ISSN 2007-9737

Interest Scenes Retrieval in Long Duration Videos Using Image to Text Codification

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

Benemérita Universidad Autónoma de Puebla,
Facultad De Ciencias de la Computación, Puebla,
Mexico

{edgar.abidan.pl,lopcorp.z, bbeltranmtz}@gmail.com, david.pinto@correo.buap.mx,
cerino_rigoberto@hotmail.com, alberto.reyesp@alumno.buap.mx

Abstract. This article presents an approach for retrieving scenes of interest in long-duration videos through image-to-text encoding. Unlike conventional approaches that often involve the use of neural networks, this method proposes a technique that avoids the use of these complex structures in order to reduce computational resource consumption. Through experiments, the feasibility and effectiveness of this technique are demonstrated, concluding that it is feasible to employ it for multimedia information retrieval, offering an efficient and economical alternative for this task.

Keywords. Information retrieval, scene identification, long-duration videos, image-to-text encoding.

1 Introduction

In recent years, due to the development and creation of better technological devices, there has been an increase in the use of digital cameras, such as cell phones, surveillance systems, specialized cameras, among others, resulting in a large volume of multimedia files.

Extracting information from these massive data volumes presents various challenges, which vary depending on the type of information to be retrieved (video, images, audio, or text). Some of the techniques used to obtain information belong to the field of *machine learning*, with deep learning standing out, as it is capable of automatically extracting features from data [6]. However, despite the growth in artificial intelligence studies, many public and private organizations still rely on human resources to perform specific tasks such as searching for scenes of interest in long-duration videos.

This method of obtaining information is slow and consumes a large number of human-hours, as the process of identifying scenes in videos is difficult and tedious, especially when dealing with large volumes of data.

In contrast, it is observed that image information retrieval techniques help reduce human costs in this task [9]. Currently, the advancement of deep learning provides us with tools to develop multimedia information retrieval systems.

However, if these types of techniques are to be implemented, the following points must be considered: a large corpus is needed to train deep learning models; training these models often takes too long; a specialized model is required; and computational resources are high (RAM, GPU, processor, and disk space).

This document proposes an approach based on text information retrieval to identify scenes of interest in long-duration videos

These techniques tend to require fewer computational resources, making them ideal for adaptation on many servers that store videos.

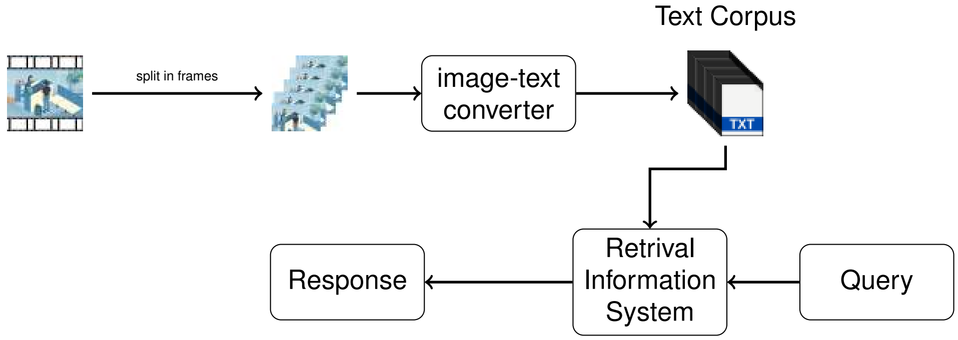

Fig. 1. Proposed methodology for identifying scenes of interest in long-duration videos using an image-to-text encoding module

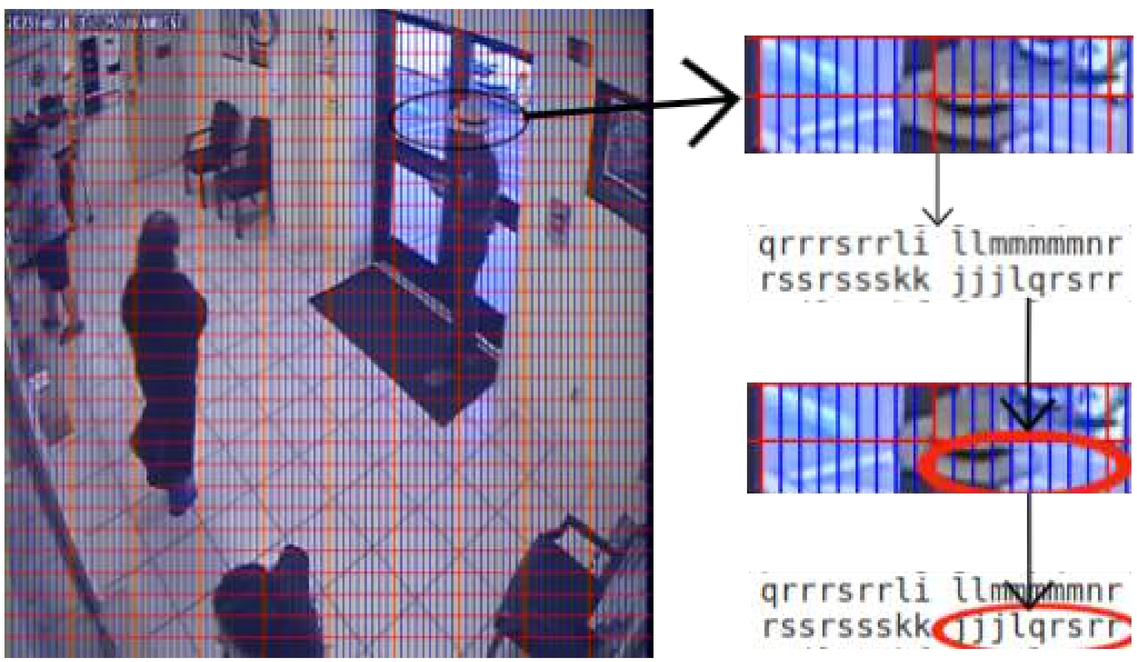

Fig. 2. Analyzing the hat band present in frame 13

2 State of the Art

Multimedia Information Retrieval Systems (MIRS) are closely linked to technological advancements. There are various techniques applied to these systems.

