Figure 4.10 shows the result of detecting terrain type and its semantic segmentation. However, elements such as the sky or cars have not been detected. It should be noted that the model is not focused on that type of element, only on the ground and possible obstacles. Figure 4.10-a shows the correct identification of the soil (grass) with an average precision of 88% concerning the original benchmark. On the other hand, Figure 4.10-b shows the recognition of the soil and an obstacle as surmountable; the average percentage of detection of the segmented space concerning the benchmark is 93%.

(a) Evaluation over dataset [192]. (b) Evaluation over dataset [90].

Figure 4.10: Assessing the Performance of the Trained Neural Network Model on Datasets [90, 192] for Terrain and Obstacle Detection.

Furthermore, a comprehensive comparison has been conducted, utilizing various metrics to evaluate the proposed method against different approaches in the state-of-the-art domain of vision systems for ground characterization, as summarized in Table 4.5. It shows that most prior research primarily focuses on terrain characterization, either with or without a subsequent step involving semantic segmentation. Notably, a noticeable gap in systems characterising specific components, such as debris, constitutes valuable information for decision-making in outdoor robotics.

PointCloud-based techniques for environment identification rely on traversability maps [108, 301]. However, this approach often falls short of providing environmental details, such as the stability of regions and the presence of obstacles. In contrast, the proposed method for terrain identification gives a more comprehensive understanding of the environment and its stability.

Table 4.5: Comparing the Proposed Terrain Identification Characterization Method with State-of-the-Art Approaches. Meets: ✓Fails: X

Work	Terrain Id	Obstacle Id	Benchmark Test	Sensor	Tested on Robots	Semantic Segmentation
[298]	✓	X	X	RGB	X	X
[289]	✓	X	✓	RGB-D	✓	X
[321]	✓	X	✓	RGB	X	✓
[284]	✓	X	X	RGB	X	✓
[292]	✓	X	X	RGB-D	✓	X
[316]	✓	X	✓	RGB	✓	X
[5]	✓	X	X	RGB	X	X
[38]	✓	X	X	RGB-D	✓	✓
[89]	✓	X	X	RGB	✓	✓
[185]	X	✓	X	stereo camera	✓	X
[162]	✓	X	X	RGB	X	✓
Proposed	✓	✓	✓	**RGB-D**	✓	✓

Comparative Analysis of Gait Pattern Adjustment Techniques

The comparative analysis is presented in Table 4.6. Previous research primarily focused on

adapting gait patterns for various robot types, not limited to quadrupeds. In the context of quadruped robots, many studies have concentrated on pattern adjustment using floor contact sensors to assess stability in a binary way (stable or not). Some studies have also incorporated RGB-D image processing to evaluate spatial depth and adapt locomotion accordingly.

In contrast, works involving hexapod and spider robots have been able to modulate their gait patterns, often employing alternate tripod gaits based on information from RGB sensors to accommodate different terrains. However, this chapter's approach, which involves gait pattern regulation through CNN-visual processing, shows a new approach in the literature.

Table 4.6: Comparison of gait pattern adjustment works.

Work	Robot	Visual Terr. Det.	Visual Obst Det.	Pattern	Test Real/Sim
[95]	Quadruped	X	X	2-2 / 3-1	Real
[320]	Quadruped	X	X	2-2	Sim
[3]	Quadruped	X	✓	2-2	Real
[317]	Hexapod	X	X	alternate tripod	Sim
[158]	Quadruped	X	X	2-2	Study
[292]	Spider	✓	X	alternate pairs	Real
[316]	Spider	✓	X	tripod	Real
[307]	Hexapod	X	X	tripod	Real
[99]	-	X	X	pattern extraction	Real
[52]	wheel-legged	X	✓	hybrid	Real
Proposed	**Quadruped**	✓	✓	**2-2/3-1**	**Real/Sim**

4.1.7 Discussion

This section introduces and substantiates a novel approach for overcoming unstructured terrain with a quadruped robot. This approach leverages pre-defined gait patterns based on the robot's model. It incorporates an automated process for recognizing and characterizing the environment, enabling autonomous adjustment of these gait patterns.

