
franck.panta@irit.fr

André Péninou
IRIT
Université Paul Sabatier
Toulouse, France
andre.peninou@irit.fr

Florence Sèdes
IRIT
Université Paul Sabatier
Toulouse, France
florence.sedes@irit.fr

ABSTRACT

This paper presents our work on forensic video analysis that aimed to assist videosurveillance operators by reducing the volume of video to analyze during the search for post-evidence in videos. This work is conducted in collaboration with the French National Police and is based on requirements defined in a project related to videos analysis in the context of investigations. Due to the constant increasing volume of video generated by CCTV cameras, one of the investigators' goals is to reduce video analysis time. For this purpose, we propose a negative filtering approach based on quality and usability/utility metadata, enabling to eliminate video sequences that do not satisfy requirements for their analysis through automatic processing. Our approach involves a data model which is able to integrate different levels of video metadata, and an associated query mechanism. Experiments performed using the developed framework demonstrate the utility of our approach in a real-world case. Results show that our approach helps CCTV operators to significantly reduce video analysis times.

KEYWORDS

Metadata; CCTV Systems; forensic video analysis; video evidence

1 INTRODUCTION

Videosurveillance has become a very important security measure in crime prevention [10]. Recent studies [8], [19] illustrate the relevance of videosurveillance that helps store owners, business owners and police to deter and respond to criminal incidents detected by this technology. One of the main functions of videosurveillance is to store images of criminal incidents and non-social behaviour to facilitate post-event analysis during investigations. The current method used by investigators (in France), consists of applying automatic processing (face detection, vehicle detection, license plate recognition...) to all video sequences collected upstream and previously indexed (conversion to the right format, camera geolocation, management of timestamps...). Automatic processing are applied to all the videos or only to a geographical or temporal subset. After these processing and data extraction, investigators conduct researches (vehicles, plates, people, faces...), view the sequences of interest and launch new processing. Recent investigations (robbery, terrorism, incivility, homicide) have required the analysis of several tens of terabytes of video data, corresponding to several tens of thousands of video hours. The use of automatic processing in these investigations enabled a time saving of a factor of 3 on average (which still too low in an operational context) compared to manual analysis. However, many videos were unusable for automatic processing (brightness due to night, optical conditions, etc.). The processing time could have been reduced if unsuitable video sequences for automatic processing could have been discarded (what we call "negative filtering"). In this context, negative filtering is therefore defined as a set of processes enabling elimination of video sequences that are not compatible with a given automatic processing, among a mass of available videos.

In this paper we focus on reducing the volume of video to be exploited (implicitly the reduction of operating time) for the search of video evidence during an investigation through the negative filtering. The goal is to improve the automatic processing speed by discarding video sequences that do not satisfy automatic processing requirements. So, it means determining whether a video sequence is able to give results with respect to a given automatic processing, in which case the automatic processing will be applied to this sequence. In the contrary case, it is useless to carry out the automatic processing, if it is already sure that no result will be obtained. Three automatic processing have been selected for this study: (i) face detection, (ii) vehicle detection, and (iii) plate detection and recognition. These automatic processing are the most commonly filters used by Scientific Technical Police in in criminal investigations which are generally done in two modes: urgent investigation and deep investigation. The first mode requires a very fast analysis of the videos. It occurs when one or more dangerous and wanted individuals are running away. The need for quick results leads to focus only on very good quality sources in the geographical and temporal area where the chances of finding the target are high. It is better to discard videos that results are approximate compared to very good quality videos that could bring significant elements to the investigation. The second mode enables an in-depth and less urgent analysis. The aim here is to get the most accurate and comprehensive possible results. The research can involve video sequences with lower quality, but whose processing is still able to provide results.

We propose a negative filtering approach that takes into account the three automatic processing methods considered and that provides results for the two investigation modes described. Negative filtering is based on features that evaluate the quality and usability/utility of videos. It is then necessary

to define for each automatic processing the criteria of quality and usability/utility of video that will allow to filter the video sequences according to the two modes of investigation (urgent and deep). More concretely, it consists of developing metrics that express the quality and usability/utility of videos based on a combination of technical metadata, metadata describing the movement and field of view of the camera (e.g. camera speed, orientation in relation to objects that could obstruct the field of view) and metadata from content analysis algorithms (e.g. describing the movement or number of people in the scene). These metadata can be represented according to different levels of semantics and granularity. Collaboration between the different levels of metadata is a challenge for this study. Therefore, we propose a generic data model to integrate all these metadata and a and supporting mechanisms for filtering large collections of video related to the research of a posteriori evidence.

To sum up, the main contributions of this paper are summarized as follows:

- We propose a generic and scalable data model enabling integration and interoperability of metadata (quality and usability/utility of the videos) required for negative filtering implementation.
- We provide a robust querying mechanism to filter out video segments that processing will be useless for the search of digital evidence.
- We conduct experimental evaluations demonstrating the usability of our approach in a real-world scenario of searching for a posteriori evidence in videos.

The rest of the paper is organized as follows. Section 2 reviews the related works; Section 3 presents a description of our approach; Section 4 shows an real-case experimentation that we performed to evaluate our methods for relevant video retrieval and content filtering. Finally, in section 5, we discuss and conclude with suggestions for future research.

2 RELATED WORK

Interest for a posteriori video analysis motivated many researchers to propose solutions for the problem of searching for "digital evidence" in video surveillance collections. Most of the proposed work focuses on the development of tools for video content analysis in order to detect and/or track objects [7], [4], to recognize actions [14], events [9] or scenes [11], [18], and to analyze human crowd behavior [20], etc. During the last few years, many solutions have been proposed and many collaborative projects have been set up both at national and European level.

CARETAKER [1] is a European project which was part of the context of the surveillance of metro stations via the exploitation of video and audio streams. The project enabled the development of techniques for automatically extracting relevant semantic metadata from video content. However, no filtering was performed before extracting the knowledge from the video streams.

VANAHEIM project (Video/Audio Networked surveillance system enhAncement through Human-cEntered adaptIve Monitoring) enabled the development of a technique for real-time automatic filtering of videos using algorithms to detect abnormal activity. But the implementation of the learning algorithms used in the filtering process seems complex for large volumes of data.

