UNIVERSITÀ
DI TRENTO

Department of Information Engineering and Computer Science

Master's Degree in
Artificial Intelligence Systems

Course of
Signal, Image And Video

Human Detection on CCTV Cameras

Professor

Rosani Andrea

Students

Arcangeli Lorenzo

Morandin Marco

Academic year 2024/2025

Contents

1 Introduction

Security cameras play a pivotal role in enhancing safety and monitoring activities in various environments, such as homes, workplaces, and public spaces. However, the continuous recording of video streams often leads to excessive memory usage and challenges in efficiently processing the footage.

This report presents a project focused on leveraging low-level computer vision techniques and machine learning models, such as image filtering, optical flow and Support Vector Machines (SVM), to identify the presence of humans in video footage.

By detecting and isolating relevant human activity, this approach aims to improve the efficiency of security systems in two key areas. First, it enhances security by providing timely detection and analysis of human presence, enabling quicker responses to potential threats or incidents. Second, it optimizes memory usage by selectively storing critical video segments, reducing the need to retain irrelevant or empty footage.

Link to GitHub repository of the project

2 Technical Background

2.1 Python OpenCV

OpenCV (Open Source Computer Vision Library) is a robust and widely used opensource library designed for real-time computer vision and image processing tasks. Some of the key features of OpenCV in Python include: image and video processing, feature detection, object detection.OpenCV is extensively used in applications such as facial recognition, motion tracking, autonomous vehicles, augmented reality, and medical imaging

2.2 Image Filtering

Image filtering is a fundamental technique in computer vision and image processing used to enhance, modify, or extract specific features from an image. By applying mathematical operations to the pixel values of an image, filtering can smooth noise, sharpen edges, detect patterns, or highlight certain regions of interest. Common types of filters include Gaussian filters for noise reduction, Sobel filters for edge detection, and median filters for preserving edges while removing noise.

In the context of security camera footage, image filtering plays a crucial role in preprocessing the video frames, ensuring that subsequent steps like human detection are more accurate and efficient. By eliminating irrelevant details and enhancing key visual cues, image filtering forms the foundation for robust and reliable computer vision systems.

2.3 Image Contouring

Image contouring is a key computer vision technique used to detect and highlight the boundaries or edges of objects within an image. Contours are curves that connect continuous points along the boundary of objects with similar intensity values, effectively outlining their shape. This technique is particularly useful for identifying objects, analyzing their structure, and isolating regions of interest in an image. Contour detection typically involves preprocessing steps like grayscale conversion, noise reduction, and edge detection. Once contours are identified, they can be used for tasks such as object recognition, motion tracking, and segmentation.

In the context of security camera applications, image contouring is invaluable for distinguishing human figures from their background, even in cluttered or complex environments.

2.4 Optical Flow

Optical flow is a computer vision technique that estimates the apparent motion of objects, surfaces, and edges in a visual scene between two consecutive frames of a video. This is achieved by calculating pixel-level motion vectors, which indicate the direction and magnitude of motion at each pixel.

The optical flow technology works under the brightness constancy assumption, which states that the intensity of a pixel remains constant as it moves between frames.

We can establish a direct mapping between pixels to images by employing optical flow computing $delta_x, delta_y$.

Two types of optical flow exist:

- Sparse optical flow: estimates motion only at specific key points
- Dense Optical Flow: estimates motion for every pixel in the image. This type is particularly useful for detecting humans in videos, as it ensures complete coverage of motion across the frame, capturing subtle movements that sparse methods might miss. So, using the dense estimation we obtain a detailed motion map of the scene, highlighting areas of dynamic activity.

Optical flow is a strong technique, but it has some limitations:

- Ambiguity in shadows and lighting changes: Optical flow may incorrectly detect motion caused by shadow movement or sudden illumination changes (due to the brightness constancy assumption)

- If an object (or a person) is not moving the Optical flow will not detect it.

2.5 Background Subtraction

Background subtraction is used for segmenting moving objects from a static or relatively stable background in video sequences. The core idea is to model the background of a scene and distinguish it from new or changing elements appearing in the foreground.

Substantially background subtraction identifies foreground objects by comparing each video frame to a reference model of the scene's background. Any significant deviation from this background model is classified as part of the foreground.

