HackVita 2.0 Hackathon Challenge 2024

Video Image Recognition and Duration Tracking

Hillol Pratim Kalita[1], Firoz Anjum Chowdhury[1], and Koyal Borbora[1]

[1]Jorhat Engineering College

March 23, 2024

Abstract

Object detection and tracking in videos play a crucial role in various computer vision applications, including surveillance, robotics, and augmented reality. This project focuses on developing a system for detecting and tracking objects within a video stream using the Scale-Invariant Feature Transform (SIFT)[1] algorithm. The methodology involves extracting keypoints and descriptors from an input image, matching them with video frames using the Brute-Force Matcher(BFMatcher), and identifying occurrences of the object based on the number of good matches. Additionally, the system tracks the duration of each occurrence, providing insights into the object's temporal presence in the video. The system demonstrates robustness to variations in object appearance, scale, rotation, and illumination changes. The results showcase accurate object detection, tracking, and duration estimation, laying the groundwork for further research and development in computer vision.

1 Introduction

Object detection and tracking in videos represent fundamental tasks in computer vision, pivotal for diverse applications such as surveillance, autonomous systems, and human-computer interaction. These tasks involve the automatic identification and monitoring of objects within video streams, enabling real-time analysis and decision-making.

In this project, we embark on the development of a robust system tailored for object detection and tracking in videos. Our approach centers on leveraging the Scale-Invariant Feature Transform (SIFT) algorithm, renowned for its ability to extract distinctive and invariant features from images, making it well-suited for handling variations in object appearance, scale, and orientation. To achieve our objectives, we employ a multi-step methodology. First, we extract keypoints and descriptors from an input image using the SIFT algorithm. These keypoints serve as salient points representing local regions of interest, while the descriptors encode essential information about the surrounding image patches.

Subsequently, we apply Brute-Force Matching (BFMatcher) to establish correspondences between the keypoints and descriptors extracted from the input image and those computed from successive frames in the video stream. This matching process enables us to identify potential instances of the object across different frames, forming the basis for object detection and tracking. A key aspect of our system is its ability to not only detect objects but also track their temporal presence within the video stream. By monitoring the duration of each occurrence, our system provides valuable insights into the spatio-temporal behavior of objects, facilitating more comprehensive video analysis and understanding. Throughout this project, we address various technical challenges associated with object detection and tracking in videos, including occlusions, changes in illumination, and complex object motions. By harnessing the power of feature-based methods, we aim to deliver a robust and reliable system capable of operating across diverse video datasets and real-world scenarios.

Ultimately, our endeavor seeks to contribute to the advancement of computer vision techniques for video analysis and surveillance applications. The developed system will serve as a valuable tool for researchers and practitioners alike, empowering them to tackle complex video analysis tasks with greater efficiency and accuracy.

Figure 1: SIFT descriptor creation and BF-Matcher

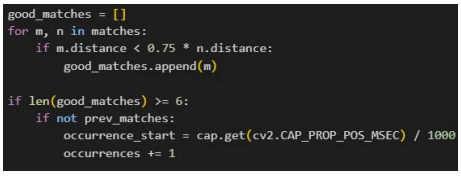

Figure 2: Feature matching

2 Method

- Data Acquisition and Preprocessing:

Input Image: We begin by selecting an input image containing the target object we wish to detect and track within the video stream. This image serves as a reference template for object matching.
Video Data: We acquire the video data containing the object of interest. The video may be obtained from various sources such as surveillance cameras, recorded footage, or simulated environments.
Preprocessing: Prior to feature extraction, we preprocess the input image and video frames to enhance their suitability for analysis. Common preprocessing steps include resizing, grayscale conversion, and noise reduction.

- Feature Extraction using SIFT:

Keypoint Detection: We employ the Scale-Invariant Feature Transform (SIFT) algorithm to identify keypoints within the input image and video frames. SIFT detects stable and repeatable keypoints that are invariant to scale, rotation, and illumination changes.
Descriptor Calculation: For each detected keypoint, SIFT computes a descriptor vector that encapsulates information about the local image patch surrounding the keypoint. These descriptors encode the unique visual characteristics of the keypoints.