In [6], authors present a video event analysis and recovery system using geospatial computer techniques. Based on target tracking and analysis of video streams from distributed camera networks, the system generates video tracking metadata for each video, represents them on a map, and merges them into a uniform geospatial coordinate. The combined metadata are stored in a spatial database where target trajectories are represented in geometry and geographic data type. The spatial database provides the system with a stable, fast, and easy-to-manage platform, which is essential for managing large amounts of video data. The spatial index provided by the database allows online querying by quickly removing unrelated data and work on data of interest. On the other hand, there is no filtering before the generation of video tracking metadata for each video, which can be very time consuming.

One of the recent works of our team [5] was on the modeling of spatio-temporal metadata associated with video content and the querying of these metadata using hybrid trajectories. This work is applicable in outdoor environments. An extension to indoor environments has been proposed in [17]. One of the perspectives of this work was to rely on relevant metadata in the context of videosurveillance to reduce space and consequently research time, and other measures that can be developed based on the metadata or on images features (e.g. image quality).

3 PROPOSED APPROACH

This section develops our negative filtering approach that helps investigators to facilitate the post-event research of evidence in videos by eliminating video sequences based on their non-eligibility for the selected automatic processing. The main goal is to speed up the process of forensic video analysis that we define as the offline analysis of video aimed at finding video or digital evidence during an investigation. The proposed approach consists of two main steps:

- *step 1* - **Metadata modeling**: Since filtering criteria are based on metadata describing the quality and usability/utility of videos, it is necessary to provide uniform modelling of this metadata to facilitate their usage.
- *step 2* - **Querying mechanism development**: the idea is to provide algorithms based on the proposed metadata model in order to implement filtering.

3.1 Metadata modeling for negative filtering

The value of digital information depends on how easily it can be located, searched and retrieved. Metadata describing the content of digital information are essential for these tasks, and without them, some digital information are considered as useless. Metadata modeled in this section describe the quality and usability/utility of videos. Image quality can refer either to the degree/level of accuracy of images (viewed as a set of signals) during acquisition, processing, storage and restitution,

or to a set of visually significant attributes of the image. Usability/utility of video refers to a set of features that determine the suitability of a video for a given situation.

3.1.1 Metadata describing the video quality.

Image quality depends on the optics of the sensor, the electronics (amplification, quantification and sequencing), as well as the environment in which the image was captured and the lighting conditions. In order to evaluate video quality, it is necessary to define a set of quality metrics by using or developing spatial image quality descriptors to describe the visible and preponderant degradations in the visual rendering of an image. Defining quality metrics is generally based on two types of approaches [16]: (i) quality measurement approaches with reference, which aim to evaluate the compliance of the target (degraded) image with respect to an original or reference image,and (ii) non-referenced quality measurement approaches based on statistical learning, natural scene statistics, or specific distortions. Video quality metadata used in this study are related to non-referenced measurement approaches. In these approaches, video-quality metadata are provided as feature vectors extracted from images.

Examples of metrics without references G-BLIINDS2 and BIQI have been developed by in [2]. As shown in Figure 1, these metrics are used to evaluate the quality of three images extracted from the TID2008 database [15], and degraded from left to right respectively by JPEG compression, JPEG2000 transmission error and noise due to the insertion of blocks of different color and intensity. MOS values (Mean Opinion Score) are provided by human observations and represent the ground truth of the image quality assessment. These values are calculated on a scale from 0 (very poor quality) to 9 (excellent quality).

3.1.2 Metadata describing the usability/utility of the video.

Usability/utility of video content can be defined based on features related to low-level image information. Features related to various acquisition artifacts such as blur, brightness, capture noise, etc. are parameters that are taken into account when defining the usability/utility criteria of the video.

The usability/utility of the video takes into account the ability to detect, recognize or identify objects in the videos. Using the "*Johnson's criteria*" is a basis for defining metrics to evaluate the usability/utility of videos. Johnson has defined thresholds, known as "*Johnson's criteria*", as the effective resolutions for detecting, recognizing or identifying targets captured by the cameras. Let's remember that "detect" is the ability to distinguish an object from the background, *recognize* is the ability to classify objects (people, vehicle, etc.), *identify* is the ability to describe the object in detail (person with a hat, reading a license plate, etc.). Thresholds defined by Johnson's criteria can be affected by factors such as field of view, spatial resolution, scene occultation, etc. This study proposes to make a subjective evaluation of detection, recognition and identification, then to introduce the performed scores into a machine learning algorithm in order to propose metadata of usability/utility of the videos.

3.1.3 Proposed data model.

Figure 2 represents a generic metadata model of video quality and usability/utility metadata proposed for negative filtering of videosurveillance content. This metadata model highlights all the entities taken into account in the modeled system, as well as the relationships between the different entities. The definition of the classes **VIDEO** and **FRAME** is essential for the subsequent definition of the other classes. In conventional video analysis, videos are divided into scenes, each one related to a different aspect of the entire video, and the scenes are subdivided into shots, each one is a single contiguous series of frames derived from one shot. Video features are computed per frame or group of frames, so the video can be divided directly into frames without having to define video segments. Video features for each frame are represented by the class **FEATURE** and each descriptor is linked to a specific processing (class **PROCESSING**). Functions ***FeatureValue()*** and ***Confidence()*** are respectively used to compute the global value and global confidence based on the quality (class **QUALITY**) or usability (class **USABILITY**) metadata attributes of the videos. New attributes (**ATTRIBUTE_QUALITY, ATTRIBUTE_USEABILITY**) can be defined for the metadata at any time. Thresholds (class **THRESHOLD**) are defined for each investigation mode (class **INVESTIGATION MODE**), in order to determine the compatibility of a processing to the chosen investigation mode. Although the required metadata for different types of video analytics could differ, the proposed metadata model is designed to be generic, enabling new metadata (inherited from **FEATURE**) to be easily integrated so that new requirements can be taken into account.

3.2 Filtering mechanism

Negative filtering is a pre-analysis module that will be integrated upstream in the overall process of massive video analysis. At the end of its computations, the negative filtering module provides two types of information:

- **Urgent analysis**: the result of the filtering will indicate for each video segment and for each automatic processing if an urgent investigation is relevant or not. The result will be displayed a two-level colour code: green if video sequence is compatible with an automatic processing in the urgent investigation mode, red if not.
- **Deep analysis**: the result of the filtering will show for each video sequence and each automatic processing a compatibility score with the deep investigation mode. The result will be presented as a three-level colour code based on predefined thresholds: Green is defined for perfect compatibility, orange for medium compatibility and red for incompatibility. Investigators will then choose whether to process the video sequences in orange or not depending on their needs and time resources.