One of the strength points of background subtraction is that the computation is really fast, making it suitable for real time system. On the other side It requires an initial period to build a reliable background model, during which detection may be less accurate.

2.6 Canny Edge Detection

The goal of the Canny algorithm is to identify edges while minimizing false positives and noise, ensuring that only meaningful boundaries are detected.

The algorithm computes the gradient intensity and direction of the image to measure how much the intensity of the image changes at each pixel, providing information about both the strength (magnitude) and orientation of edges. Weak edges connected to strong edges are retained, while isolated weak edges are discarded. This ensures that the final output contains continuous edges and minimizes fragmentation.

The main advantage of edge detection is that it can reduce false positives, but it is really sensitive to the kernel hyperparameter.

2.7 Web-Sockets

WebSockets are a communication protocol that provides full-duplex, real-time interaction between a client (such as a web browser) and a server over a single, persistent connection. Unlike traditional HTTP requests, where the client must repeatedly send requests to retrieve updates, WebSockets enable a continuous, bidirectional flow of data. This makes them highly efficient for applications that require low-latency communication, such as live chat systems, online gaming, real-time notifications, and streaming videos. Once the WebSocket handshake is established over the HTTP protocol, the connection remains open, allowing data to be sent and received without the overhead of repeatedly creating new connections.

3 Methodology

3.1 Human Detection using Contouring

The first approach begins with the conversion of frames to grayscale and resizing them so that the areas occupied by a person can be compared between frames of different sizes.
A background subtraction technique is then applied, where the current frame is compared to a reference background frame, generating a difference mask that highlights regions of change. This mask is further processed using thresholding to isolate significant differences and median blurring to reduce noise while preserving edges. Morphological operations, such as closing, are employed to fill gaps in the detected regions, enhancing the continuity of contours.

Contours, representing the boundaries of significant regions, are extracted from the processed frame. These contours are filtered based on criteria such as aspect ratio, area, and shape to ensure only relevant objects (likely humans) are retained. Bounding boxes are then created around the valid contours, marking potential human detections.

3.2 Human Detection using Optical Flow

This represents the second approach used to detect people within a video. The optical flow technique was used in combination with background subtraction and Canny edge detection. The main component of this approach is the result derived from optical flow, which is then enhanced through the use of the other two techniques mentioned above.

First, the input frames are resized to a standard resolution for consistent processing. In addition, the frames are converted to grayscale to simplify calculations for optical flow and edge detection. These steps aim to reduce computational complexity and the time required to obtain the final results. Moreover, having the same size and proportion for all the input video (and frames) is a crucial aspect. Videos of different proportions mean that the people within them have different proportions. This would make the techniques used useless.

Optical flow is computed to estimate pixel-wise motion between two consecutive frames. Using the Farneback method (dense approach), motion vectors are generated to highlight areas where significant movement occurs. To the generated mask is applied a morphology operation (MORPH_CLOSE) so that we can fill small gaps within motion regions obtaining a more compact region. Then fragmented components that belong to the same moving object, are connecting, improving the continuity of motion regions. Doing that noise is reduced, with small groups of isolated pixels removed. Moreover the detected motions under a certain threshold of magnitude are discarded since they probably represent just noise.

The next step is the background subtraction using the Mixture of Gaussian algorithm, generating the foreground mask.

The third step is the canny edge detection that aims to reduce the false positive. It focuses on structural features, such as the contours of moving objects (e.g. humans), so only regions with distinct boundaries are retained, which helps focus on well-defined objects like people.

The last step before computing the contours, is the combination of the three masks. We get some advantages doing this:

- Boundary precision: Canny edge detection mask, makes the boundaries detected by the Optical flow and background subtraction more sharp improving the accuracy of the future detections

- The combinations is able to reduce the noise (small irrelevant areas) and reduce the false positive

Once the masks are combined, bounding boxes are generated. One problem encountered was that when using optical flow, if a person did not move in its entirety, but only certain parts moved, these were identified as different bounding boxes, even though they were parts of the same person with an area of intersection between the boxes. Therefore, if the value of Intersection over Union for each pair of bounding boxes exceeds a certain threshold, they are merged. The IoU is a metric used to quantify the overlap between two bounding boxes where the Intersection is the overlapping area shared by two boxes and the Union is the total area covered by both bounding boxes.This metric is used since the result is between 0 and 1, so was easier to set the merging threshold, and different threshold can be used for different porpoise.

