Department Of Computer Engineering

STES'S SINHGAD ACADEMY OF ENGINEERING
KONDHWA BK, PUNE 411048

2023-2024

"Video Surveillance Application using Object Detection and Tracking Algorithms"

Submitted to the

Savitribai Phule Pune University

In partial fulfillment for the award of the Degree of

Bachelor of Engineering

in

Computer Engineering

By

Harsh Pashine	**COBB041**
Hritiksingh Rajput	**COBB064**
Abhishek Sinha	**COBB088**
Pravin Suthar	**COBA092**

Under the guidance of

Prof P.V. Kulkarni

Sinhgad Institutes

CERTIFICATE

This is to certify that the mini project report entitled **"Video Surveillance Application using Object Detection and Tracking Algorithms"** being submitted by **Harsh Pashine COBB041, Hritiksingh Rajput COBB064, Abhishek Sinha COBB088, Pravin Suthar COBB092** is a record of bonafide work carried out by him/her under the supervision And guidance of **Prof P . V. Kulkarni** in partial fulfillment of the requirement for **BE (Computer Engineering) – 2019 course** of Savitribai Phule Pune University, Pune in the academic year 2023-2024.

Date:

Place: Pune

Guide Principal Head of Department

This Mini Project report has been examined by us as per the Savitribai Phule Pune University, Pune requirements at **SINHGAD ACADEMY OF ENGINEERING** Pune – 411048 on . . . . . .

Internal Examiner External Examiner

ACKNOWLEDGEMENT

First and foremost, praises and thanks to the God, the Almighty, for showers of blessings throughout our project work to complete the research successfully.

We would like to express our deep and sincere gratitude to our subject professor Prof P.V. Kulkarni for giving us the opportunity to do this project and provide invaluable guidance throughout this project. Her dynamism, vision, sincerity and motivation have deeply inspired us. She has taught us the methodology to carry out the research and to present the project works as clearly as possible. It was a great privilege and honor to work and study under her guidance. We are extremely grateful for what she has offered us.

We are extremely grateful to all group members for their dedication and consistency towards this mini project. And also thankful for all the resources which are provided by each group member and which played a very crucial role in the accomplishment of this project.

Harsh Pashine Hritiksingh Rajput Abhishek Sinha Pravin Suthar

CONTENTS

Abstract

This report introduces a novel video surveillance application that utilizes object detection and tracking techniques to enhance the precision and effectiveness of surveillance systems. The system utilizes advanced algorithms to analyze video streams and identify specific objects of interest, such as people, vehicles, and other items. To accomplish this task, the system incorporates various image processing techniques, including edge detection, filtering, and feature extraction, to accurately detect and track objects. The report provides a detailed overview of the system architecture, which includes the hardware and software components utilized, as well as the design of the image processing algorithms. Furthermore, the report presents the outcomes of real-world testing of the system, demonstrating its efficiency in accurately detecting and tracking objects. Finally, the report discusses potential future developments and improvements for the system, such as the inclusion of additional sensors and technologies, to further improve its capabilities.

Introduction

The recognition and localization of specific objects within images or video streams, known as object detection and tracking, have become crucial techniques in the field of computer vision. These techniques have gained significant attention in recent years due to their diverse range of applications, spanning from video surveillance and self-driving cars to robotics and augmented reality. Object detection focuses on identifying and precisely locating objects within a given image or video frame, while object tracking involves monitoring the movement of these detected objects across multiple frames in a video stream.

The successful implementation of object detection and tracking techniques often relies on machine learning algorithms, such as convolutional neural networks (CNNs), which excel at accurately classifying objects and tracking their movements. By training these algorithms with extensive datasets, they can learn the distinguishing features and characteristics of different objects, enabling them to effectively recognize and track objects in real-time scenarios.

This report aims to delve into the fundamental concepts of object detection and tracking, exploring their applications across various domains. Additionally, it will examine different techniques and algorithms employed in object detection and tracking, with a particular emphasis on deep learning-based approaches. Furthermore, the report will address the strengths and limitations associated with these techniques and discuss potential future advancements and challenges within the field.

