Contribution report for project Scene-Detect

Development of a navigation algorithm

Sai Pavan Bolagani, s.bolagani@oth-aw.de[1]

[1]*OTH Amberg-Weiden, Study programme "Industrie 4.0 Projekt"*

Report written on May 18, 2025

Abstract

A smart home intelligent system is designed to automate, monitor, and assist daily activities, often controlled via sensors, or mobile devices. However, for blind or visually impaired individuals, current smart home solutions lack the capability to detect critical physical situations, such as a person lying on the floor or being immobile. A scene-detection system contributes a practical and extendable method for enhancing smart home safety systems, particularly for visually impaired individuals.This report outlines the design of a privacy-compliant vision-based system to detect physical presence and abnormal postures from a static image.

Keywords: *Smart Home, Visual Impairment, Scene Detection, Privacy, Pose Estimation, MediaPipe*

1. Material

The implementation of this system centers around the core principle of privacy-respecting vision processing. It gets its photos from a regular RGB webcam. However, the project's design makes sure that such sensitive visual data is only stored in system memory and isn't logged, written to disk, or displayed directly.

The image is a single-pass input, meaning it is taken, processed by a human posture estimate network, and then discarded after the extraction of structural data. The structural abstraction component was Google's MediaPipe Pose network [1]. The network uses x, y (2D position), and z (relative depth) information to identify and generate 33 human body landmarks. The landmarks do not feature any clothing, faces, or background information.

OpenCV[2] was used for picture rendering and acquisition, while Python 3.10 was used for the majority of the programming. Everything was depicted using only landmark coordinates and pose linkages on a white canvas. At no point is an original RGB image used as a source. Since the skeletal image is still abstract, it complies with privacy laws.The main component of this arrangement is using Python's del operator to manually delete the image from memory after the landmark has been extracted. This satisfies the GDPR requirements for data minimization by guaranteeing that raw imagery is not saved or maintained in memory after processing. NumPy arrays are used to create blank canvases on which the drawing is done independently.

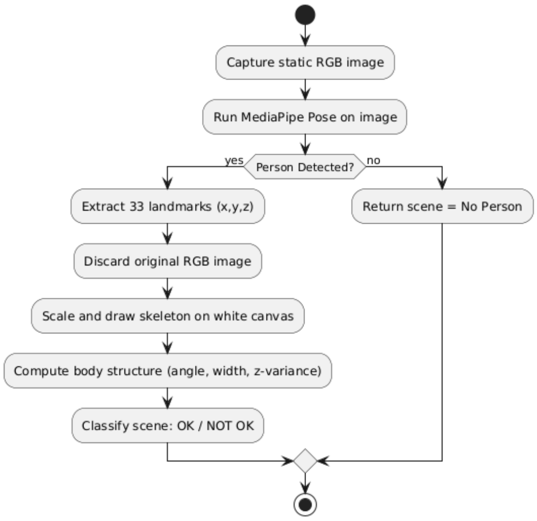

Capture static RGB image
Run MediaPipe Pose on image
Person Detected?
yes
no
Extract 33 landmarks (x,y,z)
Return scene = No Person
Discard original RGB image
Scale and draw skeleton on white canvas
Compute body structure (angle, width, z-variance)
Classify scene: OK / NOT OK

Figure 1. System Flow Diagram
It illustrates the end-to-end processing pipeline

2. Method

The three-way goal of the static scene detection pipeline's design was to balance data privacy, computational efficiency, and structural accuracy. In order to satisfy privacy requirements, the system basically uses a single frame to infer human presence and posture, extract structural positions, and remove raw images. The goal of every pipeline component is to convert unprocessed visual input into anonymous, abstract skeleton representations that can effectively and securely classify scenes. The process comprises

2.1. Acquisition of Frames

Either a camera or a test dataset can provide a static RGB image. The picture is sent to the MediaPipe Pose model right away after being instantly transformed to RGB color space.

2.2. Estimating Pose

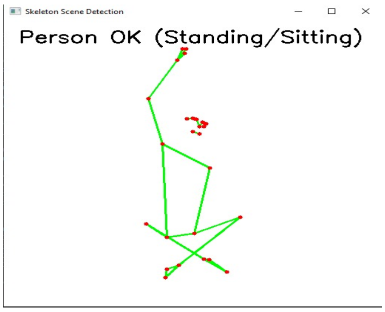
By detecting the coordinates of 33 body joint markers, MediaPipe [1] calculates human posture in static image mode. The landmarks lack any identifying information, but they are normalized both spatially and (relatively) in depth.

2.3. Discarding RGB Images

Once the pose keypoints are recovered, the original image is erased from memory. This is a deliberately planned privacy-enhancing step that is implemented by Python's del command. In order to comply with GDPR-compliant design standards, no visual representation of the input is saved or displayed.

2.4. Abstraction of Skeletons

NumPy initializes a white blank canvas of preset size (512×512). After centering and normalizing landmarks, OpenCV[2] is used to generate joints and lines. As a result, a depersonalized skeleton image with only safely visible geometric structural information is created.



Person OK (Standing/Sitting)

Figure 2. Skeleton Output
The original RGB image has been discarded and replaced with a white canvas containing only the abstracted skeleton,

References

[1] C. Lugaresi et al., *MediaPipe: A Framework for Building Perception Pipelines*, arXiv preprint arXiv:1906.08172, 2019.

[2] *OpenCV Library. https://opencv.org/*