Pedestrian Detection Using Threshold Frequency in Open-CV Python

M. Akshanth Chouhan
Dept of Computer science and Engg
Amrita school of computing
Amrita Vishwa Vidyapeetham
Chennai, India
akshanthchouhan.m@gmail.com

Madhumita K
Dept of Computer science and Engg
Amrita school of computing
Amrita Vishwa Vidyapeetham
Chennai, India
kmadhumita2004@gmail.com

Gowtham Hari S
Dept of Computer science and Engg
Amrita school of computing
Amrita Vishwa Vidyapeetham
Chennai, India
gowthamsengodan7@gmail.com

Kishore N S
Dept of Computer science and Engg
Amrita school of computing
Amrita Vishwa Vidyapeetham
Chennai, India
kishorens2005@gmail.com

Gajjelli Srimaan
Dept of Computer science and Engg
Amrita school of computing
Amrita Vishwa Vidyapeetham
Chennai, India
gajjellisrimaan37@gmail.com

R. Ishwariya
Dept of Science and Humanities and Engg
Amrita school of computing
Amrita Vishwa Vidyapeetham
Chennai, India
r_ishwariya@ch.amrita.edu

ABSTRACT:

In this research, a novel approach to real-time pedestrian recognition in video streams is presented. intended to improve safety measures, urban planning, and intelligent transportation systems. The method includes a multi-stage pre-processing pipeline intended to enhance pedestrian visibility in various scenarios, such as shifting lighting, moving cameras, and obstacles. Important steps include shape analysis to highlight pedestrian outlines, motion detection through interframe differencing, background subtraction for pedestrian isolation, color normalization for consistent extraction of features, contour extraction for accurate boundary identification, and reducing noise through spatial filtering. By taking these preliminary steps, it is possible to accurately recognize and track pedestrians and differentiate them from other moving things like cars and animals. The method is optimized for efficiency and scalability, enabling its application across different hardware platforms, including those with limited processing capabilities. This makes it particularly suited for embedded systems and edge computing devices. The forthcoming evaluation is expected to confirm the approach's effectiveness and reliability for real-world applications in urban management and security. This paper contributes a novel, efficient technique for real-time pedestrian detection promising significant advancements in environmental science applications

Keywords: -
Intelligent Transportation Systems ,Background Subtraction, Shape Analysis, Edge Computing, Video, Stream Processing, Real-Time Pedestrians Detection.

1. INTRODUCTION

This study presents an algorithm designed to enhance pedestrian detection by integrating various computer vision techniques, including grey scale conversion, blurring, dilation, motion detection and background subtraction. Aimed at improving the accuracy of identifying and tracking pedestrians in video streams, this method pre-processes images to highlight features of interest, employs motion detection to isolate moving regions, and uses contour analysis for precise pedestrian delineation. To improve feature extraction, the method first pre-processes the input images by adding blur, converting them to grey scale, and reducing brightness. Dilation techniques are then used to highlight regions of interest to perform threshold detection. By minimizing background interference and increasing the visibility of pedestrian characteristics, this pre-processing step is essential to boosting the accuracy of pedestrian detection. A method that tackles the difficulties brought about by different pedestrian behaviours and ambient circumstances. Systems for detecting pedestrians must be able to discriminate between moving items like cars, bicycles, and animals with accuracy. The method improves the discriminating capability by using sophisticated image processing techniques, which lowers false positives and raises total detection accuracy. Adaptive threshold techniques are also incorporated into the algorithm to account for variations in illumination, guaranteeing dependable performance in both daytime and evening circumstances. The method is also efficient and scalable, which makes it is right for practical implementation on a variety of hardware platforms. The proliferation of inexpensive cameras and embedded computer devices has led to an increase in demand for lightweight, computationally efficient pedestrian detection technologies. To do this, the suggested approach makes use of parallel processing methods, efficient data structures, and algorithmic optimizations. This allows real-time pedestrian detection on systems with limited resources, such as embedded systems and edge computing platforms. The algorithm is well-suited for implementation in a variety of applications, from smart traffic management systems in metropolitan regions to pedestrian safety solutions in rural areas, due to its scalability and efficiency

