An information retrieval system for fast identification of objects in long duration surveillance videos

Edgar Abidán Padilla Luis, David Pinto , Rigoberto Cerino Jiménez, Francisco José López Cortés, Alberto Esteban Reyes Peralta, Beatriz Beltrán

Facultad De Ciencias de la Computación, Benemérita universidad Autónoma de Puebla, Avenida San Claudio, Heroica Puebla de Zaragoza, 72570, Puebla, México.

*Corresponding author(s). E-mail(s): david.pinto@correo.buap.mx;
Contributing authors: edgar.abidan.pl@gmail.com;
cerino_rigoberto@hotmail.com; lopcorp.z@gmail.com;
alberto.reyesp@alumno.buap.mx; bbeltranmtz@gmail.com;

Abstract

This paper presents an information retrieval system for long-duration surveillance videos, utilizing an approach based on text processing generated by an object detection model. Our system offers a quick and efficient response, which can significantly improve the task of searching for scenes in surveillance videos. This tool has the potential to streamline security operations and facilitate the work of personnel responsible for monitoring and analyzing large volumes of visual data. However, more detailed evaluation and additional testing are required to validate its effectiveness in real surveillance environments. Ultimately, our goal is to contribute to the advancement of security and surveillance technologies through the development of innovative and practical solutions.

Keywords: information retrival system, surveillance videos, objects identifiction, YOLO-NAS.

1 Introduction

Enormous amounts of data are accumulated daily on servers, in the cloud, and even on personal devices, with a considerable portion of this data comprising multimedia files (*images*, *video*, and *audio*). Retrieving multimedia information relevant to a user is extremely complex and requires systems capable of extracting features that can be analyzed, understood, and manipulated. This situation makes multimedia information retrieval a challenging area full of opportunities [13].

Zhou's study [18] shows that contemporary methodologies for information retrieval rely on cross-modal models. Additionally, table 5 highlights the computational costs required to train these models, which tend to be prohibitively high for most companies due to the limitations of available computing resources. This adds more complexity to the task of multimedia information retrieval.

This document presents an innovative information retrieval system that quickly and cost-effectively identifies objects present in long-duration security videos. To achieve this task, the video is processed using an object detector (a pre-trained neural network model), which generates a text corpus from the obtained labels. This corpus feeds an information retrieval system that utilizes highly efficient techniques. Finally, through a web application, users can interact with the system by entering their queries in natural language and obtain the spatial and temporal location of the searched objects in the video within milliseconds.

2 State of the Art

With the aim of simplifying tasks related to video surveillance, security and monitoring systems are constantly evolving. This is evidenced in the comprehensive study conducted by Elharrouss [5], which analyzes the main components, architectures, and methodologies used in these systems. Additionally, a comparison of the functionalities of various systems is carried out.

Karbalaie [7] reviews several event detection systems in security cameras. He highlights the importance of rapid and accurate detection, as well as the implementation of effective classification models to obtain results in automatic event detection systems. He also notes that most classifiers tend to work with short videos, which is impractical in real-world situations.

On the other hand, Ingle [6] develops an MSD-CNN (Multiclass Subclass Detection Convolutional Neural Network) scene classifier that distinguishes between normal frames, such as walking, and abnormal frames, such as scenes of people with guns or knives. His results show that even with limited resources, an accuracy of 85.5% can be achieved in detecting guns or knives.

Murugesan [10] employs a multilayer perceptron recurrent neural network for anomaly detection in surveillance systems, with results indicating high performance in metrics such as precision, sensitivity, and specificity.

Adimoolam [2] introduces an innovative technique for identifying multiple objects in surveillance systems. He uses a dense feature selection convolutional neural network with hyperparameter tuning, achieving 98% accuracy in his experiments.

Pérez-Hernandez [12] proposes the use of binarization techniques in convolutional neural networks to improve the detection of small objects that can be manipulated by hand and easily mistaken for a gun or knife. His experiment demonstrates that false positives can be reduced by up to 56.50%.

Castañón [4] focuses on the retrieval of video segments in long-duration surveillance videos. He extracts features from objects and movement paths, storing them in a hash table-based tree to perform efficient queries. He also develops a system for retrieving rare, abnormal, and recurring activities through dynamic programming.

Tseng [15] proposes a system for searching and retrieving people in multi-camera surveillance systems. He uses the YOLACT++ model to segment images and a multilayer convolutional neural network to extract appearance attributes of people. This system implements an attribute matching engine and generates summaries associated with the retrieved elements.