Exploring bio-inspired locomotion systems in quadruped animals has facilitated their emulation by real robotic systems. These natural movements and gait patterns have been integrated with intelligent algorithms to adapt locomotion in unstructured environments.

A simulation stage has proven instrumental in validating the replication of dog-like gait patterns and assessing their efficacy across various terrains featuring obstacles. Consequently, it was determined that the 2-2 alternate gait pattern is the most suitable for navigating environments littered with debris, which was integrated into the central gait pattern generator, forming the fundamental basis for forward motion.

The utilization of CNN for autonomous visual terrain recognition and obstacle characterization has outstanding precision, surpassing 90%, in locating obstacles within dynamic real-world settings. Comparative assessments against state-of-the-art approaches and benchmark evaluations consistently highlight the optimal performance of this method. By leveraging RGB images, this proposed technique enhances our comprehension of the environment, particularly regarding terrain identification and stability assessment. In contrast, conventional ones use pointclouds but lack crucial environmental details, especially about stability.

4.2 Environment Advanced Analysis and Strategic Movement

This section aims to refine the robot's motion through the terrain more precisely and inform, operating one leg at a time - using as a basis the **1-3 gait pattern** outlined in Section 4.1. This approach maintains a support polygon of three legs on the ground, which generates a concerted movement in conjunction with the fourth leg to ensure the body's stability.

Extracting the maximum amount of local information from the terrain is necessary to generate this specific leg motion and traverse the unstructured area. This involves identifying not only the environment and obstacles, as discussed in the previous section, but also delving deeper into the characterization of elements and obstacles. The primary approach to identifying obstacles and selecting the optimal **foothold** for the robot to place its leg relies on a human-based criterion, considering the stability probabilities of each element that constitutes the terrain.

To address the problem of probabilistic characterization, a CNN has been trained. The images used in this training incorporate the defined stability percentage criteria as detailed in Subsection 4.2.1. For the method's applicability in the quadrupedal robot, a subsequent phase of discretization in image regions is required, grouping pixels according to detections, associating the discretized regions with the actual depth of the environment, and transforming the coordinate systems from RGB+depth-camera to world-robot.

4.2.1 Probablistic Terrain Characterization Based on Semantic Criteria

CNN Processing

Based on the initial approach followed in the preceding section, it has been empirically established that the robot exhibits varying responses when confronted with the same obstacle on different occasions. In other words, categorising an obstacle is not absolute; therefore, the distinction between "superable" and "No superable" does not accurately represent reality in the most precise manner possible. The solution implemented involves expanding these labels to make them more specific, thus transitioning from an absolute annotation to a probabilistic annotation criterion guided by a human-centric approach.

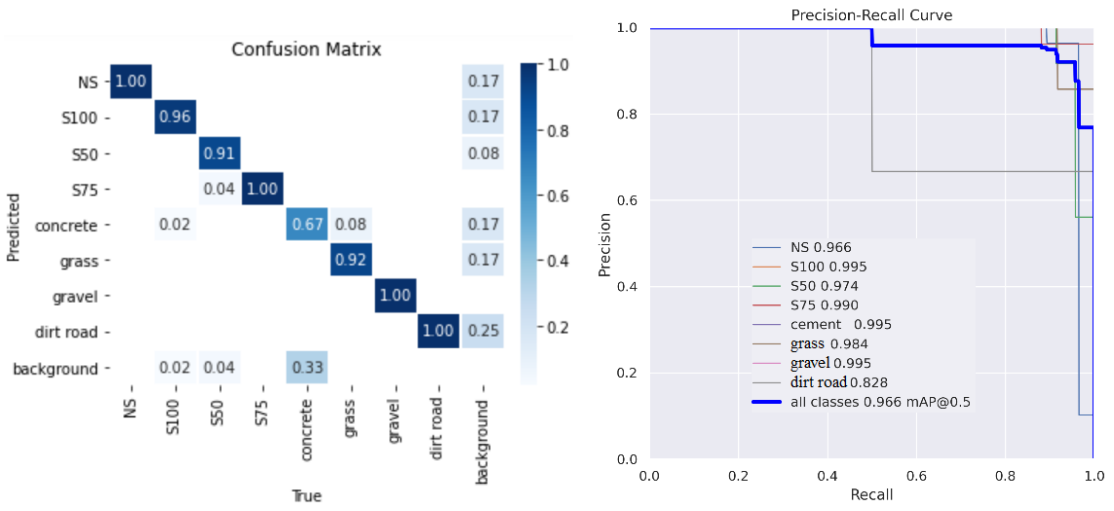
The dataset used is the same as in the previous section, available at the following URL `https://github.com/ChristyanCruz11/Terrains`. Regarding the classification of obstacles, three criteria have been employed to define them:

