IDENTIFYING IMAGES WITHIN VIDEOS UTILIZING OPENCV

R. PALANI
Research Scholar, Department of Computer and Information Science,
Annamalai University, Chidambaram, Tamil Nadu, India

Abstract: The objective of this paper is to employ OpenCV techniques to locate images within video files. This sought-after program is employed in surveillance cameras to identify specific individuals, thereby reducing unnecessary video playback and saving time. It aids in the identification of items in department stores and tracking individuals' movements within the premises.

Keywords: image searching in videos, to find image in videos

1. Introduction

Image searching in videos refers to the process of identifying specific images or patterns within a video sequence. This can involve various techniques such as object detection, feature matching, or deep learning algorithms to locate and recognize desired images or objects within the video frames.

2. Importance of Image Searching

Searching for images is vital for detecting irregularities. It involves carefully examining surveillance videos to maintain continuous observation and reduce time and effort. Yet, this endeavor can be demanding due to the presence of irrelevant footage, amplifying the difficulty. This article explores effective techniques for image searching in videos, outlining three distinct approaches:

3. Image searching types:

- **Manual Scrutiny**: Users can manually watch through the video to identify specific images or scenes of interest. However, this method is time-consuming and not practical for large video collections.
- **Metadata-based Search**: Some video management systems allow users to tag or annotate videos with metadata, including keywords or descriptions. Users can then search for specific images based on this metadata. However, this method relies on accurate and comprehensive tagging, which may not always be available.
- **Content-Based Image Retrieval (CBIR)**: CBIR systems analyze the visual content of images within videos to enable search. These systems use techniques like feature extraction, indexing, and similarity matching to locate images similar to a query image.

CBIR systems can efficiently search through large video datasets based on visual content alone.

4. Methodology

Content-Based Image Retrieval (CBIR) is a methodology used for searching and retrieving images based on their visual content. Here's a rephrased explanation of the CBIR methodology:

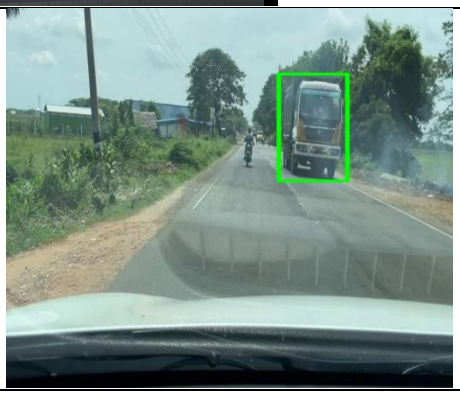
Content-Based Image Retrieval (CBIR) is a technique utilized to search for and retrieve images based on their visual characteristics rather than relying on text-based descriptions or metadata. This approach involves analyzing the pixel values, colors, textures, shapes, and other visual features present in the images. By extracting and representing these features in a numerical format, CBIR systems can compare the visual similarity between query images and the images stored in a database. This allows users to retrieve images that closely match their search criteria, making CBIR an essential tool in various applications such as image search engines, medical diagnosis, surveillance, and digital asset management.

5. Experiment setup

The experiment was initially analyzed in a CPU environment on a laptop; however, it is recommended to utilize a GPU environment for improved performance. Most of the coding was executed and validated in the Colab Research environment; an alternative platform available depending on accessibility. Google Colab offers users access to various NVIDIA GPUs. For this study, processing was conducted on a Tesla T4 processor running Torch 1.10.2+cu111 CUDA:0 with 12 GB of RAM. Files were stored on Google Drive, which has a default storage limit of 15 GB.

6. Result

The identified search images are extracted, and these frames are then copied into an output folder. The search images are delineated within the output frames. The accuracy of this outcome is manually validated and confirmed. Python coding, utilizing OpenCV libraries, is employed for this customized image processing, making it well-suited for such tailored search tasks. The Python script has been shared on a Git repository with public access for future development purposes.

Description	Image / Video
Search Image	
Video File	
Result	 

url: https://github.com/PalaniRamu/Ensemble_detection.git

7. Conclusion

The article demonstrates the successful identification of the search image within the video. The resulting frames containing the search image have been extracted and stored in the output folder. Furthermore, this script can be extended to determine the duration and location of the image within the video session.

References:
https://docs.python.org/3/reference/index.html
https://pythonprinciples.com/reference/
https://www.w3schools.com/python/python_reference.asp