

MEASUREMENT AND ANALYSIS OF PEDESTRIAN TRAFFIC CHARACTERISTICS IN REAL TIME

Ireneusz Celiński^{1,*} , Tadeusz Opasiak^{2,*} 

¹ Silesian University of Technology, Faculty of Transport and Aviation Engineering, Krasińskiego Str 8, 40-019 Katowice, Poland, e-mail: ireneusz.celinski@polsl.pl, <https://orcid.org/0000-0002-9253-0994>

² Silesian University of Technology, Faculty of Transport and Aviation Engineering, Krasińskiego Str 8, 40-019 Katowice, Poland, e-mail: tadeusz.opasiak@polsl.pl, <https://orcid.org/0000-0002-0777-2316>

* Corresponding author

Reviewed positively: 16.09.2025

Information about quoting an article:

Celiński I., Opasiak T. (2025). Measurement and analysis of pedestrian traffic characteristics in real time. *Journal of civil engineering and transport*. 7(3), 199-217, ISSN 2658-1698, e-ISSN 2658-2120, DOI: [10.24136/tren.2025.012](https://doi.org/10.24136/tren.2025.012)

Abstract – This article addresses the issue of measuring and analyzing pedestrian traffic characteristics in real time. The proposed methodology employs standard, low-cost web cameras, which are temporarily installed at selected pedestrian crossings in Katowice. In subsequent stages of the research, cameras will be deployed at signalized pedestrian intersections, with their locations optimized according to the adopted measurement procedures. In this context, the spatial positioning of the camera is a critical factor that affects the quality of data and the quality of the measurement. The primary objective of the measurements presented is to analyze pedestrian traffic dynamics within the framework of a conceptual model describing the interactions between pedestrian and vehicular flows, as well as interpersonal interactions among pedestrians at intersections, with and without traffic signals. Here are some concepts for using such a model. The theoretical model will be elaborated in future publications. The present article focuses on the development and implementation of the real-time automatic data acquisition method that supports this modeling approach. Real-time data collection was conducted using computer vision techniques implemented with the OpenCV library and the Python programming language. The software system operates on the Windows platform, but can be run on any platform: Unix, MacOS. In the analysis, a comprehensive set of traffic parameters was employed that accounts for both the spatial and functional characteristics of the observed environment, as well as the behavioral patterns of pedestrian movement.

Key words – traffic, pedestrian crossing, OpenCV, vision technique, attractiveness, POI

JEL Classification – C88, C93, L90

INTRODUCTION

This article addresses the issue of measuring and analyzing pedestrian traffic characteristics in real time at intersections with and without traffic signals. Measurements were made using standard low-cost web cameras under \$100 (e.g. Logitech 922E, chosen for its durable construction), and very cheap Creative Live Cam Sync 1080p V2 under \$50. Both were placed in locations that were available during testing (the camera was mounted on a photographic tripod). Ultimately, cameras will be installed at signal-controlled pedestrian intersections in locations optimally selected based on the computational procedures employed (mounted high on poles). Generally, it is convenient to install cameras at intersections with traffic lights when booms are present.

The purpose of the measurements is to analyze pedestrian traffic in the context of a model that describes interactions between pedestrian and vehicular traffic at various types of intersections, including roundabouts. This model accounts for such interactions both at intersections with traffic signals and at unsignalized crossings. In this article, the preparation of data for the model is presented. To measure pedestrian traffic characteristics, computer vision techniques implemented using the OpenCV library were employed [1]. Due to the complexity of the problem, a variety of computational techniques from different areas of this library were used, including processing of images recorded in the infrared. Infrared testing at certain times of the year can provide additional information.

The authors intend for this method to extract information about pedestrian movements at intersections-

distinguishing pedestrian types, direction of movement, speed, etc. In the future, the system will also provide possible exact demographic data. This is possible for high resolutions from FHD upwards. Therefore, this represents a highly detailed measurement approach that requires the application of multiple diverse procedures. In measurements based on processing moving images, the OpenCV library is frequently used across a wide range of domains, including production, road traffic, aviation, and railway systems. It is an open-source computer vision library that encompasses numerous procedures that enable advanced processing of moving images, from image acquisition, filtering, and thresholding to sophisticated computational algorithms. The primary objective of employing these procedures is to process images in real time and extract useful information. The effectiveness depends on both the procedures applied and on the computational power of the computer on which the images are processed. If the target system is intended to operate in real time, for example, to support procedures for calculating traffic signal timings, processing delays must be minimal, usually measured in fractions of processor cycles. OpenCV was developed by Intel and is well supported by the user community. The library supports implementation in multiple programming languages, including Python, C++, and Java, making it a cross-platform tool compatible with UNIX, Windows, and other operating systems.

Computer vision techniques emulate natural vision, allowing computers to interpret what is visible in an image, that is, to understand and classify the perceived scene [1–3]. From this interpretation, specific actions can be triggered that update various systems (e.g., changing traffic signal parameters, opening a dam sluice gate, raising a barrier, etc.). Image processing encompasses many different tasks, such as face recognition, motion detection, object detection through background subtraction, edge detection, and more. It is difficult to enumerate all possible applications, which are often original and highly specialized. Pedestrian detection is fundamental to autonomous vehicles (AV), and OpenCV can be used to achieve this. Properly implemented pedestrian detection procedures from the OpenCV library can be used to detect and track people in images and video streams. Every video stream is, in fact, decomposed into individual frames. In this article, two pedestrian detection methods were applied. Both methods are built into OpenCV and rely on the HOG (Histogram of Oriented Gradients) model combined with a linear SVM (Support Vector Machine) classifier, enabling pedestrian detection in images and video streams. The advantage of the procedures is that although they are only moderately effective in recognizing pedestrians, they are simple to implement, quick to operate and proven over 20 years in numerous applications. Procedures verified on a normal image and on an image recorded using an infrared camera. HOG is based on feature descriptors, which are representations of an image or image region that simplify the image by extracting useful information and discarding irrelevant data. Typically, a feature descriptor converts a three-dimensional image into a feature vector of length n . The HOG descriptor can be computed at multiple scales. In this method, gradient orientation distributions are used as features. Image gradients are useful because their magnitudes are larger around the edges and corners of objects undergoing dynamic changes. These are areas of the image with sudden variations—such as a moving car, a pedestrian, or a bird, in short, any moving object. Literally anything that moves or appears to move. It is important to emphasize that edges, especially corners, carry significantly more information about the shape of an object than flat regions [4–8]. The second method used in this study is based on Haar characteristics. Haar features represent visual scene characteristics captured by imaging devices and are used in object recognition. They were used in the first real-time face detector [9–13]. Object detection using Haar feature-based cascade classifiers was introduced by Paul Viola and Michael Jones two decades ago. In this approach, the cascade function is trained on a large set of positive and negative images. This is an example of machine learning (ML). The trained classifier is then used to locate specific objects (faces, eyes, license plates, hands, etc.) in other images. In other words, a tool is derived from the training images that can classify new images according to the specified features. To train the classifier, you can use the Cascade Trainer GUI which is available for download from: http://amin-ahmadi.com/cascade-trainer-gui/#google_vignette. The algorithm requires a training process involving sets of positive images (e.g., body, number plate, eyes, ears, or faces) and negative images (images that do not depict the targeted features, ideally containing objects lacking those features). Negative should have nothing to do with this, he presents a positive image. Essentially, it works in a manner like the human brain, which tends to find faces even with low similarity due to training. Likewise, Haar classifiers may sometimes detect non-existent features, just like the human brain. This is part of the classification process. The next step is to extract the Haar features. A feature is a single value obtained by subtracting the sum of pixel intensities. Haar features consider neighboring rectangular regions within a detection window (sampling parts of the processed scene), the sum of the pixel intensities within each region, and compute the difference between these sums. This difference is then used to categorize the image patches. Fundamentally,

the method is based on transitions and contrasts. For example, in the case of human face recognition, certain general observations can be applied. The eye region is usually darker than the cheek/forehead area, although this depends on variables such as directional lighting, color, and 3D spatial configuration. A Haar feature for face detection may consist of a pair of rectangles positioned over the eye and cheek regions. The positions of these rectangles are defined relative to the bounding box of the target object [9–13]. During detection, a target size is slid across the input image and for each subregion, the Haar feature is calculated. This value is then compared to a learned threshold that separates objects from nonobjects. Because individual Haar features are weak classifiers (only slightly better than random guessing), many of them are needed to effectively describe an object. Consequently, Haar features are organized into cascades of classifiers to strengthen the overall classification result.



Fig. 1. Haar classifier illustration: a) source material, b) positive image, c) principle of classification. Source: elaboration own

The main advantage of Haar like features, despite their relatively weak object-recognition performance (a high number of misclassified images or, more precisely, subimages) - is computational speed. Public availability of classifiers is also important. A Haar like feature of any size can be computed in constant time, requiring only a few dozen processor instructions for a two-rectangle feature. A Haar like rectangular feature can be defined as the difference between the sums of intensities in regions within a rectangle that can be located at any position and scale within the original image [9–13].