1. Firstly, concerning size, the superable percentage decreases as the obstacle becomes larger.
2. Next, the object's shape has been taken into account. Objects with more curved geometries or steeper inclines have been considered less stable.
3. Lastly, concerning the obstacle's material, it has been considered that smoother and more homogenous surfaces improve robot stability, in contrast to obstacles with higher surface irregularities.

Based on these criteria, the general terrain composition labels, such as "gravel", "dirt road", "grass", and "cement", have been extended with the following annotations:

- **"NS"** - Zero probability of obstacle traversal. This label applies to obstacles larger than the robot itself, meaning that the leg's range of motion does not allow it to be placed on the obstacle, such as trees.
- **"50%"** probability - This label indicates the possibility of obstacle traversal but with limited certainty.
- **"75%"** probability - Obstacles in this category are more likely to be surmounted than in the previous category. Still, it cannot be guaranteed that they will be surmounted on every occasion.
- **"100%"** probability - These obstacles are considered non-problematic for the robot, and it should not encounter any disruptions in its movement when facing an obstacle classified with this label.

Figure 4.11 illustrates the performance evaluation of the trained CNN model, with the confusion matrix shown in Figure 4.11-a. In this matrix, the cells on the main diagonal, representing the number of true positives, exhibit a high percentage for all labels, confirming high reliability. Furthermore, there is a very low percentage of false negatives and false positives, indicating that when the model detects an object, the probability of it making a detection error is very low. However, it is observed that there are detections made by the model that correspond to the image background, not to real terrain or obstacles. This is particularly noticeable in the case of terrains, such as soil. On the other hand, when the model fails to make a detection, the probability of there being a detection of that type is also low.

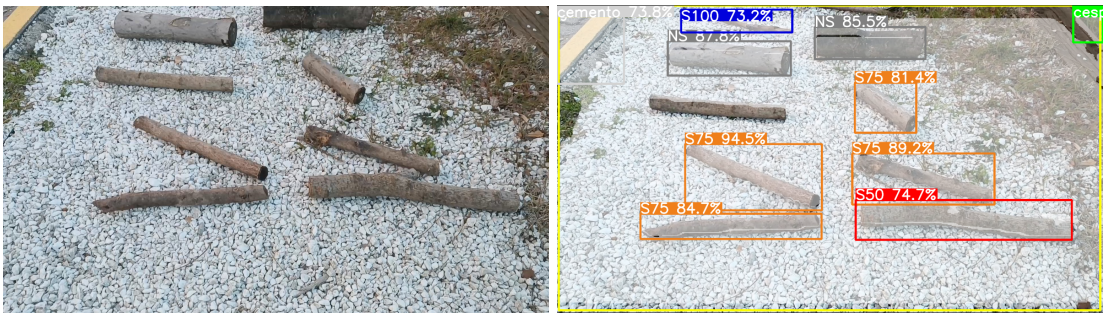


(a) Cconfusion matrix (b) Precision-Recall curve

Figure 4.11: Evaluation performance of trained CNN model.

Figure 4.12-a displays an original scene of a terrain with gravel and obstacles of various sizes, while Figure 4.12-b represents the detected classes. Initially, gravel terrain is identified, and on top of it, various obstacles are detected with surmountability percentages of 100%, 75%, and 50%, along with the corresponding precision percentages for each detection. The preliminary results are highly reliable in terms of the realism of the environment.

The results related to the training of the neural network, as well as the algorithms implemented for generating the probabilistic terrain discretization (Section 4.2.2) and the iden-



(a) Original ground terrain with obstacles

(b) Processed ground terrain, a series of obstacles and gravel terrain were identified.