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

The final step is to filter the bounding boxes. A filter is applied to the generated bounding boxes in order to identify only those that actually detected a person. The filter is based on the proportions that a bounding box should have if it is actually detecting a person. The characteristics examined are: area and aspect ratio (relationship between width and height).

3.3 Combination of Contouring and Optical Flow

This represents the final step in the project pipeline. Here, two operations are performed on the entire set of bounding boxes obtained through the two approaches are performed:

- A check over all the pari of boxes is made to see if it is necessary to merge them together. In this case, the threshold value is set to a higher value since the probability that two boxes with a low level of overlap represent the same person, is limited

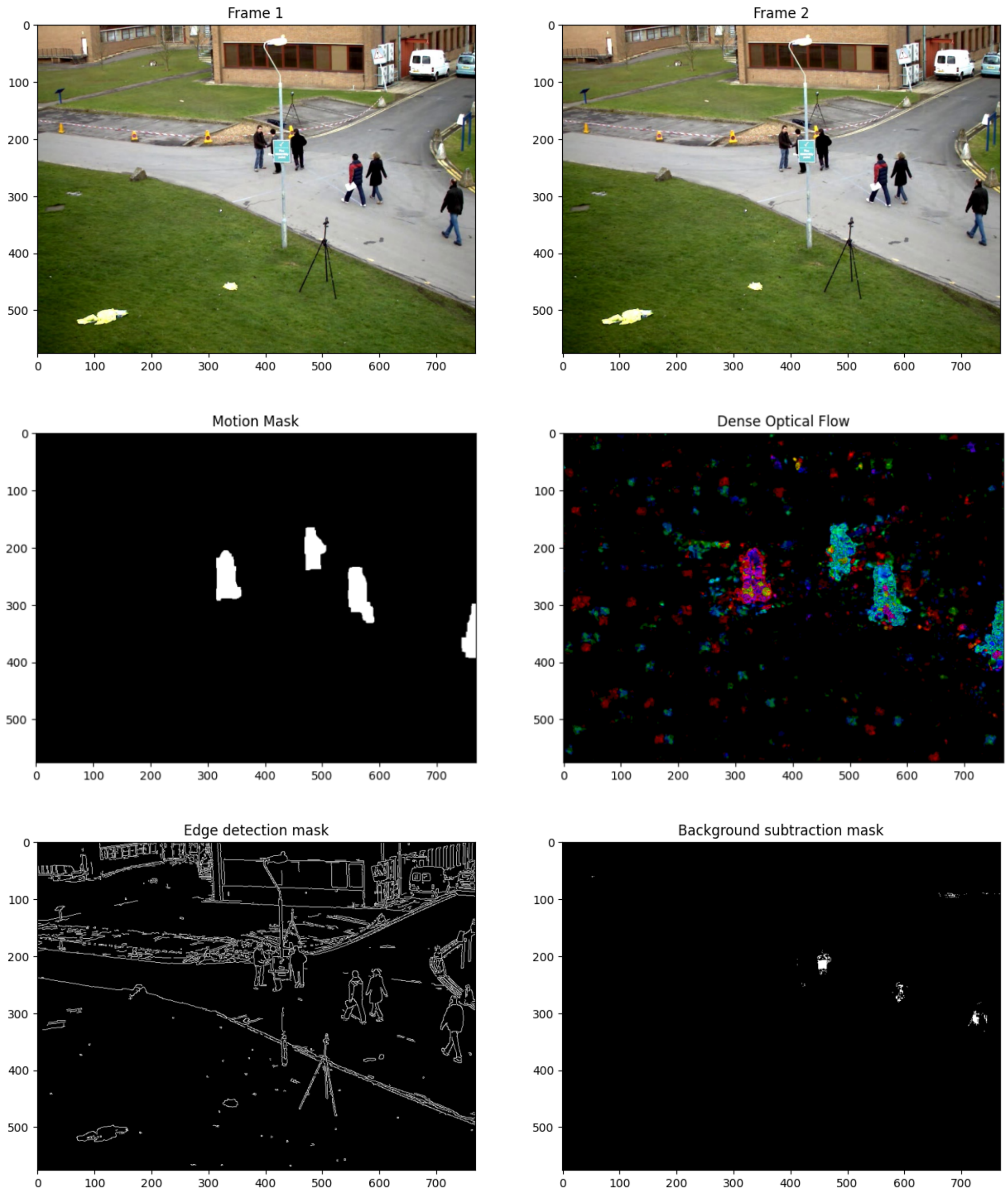
- The second step is filtering. In this case, instead of using the features mentioned above, a classifier is used to determine whether the bounding boxes actually contain a person or not.