1. MATERIALS & METHOS

To develop a method to measure and analyzing pedestrian traffic characteristics at road crossings, two computational procedures were applied, based on the HOG and Haar algorithms implemented in the OpenCV library (along with many others). Another popular technique is pedestrian extraction from the background, which is also simple and effective. These two methods were specifically used to detect pedestrians of various types. Other procedures, such as contour recognition, were used to detect bicycle and vehicle traffic for PWSN (Pedestrian Wireless Sensor Network).

The study used two USB webcams. Both runs, among other things, Windows 11. The first camera, Logitech C920e, is a work-from-anywhere HD 1080p webcam that exceeds the video and sound quality provided by most laptops. The built-in HD autofocus ensures clear video. Working with automatic light correction. She has fixed 78 ° field of view captures. Second-camera Creative Labs Live! Cam Sync 1080P V2 Webcam 2 is a full HD wide-angle video. The wide field of view of 77° allows the capture of pedestrian crossings with minimal optical distortion. The camera can rotate 360° horizontally and tilt 30 vertically and comes with a universal mounting bracket. But most importantly, both cameras are cheap.

Figure 2 illustrates the application of Haar features for pedestrian detection. The study tested multiple pedestrian crossings on various streets in Katowice, Poland. However, this article presents results based on pedestrian detection at two intersections located in the city center, which are heavily trafficked by both vehicles and pedestrians: Mickiewicza Street (an intersection that now functions as a pedestrian zone intersecting Stawowa Street; hence this case is referred to as Stawowa Street) and Moniuszki Street.

The study used video recordings of pedestrian crossing scenes captured in full HD resolution. The camera was mounted on a tripod at a height of 1.80 meters. From the attached screenshots illustrating the operation of the Haar method, especially in Figure 2c, it is evident that not every pedestrian-type object is detected using the cascade classifier. This is normal when it results from certain features of the images that were used to build the classifier saved in the XML file. In addition to pedestrian detection, Haar features were also used

Measurement and analysis of pedestrian traffic characteristics in real time

to detect faces (described later in this section). Roadside advertisements (with people's faces, as shown in Figure 2b) are often placed near intersections, and these cause false detections. Not only advertisements but also building facades or any other objects with similar visual characteristics can result in such false positives. An example is shown in Figure 2b, where an advertisement that shows images of people is mistaken for actual pedestrians, although it is not. An intermediate solution is to introduce the extraction of the traffic scene area at the pedestrian crossing and subtract the background from the traffic scene, which increases the accuracy of pedestrian identification.

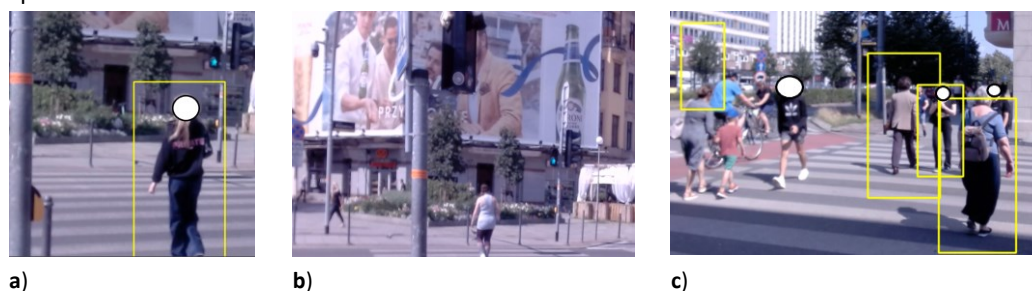


Fig. 2. Pedestrian crossing with traffic light in Katowice, Haar procedure: a) Case 1, Stawowa Str., b) Case 1, advertisement on Stawowa Str., c) case 2, Moniuszki Str. Source: elaboration own

In contrast to Haar features, the HOG algorithm is more accurate, as demonstrated by the images in Figure 3. It should be added that the accuracy of any method depends on the input parameters of the procedure. Here, knowledge of the OpenCV library and computer vision techniques is useful. However, changes to the procedure parameters are not very flexible. They should rather be combined with thresholding and motion scene image filtering techniques. Proper image pre-processing before applying detection procedures and appropriate tuning of input parameters increase the accuracy of object recognition. Ultimately, each procedure still fails to detect all the objects present for detection. Discussions of method validation and error rates are presented at the end of this article.

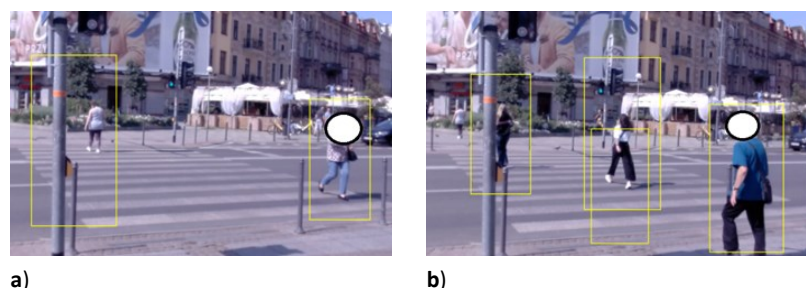


Fig. 3. Pedestrian crossing with traffic light Katowice, HOG procedure: a) Case 1, b) case 1bis. Source: elaboration own

Figure 3b shows double detection of one pedestrian in the image frame, which illustrates one of the possible errors when using these methods. When using these methods, sometimes the person in the frame is not identified at all.

During the measurements, a manual pedestrian counter was also used, independent of the cameras, although both devices shared a common time base synchronized over the Internet with an atomic clock. The purpose of the counter was to provide data for validating and calibrating the method. It was implemented as a program installed on a Microsoft Surface tablet. While recording video footage of pedestrians at the crossing, the operator clicked on the appropriate type of pedestrian and their entry point to the crossing, as shown in Figure 4a. The entry time was recorded with millisecond accuracy. This level of precision is important because it allows the multitouch mode of the tablet to register up to 10 pedestrians clicked simultaneously (at least in theory). The manual pedestrian registration process is shown in Figure 4. Using manual reading, we identify up to 8 different categories of pedestrians, and we want to use the same number using automatic detection procedures.



a) b) c) d)

Fig. 4. Hand counter for pedestrian on tablet a) program counter, b) output file, c) manual counter operator's working method, d) crossing. Source: elaboration of its own

In the program, whose main window is shown in Figure 4a, the operator clicks on the type of person moving from the curb to the pedestrian crossing, and the entry time is recorded in the output file, as shown in Figure 4b. The operator notes different types of people, including pedestrians with PWSN (people with special needs), cyclists, and others. The output file also records the direction of movement (hence the two rows of icons representing pedestrians on both curbs in Figure 4a). The manual counter operator's working method for manual counting of pedestrians is shown in Fig.4c. Additionally, the system registers instances of pedestrians crossing during a red light or leaving the crossing while the green light is flashing. However, in this study, no incidents related to traffic violations were recorded.



a) b) c)

Fig. 5. Cropping and transformatting scene: a) cropping, b) blurred, c) OTSU transformation. Source: elaboration own

As in many procedures implemented using the OpenCV library, preliminary image preprocessing is required, such as cropping the image to match the area of interest. The conversion of the visualized traffic scene to smaller dimensions is shown in Figures 5a. Blurred image on 5b. Useful and sometimes necessary image transformations include thresholding, blurring, and other operations performed using various filters. An example of a threshold transformation (Otsu's method) is presented in Figure 5c. These operations significantly speed up computational procedures, enable the detection of relevant details in the traffic scene, and, in some cases, are required by specific library procedures, such as converting the image to grayscale. Figure 6 shows other example transformations of the traffic scene image at the pedestrian crossing.



Fig. 6. Digital transformation of scene: a) gray scale, b) negative image, c) thresholding (172,255). Source: own elaboration

Figure 6a shows an example of applying the grayscale conversion operation and a blurring operation performed using the command `cv2.GaussianBlur(imgray, (5, 5), 0)`. Figure 6b presents a negative image and 6c presents a threshold with color range from 127 to 255. Generally, the procedure involves transforming the drawing into a format that is required by further procedures of the OpenCV library or that allows for the enhancement of full features of the drawing.

Similarly to other examples involving the OpenCV library, the processed image is split into several separate windows. Different computational procedures are performed in each of these windows. This is illustrated in Figure 7.