2. ALGORITHM

1. Converting the initial visuals of into grayscale and then into a blurred one
2. Applying dilation methods to processed visuals and setting visuals for threshold detection.
3. Import required libraries in python.
4. Initializing a motion detector and analysing changes in visuals frame by frame.
5. Using contour library for pedestrian identification.
6. Applying background subtraction to detect motion.
7. Detecting pedestrian count by initializing a virtual line incrementor.
8. Detection ends if video ends.

3. PEDESTRIAN DETECTION AND COUNTING SYSTEM

We develop a system that automatically detects and counts pedestrians in an image or video stream. This system will reliably identify pedestrians even in complex scenes with varying lighting, occlusions, different viewing angles, and different weather conditions. The system will provide an accurate count of pedestrians, avoiding double counting or missing individuals

The mathematical explanation includes

- Background subtraction
- Gaussian blur using a kernel
- Dilation
- Closing
- Contour detection
- Bounding rectangle
- Person counting

Background subtraction:

It is a fundamental technique to separate the foreground object from the background in a video, by subtracting the background image with the current frame. This is done by subtracting the pixel values from each other

The following equation represents background subtraction:

$$If\ (x,y) = Ic\ (x,y) - Ib\ (x,y)$$

If (x, y) = foreground image

$Ic(x, y)$ = coordinates of current frame.

$Ib(x, y)$=coordinates in the background frame.

Background subtraction mathematically calculates the difference in the matrix of the current frame and the background image. It is commonly used in object detection, where changes in the foreground objects are detected against the background.

Gaussian blur using a kernel:

In this method we convolve the image using a Gaussian kernel. The Gaussian kernel involves calculating the weighted average of the pixel values in each pixel's neighborhood.

Gaussian kernel: This method gives a higher weight to the pixel which is in the center and lower weights as it goes farther away from the center.

The formula is for 2-D Gaussian function. The standard deviation σ $c$ontrols the amount of blurring applied to the image, the larger the values larger the extent of blurring.

Gaussian kernel formula:

$$G(x, y) = 1/2\pi\sigma^2 e - x^2 + y^2/2e^2$$

Where:

$G(x, y)$ = value of the Gaussian Kernel at position (x, y)

σ is the standard deviation, controlling the speed of kernel. e is the base of the natural logarithm (e = 2.71828).

π is the mathematical constant pi (π = 3.14159).

X and Y are the distances from the center of the Kenel in the horizontal and vertical directions, respectively.

$I(x, y)$ is the intensity of the pixel at coordinates (x, y) in the input image, and let $G(x, y; \sigma)$ is the value of the Gaussian function at coordinates (x, y) with standard deviation σ.

Morphological operation:

DILATION: In our project, we have added dilation to convolve the image with our structural element, which is a binary element. Dilation is a morphological operation that is frequently used in image processing to expand the borders of objects in a binary image. The following is an expression for dilation. In this case, A is the input binary picture (the binary mask produced by background subtraction), B is the structuring element, and D is the dilated image. The operator ⨁represents the dilation process. Dilation operates practically by positioning the structuring element's origin at each pixel in the input image and determining if any of the structuring element's pixels overlap with the input image's foreground pixels. In the event of an overlap, the associated pixel (often denoted by a value of 1) in the output image is set to foreground The limits of the items in the foreground are enlarged by this technique, which is repeated for each pixel in the input image. If there is an overlap, the corresponding pixel in the output image is set to foreground (usually represented by a value of 1).

By repeating this process for every pixel in the input image, the boundaries of the foreground objects are expanded, effectively making them larger. The output image is a binary image which is larger than the input image.