For example, the Bag-of-Visual-Words method involves finding global descriptors of images by creating histograms of high dimensionality. This technique offers the ease of creating inverted indices and vector spaces to measure their distance from other images, but it tends to be less accurate in searches [26]. It has been shown that global descriptors do not capture relevant characteristics well, and local descriptors tend to yield better results [4].

In the past decade, the concept of Vector of Locally Aggregated Descriptors (VLAD) was introduced. It extracts regions using invariant detectors (algorithms for extracting

```
ikkjklllm kmjllmlll jmojklknn opnmkklil nkprpppqq qrrqqpjfg lnnmlkmnn oppppppoo oooonnnmm mmllkkkia
ijkllllno nkhhpopmm hlmkjlmoo oppmmknjl nnrsppppo mrrrsrkfg syyyyywsh ggnmkjkmn ppooooiik lmmmlllja
jkkkjlmnq oiedpnkmm ikmkkmmoo ppqooopkk msssppqp osssrkfg rvvxyyxqh gjpsswxur oopppoljk knnmmmlja
klmljlmmm mkefnoool lmmmmmloo pppqqqqqr rsrsqqqrr qssssrkef psssrrrlg gkompqsws tqppppoo oghiklmka
klmkleefk kkgfkppok lmmmmmmop ppiijnqqq qrrsqqqrr prsssrkef qrrrsrrli llmmmmmnr rqqqqpppp mddiihgga
jffefggjk kknoppppm mmmmmmmmn nkeeehnff gsssqqqrs ssssrqjde rssrssskk jjjlqrsrr qqqqqqqpp jefikjjha
fdeegkkii hloppppm mmmmmmlhk igddfihee erssqqqrs sssrpohee egjlnpshf fggglrrrr rrrqqqqqq gfihikiga
dcegikmhd imooppppn lmmmmmmhi gifffgfee eqrsrqqrr rssrqphdg spnkhggff ggfffkrrr rrrrrrqqo ffiiihhfa
hghjhjlid hnmoooooo nooooolm hkghgfggi gorsrqqrr rsssrqhdh uuuuuplff fffffinpr rrrqpqqqn ifggihga
giigfhjlj hmkmoooop pnkkkmppp nolmlhkjk jmrrrqqqr rrrsrqfdj stttttnff ffffgheem rrrokknqq qqnljhgfa
jigffghgg knooppqqq meeeefnqq pppplnnn noqrrpppq qrrrqofdg mqsssskef eeffottpq rrrollqqq qqqqppnka
jiheecddf noopppqqq hddddiqqq qrrqqqqqq qqqqqqqqq pqqonlgfe edeilogee eeejssssr rrrrrrrrr qqqqqppna
ikjkkfdde moppqqqph dddddegkq rrrrrrrrr rrrrrrrrr pmkhfeffh iihfdddde eemrrssrr rrrrrrrrr qqqqpppna
jkllmkhlj joppqrrid ddddddee iqsrrrsss rrrrrrnhf ffffffff fffhiigfe eegknrrrr rrrrrrrrr qqqqqppna
nillligjl hkqqqrrkd dddddddd gqsssssss ssssssssp lhfffffff fffffeghe eeedddmqr rrrrrrrrq qqqqqppna
llhhkmllk mprrrqrrl ddddddddg sssssssss sssssssss ssqmiffff ffffffeee eegjihqqr rrrrrrrrq qqqqpppna
hmjiknmno opprssrqr fddddddei sssssssss sssssstss ssrrrrnjf fffffffee efeeelqqq rrrrrrrqq qqqppppma
gimopommo oopqqrrrq ofddddddk sssssssrst tttttsrss sssssssrrr njffffffee eefilpqqq qqrrqqpoq qqppppppma
iginmklll opopqqqqq pidddddl rssssssss sssssssss ssssssrss rrrokgffg kossrpppp qqqqqqqpq qppppooma
jigiokjik moooppqqn fdddddddn srrssssss sssssssss srrrsssss srrqqropr ssrrqpppp ppppppqqq ppppooola
kihginnpo ppppooppn iededdddq sssrrssss sssrrrsss ssssrrrrr rsrrrqqqr qqqqqpppq pppoppppp pppooonla
lkljginpp pooooopp pndddddegq srrssrrsr rsrsssrrs sssrrrrrr rsrrrrrqq qrrrqqqqp ooppppppoo ooooonmia
klllkhiop pppppppooo oonliefgl qrrrrrrrr ssssssssr qrrssssssr rrrrrrrrr rrqqqqppp qpppoppoo oonnnmlka
klkkllhhn ppppooooon nmmmmkkon oqqrsssss srssrrrrs srrrrssss rrrrrrrrr rrrqqpqqq qqpppkijm onnmjjkia
jkkklmlhh mpooppppo nnoonmnno poprrssss srqnprsss sssrrrrrr ssssrrqqq rqqqrqqqp plhfglonk iiiihggfa
kkkkllmmi gkppppooo nnmnooooo pppppnnmn qngfffqss ssrrrrrrr sssssrrqq qqqqqqqql gfeeeegil lkiefihha
lkkkklmmm jgioonnno ooonnoooo oolfeeeee eeffffpqr rrssssssr rqrrrrrrr rrqppqohf eeeeeeedd efgkkllja
llkjjkkll lkghlmmmn nooonnnnn oofeedddd eeeeefhpr qrsssssss rqqqqrrrr rqqqqmhfe eeeeeeddd eillkkkja
lllkjjjkk kkkjhlmmm mnnnnnnoo oplfddddd dddddefiq rrqqrrrrr rrrrrqqqr qqppofeee eeeeedddd iihghijga
klllkjjjk kkkkihjlm mmnoooooon opoiifedd ddddddefm qrrqppqrr rrrrrrrqp ppppjknig edddddddd deghkiffa
```