Figure 4.12: Original and Post Processed Terrain using trained CNN.

tification of the best foothold (Section 4.2.3), are available in the ROBCIB research group GitHub repository `https://github.com/Robcib-GIT/probabilistic-terrain-perception`.

4.2.2 Probabilistic Generation of Discretized Terrain.

Adjustment 2D-3D between Processed and Depth Image

In probabilistic terrain generation, the objective is to obtain a **dynamic matrix** representing the discretized image corresponding to the perspective of the real terrain. This matrix contains information that can be used in the decision-making process for the robot's foothold. The subsequent step involves discretising the terrain-processed 2D image and projecting them following a perspective according to the depth camera.

First, groups of pixels from the original image (as exemplified in Figure 4.13-a) are associated in subregions, represented by a single point. This division is carried out to characterize that area based on its conditions. The division considers parameters such as the image size (640x480) and the segmentations previously performed by CNN. Additionally, it is known that the scale factor of the image, in meters, is 0.0017, as defined by Equation 4.13. All these pixel values update the original dynamic matrix $P.T.M[m, n]$.

Considering this value, it has been determined that an appropriate image division for analysis involves segmenting the image into **10x10 pixel** regions, roughly equivalent to 2x2 cm squares (as illustrated in Figure 4.13-b). This division could be adjusted to suit the **robot's contact area** and may be larger if more computational processing is required.

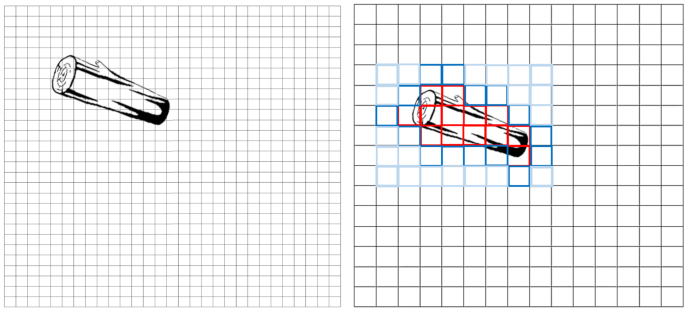
$$d_{real} = f_{scale} * number\ of\ distance\ pixels \tag{4.13}$$

Figure 4.13-b also depicts the obstacle evaluated by the CNN and the descending gradient of obstacle traversal probabilities (in colour). The region in red represents the most challenging area to overcome, as identified by the CNN, with its probability of adjusting to the obstacle (higher obstacles correspond to greater difficulty). In the surrounding areas, represented in blue and later in light blue, the probabilities of overcoming the region are higher, and they are depicted with varying intensities following the local analysis criteria defined in equation 4.14.

$$P(i,j) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} (P(i'+m, j'+n)) \tag{4.14}$$

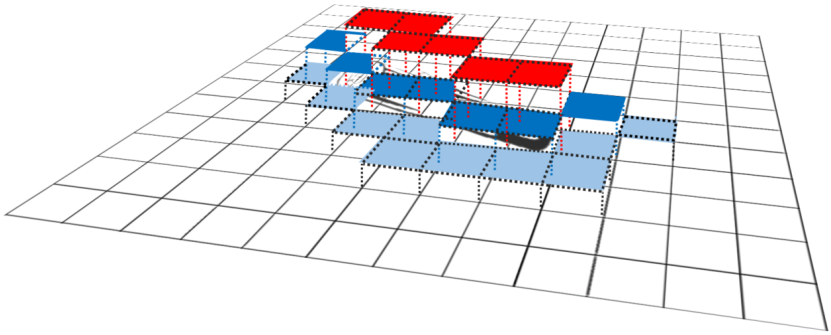
However, as the images are captured with a certain inclination, the image's perspective causes the pixels at the far end to represent a shorter real distance than the pixels at the beginning. Therefore, the parameter obtained for the real distance between pixels has been multiplied by a factor (Equation 4.15), resulting in the creation of a mesh perspective, as depicted in Figure 4.13-c. This provides a clearer view of the probabilistic gradient in the area where an obstacle is located.

$$d_{real} = perspective\_coef * d_{real} \tag{4.15}$$



(a) Squematic original mage.