In conclusion, object detection and tracking play a pivotal role in computer vision, enabling machines to identify and monitor specific objects of interest. By leveraging machine learning algorithms and deep learning techniques, these methods have proven their effectiveness in numerous domains. However, ongoing research and development are essential to overcome existing limitations and push the boundaries of object detection and tracking further.

Problem Statement

The inability to precisely identify and track objects of interest is a limitation of the conventional video surveillance systems that are utilized in a variety of industries, including retail, public safety, and transportation. In order to identify and alert security personnel to potential security threats, these systems rely on manual monitoring or pre-defined rules. This can cause delays and errors in responding to critical situations. Additionally, these systems struggle to deal with complex situations like occlusions, shifts in lighting, and moving backgrounds, which can lead to missed detections or false alarms.

A video surveillance application that makes use of cutting-edge object detection and tracking algorithms to precisely and effectively identify and track objects of interest in video streams is required to overcome these drawbacks.

Motivation

The goal of this project is to create a video surveillance system that can effectively track and identify interesting objects. Traditional video reconnaissance frameworks frequently experience the ill effects of constraints like missed discoveries, phony problems, and defers in answering potential security dangers. By utilizing object identification and following calculations, we might possibly defeat these impediments and work on the precision and productivity of observation frameworks.

Objectives

- To study and learn about Object Detection and Tracking algorithms.
- To implement a video surveillance application using Object Detection and Tracking algorithms.

Hardware and Software Requirements

- PC with Quad Core Processor and 8GB Ram recommended.
- Python 3
- OpenCV

Theory

Object Detection:

Object Detection is a computer technology related to computer vision, image processing, and deep learning that deals with detecting instances of objects in images and videos.

Object Tracking:

Object Tracking is the process of finding objects and keeping track of their trajectories in a video sequence.

Open CV:

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

Background Removal:

Background removal is a fundamental task in image processing that involves separating the foreground objects from the background. It plays a crucial role in various applications, including object recognition, image editing, and computer vision systems.

The goal of background removal is to accurately extract the main objects or subjects of interest from an image while eliminating or isolating the surrounding background. By removing the background, it becomes easier to focus on the foreground objects, analyze their properties, and perform further processing tasks.

Gaussian Blur:

Gaussian blur is a commonly used image filtering technique in image processing and computer vision. It is a linear filter that smooths or blurs an image by applying a Gaussian function to each pixel in the image. The Gaussian function describes a bell-shaped curve that is characterized by its mean (μ) and standard deviation (σ).

The Gaussian blur operation is performed by convolving the image with a Gaussian kernel, which is a matrix or a kernel with values obtained from the Gaussian function. The size of the

kernel and the standard deviation determine the amount of blurring applied to the image. A larger kernel or a higher standard deviation result in a stronger blur effect.

Algorithm:

1. Import all the required libraries – Opencv, numpy, matplotlib
2. Extract Background in Video
3. Processing a Frame
4. Converting Median Sample and sample image to grayscale
5. Performing Absolute difference between gray_frame_sample and gray_frame_median to get the moving objects only, with the background removed.
6. Performing Gaussian Blur for noise reduction and to simplify edge detection.
7. Binarizing the image – Thresholding
8. Creating contours on the threshold frame.
9. Compiling frames together for processing video.