Example: Figure 3 shows the results of negative filtering for a given automatic processing (e.g. vehicle detection) on the video "*file_001.mp4*". Color coding shows that the selected automatic processing can be applied to the video sequences "U_2" and "U_4" in the urgent analysis mode, as well as to the

G-BLINDS2 = 0,786
BIQI = 0,756
MOS =3,65

G-BLINDS2 = 0,662
BIQI = 0,714
MOS =2,67

G-BLINDS2 = 0,803
BIQI = 0,759
MOS =3,69

Figure 1: Examples of image quality scores performed using G-BLINDS2 and BIQI non-reference metrics.

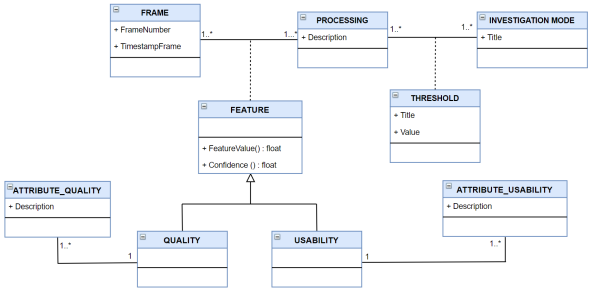


Figure 2: Generic model for video quality and usability/utility metadata.

video sequences "A_2", "A_4", and "A_6" in the deep analysis mode. Depending on their needs and time resources, investigators can analyze the video segment "A_3".

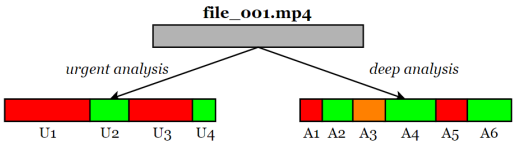


Figure 3: Example of negative filtering results.

Data definition

Definition 1: a video segment u is a sequence of successive frames f_1, f_2, ..., f_i within a video v, *debSeg(u)* gives the first frame of the segment and $finSeg(u)$ gives the last frame of the segment. A video segment $u \in v$ is defined by $u = [f_{start}, f_{end}]$ where f_{start} is the frame that represents the beginning of the segment and f_{end} is the frame that represents the end of the segment. So for the video segment u, $debSeg(u)$ and $finSeg(u)$ return f_{start} and f_{end} respectively;

Definition 2: a frame is compatible with a given automatic processing for a given analysis mode if the global features value associated is included in a range s called the compatibility threshold. Threshold s is defined by $s = [value_1, value_2]$, where $0 \leq value_1 < value_2 \leq 1$.

Definition 3: An automatic processing t refers to an algorithm for automatic video analysis and has a compatibility threshold for each video analysis mode. Automatic processing t is defined by $t = \{s(m)\}$ where each s represents the compatibility threshold of the analysis mode m.

3.2.1 *Negative filtering algorithms.*

The proposed negative filtering is based on the metadata modelling presented earlier. The goal is to define metadata-based query algorithms to automatically discard unusable video segments based on quality and video utility/usability criteria. Filtering algorithms have as parameters a video list, an automatic processing list, and customizable thresholds. Thresholds are defined for each analysis mode (urgent or deep analysis) in order to determine the compatibility of the video sequences with the different automatic processing (face detection, vehicle detection, plate detection and recognition). Compatibility to an analysis mode is determined by comparing the global features value for each frame (or group of frames) of video to the compatibility thresholds defined for different automatic processing. Then, the results of the comparisons are used to build (frame grouping) video segments.

Given a set of videos $V = \{v_1, v_2, ..., v_i\}$ (each video v_i composed of a set of frames $F_i = \{f_1^i, f_2^i, ..., f_n^i\}$) and a set of automatic processing $T = (t_1, t_2, ..., t_j)$, the result of negative filtering for each analysis mode is a set of triplets: $R = \{r = (t_j, v_i, [f_{start}^i, f_{end}^i])\}$, where $t_j \in T, v_i \in V,$ and $f_{start}^i, f_{end}^i \in F_i | f_{start}^i \leq f_{end}^i$. For **deep** analysis mode, there are compatible video segments and medium compatible (optional) video segments with automatic processing.

Algorithms 1 and 2 provide negative filtering results for both analysis modes. For these two algorithms, function *getVideoFrames*(v_i) retrieves in a list all the frames of the video v_i, and function *getFeatureValue*(f_k, t_j) retrieves for each frame f_k of this list the global feature value corresponding to the automatic processing t_j. This global features value is then compared to the different thresholds defined for each analysis mode in order to determine the compatibility of the frame with automatic processing. Algorithms run in two main steps which are frame filtering and segment composition.

Algorithm 1: Negative filtering algorithm for urgent analysis

Input: a set of processing tasks: T and a set of videos: V
Output: a list of compatible video segments for each type of processing task

foreach t_j *in* T **do**
  foreach v_i *in* V **do**
    $frameList \leftarrow getVideoFrames(v_i)$;
    foreach f_k *in* $frameList$ **do**
      $val \leftarrow getFeatureValue(f_k, t_j)$;
      if $val \geq t_j.s(u).param1$ *and* $val \leq t_j.s(u).param2$ **then**
        $add(urgentSegment, f_k)$;
        $f^i_{start} \leftarrow debSeg(urgentSegment)$;
        $f^i_{end} \leftarrow finSeg(urgentSegment)$;
      else if $urgentSegment$ *is not empty* **then**
        $addResult(t_j, v_i, [f^i_{start}, f^i_{end}])$;
        $clear(urgentSegment)$;
      else
        $clear(urgentSegment)$;
      end if
    end foreach
    if $urgentSegment$ *is not empty* **then**
      $addResult(t_j, v_i, [f^i_{start}, f^i_{end}])$;
      $clear(urgentSegment)$;
    end if
  end foreach
end foreach

Frame filtering step: is the step that consists of comparing each frame of a video to the different thresholds defined in order to determine its eligibility for a given automatic processing according to an analysis mode.