In the last decade, You Only Look Once (YOLO) was introduced by Redmon *et al.* in [14]. This architecture can predict bounding boxes and the probability that these boxes belong to certain classes. YOLO has been widely used in various tasks. Uma [16] reviews the various YOLO architectures and their applications, highlighting their potential for integration with augmented reality.

On the other hand, Abdusalomov [1] improves the performance of YOLOv3 in detecting fires in surveillance videos by creating a specialized dataset and fine-tuning. His experiment demonstrates that the neural network can accurately detect fire regions in real-time.

In addition to object identification and segmentation, YOLO is a very versatile tool that can be used for multimedia information retrieval. Xin [17] uses a specialized dataset to train YOLOv5, and through the obtained features, can retrieve images similar to an input image.

Nath [11] generates data through web mining to train YOLO and improve its accuracy on various test sets. For his part, Kumar uses the labels provided by YOLO to retrieve and display information from the web [8].

After reviewing the state of the art, it was noted that convolutional neural network-based video surveillance systems are used in classification, object, or scene detection tasks and, to a lesser extent, are used to create information retrieval systems. In particular, of all the articles reviewed, none were found that use YOLO for the task of information retrieval.

3 Methodology

Figure 1 shows the methodology that constitutes the information retrieval system presented in this paper. This system has three main components: the first creates a text corpus from a long-duration video, the second is a text information retrieval system that is fed by the created corpus, and the third is a web application that facilitates the interaction between the user and the system.

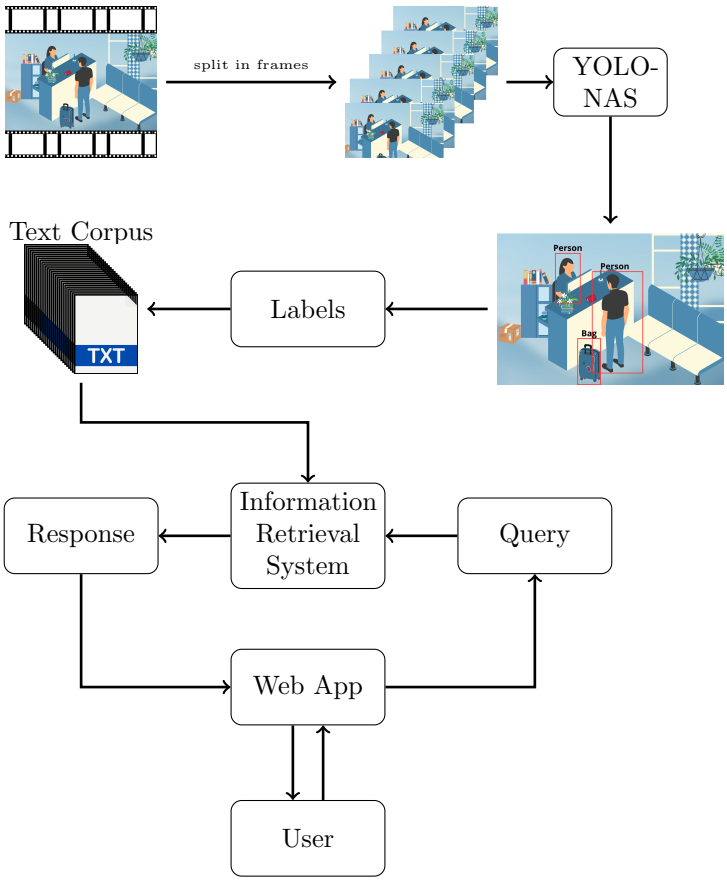

Fig. 1: Methodology of the information retrieval system for fast identification of objects in long-duration videos.

The system operates as follows: first, a frame is extracted from a long-duration video at specified intervals. This image is processed by an object detection model (YOLO-NAS was chosen for this experiment due to its high accuracy and low computational cost). The labels of the detected objects in the frame are obtained and saved in a text file. The collection of these files forms the corpus for the information retrieval system. Meanwhile, the user submits a text query through a web application, and the system generates a response to the query.

4 Development of the Information Retrieval System

4.1 Creation of the Text Corpus

The following procedure was carried out to create the text corpus: a frame is extracted from a long-duration video at specified intervals, each frame is processed by an object

detection model, the labels of the detected objects are obtained and saved in a text file, so each frame will have a unique associated text file. The collection of all generated text files is used as the text corpus, which is subsequently indexed to feed the information retrieval system.