Code and Output

jupyter Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[1]:
```
import cv2
import numpy as np
from object_detection import ObjectDetection
import math

# Initialize Object Detection
od = ObjectDetection()
cap = cv2.VideoCapture("car2.mp4")

# Initialize count
count = 0
center_points_prev_frame = []

tracking_objects = {}
track_id = 0

paused = False  # Flag to indicate whether the video is paused or not

while True:
    if not paused:
        ret, frame = cap.read()
        count += 1
        if not ret:
            break

        # Detect objects on frame
        (class_ids, scores, boxes) = od.detect(frame)

        # Point current frame
        center_points_cur_frame = []

        for box in boxes:
            (x, y, w, h) = box
            cx = int((x + x + w)/2)
            cy = int((y + y + h)/2)
            center_points_cur_frame.append((cx, cy))
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

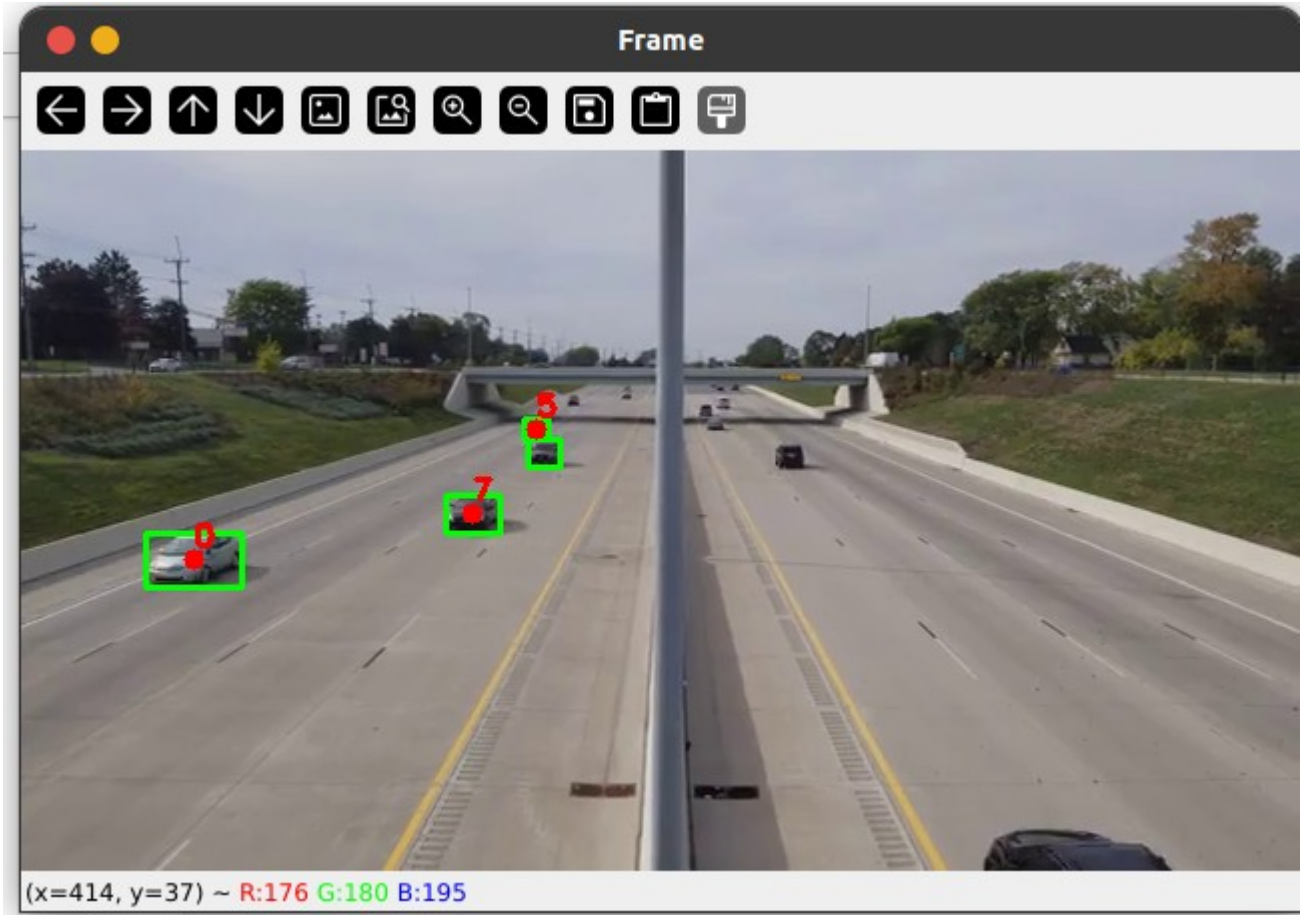
jupyter ip Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