Video segment composition step: since a video segment is defined as a consecutive sequence of frames, this step groups together the frames of a video chronologically and according to the classification (eligibility for a given processing in a given analysis mode) made in the filtering step, in order to build indexed video segments, i.e. eligible or not eligible for a given processing according to a chosen analysis mode.

3.2.2 Example of a negative filtering application case.
Let's consider as inputs for our algorithms:

- video "*video_001.mp4*" shown in figure 4. This video is composed of 20 frames (f_1, f_2, f_3, ..., f_{20});
- automatic processing $t_1, t_2,$ and t_3 whose compatibility thresholds for each analysis mode are defined in the table of Figure 5.

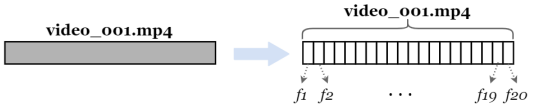


Figure 4: Example of video input for negative filtering.

The global features values for each frame of the video and related automatic processing are presented in the table 1.

Algorithm 2: Negative filtering algorithm for deep analysis

Input: a set of processing tasks: T and a set of videos: V
Output: a compatible list and an optional list of video segments for each type of processing task

foreach t_j *in* T **do**
  foreach v_i *in* V **do**
    $frameList \leftarrow getVideoFrames(v_i)$;
    foreach f_k *in* $frameList$ **do**
      $val \leftarrow getFeatureValue(f_k, t_j)$;
      if $val \geq t_j.s(a).param1$ *and* $val \leq t_j.s(a).param2$ **then**
        $add(indepthSegment, f_k)$;
        $f^i_{start} \leftarrow debSeg(indepthSegment)$;
        $f^i_{end} \leftarrow finSeg(indepthSegment)$;
      else if $val \geq t_j.s(o).param1$ *and* $val \leq t_j.s(o).param2$ **then**
        $add(OptionalSegment, f_k)$;
        $f^i_{start} \leftarrow debSeg(OptionalSegment)$;
        $f^i_{end} \leftarrow finSeg(OptionalSegment)$;
      else
        if $indepthSegment$ *is not empty* **then**
          $addResultA(t_j, v_i, [f^i_{start}, f^i_{end}])$;
          $clear(indepthSegment)$;
        else
          $clear(indepthSegment)$;
        end if
        if $OptionalSegment$ *is not empty* **then**
          $addResultO(t_j, v_i, [f^i_{start}, f^i_{end}])$;
          $clear(OptionalSegment)$;
        else
          $clear(OptionalSegment)$;
        end if
      end if
    end foreach
    if $indepthSegment$ *is not empty* **then**
      $addResultA(t_j, v_i, [f^i_{start}, f^i_{end}])$;
      $clear(indepthSegment)$;
    end if
    if $OptionalSegment$ *is not empty* **then**
      $addResultO(t_j, v_i, [f^i_{start}, f^i_{end}])$;
      $clear(OptionalSegment)$;
    end if
  end foreach
end foreach

Example of urgent analysis

The two steps of algorithm 1 are presented in Figure 6. At step 1, the global feature value associated to each frame and related to an automatic processing is retrieved and compared to the threshold defined for this automatic processing. For example automatic processing t_1 is applicable to a frame if the global features value for the frame is in interval $[0.85, 1]$ (see Figure 5). The global features value for frame f_1 for automatic processing t_1 is 0.71 (see table 1), so processing t_1 is not applicable to frame f_1 in urgent mode. Therefore, on Figure 6, frame f_1 is represented in red color for processing t_1. However, automatic processing t_2 is applicable to frame f_1 in urgent

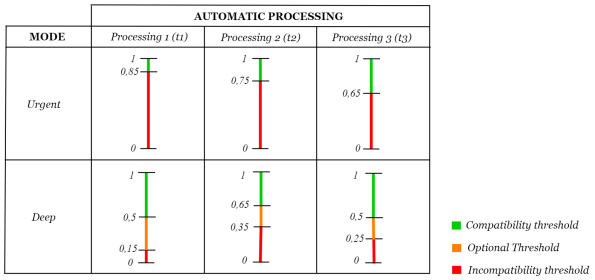


Figure 5: Examples of compatibility thresholds for different automatic processing.

Table 1: Global feature values for each automatic processing.

	AUTOMATIC PROCESSING		
FRAMES	processing t_1	processing t_2	processing t_3
f_1	0.71	0.76	0.67
f_2	0.68	0.80	0.71
f_3	0.74	0.83	0.74
f_4	0.11	0.79	0.68
f_5	0.12	0.32	0.76
f_6	0.14	0.29	0.82
f_7	0.13	0.30	0.74
f_8	0.87	0.24	0.91
f_9	0.91	0.41	0.89
f_{10}	0.94	0.52	0.21
f_{11}	0.98	0.49	0.23
f_{12}	0.16	0.61	0.19
f_{13}	0.18	0.83	0.24
f_{14}	0.14	0.78	0.53
f_{15}	0.19	0.84	0.61
f_{16}	0.25	0.90	0.51
f_{17}	0.35	0.87	0.55
f_{18}	0.56	0.79	0.62
f_{19}	0.67	0.87	0.59
f_{20}	0.78	0.93	0.64

mode, because the global feature value of f_1 for this automatic processing ($value = 0.76$) fits the defined threshold (range $[0.75, 1]$) for compatibility in urgent mode. This is illustrated at Figure 6 by the representation of frame f_1 in green color for automatic processing t_2.

At step 2, frames are grouped together to create video segments, while distinguishing for each processing the video segments that are compatible or not with the urgent analysis mode. For example, processing t_2 can be applied to video segments V_1 (composed of frames f_1, f_2, f_3, f_4) and V_3 (composed of frames f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}) for an urgent analysis of the video "*video_oo1.mp4*".

Negative filtering result in urgent mode for our example is the set: {(t_1, *video_001.mp4,* [f_8, f_{11}]), (t_2, *video_001.mp4,* [f_1, f_4]), (t_2, *video_001.mp4,* [f_{13}, f_{20}]), (t_3, *video_001.mp4,* [f_1, f_9])}.