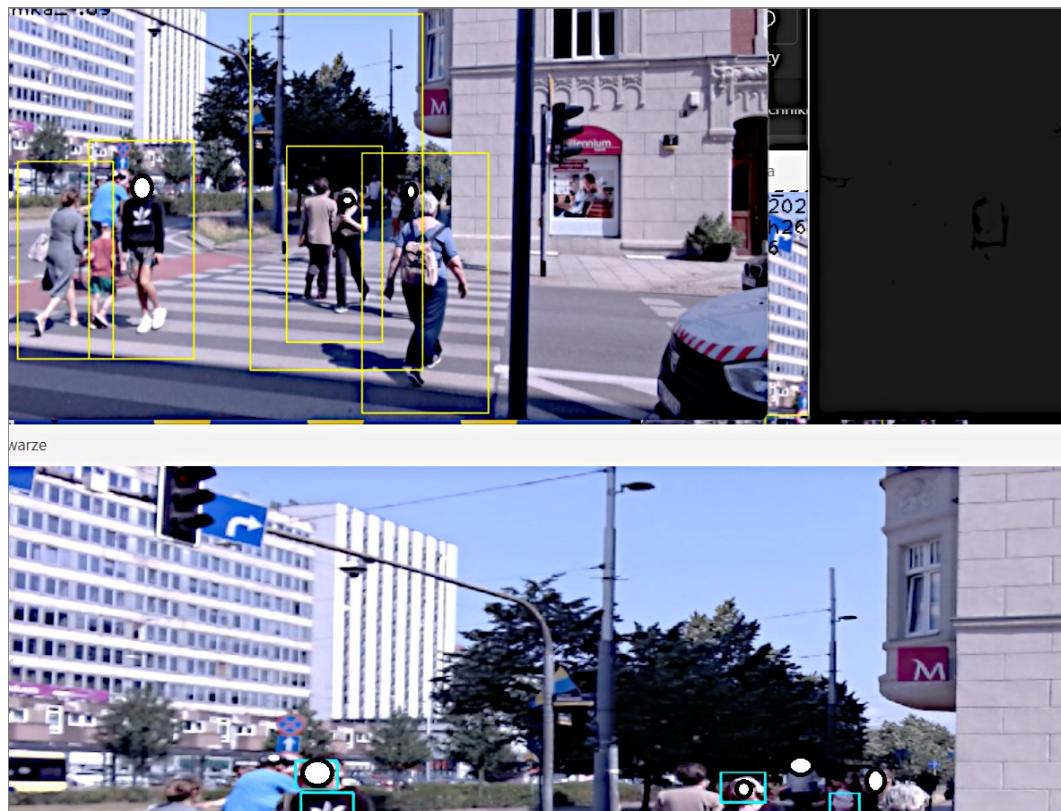


Fig. 7. Multi-image processing a) pedestrian, b) bicycles, c) faces. Source: elaboration own

In Figure 7a, the HOG procedure is applied to the video material, with detected pedestrians marked by yellow boxes. Next to it, in the window on the right side of the screen, after converting the frame to a grayscale image, wheels of vehicles such as bicycles and wheelchairs (used by PWSN) are detected. At the bottom, Haar features are used to recognize the faces of pedestrians crossing the street. In this way, the movement of two opposite pedestrian traffic streams at the crosswalk is isolated. Some go to the camera, others away from the camera. Recognized faces are marked in blue (using squares or blue boxes). During heavy traffic, pedestrians (including people and, occasionally, vehicles that block the view) often become obscured. This is why a high camera mounting position will be used in the commercial solution. This not only increases detection reliability but also reduces the number of artifacts falsely recognized as pedestrians or their faces (mainly originating from the scene background).

These artifacts can be eliminated by extracting objects from the background, but this slows down the proposed method and requires frequent background updates to maintain stable object detection. This conceptual system installation model will be presented in subsequent publications.

2. RESULTS

In Figures 8a–d, the results of the manual pedestrian count are presented using software installed on a tablet. These results come from data recorded in the output file, where information is saved about the direction of movement, the type of pedestrian (types listed in the legend in Figure 8), the time of entry onto the roadway and any traffic law violations. The manual notation procedure is suitable for low volume pedestrian traffic. It fails when pedestrian traffic is high, and intersections occur.

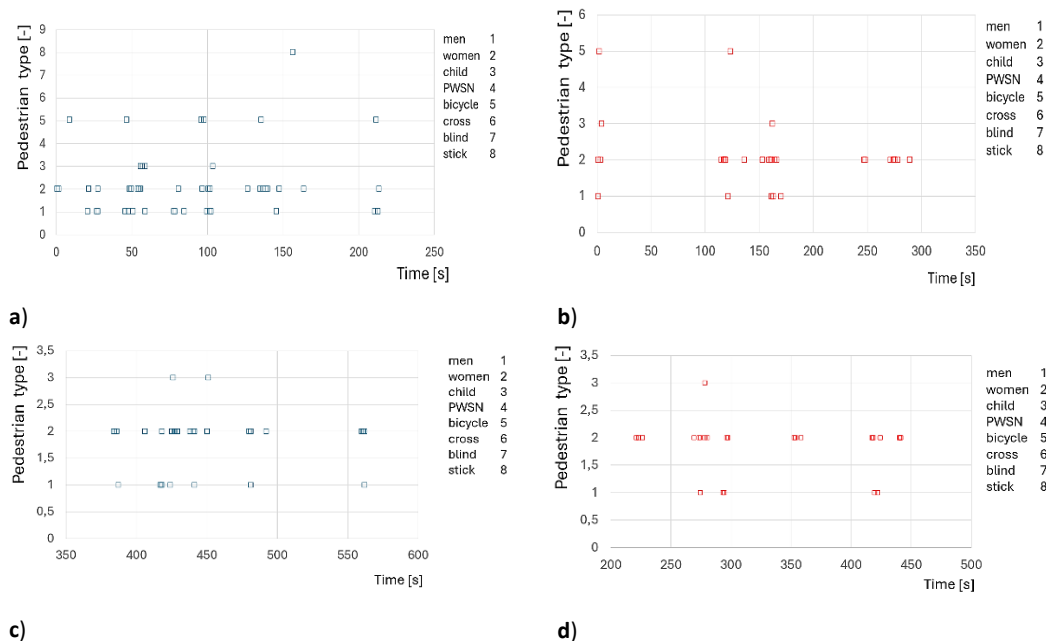


Fig. 8. Data from manual counter: a, b) pedestrian type, case 1 c, d) pedestrian type, case 2. Source: elaboration own

In Figure 8, it can be observed that pedestrians move according to signal cycles; this is illustrated by clusters of points visualized in the graphs (visible in the form of characteristic gaps between pedestrians). On these graphs, the horizontal axis represents time measured with millisecond precision, while the vertical axis indicates the specific type of pedestrian (the recording time of entering the road varies depending on the reaction time, which can range from 0.3 to 0.9seconds). The types of pedestrians are numbered and explained in the figure legends. The measurement tablet supports multitouch functionality, allowing up to 10 simultaneous touch points [14]. The main limitation lies in the reaction time of the operator and his multitasking ability (probably

Measurement and analysis of pedestrian traffic characteristics in real time

6 to 9 pedestrians in one time, depending on the skills of the handheld recorder). The tablet itself is used solely as a device for calibrating the results of the automatic method.

In the future, the system is planned to be calibrated separately for the two directions of traffic at the crosswalk. Interestingly, dynamic pedestrian registration presents more challenges than vehicle detection, for example, pedestrians turning around or changing direction mid-crossing are often observed. It should be noted that the calibration using the tablet and manual counter is only approximate. Manual measurement records the number of pedestrians, their type, and the time of entry on the roadway. These coarse characteristics are used solely to calibrate the measurement procedures of the automated method.

The automatic method, based on the motion captured by the camera and implemented using OpenCV, has a fundamentally different nature and a level of data accuracy. The method can utilize video recorded at frame rates ranging from 15 fps up to 300 fps. However, cameras above 120 fps are not expected to be used due to cost considerations (cheap cameras are up to \$100). The system is designed to operate at extremely low implementation costs, calculated per individual crosswalk. We estimate that, excluding software development costs for the end-user version, the hardware and installation costs per intersection will not exceed approximately \$150. Measurement system could also be used to update traffic signal programs. For example, in Katowice, on Francuska Street, pedestrians often wait at an intersection despite forming a significantly larger group than drivers for extended periods of the day (at green waves). This method is formalized to resolve such doubts as well.