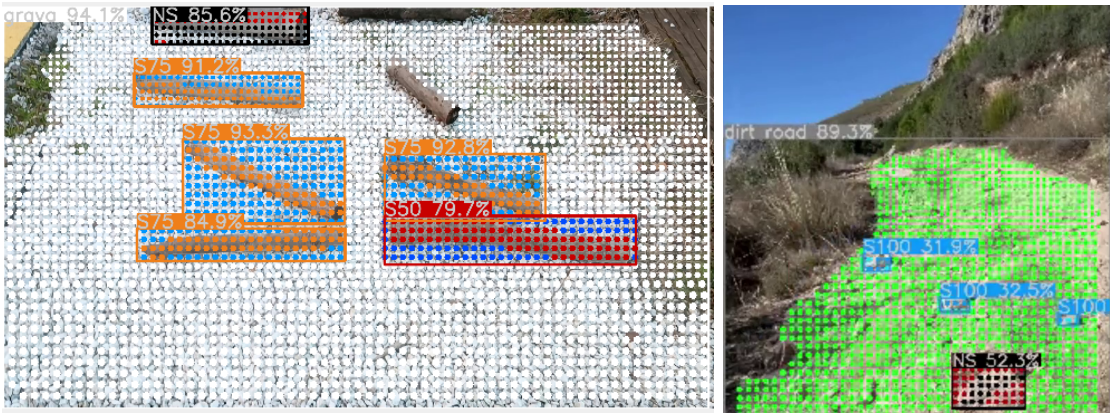
(b) Discretization, ostacle association and gradient in the surrondings representation.



(c) Projection of the discretized terrain associated with the real depth of the environment.

Figure 4.13: Descriptive process of generating the probabilistic terrain map.

The visual representation of this discretization is shown in Figure 4.14. It is important to emphasize that the points represent the centres of each region, not their surface area. Additionally, it's notable that a perspective is associated with the areas corresponding to the upper part of the image (background of the scene). Both terrain (gravel, dirt road, etc.) and obstacles are characterized probabilistically (0 - 100% probability of overcoming), generating a gradient in their surrounding areas. This information serves as a basis for selecting footholds for legged robots.



(a) Probabilistic Discretized Gravel with Obstacles Terrain from Figure 4.12-a

(b) Probabilistic Discretized Dirt Road Terrain with Rocks (Scene Figure 4.8-d).

Figure 4.14: Probabilistic Terrain Discretized from depth information and CNN-preprocessing. Results available in link.

4.2.3 Advance based on Foothold Definition

Kinematic Adjustment between Processed Terrain and Robot

The following reference systems and their relationship are defined to facilitate advance through the environment. Figure 4.15 provides a detailed representation of the locations of these reference systems:

1. **Robot reference system:** All positions and orientations of the robot joints are referred to concerning this system, as described in Section 4.1.1. The origin of this system is located at the robot's torso; however, the point of contact is situated on the robot's legs. As a result, an additional transformation is necessary to account for this discrepancy.
2. **Camera coordinate system:** The centre of this reference system is positioned at coordinates (250, 0, 0) [mm] concerning the global reference system. This corresponds to the distance between the robot's centre and the edge in the "x" direction at the front end.
3. **Depth sensor coordinate system:** Depth calculations (real-world information) for a specific point in the image are provided about a coordinate system imposed by the camera manufacturer. Although the RGB camera and the depth sensor are part of the RGB-D set, a fixed separation distance exists between them, as detailed in Figure 4.15.

The camera's reference system is rotated 90° counterclockwise around the y-axis of the global reference system. This arrangement makes the y-axis of all three reference systems parallel. In contrast, the z-axis of the camera coordinate system and the depth sensor coordinate system is rotated 90° counterclockwise concerning the global reference system.

Once the coordinate systems are defined, the task at hand is to transform a point in the image, initially described as a two-dimensional pixel in the reference system of the depth

sensor (Frame **RGB-D cam** in Figure 4.15), into a point in three-dimensional space, about the robot's coordinate system (Frame **World** in Figure 4.15).