In each of the aforementioned steps, various parameters must be defined, such as the time interval for extracting a frame or the preprocessing performed on the image before being processed by the object detection model. It is important to study the variation of these parameters and their consequences; however, this discussion is not presented in this document, as the objective of this article is to present the information retrieval system. Nonetheless, in a later section, the specific details used to conduct the experiment presented in this article are provided.

4.2 Fault-Tolerant Boolean Information Retrieval System with Synonym Search

Figure 2 details the functioning of the implemented text retrieval system. Initially, the created corpus goes through a text preprocessing module before generating an inverted index during the indexing stage. This inverted index is crucial for the speed of the system, as it allows queries to be executed with a complexity of $O(1)$.

When a user enters a text query through the web application, the system determines if the query is Boolean or not. The query then undergoes the same preprocessing used for the text corpus to homogenize the query and the text present in the corpus. If a Boolean query is identified, the corresponding Boolean operator is applied. On the other hand, if the query is not Boolean, matches are searched for in the inverted index.

To enhance the robustness of the system, two additional modules are implemented. The first is the fault-tolerant retrieval, which attempts to find matches in the inverted index using the Levenshtein distance [9]. Thus, if the entered query is not found in the inverted index (due to typographical errors), elements within a certain distance are retrieved, meaning the system will find the closest word using the Levenshtein algorithm. The second module implements synonym search; if a synonym of a term in the inverted index is entered, the system returns all elements associated with the term in the index. Therefore, a fault-tolerant Boolean information retrieval system with synonym search is created.

The web application was developed using Django, a web development framework in Python known for its ability to simplify and accelerate the creation of web applications. This framework was chosen due to its solid structure, the availability of reusable components, and its scalability. Additionally, Django offers practical and optimized template systems for result visualization. This allowed for the creation of a simple and intuitive yet powerful web application.