Fig. 3. Textual representation obtained from frame 13

characteristics from an image, robust against geometric transformations) and then characterizes them by SIFT descriptors (Scale-Invariant Feature Transform). The results are classified into clusters, which for information retrieval are associated with a vocabulary [12].

These types of systems are known as Content-Based Image Retrieval (CBIR) systems and are used in various fields, such as medical applications [3]. In recent years, methods for obtaining image descriptors have been improved.

For example, Alsmadi in [1] obtains color, shape, and texture information, and by combining this information, presents results with good accuracies. Some methods additionally create hash codes (bit strings) from the constructed descriptors, which are used to index images and apply IR algorithms based on these indices [16].

However, the rise of artificial intelligence, specifically the use of Convolutional Neural Networks (CNNs), has shown that descriptors can be found almost automatically [24]. Hash codes are used to simplify and accelerate searches; currently, CNNs are trained to obtain these codes [10], and studies are beginning to be conducted with other architectures like transformers to generate these codes [7].

It is worth mentioning that the use of CNNs requires a dataset to match images with text. CNNs have shown the ability to associate images with text and text with images by projecting the text and the image into a single feature subspace [23, 8, 11].

There are various studies focused on the medical field that use CNNs to create specialized Multimedia Information Retrieval Systems (MSIRS). Shamna and Aziz train a magnetic resonance classification model using CNNs and, through a similarity comparison, retrieve related documents that will assist the physician in making a diagnosis [19].

In the same vein, Zhang et al. employ neural networks to classify resonances and match them with text related to medical diagnosis, thus providing specialists not only with retrieved images but also with potential diagnoses [25]. In the

Frame 11 of the video

Frame 20 of the video

Fig. 4. Frames close to frame 13

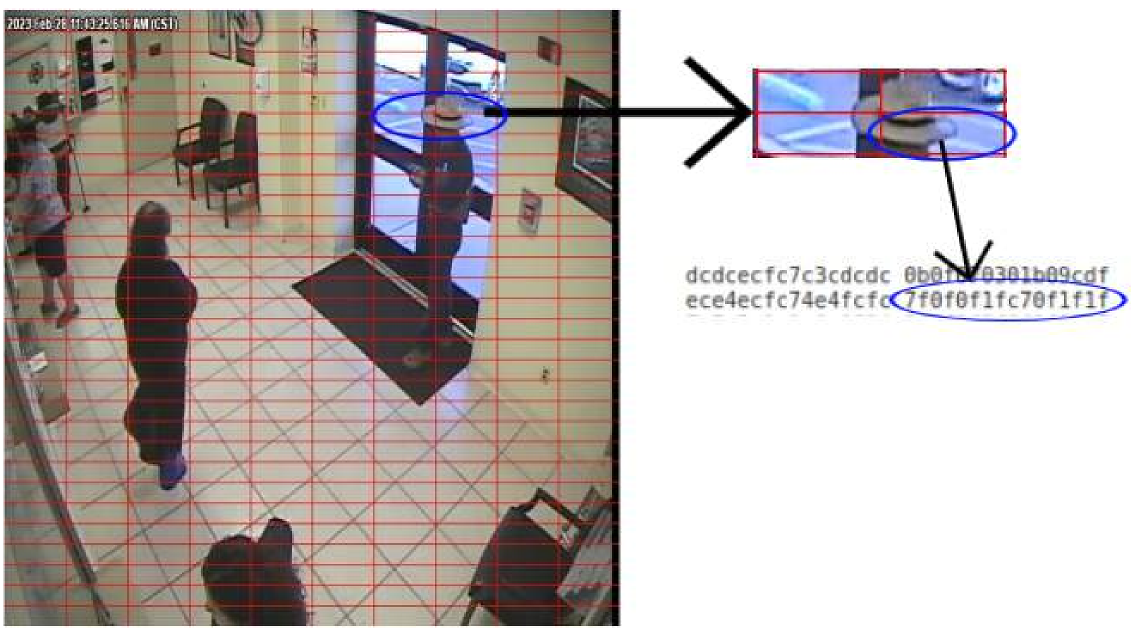

Fig. 5. Analyzing frame 13 with hash code

fashion domain, Whu and Gao introduce the first dataset to support the advancement of image retrieval systems for fashion [23], leading to the development of the first image retrieval models referencing this database [5, 18]. On the other hand, geolocation has gained great importance, from the automation of unmanned vehicles to tourist use.