Example of deep analysis

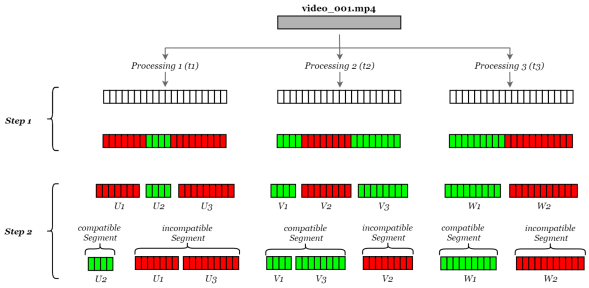


Figure 6: Example of negative filtering in urgent analysis mode.

Figure 7 shows the 2 running steps of algorithm 2. At step 1 of this algorithm, a new threshold is taken into account in the comparison of global feature values associated to each frame for a given automatic processing. This is the medium compatibility threshold. At step 1 of the urgent analysis mode, a frame could have either the "compatible" state (shown in green) or the "incompatible" state (shown in red) for a given automatic processing. With the deep analysis mode, a new "medium compatible" or "optional" state (shown in orange) is taken into account. For example, for the deep analysis mode, automatic processing t_1 is optional for a frame if the global feature value of the frame is in the range $[0.15, 0.5]$ (see Figure 5). The global feature value of frame f_{12} for automatic processing t_1 is 0.16 (see table 1), so automatic processing t_1 is optional for frame f_{12} in the deep analysis mode. Therefore, on Figure 7, frame f_{12} is shown in orange color for automatic processing t_1.

At step 2, video segments that are medium compatible or optional for an automatic processing can be constructed. For example, video segment U_4 (composed of frames f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}) is medium compatible with processing t_1 (or optional) for a deep analysis mode of video "*video_oo1.mp4*".

Negative filtering in deep analysis mode for our example consist of two sets:

- the set composed of video segments compatible with deep analysis mode: {(t_1, *video_001.mp4,* [f_1, f_3]), (t_1, *video_001.mp4,* [f_8, f_{11}]), (t_1, *video_001.mp4,* [f_{18}, f_{20}]), (t_2, *video_001.mp4,* [f_1, f_4]), (t_2, *video_001.mp4,* [f_{13}, f_{20}]), (t_3, *video_001.mp4,* [f_1, f_9]), (t_3, *video_001.mp4,* [f_{14}, f_{20}])}.
- the set composed of video segments that are medium-compatible or optional for deep analysis mode: {(t_1, *video_001.mp4,* [f_{12}, f_{17}]), (t_2, *video_001.mp4,* [f_9, f_{12}])}.

4 EXPERIMENTS AND RESULTS DISCUSSION

4.1 Architecture of the proposed framework

Data model and algorithms previously proposed have enabled the development of a framework for negative filtering of large volumes of video collected from CCTV systems. Figure 8 illustrates the architecture of the proposed framework. The

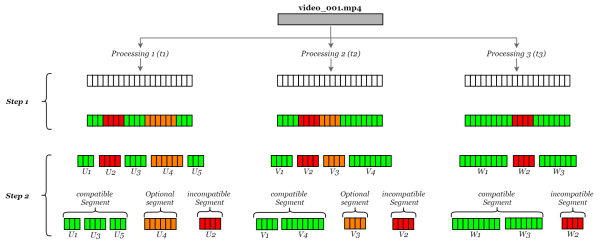


Figure 7: Example of negative filtering in deep analysis mode.

prototype has been developed in Java and communicates with an Oracle database that integrates spatial extension to store spatial data and perform spatial queries. This framework is composed of three modules: metadata collection module, user interface module, and metadata management and processing module.

Metadata collection module. Metadata sources considered in this work are multiple can be classified into two main groups: metadata from sensors, metadata from video analytics tools.

User interface module. User interface enables to define user's queries or to use queries already defined in JSON format files, to view the data stored in the database and to view the results.

Metadata management and processing module. This is the most important part of the framework. It has three components: metadata storage, query interpreter, and negative filtering. Component *Metadata repository* enables to store metadata collected for our experiments based on the proposed data model. Component *Query interpreter* interprets the user's query to make it usable by the framework. It takes as input a JSON file containing the different elements of the query such as location, time, analysis mode, etc. Component *Negative filtering* implements the proposed negative filtering algorithms and returns the results to the user through User interface module.

4.2 Data set and experiments

4.2.1 Data set presentation.

Experiments were performed on the ToCaDa dataset [12], which contains a collection of videos recorded on the campus of the University of Toulouse III - Paul Sabatier. A detailed description of the ToCaDa dataset (Toulouse Campus surveillance Dataset) is presented in the work of our team and the laboratory [12]. Here we provide a useful summary for understanding our experiments. This dataset contains two sets of 25 time-synchronized videos corresponding to two predefined scenarios. The videos were recorded on July 17, 2017 at 9:50 a.m. for the first scenario and at 11:04 a.m. for the second scenario. Cameras were installed as follows:

- 9 cameras were located inside the main building and were shooting from the windows of the different floors. All these cameras were focused on the parking and the path leading to the main entrance of the building with large overlapping fields of view.
- 8 cameras were located in front of the building, with large overlapping fields of view also (these 9+8=17 cameras with overlapping fields of view are visible in Figure 9).
- 8 cameras were placed further away, scattered across the university campus (see Figure 10). The fields of view of these cameras are disjointed.

About 20 actors were invited to follow two realistic scenarios by performing predefined actions, such as driving a car, walking, entering or leaving a building, or holding an object in their hands during filming. In addition to ordinary actions, some suspicious behaviour are present. Specifically:

- In the first scenario, a suspicious car (C) with two men inside (D the driver and P the passenger) arrives and parks in front of the building (in view of the overlapping cameras). P gets out of car C and enters the building. Two minutes later, P leaves the building holding a package and goes into C. C leaves the parking lot and drives away from the university campus (passing some of the cameras with disjointed fields of view).
- In the second scenario, the situation is similar with a suspicious car (C) and two men inside (D the driver and P the passenger) arriving and parking in front of the building (again in view of the overlapping cameras). P gets out of C and enters the building. A minute later, a woman complains to D about her poor parking. C moves away quickly and stops in the field of view of camera 8. About a minute later, P leaves the main building holding a package and runs away. P meets C a little further away (in the field of view of camera 8), enters C, and C quickly leaves the university campus (passing within the fields of view of most cameras).