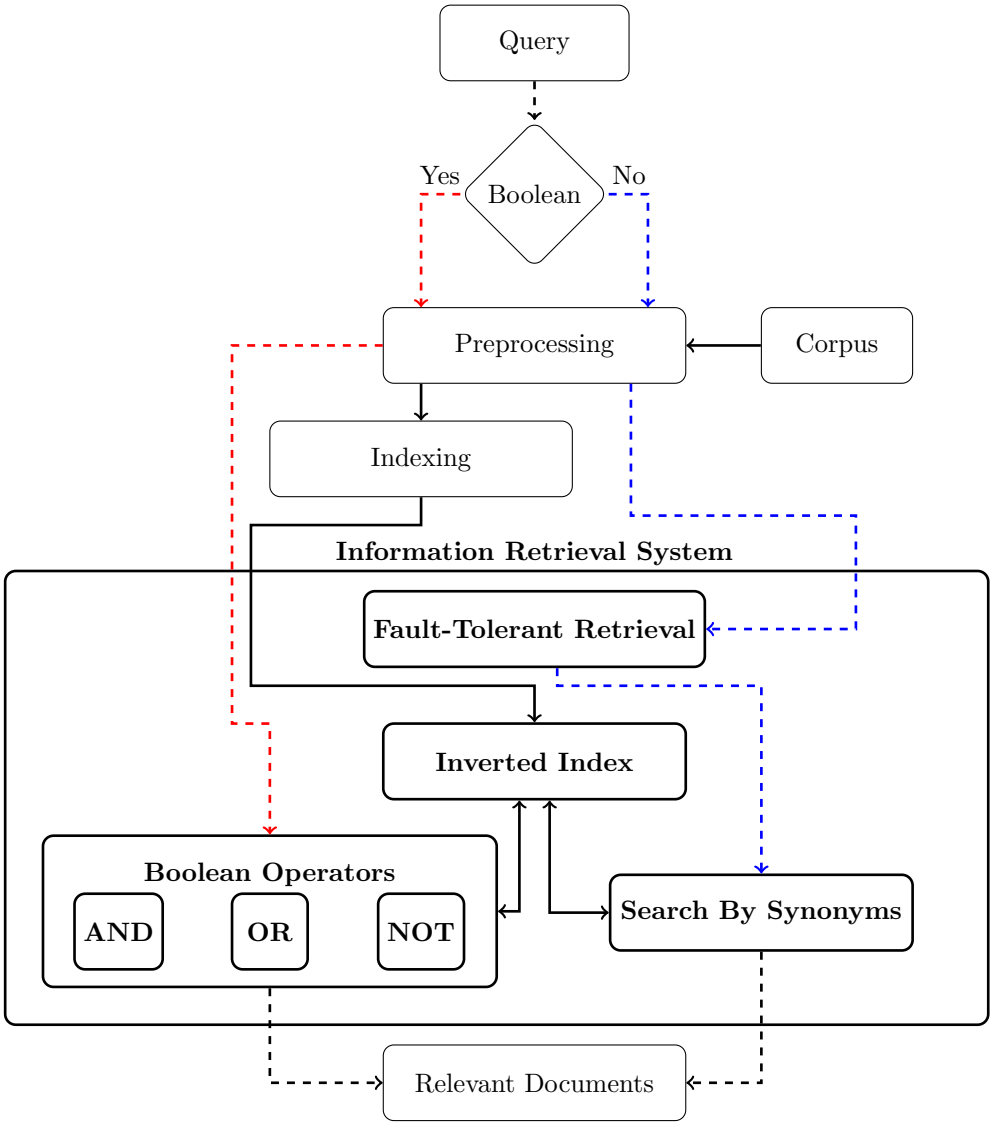

Fig. 2: Implemented Fault-Tolerant Boolean Information Retrieval System with Synonym Search.

5 Experiment

For the experiment, a 5-hour, 6-minute, and 4-second video from the internet was used. The video shows the lobby of an office, and it is of our interest to identify the following common objects in an office: people, backpacks, phones, and computers. Additionally, it is also of interest to see how our system works on a computer that does not have a

GPU. For this purpose, the experiment is conducted on hardware with the following specifications:

- **Hardware Model**: Apple Inc. MacBookPro9,2. (2012).
- **Operating System**: Ubuntu 22.04.4 LTS 64-bit.
- **Processor**: Intel® Core™ i5-3210M CPU @ 2.50GHz × 4.
- **Memory**: 8GB
- **Graphics Card**: Intel® HD Graphics 4000 (IVB GT2).
- **Storage**: 500.1 GB.

The experiment presented in this article has the following peculiarities: every 5 seconds, a frame is captured from the video. This image is converted to grayscale and resized to 680×680. Subsequently, this frame is processed with a precision of 0.7 by the YOLO-NAS model pretrained with the COCO dataset. The model obtains the bounding boxes and the classes to which they belong. The bounding boxes are drawn on the frame to visualize the objects in the image, and the labels are saved in a text file with a name that associates them with the time the frame was captured. This way, the objects are located in time and space within the video.

Once the text corpus is obtained, an inverted index is created. This index takes approximately 124 milliseconds to be created and is saved in a JSON file for easy use in the web application. The JSON file weighs 162.2 KB, which can be loaded into memory without any issues even on computers with lower RAM capacity. The posting list created is the basis of our previously described information retrieval system.

The next section presents a discussion of the system's responses to various queries, the execution of the implemented modules, and the time it takes to present the results.

6 Discussion

It is natural to wonder about the time it takes for the system to build the text corpus, which largely depends on two factors. The first factor is choosing the time interval to capture a frame from the video, and the second important factor is the object detection model to be used. For this experiment, the YOLO-NAS model [3] was employed. The advantage of this model lies in its architecture as it employs quantization-aware blocks, optimizing performance and increasing accuracy compared to other YOLO models. Additionally, it is a model that requires low computational resources. Table 1 shows the chosen time interval to capture a frame from the video, the approximate time in minutes it takes the system to create the text corpus, the number of text files generated (number of processed frames), and the disk memory occupied by the generated text corpus. As expected, the time interval is a decisive factor in the time it takes the system to generate the corpus, as a shorter time interval results in a higher number of images to be processed. However, it should be noted that the weight of the generated corpus is relatively low. Consequently, generating an inverted index is relatively fast, and loading this index into memory has a very low cost. If we divide the time in seconds it takes for the system to generate the text corpus by the number of text documents obtained, we will have the time to create each text file. Upon performing this division, it is noticed that approximately 1 second is required to create each text file. This data

is important as it indicates that the proposed system can be implemented in real-time, and a frame can be obtained every 1.5 seconds from a video surveillance system. Before obtaining the next frame, the system will have processed, saved, and updated the inverted index with the new information.

Interval	Time to generate corpus	Text files	Disk memory
5	53 minutes	3672	77.1 KB
10	31 minutes	1837	38.2 KB
15	18 minutes	1224	25.8 KB

Table 1: Data obtained when generating the text corpus by taking different time intervals to capture a frame from the video.