Fig. 6. Partition of frame 13 in the second experiment

Table 1. Documents retrieved by applying fault-tolerant information retrieval

File	Position	Distance
frame_13.txt	(5, 7)	0
frame_12.txt	(5, 7)	
frame_14.txt	(5, 7)	1
frame_11.txt	(5, 7)	
frame_15.txt	(5, 7)	2
frame_16.txt	(5, 7)	
frame_1176.txt	(10, 1)	3
frame_19.txt	(5, 7)	
frame_1223.txt	(10, 1)	
frame_18.txt	(5, 7)	
frame_17.txt	(5, 7)	

Tang et al. present an information retrieval model that does not rely on a global satellite navigation system.

They use CNNs to predict images of the environment and, through a RIM system, retrieve possible geographic areas in which they are located [22]. In terms of security, video surveillance is of utmost importance. Prathiba introduces a multimodal retrieval system using a promising clustering algorithm for human-computer interaction.

This algorithm extracts features from the frames comprising the video, applies the nearest neighbor-based algorithm, and based on this, calculates the distance between frames to retrieve the most similar images [17]. A quick review of the state of the art highlights that neural networks lead the way in tasks related to multimedia information retrieval [2, 20, 7, 8, 10, 11, 15, 17, 19, 21, 22, 23, 23, 24, 25].

3 Methodology

The methodology proposed in this document is illustrated in Fig. 1. This methodology consists of two main components: firstly, a module is included to generate the text corpus, which is carried out through an image-to-text conversion algorithm; secondly, a text information retrieval system is implemented, allowing text queries to be made and relevant documents associated with the query to be obtained as a response.

3.1 Image-to-text Conversion Algorithm

After obtaining the video frames, a modification to the steps for extracting feature sequences presented by Luo et al. [14] is applied. This algorithm is used to hide information in images; however, a text corpus is created using the following steps: from the video, images are extracted at regular intervals t, referred to as I_t.

Each I_t is divided into $m \times n$ blocks, where each block is part of the set $B_t = \{B_1, B_2, \ldots, B_{mn}\}$. A character string α is assigned to each $b_i \in B_t$ through some mapping f, such that $f(b_i) = \alpha$. This document presents two mappings: conversion to characters by averaging grayscale and image-to-hash code conversion. Thus, each image is represented by a text file, and the generated text files form the corpus that feeds the information retrieval system.

Table 2. Strings with Levenshtein distance less than or equal to 3

File	Position	Distance
frame_13.txt	(5, 7)	0
frame_11.txt	(5, 7)	1
frame_12.txt	(5, 7)	
frame_967.txt	(17, 1)	3
frame_1106.txt	(20, 0)	
frame_1114.txt	(20, 0)	
frame_1113.txt	(20, 0)	
frame_1109.txt	(20, 0)	
frame_1111.txt	(20, 0)	
frame_1115.txt	(20, 0)	
frame_1110.txt	(20, 0)	
frame_1112.txt	(20, 0)	
frame_1107.txt	(20, 0)	
frame_1108.txt	(20, 0)	
frame_501.txt	(18, 4)	
frame_14.txt	(5, 7)	
frame_862.txt	(19, 4)	
frame_500.txt	(20, 4)	
frame_503.txt	(25, 4)	

4 Experiments

For the following experiments, a 5-hour video recording an office at a tax agency in the United States was downloaded from the internet.

4.1 First Experiment: Conversion to Characters by Averaging Grayscale

Based on the idea presented in section 3 for obtaining textual representations, the following process is carried out to obtain a text corpus:

1. A frame is obtained every 1.5 seconds.
2. A partition of each frame is obtained as follows:
   (a) Each image, I_i, is divided into N rows and M columns, forming a matrix of sub-images.
   (b) Each sub-image, $S_{n,m}$, is divided into P elements.
3. Each element, P_p, undergoes a mapping f, where this mapping transforms each element into grayscale values, then the average of the grayscale colors is obtained, and this average is associated with one of the 26 letters of the English alphabet.
4. The letters are joined to form a character string.
5. The document will have N rows with M character strings.

For this experiment, the parameters are as follows: $N = 30$, $M = 10$, and $P = 9$. In Fig. 5, the process for partitioning frame 13 (Fig. 5) obtained from the video is shown. For this purpose, the frame is divided into 30 rows (Fig. 5) and 10 columns (Fig. 5). Subsequently, each sub-image is divided into 9 elements (Fig. 5). This division can be interpreted as a text document consisting of 30 lines, with each line containing 10 words composed of 9 letters.

Once the division process is complete, the mapping f is applied, thus obtaining a textual representation for each sub-image and a text document by concatenating all the representations following steps 5 and 6 presented in this section. The document obtained after applying this process is shown in Fig. 3.

After setting frame 13 as the scene of interest, a manual search was conducted in the nearby frames. As a result, it was found that the scenes from frame 11 to frame 20 are very similar. Observing Fig. 4, it is inferred that the only significant change occurring from frame 11 to frame 20 is the movement of a person.

In frame 13, focus is placed on the hat of the person near the door, specifically on the hat band. Fig. 2 shows this focus, and the associated string for this element is **jjjlqrsrr**. When performing a search in the information retrieval system, it is found that the only file containing the string **jjjlqrsrr** is frame 13, and this string is found on line 5 and column 7. If a search is conducted for the tokens representing this band, it is noticed that this band appears in the same position in frames 11 and 20 (the extreme frames of the manual search). The strings in that position are:

– Frame 11 -> **jjjlqrssr**.
– Frame 20 -> **jkkkoqssr**.