In our experiments, we do not take into account all the cameras located in front of and inside the main building, as they were deployed for the purpose of reconstructing 4D scenes, which explains the overlapping of their fields of view. Among these cameras, we have selected three (cameras 2, 5 and 25 in Figure 9) whose fields of view allow maximum coverage of the desired area. All cameras in Figure 10 are included in the experiments. Cameras 2, 5, 25 of Figure 9 have respectively for identifier *camA*, *camB*, *camC*, and cameras 6, 8, 9, 10, 11, 12, 13, 14 of Figure 10 have respectively for identifier *camD*, *camE*, *camF*, *camG*, *camH*, *camI*, *camJ*, *camK*. Each camera filmed for a total of 10 minutes 28 seconds (4 minutes 48 seconds for Scenario 1 and 5 minutes 40 seconds for Scenario 2). The videos from the 11 cameras (*camA* to *camK*) are respectively identified by V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}.

Tests were conducted to measure the impact of our proposition on video processing times. In this experiment we run our negative filtering algorithms on videos from the eleven cameras selected for the experiments. As the quality of the videos from the selected cameras was perfect, we randomly deteriorated some video segments in order to filter based on quality criteria.

Quality of each frame was evaluated using the BRISQUE (Blind Referenceless Image Spatial Quality Evaluator) metric [13]. BRISQUE metric is the most commonly used metric in

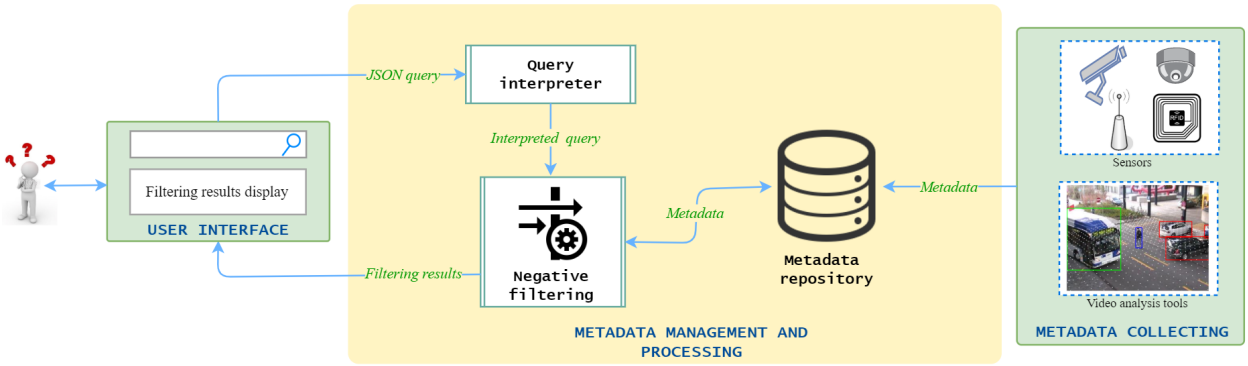


Figure 8: Architecture of the proposed framework.

Figure 9: Main building with 17 cameras with overlapping fields of view [12].

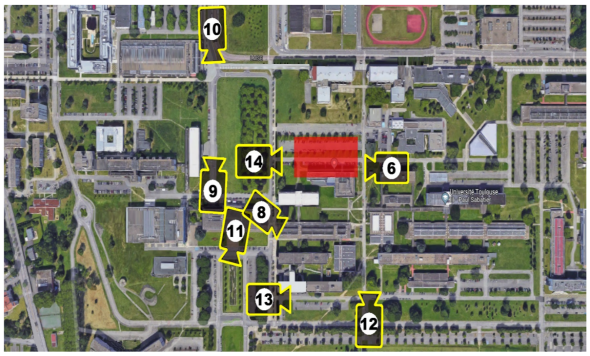


Figure 10: Camera positions on the campus of the Université Paul Sabatier. These 8 cameras have disjointed views [12]. La zone rouge correspond à la Figure 9

Image Quality Assessment (IQA) without a reference [3], i.e. not requiring a reference image with good quality. Extracting this metric for each frame of the degraded video has been implemented in Python using the "pybrisque" module proposed by Kushashwa Ravi Shrimali (under MIT license)[1]. Extracted quality metadata were saved in the database.

4.3 Results

Negative filtering for a given video, returns for each frame of the video its compatibility (*compatible, optional, incompatible*) with each processing (*face detection, vehicle detection, plate detection and recognition*) according to the different analysis modes (*urgent, deep*). Figures 11, 12, and 13 represent the negative filtering of video V_1 for the three automatic processing according to each analysis mode. On these figures, green frames are *compatible*, orange frames are *optional*, and red frames are *incompatible* with the automatic processing chosen in the given analysis mode.

Negative filtering results are returned as video segments consisting of successive frames of same color. For example, for automatic processing *"Face Detection"*, result for video V_1 is:

- **Urgent analysis mode**
  - Compatible video segments: $[V_1F_1, V_1F_{1963}]$, $[V_1F_{2747}, V_1F_{5031}]$, and $[V_1F_{9274}, V_1F_{11383}]$.
  - Incompatible video segments: $[V_1F_{1964}, V_1F_{2746}]$, $[V_1F_{5032}, V_1F_{9274}]$, and $[V_1F_{11384}, V_1F_{18840}]$.
- **Deep analysis mode**
  - Compatible video segments: $[V_1F_1, V_1F_{4927}]$, $[V_1F_{9275}, V_1F_{11561}]$, and $[V_1F_{16532}, V_1F_{18840}]$.
  - Optional video segments: $[V_1F_{14242}, V_1F_{16531}]$.
  - Incompatible video segments: $[V_1F_{4928}, V_1F_{9274}]$, and $[V_1F_{11562}, V_1F_{14241}]$.

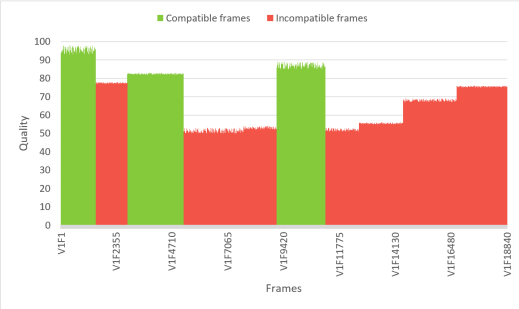
4.4 Evaluation

We evaluated the time savings gained using the proposed approach. This time saving is evaluated for each automatic processing in different analysis modes.