Typically, for FHD image recording at 30 fps, the accuracy of pedestrian location on the crosswalk is about 0.033 seconds. When using 300-fps cameras, this improves to 0.003 seconds. A comparison of the method at different frame rates (e.g., 30 fps vs. 120 fps) will be presented in a separate study. The purchase of 300-fps cameras is not planned, as they are not economically justified for commercial deployment. It's worth noting an important difference between manual and automatic measurement data. Manual measurements register only the event of a pedestrian entering the roadway and record their type and direction. In contrast, automatic measurements provide full tracking of the pedestrian's movement path within the camera's field of view. These precise positions enable the study of interactions such as pedestrian–pedestrian, pedestrian–cyclist (or scooter rider), and pedestrian–vehicle driver. The characteristics recorded in this way show the movement paths of individual spaces. This allows us to determine the probability of disruption to pedestrian traffic at the crossing. Probability calculated when a pedestrian's trajectory changes while crossing the road. From here, we can calculate traffic flow characteristics at the pedestrian crossing, known from road and rail traffic engineering literature as traffic smoothness function:

$$F(q) = (1 - p(q))q \quad (1)$$

Where:

$F(q)$ - traffic flow smoothness function, the number of pedestrians who passed smoothly through the crossing, excluding any stops, which is inconsistent with the law [pedestrian],

q - intensity of pedestrian traffic [pedestrian/min.],

$p(q)$ - probability of regulating pedestrian traffic at a known intensity [-]

Probability of regulating pedestrian traffic at a known intensity is calculated by qualification changing the trajectory of each pedestrian in both directions of movement, although the procedure can be split for both directions of movement. Hence, the regulation probability is calculated as the sum of the deviations of each pedestrian trip through the crossing:

$$p(q) = \frac{n}{m} \quad (2)$$

Where:

n - pedestrian trajectories disrupted [-],

m - pedestrian trajectories non disrupted [-],

Knowing the traffic flow for the streams intersecting pedestrian traffic, we can use the analogy for the product of traffic at railway crossings to calculate the expected traffic flow at a pedestrian crossing colliding, or non-colliding with road traffic streams:

$$F(q, q_v) = F(q) * F(q_v) \quad (3)$$

Where:

q_v - intensity of vehicle traffic in common traffic light phase or separate [vehicle/min.].

This analysis opens new opportunities in traffic management, signal control, modification of traffic organization, and infrastructure design. The same approach can be extended to railway crossings, where pedestrian violations of traffic laws are also an important subject of research. The accuracy of pedestrian positioning depends on the type and mounting of the camera. With standard 30-fps recording, updates occur every 0.03 seconds; with 300-fps systems, this interval is reduced to 0.003 seconds. This process creates trajectories of moving objects associated with pedestrians crossing at designated crossings. These trajectories form a kind of VST (Velocity-Space-Time) diagram. Figure 9 shows examples of such trajectories.

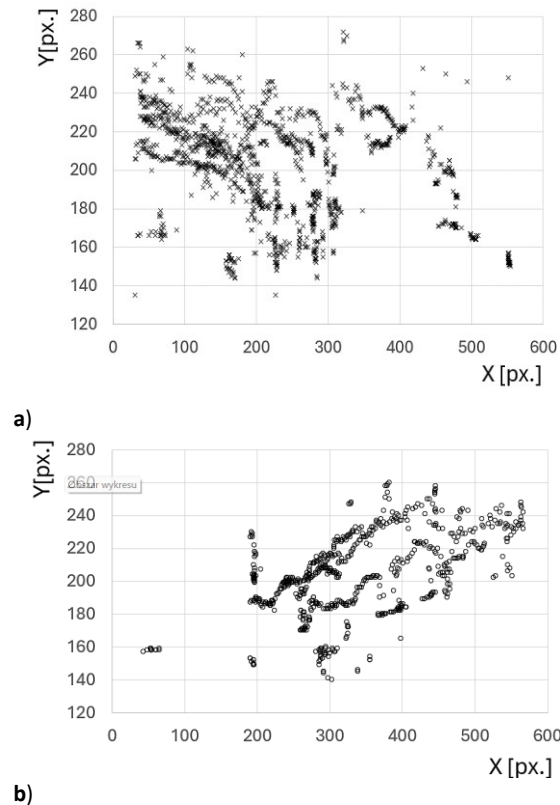


Fig. 9. Path of pedestrians a) case1, Moniuszki b) case 2, Stawowa. Source: elaboration own

Figures 9a and 9c show pedestrian trajectories with characteristic snake-like patterns. These are generally areas of disruption to the flow of traffic when accompanied by an object represented by another pedestrian. As seen in Figure 9, pedestrian crossings on Moniuszki Street are slightly more chaotic than those on Stawowa (this is due, among other things, to the greater number of pedestrians now). Pedestrians more frequently interfere with one another's movement paths. This results from a higher pedestrian density in Moniuszki despite similar times of day (morning, 07 July 2025). The Moniuszki crossing is also slightly shorter (approximately 10.20 and 6 m) compared to the Stawowa crossing (9.96 and 6 m). Furthermore, the structure of both vehicle and pedestrian traffic is more complex in Moniuszki. The graph in Figure 9 effectively highlights areas with increased pedestrian interaction. So far, interactions between pedestrians and vehicles during green arrow phases have not been analyzed. Based on our observations, we argue that the dynamics of these interaction points between pedestrians in the transition region of the level crossing can be the basis for their classification in future studies.

The trajectories shown in Figure 9 are affected by perspective distortion, caused by the placement of the camera. However, they can be corrected by considering the exact mounting location within the scene. These trajectories, in addition to measuring movement speeds, allow a detailed analysis of pedestrian–pedestrian

Measurement and analysis of pedestrian traffic characteristics in real time

interactions. At busy crossings, numerous such interactions occur within counter flowing pedestrian streams. These include changes in the direction and altered movement paths, which are also recorded manually when a pedestrian crosses unusually. In this context, we plan to investigate pedestrian decision-making strategies at such sites in future research. Figure 10 presents another characteristic of pedestrian movement. It enables the estimation of the type of pedestrian based on size (more precisely, the footprint of the pedestrian in the camera's image). This refers to the area of the image, expressed in pixels, in which a pedestrian was detected.

Figure 10 presents a characteristic related to the execution of the HOG procedure. This feature relates to the execution of the HOG procedure, which detects point changes on the screen that correspond to the position of a pedestrian (if properly recognized). Simultaneously, the surface area associated with these changes (in pixels) is calculated. Based on this area, the type of pedestrian is estimated by comparing it with statistical characteristics in the dataset, such as mean, median, and mode. Since elongated objects often have wheel detected by separate procedures, these calculations exclude such objects. Therefore, it becomes possible to approximate the height and classify individuals into groups such as adult or child. In Figure 10, the height of the object at consecutive measurement points corresponds to its detected surface area in pixels. To determine the direction on the crosswalk, the Haar cascade procedure is used for face recognition. Detection accuracy depends on many factors, including camera mounting. In a commercial system, a separate camera may be dedicated for this purpose, mounted in the opposite direction. There is no difficulty in implementing multicamera systems within the proposed method, as the authors have relevant experience in this area.

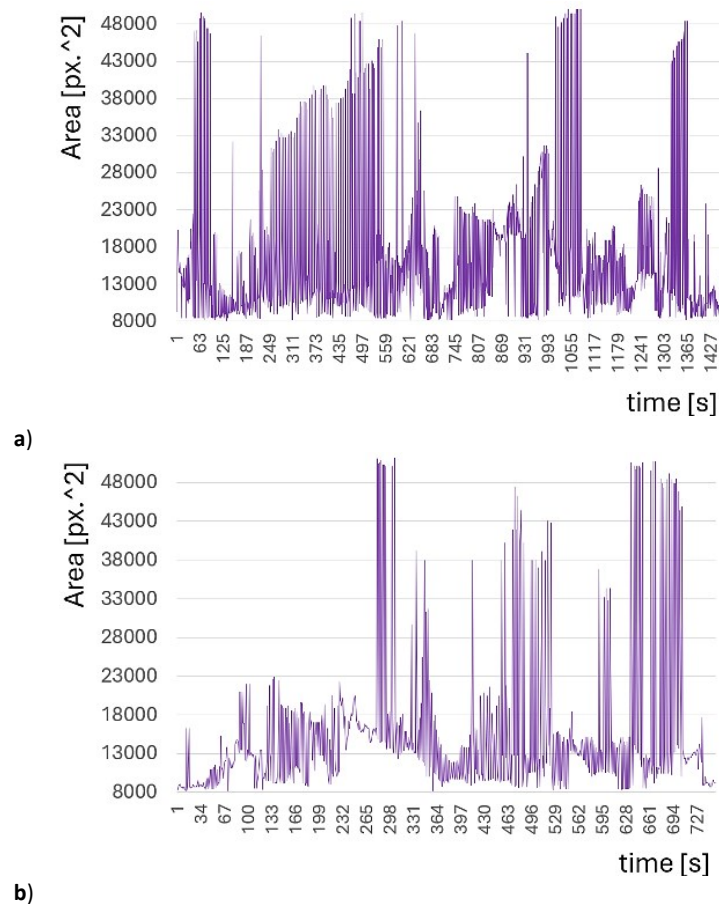


Fig. 10. Measurement surface [in pixel²] area of detected object a) case 1 b) case 2. Source: elaboration own

Data to observe pedestrian crossings in different traffic light cycles are presented below.