It is observed that the only difference between the string obtained from frame 11 is the letter 's' (highlighted in red), while in frame 20 there is a greater difference in the strings.

If it is considered that the string **jjjlqrsrr** obtained from frame 13 is a word and that this word is correctly spelled, it can be considered that the strings obtained in frames 11 and 20 are misspelled words, and their correction should be the string **jjjlqrsrr**.

In this case, we could apply an algorithm present in fault-tolerant information retrieval, which calculates the Levenshtein distance [13]. For example, the Levenshtein distance between the strings **jjjlqrssr** and **jjjlqrsrr** is 1, as only the letter 's' (highlighted in red) needs to be changed to 'r' in the first string to obtain the second string.

All documents containing tokens with a Levenshtein distance less than or equal to 3 from the token **jjjlqrsrr** were retrieved, resulting in 19 tokens. Table 2 shows the results. Out of the 19 results, only 4 documents are within the desired range.

4.2 Second Experiment: Conversion to Hash Code

For the second experiment, the following steps are followed:

1. A frame is obtained every 1.5 seconds.
2. Each frame is partitioned as follows: Each image, I_i, is divided into N rows and M columns, forming a matrix of sub-images.
3. Each sub-image, $S_{i,j}$, is mapped using function f to obtain a hash code.

Fig. 6 shows the partition of frame 13 into 30 rows and 10 columns. Fig. 7 shows the text document associated with frame 13. Again, the section of frame 13 with a person wearing a hat is analyzed, focusing on the hash code representing the hat's stripe. Fig. 5 shows this image segment to be analyzed.

Searching for the string **7f0f0f1fc70f1f1f** in the information retrieval system retrieves two documents: frame 13 and frame 12. Similar to the previous experiment, the hat stripe appears in that position from frame 11 to frame 20. The corresponding hash codes are as follows:

– Frame 11 -> **5f0f0f1fc70f1f1f**.
– Frame 20 -> **7e0f0f0fc7670f1f**.

Again, if we consider that the strings in frames 11 and 20 are misspelled, we can apply the Levenshtein algorithm. Searching for tokens with a distance less than or equal to 3 from the token **7f0f0f1fc70f1f1f** retrieves 11 results, which are presented in Table 1. Out of the 11 results, 9 fall within the manually found range.

5 Discussion

It is important to consider the following points regarding resource usage and time. For the first experiment:

– 1,237 text files were obtained.
– The process of obtaining the corpus took approximately 15 minutes and 40 seconds, meaning each image took 0.75 seconds to process.
– The total weight of the text files is approximately 3.7 MB.

For the second experiment, the following information is available:

– 1,237 text files were obtained.
– The process took approximately 14 minutes and 32 seconds, approximately 0.70 seconds per frame.

```
070f00f3554d7f00 fcfe001cff27f300 0363016d7b6ff703 e1e0e0e0e4e4e4e4 3f3f3f3f3f3f3c3c f8f8f8f8f8f8fc7c ff7f1f0f07c3e0f0 f0f0f0f8f8fcfe7e f0f0f0f0f0f8f8f8 fcfcfcfcfcfcfcfc
0f0f1f1f3b333347 feffcf8f838f8f8f 677f7f7f37274747 e5e1e1f1f8f8f9fc 3434343232727673 78787878f8f8f8f8 787c7e7e7e7e7e7e 3f1f0f0703637170 fffff9f8f8f8f8f8 fcfcfcfcfcfc7c7c
0f0f1f1f27272727 8f8f8f8f8989898b 07473f3f3f3f3b3b c8e0f0fcfcfeffff 737272737072f0f0 f8f8f8f878787878 fefefefe7c3c1c0c 3c3c3e3f17070f07 7838383cfefeffff 7c7c7cfcfcfcfcfc
27266667677f7f7d 0e8e8f8f8f8f9a96 23030303c3434343 1f1f1f3f1f1f1f2f f2f1f1f1f13133f3 78f8f8f8f8f8f8f8 ccf4f0fcfcfc7c7c 0f7f3f1f03230301 c0c0c0a0a0e0e0f0 fefebe9e9e8e8686
f9f1f1f1f1f2c8c0 ae8e8e0e8f0f0f1f 0303030303030303 ffffcfcfc7c78786 f39393b372737373 f8f878f8f8f8f8f8 dcdcecfc7c3cdcdc 0b0f070301b09cdf a0a02060f078f8f8 82b290989c9c9e9e
e1a181838f8f0f07 0e1e1e3e3e3e3e3e e3e3e3e3c3c2c0c4 c6c6c6c68404c4c4 7373737377777373 f8f8f8f8f8f8f8f8 ece4ecfc74e4fcfc 7f0f0f1fc70f1f1f 6070f0f0f8f8f8f8 9e9e9e9ebebebebe
0f0f070e0e1c1c0d 3e3e3e3e3e3e3e3f fdfdfdfdfcfcf8f8 cccc8c00040c0c0c 7f7f7f7f7f7f7f7f f8f8f8f8f8f8f8f8 7e7e3c1c0c0c0604 1f0f0f0f0f0f0707 f0f0f0f8f8f8f8fc bebe3e3e2c3c3e3c
0e0e1e0e1e1e1e3e 3f3f3f3f3f3f3e3e f8f0f4ecdcbd7d7d 0c0c1830e0c0c0c0 7f7f7f7f7f7f7f7f f8f8f8f8f8f8f8f8 0080c0c0e0e4f0f0 0707070707070707 f8fcfcfefefefefe 3e7e7e7268341058
3e3e3e3e2626ae2e 3f3f3f3f5f5f5f17 1c3c64fcfcff7fff c0c04040c1e7f7ff 7f7f3f3f3f3f3f3f f8f8f8f8f8f8f8f8 f8f8f0f8f8fcfcfc 0707070707030301 fefefefefeefe7e3 682c34361a2aac9c
0e2307c7c3436767 1f1f1f1f0f0f1f1f ffffcfcf87878783 5452d0d0d0c0f1f3 3f3f3f3f3f3f3f3f f8f8f8f8f8f8f8f8 fcfcfcfcfcfcfcfc 090909191191392d e3e3e3e3e3e3e3e3 c0c0e0e0f0f0f8f8
c7c7cec6c480c0c0 0f0f0f0f0f0f1f3f 8787878787070707 f0f0f0f0f0f8ffff 3f3f3b3939397071 f8f8f8f8f8f8f8f8 fcfcf8fcfc7c7838 0d0d0f0f1f0f0f1f e3e3e3e3e3e3e3f7 f8fcfcfefefefefe
f0f1e1e0e0e0c0d0 17171f1f1f1f1f1f 0707070707070707 c0e4e4e7f2f2fa7c b0a0909c2e6ee4f5 f8f8f8f8f8f8f8f0 3c3c1c1c0c0c0000 0f0f1f1f1f1f1f1f e0f0f8fcfcf8f8f8 fcfcfcfcfcfcfcfc
90d0f0f0f8f8f8f9 3f3f3e3e3e3e3e7c 0707030303010101 607830b2a6cf9f9f f0f3f3ebc95c3d3e f3f1e1c0c0808000 8080c0c0c0e06070 1f1f3f3f3f3f3f3f f8787878f8f8f8f8 fcfcfcfcfcfcfcfc
787a787a3a3b3bbb 7e7c7c7c7c7c7c7c 0707030303030101 ff7f7f7f7f3f373f fefcf8f8f8fcfcfc 78f8f8f8f8f8f8f8 303018181c0c0e06 3f3f1f1f0f0f0707 f8f8f8f8f8f8f8f8 fcfcfcfcfcfcfcfc
bbb9b9b1b1b9baba 7c7c7c3c3c3e3e3e 03030303070f0f0f 3f3f3f3f7f7f7e7e ecd696b33b7bfdfc 000000008080c0c0 0606030300000000 0703030703070707 f8f8f8f8f8f8f8f8 fcfcfcfcfcfcfcfc
a888cccccfcfcf4f 3e3e3e3a3e3e3e3f 0001010101818181 7c393bb3a78fcf9f fcfc787a73a7a78f c0c0e0e0e0f0f0f0 fefefefefcfcf8f8 27271717171f0f07 fcfcfcfc7c7c7c7c fcfcfcfcfcfcfcfc
47474747434f4f1f 1f1f1f1f1d1d1d19 0101010181818181 8f2f2f6f67e6f4f4 9f9f9f0f6f6fe7f7 f8f8f8fcfcfcfefe a0a0a0a0c4cecfdf 0f07070707070707 fcfcfcfc78787070 fcfcfcfcfcfcfcfc
1f3f3f3b33313931 1f1f2f0e1e1f1f1f 818181818181c1c1 f1f9f3f3e3e9c9dd f6f0f8f9f9f8f1f5 e2f3f3f2faf8fcf9 0c0c8c80c0c0c0e0 07070f0f1f1f1f3f fffffdfdf8f8f9f9 fcfcfcfcfcfcfcfc
3f3f3f3f17131210 1f1f1f1f1d0b0b03 c1c1c1c181818181 9cbc3e7e7e7b7b7e e4ccdc9ebe3e7e7e f8f2f2f6676f4f1f e0e0e0f0f0f0f8f9 3a3b7b797cfcfcf8 f97c7e3e3ebe9e1e fcfcfcfcfcfcfcfc
1818181818180d89 6f6f4f4f0e8e2e26 8181810101010101 3ebcbcb99b9397c7 3f3fbfbf9ededccc 9f9f3f1f5fcfcfe7 f9e9cfcfdfcfcfa7 b8f0f0f0e0e0e0e0 5f4f6f2727131303 fcfcfcfcfcfcfcfc
0b0f0f0f8f878787 66e6e6f6f7f5f1f3 0101010181818181 cfcfcf8fa7276767 cde9e3e3e7e7e3c3 e7e7f7f2f2f8f8f9 a47072f2f8f8f8fc e2d2d2dab8b83878 ebe3edcdc4c2d298 fcfcfcfcfcfcfcfc
07076747e7a3a323 f3f3e3e3cbd99939 81c1c1c1c1c1c1c1 f7f7f3f39b190818 d39b9939397d7cfc f9f9f0f4e4eeeecf f4fcfafcfcfcf879 70f0f8f8787838ba bcbc3c3c7c7c3cbc fcfcfcfcfcfcfcfc
6373737373535151 3979f9fcfcfcfcf8 e1f1f0f0f0f1f1f1 7d3d393a3636262f fefefefe7f7fbe7e de9e3e3f7f7f3f3f 7b73b3b78787cfcf be9cdcdcc9e9e9e2 9cdcccece8e0e0f0 fcfcfcfcfcfcfcfc
f9f9f9f95959587c fcfcf8f8f2f6e6ee f1f1d39b1f1f7f7f 0f1f1f1f3f3f3f3f 7c7c7d79bbb3b7a7 3f9f9f1f4fcfcfee cf8f8fa727777373 f2f2f2f0e0e9e9cc fffffbf9f9f8f0f0 fefcf6f6f6f2e0e0
7c7c7c7c7c7c7e7e cede5f7f7f7f7e7e 3475f1f9b3b3a7a7 3f1f1f1f1f0f1f0f efcfdfdf9faf4f47 eee6e4f5f1f1f3f3 f8f9f9f8fcfcfcfa ccdcde9ebcb83878 e4c6c6cf8f8f8787 fc78787832b296c6
0e0e1e1e1e3e3f3f 7f7f3f3f3f3f3f3e 8fcf9f9f1f1f4f4f fffffffffdf8f0e0 c7c7c7c783838303 f1f1e5ecccccde9e fcfcfcfcfcfcf8f9 7f7f7efebebebedc 8787830301010000 c4e4e4eececececede
0f07078787878787 3f3f3f1f1b171f1f c3c3c3e3e7e7e7e7 e0e0e0e0e0c0c0c0 0707070707070707 9e3e3e7e7efefefe 797b733337a7a78f fefcfcfcfcfcfcf8 f0f0f0f0f0f0f0f0 ce1e1e1e3e3e3e3e
c7c7c3c3c3c3c3c0 9f9f9f8f8f0f4fcf e7f7f2f2f475717b c0c0c0c0c0c0c0c0 0707030303030303 7e7f7f7f3f3f3f3f cfcfcf8f8fafa727 f8f8f8f8f8f8f8f0 f0f0f0f0f0f0f0f0 3e3e3e7e7e7e7efa
e0e0e1e1e1f1e3e1 8f0f0f2767676727 7373272f0d1d1d3f c0c0e0e0e0e0e0e0 0303030303030301 bf9f9e9edccccdc9 7773f3fbf9f8f8fc f0f0f0f0f0f8f8f0 f0f0f1d8d8e8e060 fefee686840c0c2c
e1e1e0e0f0f0f0f0 e7e7e3e3e36363e1 3d3f1e1e1e1e1f1f e0c0e0f0f0e8e8e8 0303030301010101 e9e36377f7e3e3eb fcfcf8f8f8f8f8f8 f0f4f4f4f4e4e4e4 7c7e3a18088081c0 3c3c3c3c7878583a
```