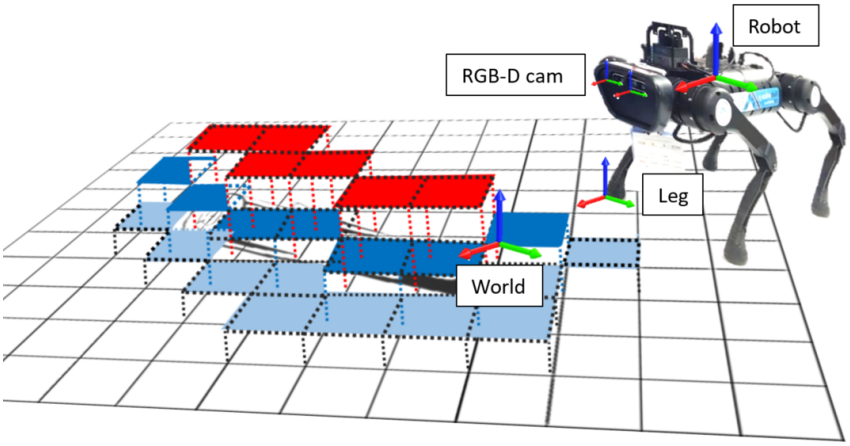


Figure 4.15: Definition of the reference systems involved in the movement process.

Using α as the depth value for a pixel to which the scale factor has already been applied, equations 4.16–4.18 are employed to calculate its depth coordinates:

$$x_{world} = \alpha . \frac{(x_p - x_f)}{f_x} \tag{4.16}$$

$$y_{world} = \alpha . \frac{(y_p - y_f)}{f_y} \tag{4.17}$$

$$z_{world} = \alpha \tag{4.18}$$

Where (x_p, y_p) correspond to analized pixel, (x_f, y_f) focal point coordinates, (f_x, f_y) focal distance and $(x_{world}, y_{world}, z_{world})$ are the coordinates concerning the Depth camera frame. Following this, based on the coordinate systems that have been defined, it will be necessary to perform a translation (according to equation 4.19) from the depth sensor's reference system to the RGB camera's reference system.

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} = \begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \end{bmatrix} - \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{4.19}$$

A translation and rotation of the camera's reference system to the global reference system (of the robot) are required. Consequently, the problem must be divided into a rotation (equation 4.20) and a translation (equation 4.21).

$$\begin{bmatrix} x_{c\_rot} \\ y_{c\_rot} \\ z_{c\_rot} \end{bmatrix} = \begin{bmatrix} cos(\varphi) & -sin(\varphi) & 0 \\ sin(\varphi) & cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\beta) & -sin(\beta) \\ 0 & sin(\beta) & cos(\beta) \end{bmatrix} . \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} \tag{4.20}$$

$$\begin{bmatrix} x_{ARTUR} \\ y_{ARTUR} \\ z_{ARTUR} \end{bmatrix} = \begin{bmatrix} x_{c\_rot} \\ y_{c\_rot} \\ z_{c\_rot} \end{bmatrix} - \begin{bmatrix} m \\ n \\ p \end{bmatrix} \tag{4.21}$$

Where $(x_{c\_rot}, y_{c\_rot}, z_{c\_rot})$ are the coordinates of the point concerning the rotated camera reference system, and $(x_{_{ARTUR}}, y_{_{ARTUR}}, z_{_{ARTUR}})$ are the coordinates of the point to the global robot reference system.

Foothold Selection

The developments described in the preceding section have effectively characterised the environment. However, in implementing it for the robot's traversal through the terrain, it is necessary to devise an algorithm for decision-making that guides the selection of potentially stable zones to facilitate the robot's movement.

To determine the optimal **foothold**, the starting point will be the dynamic matrix $P.T.M$, which stores the characteristics of the discretized terrain, primarily stability probabilities and their respective distances relative to the camera's reference system. For this purpose, an algorithm based on point scores and matrix attributes has been implemented, following these three criteria:

1. **Distance from the leg to the analyzed point**

The robot's maximum feasible stride length for maintaining controlled locomotion is 8 cm. Ideally, the robot will always take steps of this length. However, depending on other variables, it will decide whether taking steps of this length is advantageous or safer to shorten the stride. A logic system has been devised that serves two functions to achieve this: first, it filters out points more than 8 cm away from the leg about to move, automatically discarding them. Then, the point undergoes a logic process that assigns a specific score based on its distance from the leg, following Table 4.7. Consequently, the greater the distance between the point and the leg, as long as it is less than 8 cm, the higher the score it receives. Figure 4.16 represents the concept for the first step to any of the black points calculated in the defined range.