- Feature Matching with BFMatcher:

Matching Process: We use the Brute-Force Matcher (BFMatcher) algorithm to establish correspondences between the keypoints and descriptors extracted from the input image and those obtained from successive frames in the video stream.
Matching Criteria: The matching process aims to identify potential matches between keypoints based on similarity measures such as Euclidean distance or Hamming distance. We may employ additional criteria, such as ratio testing, to refine the matches and improve robustness.

- Object Detection and Tracking:

Thresholding: To detect the presence of the object within video frames, we apply a threshold to the number of good matches between the input image and video frames. If a sufficient number of matches are found, it indicates the occurrence of the object.
Duration Tracking: Upon detecting an occurrence, we track the temporal duration of the object's presence within the video stream. We record the start time of the occurrence and monitor subsequent frames to determine the duration until the object exits the scene.

- Implementation and Evaluation:

Programming Environment: The methods are implemented using Python programming language and the OpenCV library, a popular toolkit for computer vision tasks.
Evaluation Metrics: We assess the performance of the system based on metrics such as accuracy, precision, recall, and computational efficiency. Evaluation may involve testing the system on diverse video datasets and benchmarking against ground truth annotations.

3 Scope

Object detection and tracking in videos constitute a vast field of research with a wide range of existing methods and techniques. The scope of this project extends beyond the current implementation to explore scalability and opportunities for further improvement in the following areas:

Scalability of Object Detection:
Methods for enhancing the scalability of object detection algorithms to efficiently handle large-scale video datasets can be explored.
Parallel processing techniques and distributed computing frameworks can be investigated to accelerate object detection tasks in real-time or near-real-time scenarios.

Feature-Based Methods:
Alternative feature extraction techniques beyond SIFT, such as Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB), and Histogram of Oriented Gradients (HOG), can be evaluated.

Deep Learning Approaches:
The applicability of deep learning models, particularly convolutional neural networks (CNNs), for object detection and tracking in videos can

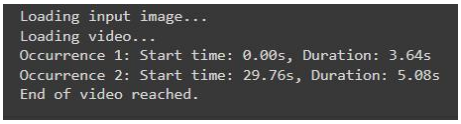

Figure 3: Occurrences and duration

be investigated.
State-of-the-art architectures such as You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD), and Region-based CNNs (R-CNNs) can be explored for improved accuracy and efficiency.

Semantic Segmentation and Instance Segmentation:
Deep learning-based segmentation models such as Fully Convolutional Networks (FCNs) and Mask R-CNN can be investigated for pixel-level object segmentation.

4 Discussion

The utilized method for object detection and tracking in videos is subject to potential errors and limitations that warrant consideration for further improvement. Sensitivity to noise in video frames, limited generalization across diverse datasets, and the arbitrary selection of thresholds for feature matching and detection are notable challenges. Moreover, the computational complexity of the method may hinder its scalability, particularly in processing large-scale video datasets or real-time applications. Assumptions about object appearance, such as static or consistent features, may not hold true in dynamic environments, necessitating adaptive mechanisms for feature extraction and tracking. To address these limitations, future improvements could focus on integrating contextual information, enhancing multi-object tracking capabilities, optimizing the method for real-time performance, implementing adaptive feature selection mechanisms, and adopting incremental learning techniques for continuous refinement of detection and tracking performance. By addressing these challenges and embracing further advancements, the method can evolve to better meet the demands of diverse video analysis tasks.

Conclusions

In conclusion, our exploration of object detection and tracking methods in videos has shed light on both their capabilities and limitations. We have seen how traditional feature-based methods like SIFT, as well as advancements in deep learning approaches such as YOLO and Mask R-CNN, offer valuable tools for detecting and tracking objects in diverse video datasets. However, these methods are not without their challenges, including sensitivity to noise, limited generalization, and computational complexity. By addressing the existing challenges and embracing further advancements in computer vision and machine learning, we can continue to advance the state-of-the-art in object detection and tracking, unlocking new possibilities for applications in fields such as surveillance, robotics, and augmented reality.

References

[1] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.