Fig. 7. Text associated with frame 13 in the second experiment

– The total weight of the text files is approximately 6.3 MB.

This data indicates that the use of computational resources is very low compared to the use of neural networks since it is not necessary to store large amounts of images or use a GPU for training neural network models.

Additionally, the required time is less since not only is the construction of a specialized dataset for the task avoided, but also the training time and adjustment of deep learning models.

When analyzing the results presented in Table 2 (referring to the first experiment where the average grayscale is used as mapping), it is observed that 4 out of the 19 recovered frames correspond to the scene in the video where the hat's stripe appears.

On the other hand, in Table 1 (referring to the second experiment which applies a Hash code as mapping), it is observed that 9 out of the 10 recovered frames correspond to the scene of interest. This suggests that converting image to text using Hash code leads to better performance in the retrieval system.

By using an evaluation metric, it is possible to obtain a more concrete data to assess these two experiments. The precision metric is used to compare their results, and it is given by the following formula:

$$Accuracy = \frac{|\text{relevant docs} \cap \text{recovered docs}|}{\text{recovered docs}}. \quad (1)$$

Considering that there are 10 relevant documents (from frame 11 to frame 20) and taking into account the results from Tables 2 and 1, for the first experiment we have a precision of:

$$\text{Accuracy} = \frac{4}{19} = 0.21. \quad (2)$$

And for the second experiment we have a precision of:

$$\text{Accuracy} = \frac{9}{11} = 0.81. \quad (3)$$

This indicates that using Hash code as mapping yields better results compared to using the average grayscale as mapping.

6 Conclusions

Currently, the development of artificial intelligence greatly contributes to the advancement of technologies. However, it is important to note that not all organizations and individuals have access to the computational resources required for these algorithms, as well as the time invested in this area. Therefore, it is vital to continue researching alternative techniques.

This article demonstrates that it is possible to find parts of an object of interest in an image through information retrieval. However, with the results presented here, it is considered possible to find objects in their entirety and not just a section of them, without the need for deep learning. With the results generated through this research, a possible efficient method for retrieving scenes of interest in long-duration videos is opened without the need for high computational resources.

References

1. **Alsmadi, M. K. (2020).** Content-based image retrieval using color, shape and texture descriptors and features. Arabian Journal for Science and Engineering, Vol. 45, No. 4, pp. 3317–3330. DOI: 10.1007/s13369-020-04384-y.

2. **Butt, H., Raza, M. R., Ramzan, M. J., Ali, M. J., Haris, M. (2021).** Attention-based CNN-RNN arabic text recognition from natural scene images.

3. **Coelho, F., Ribeiro, C. (2010).** Evaluation of global descriptors for multimedia retrieval in medical applications. 2010 Workshops on Database and Expert Systems Applications, pp. 127–131. DOI: 10.1109/DEXA.2010.44.

4. **Deselaers, T., Keysers, D., Ney, H. (2008).** Features for image retrieval: An experimental comparison. Information retrieval, Vol. 11, pp. 77–107. DOI: 10.1007/s10791-007-9039-3.

5. **Dodds, E., Culpepper, J., Srivastava, G. (2022).** Training and challenging models for text-guided fashion image retrieval. DOI: 10.48550/arXiv.2204.11004.