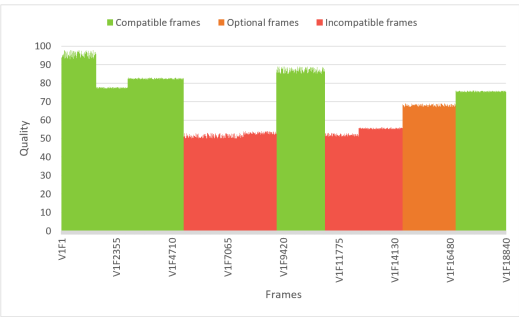
The time savings G_{time} for a given video is the ratio of the total processing time for the non-filtered video T_{total} over the total processing time for the filtered video $T_{approach}$.

$$G_{time} = \frac{T_{total}}{T_{approach}} \quad (1)$$

[1]https://github.com/krshrimali/No-Reference-Image-Quality-Assessment-using-BRISQUE-Model

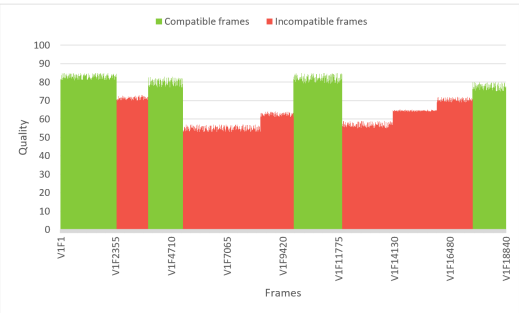


(a) Urgent analysis mode.

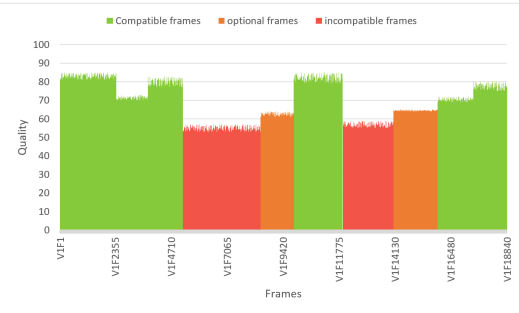


(b) Deep analysis mode.

Figure 11: Face detection for video V_1.



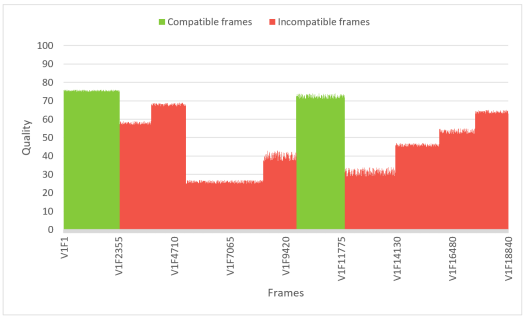
(a) Urgent analysis mode.



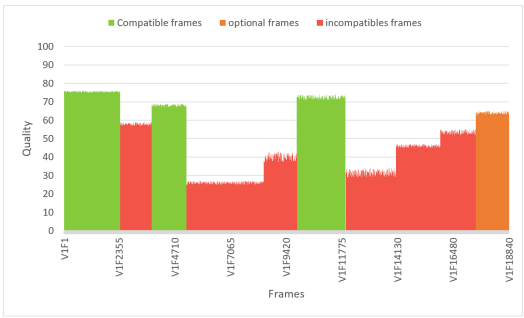
(b) Deep analysis mode.

Figure 12: Vehicle detection for video V_1.

where $T_{total} = n_{total} * t_{processing}$ and $T_{approach} = (n_{total} * t_{filtering}) + (n_{rest} * t_{processing})$, with:



(a) Urgent analysis mode.



(b) Deep analysis mode.

Figure 13: Plate detection and recognition for video V_1.

- n_{total} is the number of frames to process before filtering,
- n_{rest} is the number of frames to process after filtering,
- $t_{processing}$ is the processing time of a frame by an automatic processing (e.g. vehicle detection),
- $t_{filtering}$ is the filtering time of a frame by our negative filtering algorithm.

Equation 1 becomes:

$$G_{time} = \frac{n_{total} * t_{processing}}{(n_{total} * t_{filtering}) + (n_{rest} * t_{processing})}$$

$$= \frac{\frac{t_{processing}}{t_{filtering}}}{1 + \frac{n_{rest}}{n_{total}} * \frac{t_{processing}}{t_{filtering}}}$$

$$G_{time} = \frac{G_{filtering}}{1 + \frac{G_{filtering}}{G_{frame}}} \quad (2)$$

where:

- $G_{filtrage} = \frac{t_{traitement}}{t_{filtrage}}$ is a saving of filtering time (to be distinguished from the total time saving G_{time},
- $G_{frame} = \frac{n_{total}}{n_{reste}}$ is a saving of frame.

Evaluation of the time saving for automatic processing "face detection", "vehicle detection", and "plate detection and recognition" corresponding to the 11 videos taken into account in this experiment is illustrated respectively by the figures 14, 15, and 16.

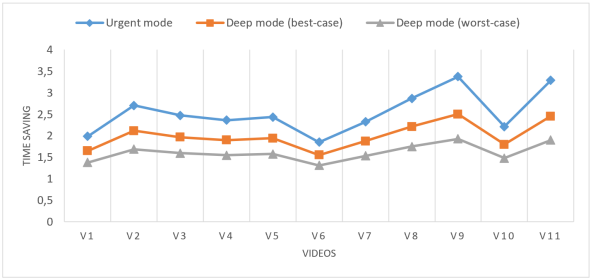


Figure 14: Time saving for automatic processing "face detection".

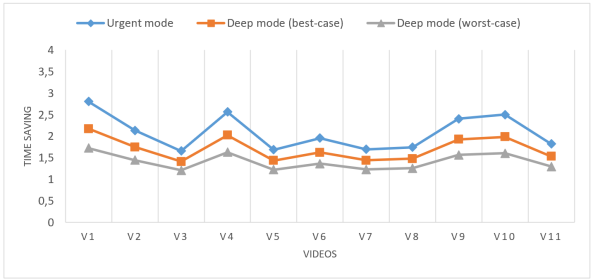


Figure 15: Time saving for automatic processing "vehicle detection".