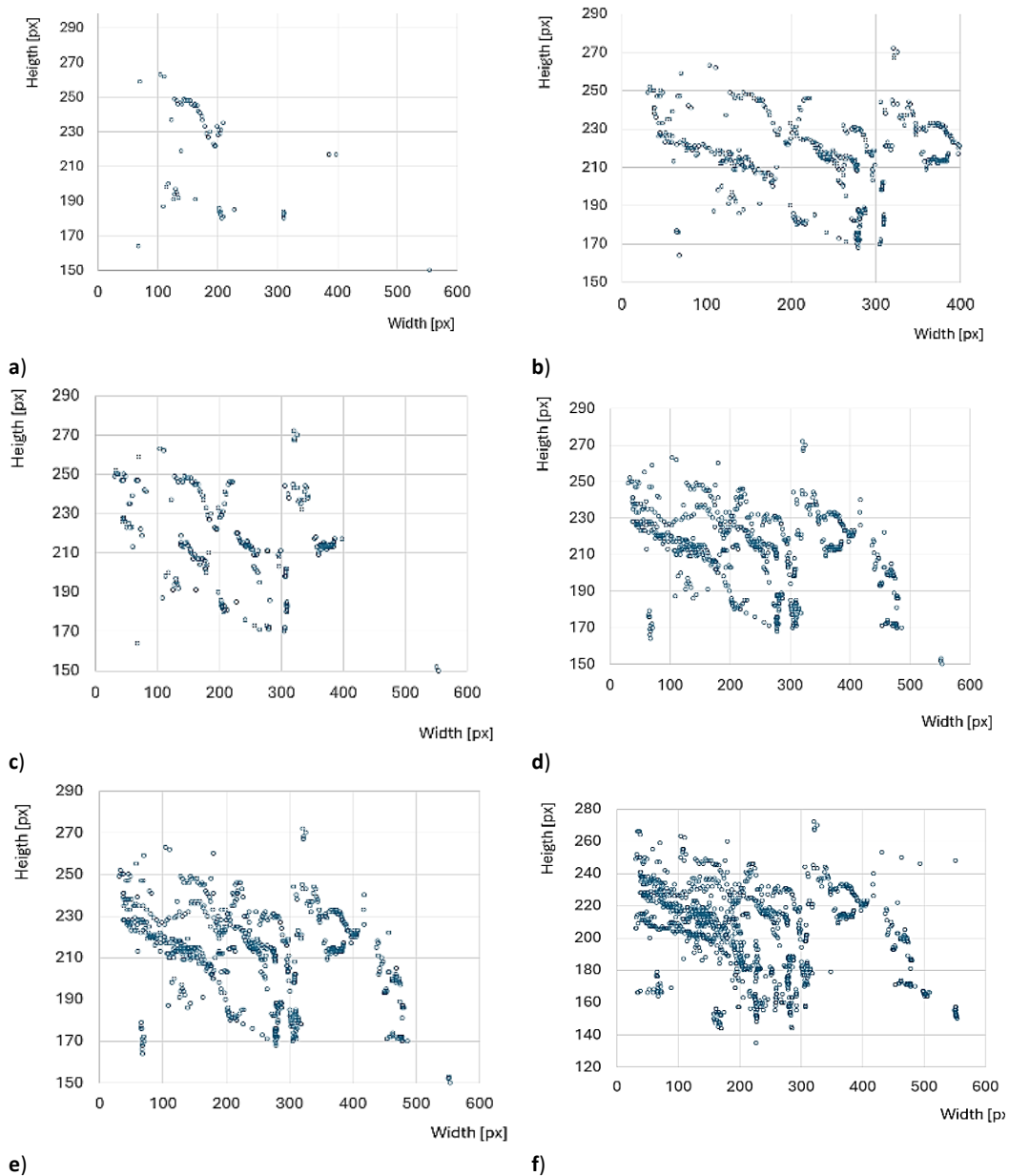


Fig. 11. Detailed data on the object in case 1, on each image separate cycle signalization data. Source: elaboration own

Figure 11 a-f presents the trajectories of detected pedestrians as they cross the crosswalk. Different, yet similar, patterns emerge in each signal cycle. However, machine learning should be used to recognize them. These drawings show characteristic shapes, e.g. the letter V and other types of clusters.

Measurement and analysis of pedestrian traffic characteristics in real time

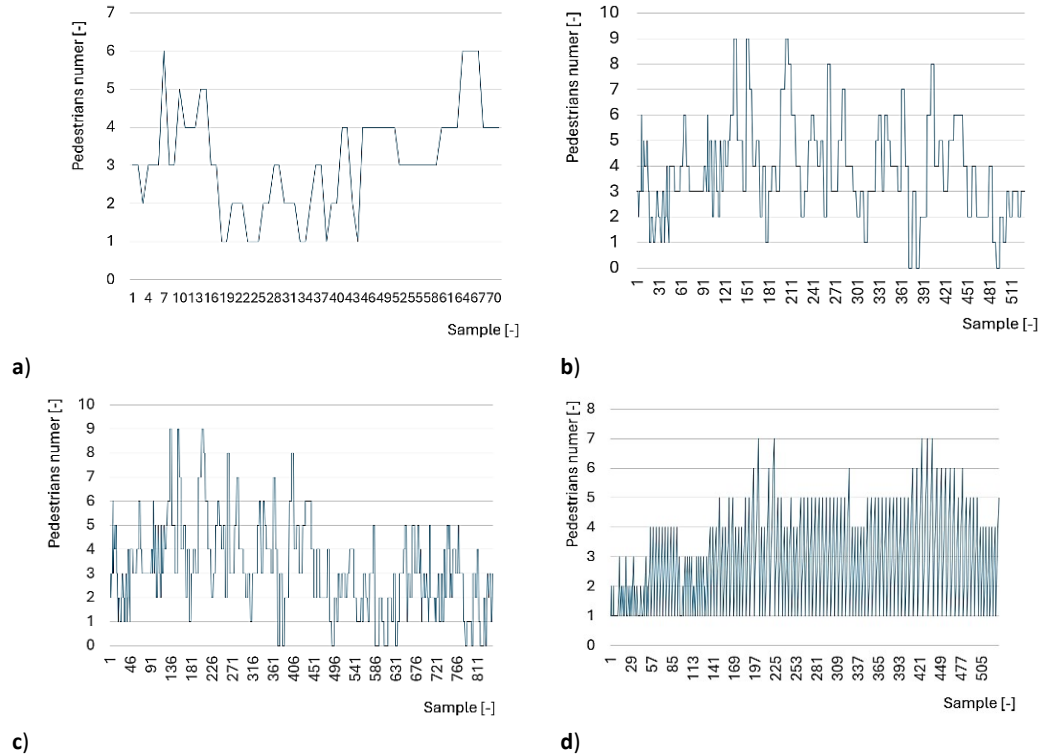
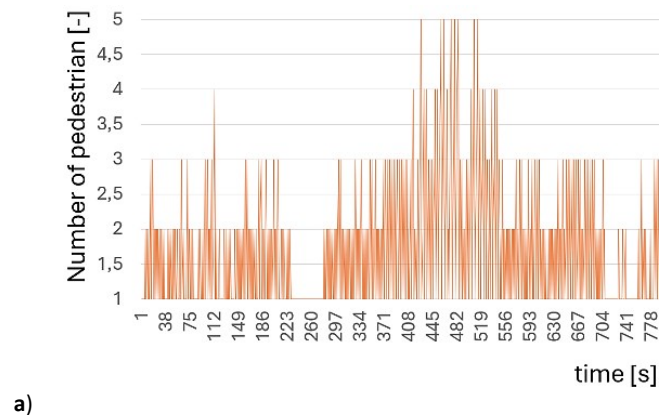


Fig. 12. Number of pedestrians, each image separate cycle signalization based on data from figure 11a-d a) from Fig. 11a, b) from Fig. 11b, c) from Fig. 11c, d) from Fig. 11d. Source: elaboration own

Figures 13a and 13b show pedestrian counts over time. Figure 13c presents the corresponding number of detected faces of pedestrians moving towards the camera. Differences in these metrics result from imperfections in the Haar face detection procedure. The method is continually refined, and procedures are calibrated based on the basis of control variables to improve detection accuracy. This can be addressed by selecting appropriate Haar parameters, choosing a suitable training sample, and using the correct cascade definition file (typically in XML format, e.g., haarcascade.xml). This topic is discussed in the method validation section, where we examine the number of falsely recognized faces and pedestrians. The Haar procedure, as mentioned, is weaker in detection.