Closing:

We apply closing to the dilated image, to close small holes or gaps in the foreground objects. This is achieved by the method of erosion. Erosion in morphological image processing is a mathematical operation that systematically reduces the size of foreground objects while preserving their overall shape. Mathematically, erosion is defined as the intersection of the input binary image with the translated structuring element, where the structuring element is traversed over the image to check for complete overlap with the foreground pixel at each position.

$$A\O B = \{z|(\hat{B})z; \subseteq A\}$$

Here's a breakdown of the mathematical components:

- Ais the input binary image.
- B is the structuring element (kernel).
- B is the reflection of the structuring element about its origin.
- $(\hat{B})_z$ represents the translated version of B by vector z.
- Ø denotes the erosion operator.

This involves considering all possible positions of the origin of the structuring element relative to the image. At each position, the intersection operation is applied to determine whether all the white pixels in the structuring element overlap with white pixels in the image. The resulting eroded image contains only those positions where the structuring element entirely overlaps with the foreground region in the input image.

Contour detection:

Contours are detected in the processed image using a contour detection algorithm. We have used a variation of Moore-Neighbor Tracing algorithm, also known as the contour tracing algorithm, for contour extraction. Our model uses cv2.findContours(). The open cv prefers Moore's more than Suzuki's. Thus, we ended up using Moore's as we are using only defined contours, Suzuki's is preferred for more complex irregular objects.

Bounding Rectangles and Centers:

Bounding rectangle is used to encapsulate the contours for further analysis. The center of the rectangle can be found by calculating the midpoint of its width and height.

$$Center\ x = x + \frac{w}{2}$$

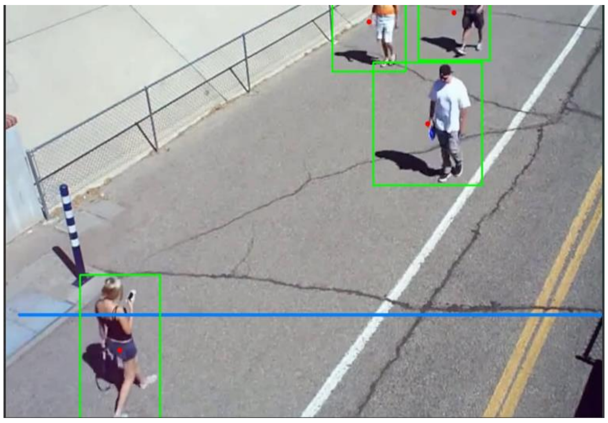
$$Center\ y = y + \frac{h}{2}$$

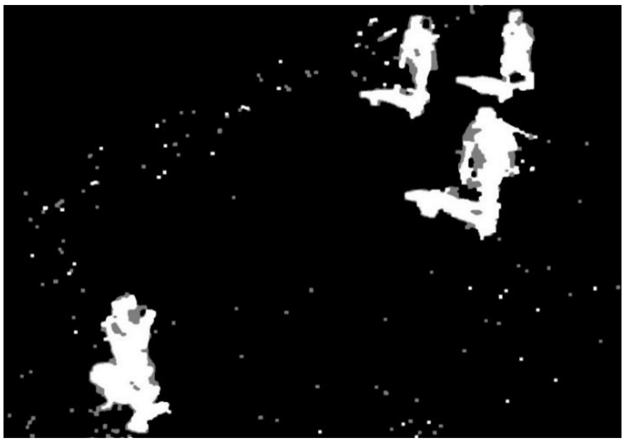
Where $(Center\ x, Center\ y)$ are the midpoints of the encapsulated rectangle. (x, y) are the coordinates of the top-left corner. The width and height of the bounding rectangle are represented by w and h, respectively. By calculating the center of the bounding rectangle, we can determine a representative point for the contour, which we used for object tracking.