With the parameters defined in Section 5, the following searches were performed. We input the query *person*. In Figure 3, it is observed that the system returns 2913 images containing a person recognized by the object identification model. It should be noted that, when presenting the results, the user is notified of the exact moment when the presented image can be located via a caption.

Fig. 3: Web interface created for interaction between a user and the information retrieval system. Example query: *person*.

In Figure 4, it is noted that upon clicking **Go to video**, the image from the video at time 03:08:15 (caption below the image) is shown. Several bounding boxes are observed, three of them correspond to people, and the other two correspond to other identified objects. This is because the bounding boxes are drawn when processing the image by the object detection model and not during the query, allowing the

system to retrieve elements in milliseconds. It is important to note that the object detection model fails to detect a fourth person. This could change if the precision level of the model is modified. Additionally, the low quality of the video may affect the object detection model. If we type the word *perosn* into the search engine, simulating a misspelled version of the word *person*, in Figure 5 it can be seen that the information retrieval system displays the same number of results (2913) and the same time (03:08:15) for the first result offered by the *person* query.

Fig. 4: First image retrieved from the query *person*.

Fig. 5: Query with a misspelled word: *perosn*.

If we click on **Go to video**, the system returns the image presented in Figure 4. Additionally, the system displays a caption below the number of results obtained (Potentially interesting: person), indicating that it could refer to the *person* query.

Generally, boolean queries are entered by individuals who are knowledgeable about logical operators and are often implemented in search engines under advanced tools. However, it is very common for people to use phrases with connectors expressing a logical operation. This system is capable of identifying some grammatical connectors that are usually boolean operations through the use of the following dictionary:

```
{ "and": "AND", "or": "OR", "not": "NOT", "with": "AND",
  "including":"AND", "also":"AND", "plus":"AND",
  "besides":"AND", "moreover":"AND", "furthermore": "AND",
  "alternatively": "OR", "instead": "OR", "otherwise": "OR",
  "else": "OR", "either": "OR", "nor": "OR", "elsewhere": "OR",
  "except": "NOT", "excluding": "NOT", "without": "NOT",
  "minus": "NOT" }
```

In this way, it is not necessary to understand how logical operators work; it suffices for the system to identify logical connectors present in the dictionary in queries such as *person with phone* to perform the boolean search *person AND phone*. The result of the *person with phone* query is shown in Figure 6.

In Figure 7, the result for the boolean query *person AND phone* is shown, which displays the same results as those observed in Figure 6.

Boolean queries emerge as a module of great importance, as it is possible to track moments when a person carries a cell phone, a laptop, or a backpack. These queries can be of interest in video surveillance tasks.

Fig. 6: Query using the conjunction *with*: *person with phone*.

Fig. 7: Query using the logical operator *AND*: *person AND phone*.

It was of paramount importance to add the synonym query expansion module, as neural network models generally only associate one label with one class. Thus, if a user is unfamiliar with the classes of the model with which the objects were detected, they may not find the objects they are looking for. By using a dictionary of synonyms, we can associate synonyms or similar words to the labels of the object detection model classes. This way, the user can perform searches like *individual* or *human* and obtain the same results as the *person* query. In Figure 8, we can see the result of the *human* query, which is identical to the *person* query presented in Figure 3.