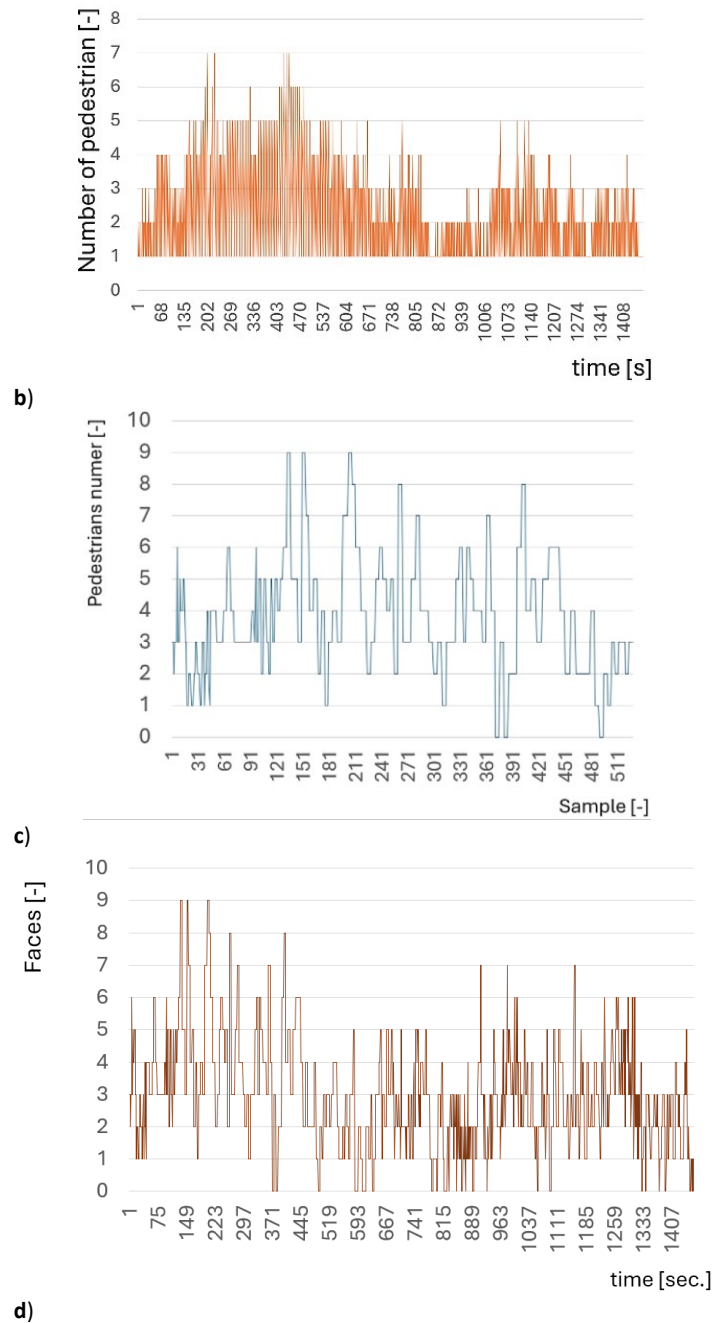


Fig. 13. Number of pedestrians: a) case 1, on the street Stawowa b) case 2, on the street Moniuszki c) number of faces on the street Stawowa, d) number of faces on the street Moniuszki. Source: elaboration own

As shown in Figure 13, each crossing records up to a dozen pedestrians simultaneously, which is consistent with the data from the manual measurement files. However, there is considerable heterogeneity in pedestrian

Measurement and analysis of pedestrian traffic characteristics in real time

traffic volume. This variation is due to the interactions between adjacent traffic signals at nearby pedestrian crossings. Interestingly, the influence of traffic light timing is quite limited, given the short distances between crossings. As demonstrated throughout the figures presented, the proposed method provides detailed characteristics of pedestrian traffic. These allow classification by type, direction, accompanying means of transport (e.g., walking or riding a bicycle), level of mobility, and more. The method will be further developed to estimate, with a certain degree of probability, the gender of the pedestrian, behavioral aspects (e.g. calm walking, hurried movement, walking in groups), and age. Yes, age can also be estimated, and this will be the subject of future research. The resulting demographic structure of pedestrian traffic can be used in analyses of the functional and economic aspects of urban areas, as discussed in the conclusions. Based on such analyses, the location of Points of Interest (POI) can be inferred. Another promising research direction involves changes in pedestrian dynamics due to temporary environmental conditions, including interactions with emergency vehicles.

3. VALIDATIONS

The validation procedure involves saving the detected objects (to a selected directory on disk) recognized by the HOG and Haar algorithms, followed by counting the correctly identified faces and pedestrians. The results of the validation procedure are presented in Table 1.

Table 1. Correctness of pedestrian and face recognition

Result	Pedestrian	Face
Stawowa	77%	69%
Moniuszki	69%	76%

These methods have a 20-year history of research, as mentioned, they are fast but not very accurate. The accuracy is therefore as indicated by other researchers (60-80%), and here it will be increased by introducing additional control procedures. At this level it is already sufficient for full research into the impact of pedestrian and vehicular traffic. The procedure is simple. At this point, it introduces a significant error, especially with heavy pedestrian traffic. But its advantage is the automation of these operations. Improving the accuracy of the calculations is a matter of the resolution of the filmed image, positioning the camera in the motion scene relative to the transition, and improving the calculation procedures, which will be consistently improved. Furthermore, comparing the calculations performed using different OpenCV library procedures is highly beneficial. In the clearings, we also compare the data with those of a plane Lidar, such as RPLidar A2M12 or similar.

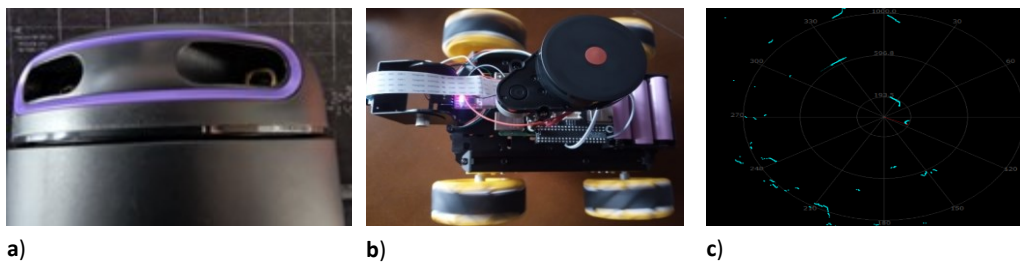


Fig. 14. LIDAR used in research: a) Lidar module A2M12, b) mobile robot with similar Lidar, Waveshare D200 c) sample example results. Source: elaboration own

In addition, we are in the process of training our own Haar classifier. More precisely, several classifiers in different views from the side and top of pedestrian crossing, with respect to the season. During the operation of the algorithm, areas likely to contain a pedestrian and areas containing a pedestrian's face (for pedestrians moving toward the camera) are identified. These image segments, extracted while processing video recordings from pedestrian crossings, are saved into separate folders. These folders are then reviewed manually; each file is inspected to determine whether the pedestrian or face was correctly identified or if the extracted area contains other artifacts from the traffic scene. This process allows for determining the accuracy of pedestrian and face recognition. It enables verification of proper functioning, that is, its validation, by calculating the

percentage accuracy and the types of recognized objects. Many such procedures also help to fine-tune the parameters of OpenCV library functions used in the presented method. The problem of direction recognition will be solved with hardware in the final version of the system. The study found that such cases are sporadic at pedestrian crossings at intersections (locations with traffic signals).

Sometimes, a pedestrian walking toward the camera turns his head and is identified as walking in the opposite direction. This issue will be addressed with hardware in the final version of the system. The study found that these cases are also sporadic at pedestrian crossings at intersections.

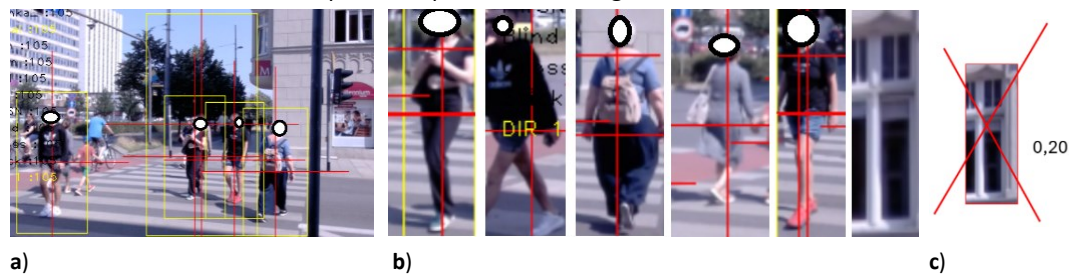


Fig. 15. Illustration of the validation method: a) camera image, b) object captured according to the information from the HOG procedure, c) result of comparing the data with the image data from the camera. Source: elaboration own

The scheme involves assessing the percentage of recognized objects that are incorrectly identified. It should be remembered that other moving objects may appear within the camera field of view, such as animals or cars. Often, parts of building facades are mistakenly detected as positive hits, especially in Haar-based procedures. In urban intersection environments, advertising elements frequently cause false detections. It is easy to imagine this by realizing how many and in what situations we see faces in surroundings that are not actually faces.

Next, we propose a comparison of computational data obtained using data from a visible light camera and an infrared camera. For this purpose, we used the Flir On Pro camera. The camera can find invisible data offering native resolution 160x120 px. (thermal image). Measure temperatures up to 400°C (752°F) with a sensitivity that detects temperature differences down to 70 mK. The camera is available with a USB-C connector for Android or iPhone. The camera boosts images to 1440x1080 resolution.