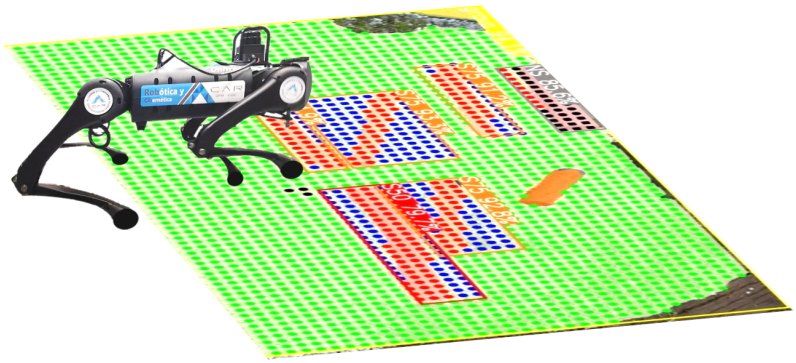


Figure 4.16: Movement of the front-right leg to the black calculated point in the probabilistic terrain.

Table 4.7: Scoring criterion based on distance.

Distance [cm]	0<d<1	1<d<2	2<d<3	3<d<4	4<d<5	5<d<6	6<d<7	7<d<8
Score	5	10	15	20	30	40	45	50

This implementation would be straightforward from the perspective of algorithm **adaptability to any quadrupedal robot model**. In this regard, one would need to adjust

the distances based on the specific robot's maximum range and establish scoring based on experimentation and testing.

2. **Stability percentage related the point**

Based on the results obtained in the previous section, different terrains have been classified according to their suitability for locomotion. A scoring system has been developed based on the segment to which the point belongs (Table 4.8). Additionally, a screening process has been implemented so that if the point's detection corresponds to the "NS" (Not Suitable) label, it is automatically discarded and no longer considered a candidate for a support point. This approach ensures that only points associated with terrain segments deemed suitable for locomotion are considered in the decision-making process regarding the foothold for the robot's movement.

Table 4.8: Scoring criteria based on detection.

Class Detected	**Score**	**Class Detected**	**Score**
Grass	50	Cement	40
Dirt Road	30	Gravel	20
Near S100	16	S100	10
Near S75	8	S75	5
Near S50	4	S50	2
Near NS	2	NS	–

3. **Stability percentage related to the surrounding points**

First, it was necessary to define a distance threshold beyond which nearby points are no longer considered influential in the robot's motion. The computational capacity of a computer in anticipating chess moves served as a reference for this determination.

In this context, it was concluded that considering up to 12 moves ahead is an acceptable number. Given a maximum reach of 8 [cm], this implies that the algorithm will analyze all points located within a range of fewer than 96 [cm]of distance (in a straight line and in all directions) from the robot. These points are considered concerning the lower centre of the image, the region of overlap between the camera and the robot's vertical axis.

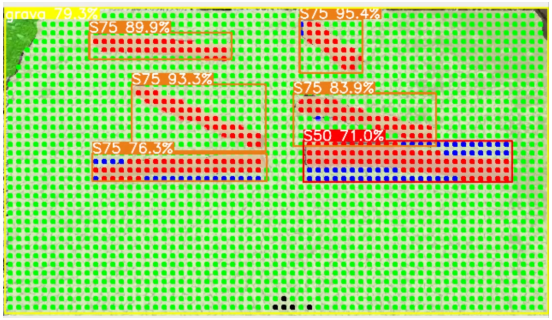
Furthermore, this function of surrounding points would deduct scores in cases with unstable areas or insurmountable obstacles near the analyzed point. The points to be deducted are the same as those added when evaluating the point's colour, but they are multiplied by a factor based on the distance from the point. As expected, this factor is closer to 1 as the distance between points decreases.

Pseudocode 2 summarizes the logic of the algorithms implemented for this section and the sequence of their execution for proper functionality. Since the generation and implementation of [1-3] gait patterns were detailed in the previous section, this section focuses on creating probabilistic terrain maps. Each function specifies the input data it receives and the output it provides.