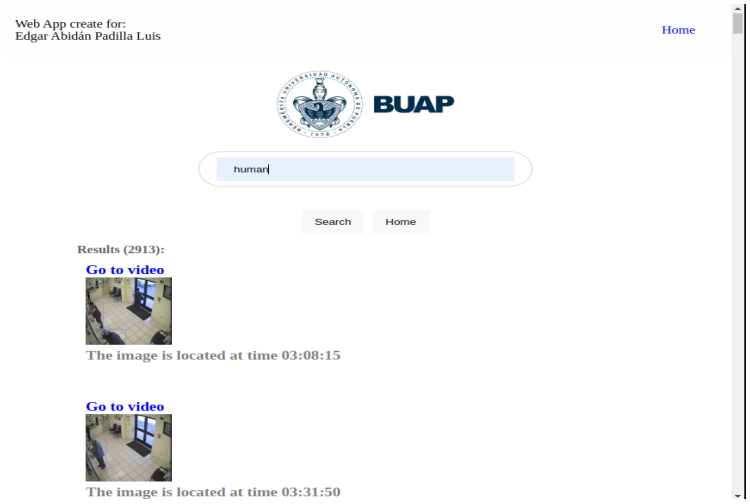

Fig. 8: Query using a synonym of the word *person*: *human*

7 Conclusions

This article presents an information retrieval system for for fast identification of objects in long-duration surveillance videos. To achieve this, a text corpus of object detection model class labels is formed. With this corpus, an information retrieval system based on text retrieval techniques is built, making the system efficient and capable of providing queries in milliseconds that the user can visualize in a user-friendly web application. It has been demonstrated that it is feasible to adapt the system to run in real-time on equipment with limited computational resources. Furthermore, it has been shown that through the use of dictionaries, a non-specialized user can make specialized queries, making the system very intuitive and user-friendly. Finally, it is important to note that in this article, it is explained how the boolean search module can be used for tasks related to video surveillance.

It is worth noting that the methodology used can be adapted to various widely used models that require low resources and are used for various tasks such as facial recognition. This makes this article a basis for a wide range of applications.

References

[1] Akmalbek Abdusalomov, Nodirbek Baratov, Alpamis Kutlimuratov, and Taeg Keun Whangbo. An improvement of the fire detection and classification method using yolov3 for surveillance systems. *Sensors*, 21(19):6519, 2021.

[2] M Adimoolam, Senthilkumar Mohan, Gautam Srivastava, et al. A novel technique to detect and track multiple objects in dynamic video surveillance systems. 2022.

[3] Shay Aharon, Louis-Dupont, Ofri Masad, Kate Yurkova, Lotem Fridman, Lkdci, Eugene Khvedchenya, Ran Rubin, Natan Bagrov, Borys Tymchenko, Tomer Keren, Alexander Zhilko, and Eran-Deci. Super-gradients, 2021.

[4] Gregory Castanon, Mohamed Elgharib, Venkatesh Saligrama, and Pierre-Marc Jodoin. Retrieval in long-surveillance videos using user-described motion and object attributes. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(12):2313–2327, 2015.

[5] Omar Elharrouss, Noor Almaadeed, and Somaya Al-Maadeed. A review of video surveillance systems. *Journal of Visual Communication and Image Representation*, 77:103116, 2021.

[6] Palash Yuvraj Ingle and Young-Gab Kim. Real-time abnormal object detection for video surveillance in smart cities. *Sensors*, 22(10):3862, 2022.

[7] Abdolamir Karbalaie, Farhad Abtahi, and Mårten Sjöström. Event detection in surveillance videos: a review. *Multimedia tools and applications*, 81(24):35463–35501, 2022.

[8] B Vinoth Kumar, S Abirami, RJ Bharathi Lakshmi, R Lohitha, and RB Udhaya. Detection and content retrieval of object in an image using yolo. In *IOP conference series: materials science and engineering*, volume 590, page 012062. IOP Publishing, 2019.

[9] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[10] M Murugesan and S Thilagamani. Efficient anomaly detection in surveillance videos based on multi layer perception recurrent neural network. *Microprocessors and Microsystems*, 79:103303, 2020.

[11] Nipun D Nath and Amir H Behzadan. Deep learning models for content-based retrieval of construction visual data. In *ASCE International Conference on Computing in Civil Engineering 2019*, pages 66–73. American Society of Civil Engineers Reston, VA, 2019.

[12] Francisco Pérez-Hernández, Siham Tabik, Alberto Lamas, Roberto Olmos, Hamido Fujita, and Francisco Herrera. Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance. *Knowledge-Based Systems*, 194:105590, 2020.

[13] Guoping Qiu. Challenges and opportunities of image and video retrieval. *Frontiers in Imaging*, 1:951934, 2022.

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[15] Chien-Hao Tseng, Chia-Chien Hsieh, Dah-Jing Jwo, Jyh-Horng Wu, Ruey-Kai Sheu, and Lun-Chi Chen. Person retrieval in video surveillance using deep learning–based instance segmentation. *Journal of Sensors*, 2021:1–12, 2021.

[16] M Uma, S Abirami, M Ambika, M Kavitha, S Sureshkumar, and R Kaviyaraj. A review on augmented reality and yolo. In *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, pages 1025–1030. IEEE, 2023.

[17] Junwei Xin, Famao Ye, Yuanping Xia, Yan Luo, and Xiaoyong Chen. A new remote sensing image retrieval method based on cnn and yolo. *Journal of Internet Technology*, 24(2):233–242, 2023.

[18] Kun Zhou, Fadratul Hafinaz Hassan, and Gan Keng Hoon. The state of the art for cross-modal retrieval: A survey. *IEEE Access*, 2023.