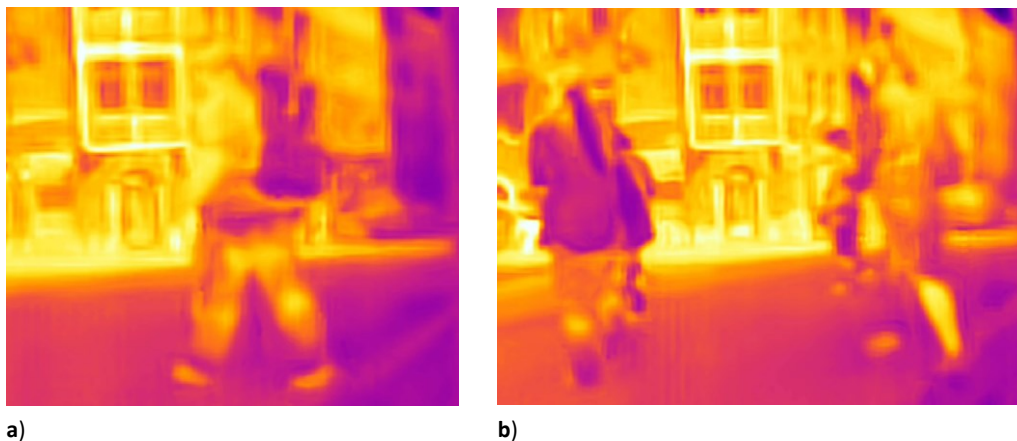


Fig. 16. Sample data from an infrared camera: a) one pedestrian, b) a group of five pedestrians. Source: elaboration own

As shown in Figures 16a and 16b, pedestrians are also visible in infrared images. The images were taken in July at temperatures above 25 degrees Celsius. As temperatures drop, pedestrians will be more visible in the video. However, the HOG procedure still can recognize pedestrians in these conditions, although the recognition accuracy is lower.

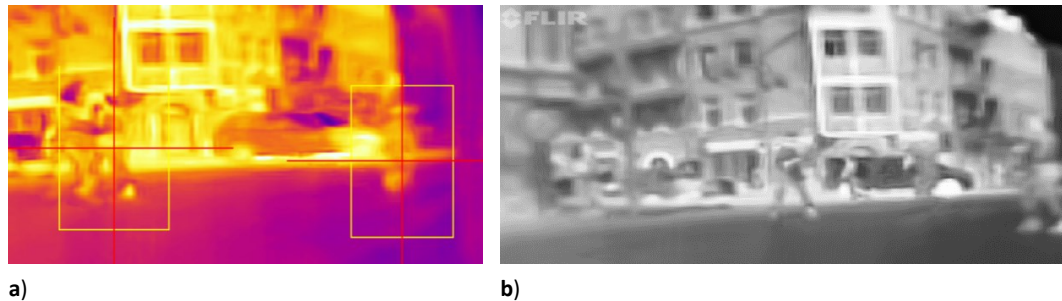


Fig. 17. Sample data from image analysis of pedestrians recorded with an infrared camera: a) identified pedestrians, b) another view of the crossing. Source: elaboration own

As shown in Figures 17 (a) and (b), pedestrians in infrared images are detected using HOG procedures. Interestingly, a procedure based on face detection implemented using the Haar classifier also works. We will present procedures for processing films recorded in the infrared range in another publication.

Example data from traffic flow calculations, calculated based on equations 1-3, are presented in Table 2.

Table 2. Product of traffic smoothness function

Street	Product of traffic smoothness function [pedestrians]	Pedestrian intensity q	Vehicle intensity q_v	$p(q)$	$p(q_v)$	$1 - p(q)q$	$1 - p(q_v)q_v$
Stawowa	466	44	16	0,14	0,23	37,84	12,32
Moniuszki	787	51	23	0,08	0,27	46,92	16,79

Such calculations, as presented in Table 2, significantly simplify the problem. Pedestrian speed is not measured while crossing the road, which affects the traffic flow function for both pedestrian and vehicular traffic flows. Nevertheless, these calculations demonstrate how the influence of two conflicting traffic flows on each other can be examined, both when the vehicular traffic flow is stopped at a red light or when the pedestrian and vehicular traffic flows intersect in a single phase, for example, at a green arrow. The assessment of the usefulness of the method requires mass calculations of dozens of intersections within the scope of the proposed methodology to compare different cases of pedestrian and vehicular traffic impact.

CONCLUSIONS

The method presented in this article is not new; pedestrian detection techniques using the OpenCV library are commonly applied in autonomous vehicle (AV) technology [16–20]. They are also used in many other solutions, such as urban surveillance or the tracking of wanted persons [21–22]. The most common application of this library is access control to restricted traffic zones (which does not necessarily involve pedestrian control, but, for example, the recognition of license plates on parking barriers) [23–25].

However, the proposed method has different long-term goals and focuses on studying the interactions in pedestrian traffic at various types of crossings [26–27]. This includes both signalized and non-signalized crossings, as well as roundabouts (although it can be used anywhere appropriate camera mounting is possible). To characterize traffic at a given crossing, it is not necessary to process the data in detail; conclusions can be drawn from the basic characteristics presented in this article. We are most interested in pedestrian interactions at crossings, as they allow us to determine whether they have been well designed and whether pedestrian trajectories do not conflict. These collisions can also result from traffic light parameters that are not matched to traffic patterns. To this end, traffic flow functions at intersections can be introduced, considering the smoothness function of both conflicting traffic flows. This is a direct analogy to the traffic flow product at rail-road crossings, known from the instructions in force on Polish national railways. This is a completely different perspective than the one currently used to assess intersections on Polish road networks.

In autonomous vehicles (AV), pedestrian detection is crucial due to traffic safety concerns (fatal accidents still occur in this area). In this method, we introduce a variety of additional solutions, such as determining the

type, direction of movement, and specific behaviors of the pedestrian while crossing, including violations of traffic laws. This is not a simple calculation known from many works in the field of vision techniques. The pedestrian violation counter itself is an added value to these algorithms. Unlike manual measurements, we know the position eight types of pedestrian traffic participants with high space and time accuracy, which can theoretically be updated every 3 milliseconds. The accuracy of the pedestrian location in space and time will be highest when the cameras are located directly above the pedestrian crossing. This opens entirely new possibilities for analyzing interactions at the interfaces: pedestrian-pedestrian, pedestrian-cyclist, and pedestrian-driver. As shown in this text, it can be used to calculate pedestrian disturbance parameters, although here it is presented in the simplest possible way.

Based on traffic safety statistics in Poland, we know that up to 30% of accidents involve unprotected road users [28–30]. The system can also be applied beyond crossings, on selected segments of pedestrian routes, allowing the construction of origin–destination (OD) matrices for pedestrian movements in cities [31–34]. Thus, it has direct commercial applications, such as assessing the commercial attractiveness of specific urban locations [35–37]. Understanding the demographic structure of pedestrians allows a range of activities related to the commercialization of urban areas. Pedestrian conflicts can also be studied in the impact zones of large-scale stores. Increasing the frame rate of recorded footage at pedestrian crossings, within reasonable costs for purchasing better cameras, is possible up to 120 fps, and with a higher budget up to 300 fps. This enables very precise analysis of interactions at pedestrian crossings and large-scale stores. Especially in the latter case, incredibly complex patterns of interactions and impacts in pedestrian traffic are observed.

This will be the subject of further publications (technical challenges arise for high resolutions and speeds). Where else can this method be applied? It can be used in complex ticketing gates, where queues are spatially structured by infrastructure elements, but people often line up in their own way. In large supermarkets around checkout areas, in tunnels and underground passages (especially at railway stations), in schools (in their halls), waiting rooms and clinics. Also, in the organization of large mass events, such as stadiums and sports arenas. This will allow for a better understanding of pedestrian traffic interactions not only at the interface between pedestrian and vehicular traffic, but also at the specific locations mentioned above. A particularly interesting issue, important due to the presented methodology, and practically not used in Poland, is the geometric shaping of pedestrian crossings. Designed to increase safety and traffic flow in these types of facilities. Any large commercial and service facilities with large, irregular pedestrian flows within these facilities can benefit from applying the presented methodology. Unlike other studies in this field, we propose the use of a wide catalogue of pedestrian traffic categories in these analyses: gender, age, degree of disability, counting microbial means of transport, etc.

ABBREVIATIONS

1. **OpenCV** - Open Computer Vision,
2. **AV** - automated vehicle,
3. **FPS** - frame per second,
4. **HOG** - histogram of gradient,
5. **Haar** - Haar-like features are digital image features used in object recognition.