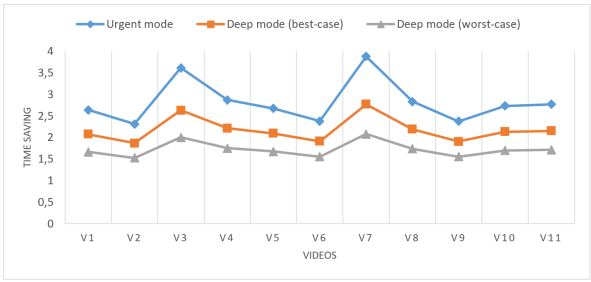


Figure 16: Time saving for automatic processing "plate detection and recognition".

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed a so-called "negative" filtering approach enabling, in the context of a posteriori video analysis, to save processing time by eliminating, among the mass of videos to be analysed, video sequences that are unusable based on the quality and usability/utility of the videos. The defined filtering algorithms are based on a generic metadata modeling of video quality and usability/utility. The proposed approach has been validated in the French project ANR FILTER2 and experimental results achieved with the developed framework shown the relevance of our approach and its feasibility in a real case. Although it is functional, improvements can still be added to the proposed framework. In order to improve the negative filtering of large video collections related to post-research of evidence, a perspective of this work is to into account contextual information (open data, mobility, social media, etc.) in order to filter according to new parameters or to enrich videos.

REFERENCES

[1] Cyril Carincotte, X Desurmont, Bertrand Ravera, François Brémond, J Orwell, SA Velastin, Jean-Marc Odobez, B Corbucci, J Palo, and J Cernocky. 2006. Toward generic intelligent knowledge extraction from video and audio: the EU-funded CARETAKER project. *IET Conference on Crime and Security* (2006), 470–475.

[2] Christophe Charrier. 2011. *Modélisation statistique et classification par apprentissage pour la qualité des images.* Ph.D. Dissertation.

[3] Christophe Charrier, Abdelhakim Saadane, and Christine Fernandez-Maloigne. 2015. Comparison of no-reference image quality assessment machine learning-based algorithms on compressed images. In *Image Quality and System Performance XII*, Vol. 9396. International Society for Optics and Photonics, 939610.

[4] Yi-Ling Chen, Tse-Shih Chen, Tsiao-Wen Huang, Liang-Chun Yin, Shiou-Yaw Wang, and Tzi-cker Chiueh. 2013. Intelligent urban video surveillance system for automatic vehicle detection and tracking in clouds. In *2013 IEEE 27th international conference on advanced information networking and applications (AINA)*. IEEE, 814–821.

[5] Dana Codreanu. 2015. *Modélisation des métadonnées spatio-temporelles associées aux contenus vidéos et interrogation de ces métadonnées à partir des trajectoires hybrides : Application dans le contexte de la vidéosurveillance.* Ph.D. Dissertation. Université Paul Sabatier.

[6] Hongli Deng, Mun Wai Lee, Asaad Hakeem, Omar Javed, Weihong Yin, Li Yu, Andrew Scanlon, Zeeshan Rasheed, and Niels Haering. 2010. Fast forensic video event retrieval using geospatial computing. In *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application*. ACM, 1–8.

[7] Gerda Edelman and Jurrien Bijhold. 2010. Tracking people and cars using 3D modeling and CCTV. *Forensic science international* 202, 1-3 (2010), 26–35.

[8] Philippa Fletcher. 2011. Is CCTV effective in reducing anti-social behaviour. *Internet Journal of Criminology. UK: Lancaster University, Unpublished dissertation* (2011).

[9] David Gerónimo and Hedvig Kjellström. 2014. Unsupervised surveillance video retrieval based on human action and appearance. In *2014 22nd International Conference on Pattern Recognition*. IEEE, 4630–4635.

[10] Martin Gill and Angela Spriggs. 2005. *Assessing the impact of CCTV*. Vol. 292. Home Office Research, Development and Statistics Directorate London.

[11] HC Lee and EM Pagliaro. 2013. Forensic evidence and crime scene investigation. *Journal of Forensic Investigation* 1, 1 (2013), 1–5.

[12] Thierry Malon, Geoffrey Roman-Jimenez, Patrice Guyot, Sylvie Chambon, Vincent Charvillat, Alain Crouzil, André Péninou, Julien Pinquier, Florence Sèdes, and Christine Sénac. 2018. Toulouse campus surveillance dataset: scenarios, soundtracks, synchronized videos with overlapping and disjoint views. In *Proceedings of the 9th ACM Multimedia Systems Conference*. 393–398.

[13] Anish Mittal, Anush K Moorthy, and Alan C Bovik. 2011. Blind/referenceless image spatial quality evaluator. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*. IEEE, 723–727.

[14] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. 2008. Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision* 79, 3 (2008), 299–318.

[15] N Ponomarenko, V Lukin, K Egiazarian, Jaakko Astola, Marco Carli, and Federica Battisti. 2008. Color image database for evaluation of image quality metrics. In *2008 IEEE 10th workshop on multimedia signal processing*. IEEE, 403–408.

[16] Michele A Saad, Alan C Bovik, and Christophe Charrier. 2010. A DCT statistics-based blind image quality index. *IEEE Signal Processing Letters* 17, 6 (2010), 583–586.

[17] Florence Sèdes and Franck Jeveme Panta. 2017. (Meta-) data modelling: gathering spatio-temporal data for indoor-outdoor queries. *SIGSPATIAL Special* 9, 1 (2017), 35–42.

[18] Claire AJ van den Eeden, Christianne J de Poot, and Peter J Van Koppen. 2016. Forensic expectations: Investigating a crime scene with prior information. *Science & justice* 56, 6 (2016), 475–481.

[19] Brandon C Welsh and David P Farrington. 2008. Effects of closed circuit television surveillance on crime. *Campbell systematic reviews* 4, 1 (2008), 1–73.

[20] B Yogameena and K Sindhu Priya. 2015. Synoptic video based human crowd behavior analysis for forensic video surveillance. In *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*. IEEE, 1–6.