After the two operations, the bounding boxes are definitely retained or discarded and the contours are placed in the frame in which they were detected.

3.4 Video Streaming with Web-sockets

In order to simulate video streaming from security cameras, a streaming server based on Flask and Flask-Socket.io is implemented.
This service utilises websockets for client-server communication, enabling real-time interaction. The client informs the server of the desired camera via the websocket, and the server responds by streaming frames and the number of boxes in each frame. The system is designed to support multiple clients concurrently, with the number of concurrent connections dependent on the server's performance, as each client is treated as a separate thread.

4 Use Case



Frame 1
Frame 2
Motion Mask
Dense Optical Flow
Edge detection mask
Background subtraction mask
0
100
200
300
400
500
600
700

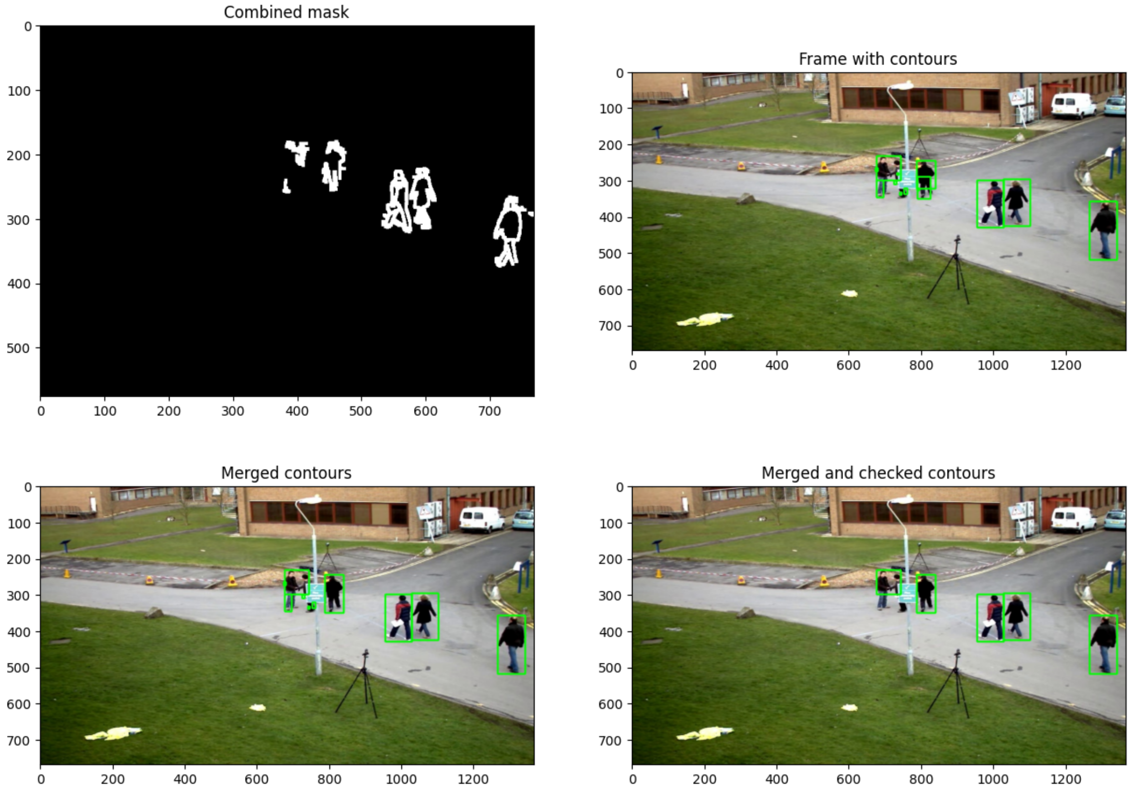
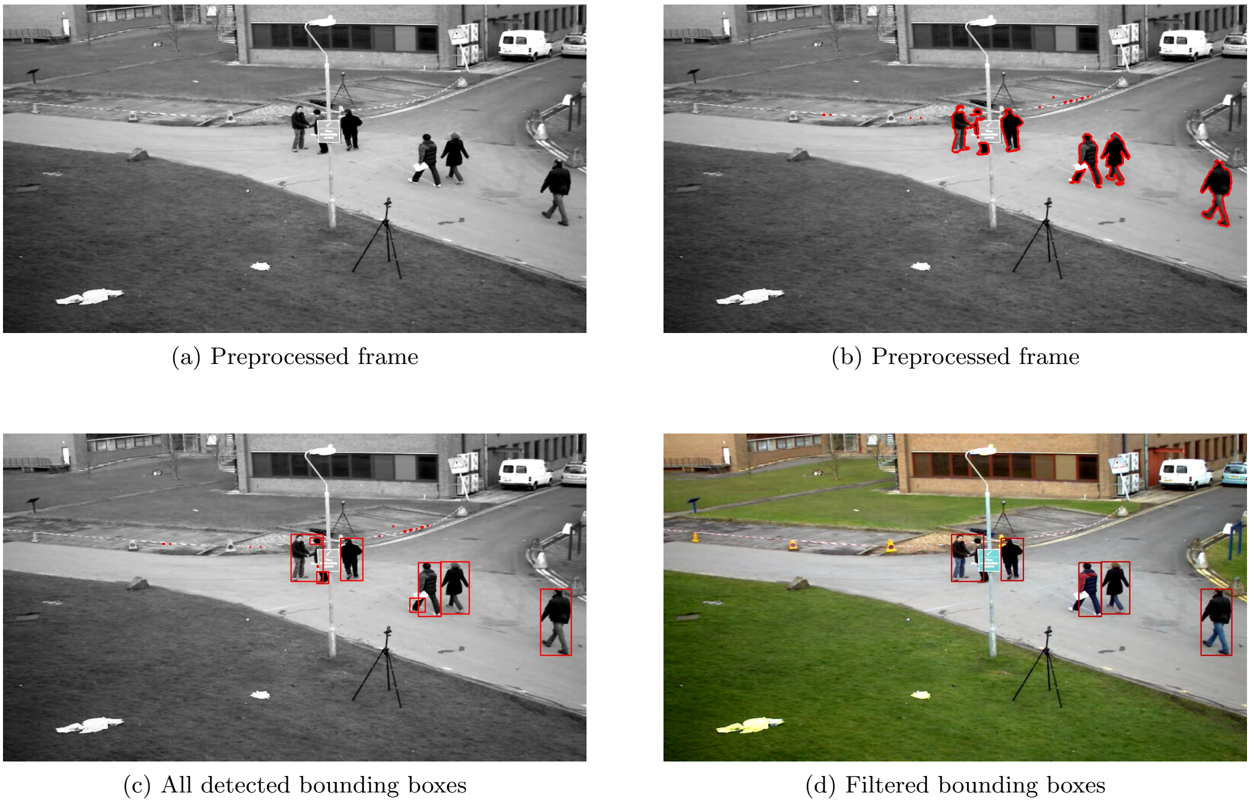


Figure 4.1: Part 2: Optical Flow Approach



(a) Preprocessed frame

(b) Preprocessed frame

(c) All detected bounding boxes

(d) Filtered bounding boxes

Figure 4.2: Image filtering and contouring approach

5 Conclusions

This project has demonstrated the successful implementation of a human detection system for CCTV applications by combining multiple low level computer vision techniques. The integration of image contouring and optical flow analysis, enhanced by background subtraction and edge detection, provides robust detection capabilities for both static and moving subjects. The addition of machine learning-based filtering and WebSocket streaming technology makes the system practical for real-world surveillance applications. While the current implementation achieves its goals of enhancing security and optimizing memory usage, there remains potential for further improvements in classification accuracy and environmental adaptability.