POMIAR I ANALIZY CHARAKTERYSTYK RUCHU PIESZEGO W CZASIE RZECZYWISTYM

Niniejszy artykuł porusza problematykę pomiaru i analizy charakterystyk ruchu pieszych w czasie rzeczywistym. Proponowana metodologia wykorzystuje standardowe, tanie kamery internetowe, które są tymczasowo instalowane na wybranych przejściach dla pieszych w Katowicach. W kolejnych etapach badań kamery zostaną rozmieszczone na skrzyżowaniach z sygnalizacją świetlną, a ich lokalizacja zostanie zoptymalizowana zgodnie z przyjętymi procedurami pomiarowymi. W tym kontekście przestrzenne umiejscowienie kamery jest kluczowym czynnikiem wpływającym na jakość danych i pomiarów. Głównym celem prezentowanych pomiarów jest analiza dynamiki ruchu pieszych w ramach modelu koncepcyjnego opisującego interakcje między ruchem pieszych i pojazdów, a także interakcje interpersonalne między pieszymi na skrzyżowaniach, zarówno z sygnalizacją świetlną, jak i bez niej. Przedstawiono tu kilka takich koncepcji wykorzystania takiego modelu. Model teoretyczny zostanie rozwinięty w przyszłych publikacjach. Niniejszy artykuł koncentruje się na opracowaniu i wdrożeniu automatycznej metody akwizycji danych w czasie rzeczywistym, która wspiera to podejście modelowania. Gromadzenie danych w czasie rzeczywistym przeprowadzono z wykorzystaniem technik wizji komputerowej, zaimplementowanych w bibliotece OpenCV i języku programowania Python. System oprogramowania działa na platformie Windows, ale może być uruchomiony na dowolnej platformie: Unix, macOS. W analizie wykorzystano kompleksowy zestaw parametrów ruchu, uwzględniający zarówno charakterystykę przestrzenną i funkcjonalną obserwowanego środowiska, jak i wzorce zachowań pieszych.

Słowa kluczowe: ruch drogowy, przejścia dla pieszych, OpenCV, techniki wizyjne, miejsca atrakcyjności, punkty zainteresowania POL.

REFERENCES

- [1] Dawson-Howe, K. (2014). *A practical introduction to computer vision with OpenCV*. Wiley.
- [2] Minichino J., Howse J. (2015). *Learning OpenCV 3: Computer vision with Python* (2nd ed.). Packt Publishing.
- [3] Bueno García G., et al. (2015). *Learning image processing with OpenCV*. Packt Publishing.
- [4] Dalal N., Triggs B. (2005). Histograms of oriented gradients for human detection. <https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>.
- [5] Kachouane M., Sahki S., Lakrouf M., Ouadah N. (2012). HOG based fast human detection. In *2012 24th International Conference on Microelectronics (ICM)*, 1–4. IEEE. <https://doi.org/10.1109/ICM.2012.6471380>.
- [6] Heisele B., Wöhler C. (1998). Motion-based recognition of pedestrians. In *14th International Conference on Pattern Recognition*, 2, 1325–1330. Brisbane, Australia.
- [7] Viola P., Jones M., Snow D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2), 153–161.
- [8] Dalal N., Triggs B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1, 886–893. IEEE.
- [9] Arya Z., Tiwari V. (2020). Automatic face recognition and detection using OpenCV, Haar cascade and recognizer for frontal face. *International Journal of Engineering Research and Applications*, 10(6-V), 13–19.
- [10] Karpiuk N., Klym H., Vasylychyn I. (2023). Facial recognition system based on the Haar cascade classifier method. <https://indico.global/event/10039/papers/96116/files/3859-2023257490.pdf>
- [11] Sen J. (2020). Face detection using OpenCV and Haar cascades classifiers [Master's project]. <https://doi.org/10.13140/RG.2.2.26708.83840>.
- [12] Reinius S. (2013). Object recognition using the OpenCV Haar cascade-classifier on the iOS platform [Bachelor's thesis, Uppsala University]. <https://uu.diva-portal.org/smash/get/diva2:601707/FULLTEXT01.pdf>
- [13] Phase T.R., Patil S.S. (2019). Building custom HAAR-cascade classifier for face detection. *International Journal of Engineering Research & Technology (IJERT)*, 8(12). <http://www.ijert.org>.
- [14] Microsoft. (n.d.). Dane techniczne i funkcje urządzenia Surface Go 2. <https://support.microsoft.com/pl-pl/surface/dane-techniczne-i-funkcje-urz%C4%85dzenia-surface-go-2-0fc6a657-2851-484f-6f82-bd3c589ed92c>.
- [15] Öztürk G., Eldogan O., Köker R. (2024). Computer vision-based lane detection and detection of vehicle, traffic sign, pedestrian using YOLOv5. *Sakarya University Journal of Science*, 28. <https://doi.org/10.16984/aufenbilder.1393307>.
- [16] Favorskaya M. N., Andreev V. V. (2019). The study of activation functions in deep learning for pedestrian detection and tracking. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W12, 53–59. <https://doi.org/10.5194/isprs-archives-XLII-2-W12-53-2019>
- [17] Raju T., et al. (2024). IoT based automatic vehicle accident detection and rescue system. *International Journal for Modern Trends in Science and Technology*, 10(3), 94–99. <https://doi.org/10.46501/IJMTST1003016>.
- [18] Xinyu W., Tingting L. (2024). A deep learning-based car accident detection approach in video-based traffic surveillance. *Journal of Optics*. <https://doi.org/10.1007/s12596-023-01581-4>.
- [19] Radojicic, V., et al. (2023). Advancements in computer vision applications for traffic surveillance systems. In *Sinergija University Scientific Conference with International Participation*. Bijeljina.
- [20] Camara, F., Bellotto N., Cosar S., Nathanael D., Althoff M., Wu J., Ruenz J., Dietrich A., Fox, C. (n.d.). Pedestrian models for autonomous driving. Part I: Low-level models, from sensing to tracking. *IEEE Transactions on Intelligent Transportation Systems*.
- [21] Geetha Rani E., et al. (2024). OpenCV based enhanced criminal identification mechanism. In *2024 International Conference on Signal Processing and Advance Research in Computing (SPARC)*. 1–6. IEEE. <https://doi.org/10.1109/SPARC61891.2024.10828801>.
- [22] Ndubuisi O. J., et al. (2024). Digital criminal biometric archives (DICA) and public facial recognition system (FRS) for Nigerian criminal investigation using HAAR cascades classifier technique. *World Journal of Advanced Engineering Technology and Sciences*, 11(2), 29–43.
- [23] Agbemenu A., Yankey J., Ernest O. (2018). An automatic number plate recognition system using OpenCV and Tesseract OCR engine. *International Journal of Computer Applications*, 180, 1–5. <https://doi.org/10.5120/ijca2018917150>.
- [24] Gupta V., Dadsena S.S., Rao M. K. (2025). Automatic-number-plate-recognition ANPR. *International Journal of Innovative Science and Research Technology*, 10(6), 46–48. <https://doi.org/10.38124/ijisrt/25jun057>.
- [25] Komarudin A., Satria A. T., Atmadja W. (2015). Designing license plate identification through digital images. *Procedia Computer Science*, 59, 468–472.

- [26] Celiński I., Sierpiński G. (2019). Roundabouts as safe and modern solutions in transport networks and systems. In E. Macioszek R. Akcelik, G. Sierpiński (Eds.), *Transport Systems. Theory and Practice* 2018, 24–39. Springer. https://doi.org/10.1007/978-3-319-98618-0_3.
- [27] Celiński I. (2019). Metoda oceny oddziaływań między ruchem drogowym wewnątrz i poza strefą sterowania obszarowego [PhD dissertation, Politechnika Warszawska].
- [28] Daszykowski M., Siedlecka S. (2021). Analysis of pedestrian accidents in Poland. *The National Transport University Bulletin*, 1, 11–21. <https://doi.org/10.33744/2308-6645-2021-3-50-011-021>.
- [29] Budzyński M., Jamroz K., Mackun T. (2017). Pedestrian safety in road traffic in Poland. IOP Conference Series: Materials Science and Engineering, 245, 042064. <https://doi.org/10.1088/1757-899X/245/4/042064>.
- [30] Sicińska K., Zielińska A. (2022). Pedestrians' safety in Poland and use of reflective materials. *Transport Problems*, 17(1). <https://doi.org/10.20858/tp.2022.17.1.11>
- [31] Fekih M., et al. (2020). A data-driven approach for origin-destination matrix construction from cellular network signalling data: A case study of Lyon region (France). *Transportation*. <https://doi.org/10.1007/s11116-020-10108-w>.
- [32] Syffen G., Khalil A., Ramadan I. (2023). Origin destination matrix estimation based on traffic counts using fmincon function in MATLAB. *Journal of Al-Azhar University Engineering Sector*, 18(66), 57–73.
- [33] Plačiak J. (2023). Single-route OD matrix estimation using automatic passenger counting data: A case study in Geneva [Master's thesis, University of Twente].
- [34] Alberola P. (2024). Places and residential attractiveness: A systematic literature review. In Place attractiveness and image. A research agenda, pp. 97–121. <https://hal.science/hal-04594612>.
- [35] Sztuk A. (2023). Cities' attractiveness factors from the perspective of digital nomads. *Scientific Papers of Silesian University of Technology. Organization and Management Series*, 2023, 323–336. <https://doi.org/10.29119/1641-3466.2023.174.23>.
- [36] Hidalgo M. C., Berto R., Galindo M. P., Getreivi A. (2006). Identifying attractive and unattractive urban places: Categories, restorativeness and aesthetic attributes. *Medio Ambiente y Comportamiento Humano*, 7(2), 115–133.