```
        # Object Tracking
        temp_tracking_objects = {}

        for pt in center_points_cur_frame:
            matched = False
            for object_id, pt_prev in tracking_objects.items():
                distance = math.hypot(pt_prev[0] - pt[0], pt_prev[1] - pt[1])
                if distance < 40:
                    temp_tracking_objects[object_id] = pt
                    matched = True
                    break

            if not matched:
                temp_tracking_objects[track_id] = pt
                track_id += 1

        tracking_objects = temp_tracking_objects.copy()

        # Display tracked objects
        for object_id, pt in tracking_objects.items():
            cv2.circle(frame, (pt[0], pt[1]), 5, (0, 0, 255), -1)
            cv2.putText(frame, str(object_id), (pt[0], pt[1] - 7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)

        cv2.imshow("Frame", frame)

    key = cv2.waitKey(1)
    if key == 27:  # Esc key
        break
    elif key == ord('z'):  # 'z' key
        paused = not paused
    elif key == ord('q'):  # 'q' key
        break

cap.release()
cv2.destroyAllWindows()
```

```
Loading Object Detection
Running opencv dnn with YOLOv4
```


Frame
0
3
7
(x=414, y=37) ~ R:176 G:180 B:195

Future Scope

Object detection and tracking are rapidly advancing fields in computer vision with significant potential for future applications. Here are some potential future directions for these techniques:

1. **Real-time Object Tracking:** One of the most significant challenges in object detection and tracking is accurately tracking objects in real-time, especially in complex scenarios. Future research could focus on developing more robust and efficient object-tracking algorithms that can handle complex scenarios such as occlusions, changes in lighting conditions, and moving backgrounds in real-time.
2. **Multimodal Object Detection:** Current object detection and tracking techniques are primarily based on visual data. However, incorporating other modalities such as audio and depth information could potentially enhance the accuracy and reliability of object detection and tracking algorithms, especially in noisy environments.
3. **3D Object Detection and Tracking:** Object detection and tracking algorithms are mainly designed for 2D images or videos. Future research could focus on developing 3D object detection and tracking algorithms that can accurately detect and track objects in a 3D space, enabling more precise localization and tracking of objects.
4. **Unsupervised Object Detection:** Currently, object detection and tracking algorithms rely on supervised learning techniques that require a large amount of labeled data. However, developing unsupervised object detection and tracking algorithms that can learn and identify objects without the need for labeled data could potentially make these techniques more widely applicable.
5. **Applications in Robotics and Autonomous Systems:** Object detection and tracking techniques have significant potential for applications in robotics and autonomous systems. Developing advanced object detection and tracking algorithms could enable robots to navigate and interact with their environments more effectively and autonomously.

Conclusion

In conclusion, the implementation of visual surveillance applications utilizing object detection and tracking algorithms has proven to be highly effective in improving the accuracy and efficiency of surveillance systems. By leveraging these advanced algorithms, it becomes possible to detect and track moving objects in real-time video streams, enhancing the overall surveillance capabilities.The use of object detection and tracking algorithms allows for the analysis of video data and the identification of specific objects of interest, such as people, vehicles, or other items. These algorithms employ various image processing techniques, including edge detection, filtering, and feature extraction, to accurately identify and track objects.The results obtained from testing the system in real-world scenarios demonstrate the effectiveness of object detection and tracking algorithms in accurately detecting and tracking moving objects. This leads to improved surveillance operations, faster response times, and enhanced security measures.In summary, the integration of object detection and tracking algorithms in visual surveillance applications has revolutionized the field, enabling more accurate and efficient detection and tracking of moving objects. With ongoing research and development, we can expect even more advanced and sophisticated systems that will greatly contribute to public safety, transportation management, and security in various domains.

References

- Image Processing Text Book Techknowledge Publication
- Deep Learning Text Book Techknowledge Publication
- https://www.geeksforgeeks.org/
- https://opencv.org/
- https://pysource.com/
- https://www.tutorialspoint.com/opencv/opencv_gaussian_blur.htm
- https://docs.opencv.org/4.x/da/d0c/tutorial_bounding_rects_circles.html