Line intersection:

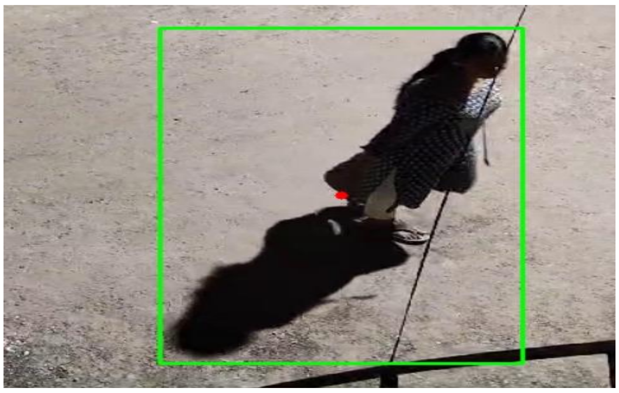
Check if the Y-coordinate of the center of each detected contour falls within a specified range around the line position.

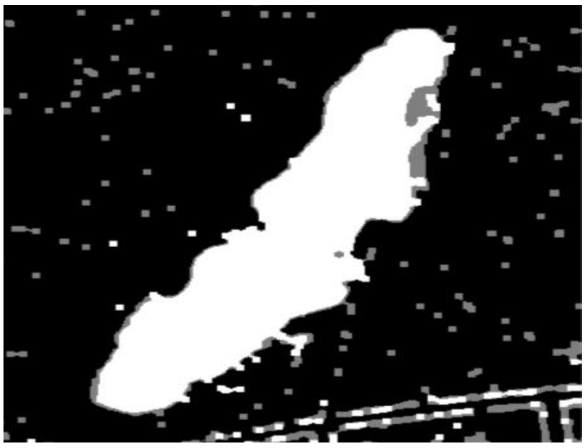
This involves comparing the Y-coordinate of the center with the Y-coordinate of the line position, considering an offset for tolerance.





This is the test case for Pedestrian detection in the Morning time





This is the test case for Pedestrian detection in the Night time

4. APPLICATIONS

Pedestrian detection and counting systems find use in various areas:

- Traffic management: Optimizing traffic light timing and monitoring pedestrian flow in busy areas.
- Urban planning: Understanding pedestrian movement patterns to improve infrastructure design.
- Retail analytics: Tracking customer traffic in stores to optimize store layout and marketing strategies.
- Security and surveillance: Monitoring public spaces for crowd control and anomaly detection.

5. CONCLUSION

Summing up, the algorithm presented in this paper provides a strong solution for improving pedestrian detection in video streams. By combining different computer vision methods such as grayscale conversion, blurring, dilation, motion detection, and background subtraction, the algorithm shows notable enhancements in accuracy and dependability. Its capability to effectively deal with various pedestrian actions and environmental situations, while reducing incorrect identifications and adjusting to different lighting conditions, makes it a versatile tool for real-life use. Furthermore, the algorithm's effectiveness and scalability are major benefits, enabling easy application on different hardware systems, starting from embed.

The algorithm is able to work well in different situations, from busy city streets to quiet rural areas. This shows that it can be used in many different ways. It is also very quick and can be used on a variety of devices, from small chips to powerful computers. This means it can be used in many different projects to improve safety for people walking and to help plan cities better.

6. REFERENCES

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detectionand people-detection-by-tracking. In CVPR, 2008.

[2] N. Dalal. Finding People in Images and Videos. PhD thesis, InstituteNational Polytechnique de Grenoble, 2006.

[3] Dollar P, Wojek C, Schiele B, & Perona P. (2009). Pedestrian detection: A benchmark.

[4] N. Dalal and B. Triggs. Histogram of oriented gradient for humandetection. In CVPR, 2005.

[5] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection andtracking from a moving vehicle. IJCV, pages 41–59, 2007.

[6] Walk S, Majer N, Schindler K, & Schiele B. (2010). New features and insights for pedestrian detection.

[7] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented

[8] histograms of flow and appearance. In ECCV, 2006

[9] [8] C. Wren, A. Azarhayejani, T. Darrell, and A.P.Pentland. Pfinder: real-time tracking of the human body. (2004)

[10] Y. Benezeth, Pierre-Marc Jodoin, Bruno Emile, Helene Laurent, and Christophe Rosenberger "Comparative study of background subtraction algorithms,"

[11] Massimo Piccardi. Background subtraction techniques: a review, 2004