6. **Dubey, S. R. (2021).** A decade survey of content based image retrieval using deep learning. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 32, No. 5, pp. 2687–2704. DOI: 10.1109/TCSVT.2021.3080920.

7. **Dubey, S. R., Singh, S. K., Chu, W. T. (2022).** Vision transformer hashing for image retrieval. 2022 IEEE International Conference on Multimedia and Expo, pp. 1–6. DOI: 10.1109/ICME52920.2022.9859900.

8. **Gordo, A., Almazan, J., Revaud, J., Larlus, D. (2017).** End-to-end learning of deep visual representations for image retrieval. International Journal of Computer Vision, Vol. 124, No. 2, pp. 237–254. DOI: 10.1007/s11263-017-1016-8.

9. **Haering, N., Venetianer, P. L., Lipton, A. (2008).** The evolution of video surveillance: an overview. Machine Vision and Applications, Vol. 19, No. 5–6, pp. 279–290. DOI: 10.1007/s00138-008-0152-0.

10. **Han, L., Li, P., Bai, X., Grecos, C., Zhang, X., Ren, P. (2019).** Cohesion intensive deep hashing for remote sensing image retrieval. Remote Sensing, Vol. 12, No. 1, pp. 101. DOI: 10.1007/s00138-008-0152-0.

11. **Iscen, A., Caron, M., Fathi, A., Schmid, C. (2023).** Retrieval-enhanced contrastive vision-text models. DOI: 10.48550/arXiv.2306.07196.

12. **Jégou, H., Douze, M., Schmid, C., Pérez, P. (2010).** Aggregating local descriptors into a compact image representation. 2010 IEEE computer society conference on computer vision and pattern recognition, pp. 3304–3311. DOI: 10.1109/CVPR.2010.5540039.

13. **Levenshtein, V. I. (1966).** Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady, Vol. 10, No. 8, pp. 707–710.

14. **Luo, Y., Qin, J., Xiang, X., Tan, Y., Liu, Q., Xiang, L. (2020).** Coverless real-time image information hiding based on image block matching and dense convolutional network. Journal of Real-Time Image Processing, Vol. 17, pp. 125–135. DOI: 10.1007/s11554-019-00917-3.

15. **Medina-Cortes, A., Saldaña-Pérez, M., Sossa-Azuela, H., Torres-Ruiz, M., Moreno-Ibarra, M. (2021).** An application of deep neural network for robbery evidence using face recognition approach. Research and Innovation Forum, pp. 23–36. DOI: 10.1007/978-3-030-62066-0_3.

16. **Öztürk, Ş. (2020).** Stacked auto-encoder based tagging with deep features for content-based medical image retrieval. Expert Systems with Applications, Vol. 161, pp. 113693. DOI: 10.1016/j.eswa.2020.113693.

17. **Prathiba, T., Kumari, R. S. S. (2021).** Content based video retrieval system based on multimodal feature grouping by KFCM clustering algorithm to promote human–computer interaction. Journal of Ambient Intelligence and Humanized Computing, Vol. 12, No. 6, pp. 6215–6229. DOI: 10.1007/s12652-020-02190-w.

18. **Saito, K., Sohn, K., Zhang, X., Li, C. L., Lee, C. Y., Saenko, K., Pfister, T. (2023).** Pic2word: Mapping pictures to words for zero-shot composed image retrieval. n Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19305–19314.

19. **Shamna, N., Aziz-Musthafa, B. (2023).** Feature extraction method using HoG with LTP for content-based medical image retrieval. International journal of electrical and computer engineering systems, Vol. 14, No. 3, pp. 267–275. DOI: 10.32985/ijeces.14.3.4.

20. **Suaña-Chambi, J. L., Gutiérrez-Cáceres, J. C., Beltrán-Castañón, C. A. (2022).** Densenet 3d for violent action recognition in surveillance video sequences. Proceedings of the 41st International Conference of the Chilean Computer Science Society, pp. 1–8. DOI: 10.1109/SCCC57464.2022.10000272.

21. **Tahmasebzadeh, G., Kacupaj, E., Müller-Budack, E., Hakimov, S., Lehmann, J., Ewerth, R. (2021).** GeoWINE: Geolocation based wiki, image, news and event retrieval. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2565–2569. DOI: 0.1145/3404835.3462786.

22. **Tang, J., Gong, C., Guo, F., Yang, Z., Wu, Z. (2022).** Automatic geo-localization framework without GNSS data. IET Image Processing, Vol. 16, No. 8, pp. 2180–2195. DOI: 10.1049/ipr2.12482.

23. **Wu, H., Gao, Y., Guo, X., Al-Halah, Z., Rennie, S., Grauman, K., Feris, R. (2021).** Fashion iq: A new dataset towards retrieving images by natural language feedback. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11307–11317.

24. **Yue-Hei-Ng, J., Yang, F., Davis, L. S. (2015).** Exploiting local features from deep networks for image retrieval. Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 53–61.

25. **Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., Langlotz, C. P. (2022).** Contrastive learning of medical visual representations from paired images and text. Machine Learning for Healthcare Conference, pp. 2–25.

26. **Zhou, Q., Wang, C., Liu, P., Li, Q., Wang, Y., Chen, S. (2016).** Distribution entropy boosted VLAD for image retrieval. Entropy, Vol. 18, No. 8, pp. 311. DOI: 10.3390/e18080311.

Article received on 11/04/2024; accepted on 12/07/2024.
**Corresponding author is Edgar Abidán Padilla-Luis.*