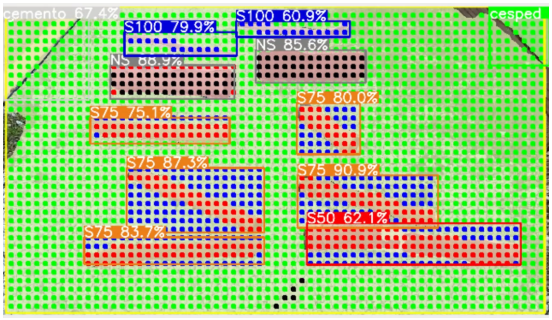
Figure 4.17 depicts the sequence of advances through the scenario shown in Figure 4.12-a. Different instances are illustrated, and in each of them, the algorithm-selected potential candidate points for leg placement are marked in black.

Algorithm 2: Quadruped Movement based on Probabilistic Terrain Analysis

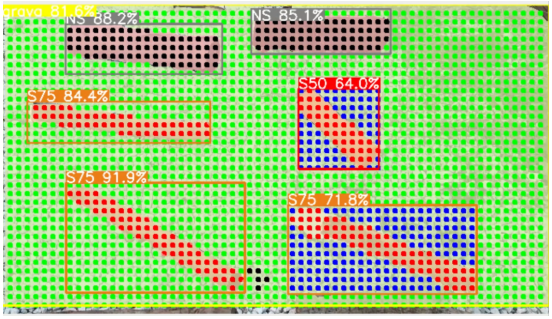
Data:
$Robot_{Joints-pose} \leftarrow q_{[1-12]}$
$Robot_{pose(xyz)-orient(rpy)} \leftarrow IMU_{estimation}$
$im_{Depth} \leftarrow [1280x720 - 16 - bit]$ depth resolution
$im_{RGB} \leftarrow$ RGB image $[640x480]$
Result:
$[q_{d_{[1-12]}}, \dot{q}_{d_{[1-12]}}, \tau_{[1-12]}]$
Function *Terrain_Characterization* (im_{RGB}) ▷ *CNN vision-based characterization*
  $CNN_{based-algorithm} \leftarrow im_{RGB}$
  return $[terrain_{ID}, n\_obstacles_{[class,size,pose]}]$
end
Function *Prob_Terrain* ($im_{Depth}, CNN\_results$)) ▷ *Dynamic Probabilistic Terrain*
  $Dynamic\_Matrix \leftarrow$ Project ($im_{[Depth]}$ in $im_{[RGB]}$)
  $Dynamic\_Matrix \leftarrow$ Regions discretize($Dynamic\_Matrix$)
  foreach *n_obstacle* of *boundingBoxes* **do**
    $Local\_Points_{[mxn]} \leftarrow gradient\_points\, araund\, Obstacle_{[i]}$
    $Dynamic\_Matrix \leftarrow append(Local\_Points)$
  end
  return $Dynamic\_Matrix$
end
Function *Foothold_Selection* ($Dynamic\_Matrix$) ▷ *Foothold Point Selection*
  $local\_points_{[1,..,n]}$ in $Dynamic\_Matrix_{[320+\Delta i, 480-\Delta i]}$ ▷ 1. Dist. criteria
  $Foothold\_Point_{[X,Y,Z]} \leftarrow Ponderation\_Alg(local\_points_{[1,..n]})$ ▷ 2-3. Prob. criteria
  return $Foothold\_Point_{[X,Y,Z]}$
end
Function *Axis_Adjustment* ($Foothold\_Point_{[X,Y,Z]}$) ▷ *Coordenate Systems Adjustment*
  $Foothold\_Adjust \leftarrow (Robot\_Coord, Camera\_RGB\_D\_Coord, World\_Coord)$
  return $Foothold\_Adjusted_{[X,Y,Z]}$
end
Function *Gait_Pattern* ($Foothold\_Adjusted_{[X,Y,Z]}$) ▷ *1-3 Gait Pattern Generator*
  $[q_{d_{[1-12]}}, \dot{q}_{d_{[1-12]}}, \tau_{[1-12]}] \leftarrow IK\_Solver(Foothold\_Adjusted_{[X,Y,Z]})$
  return $[q, \dot{q}, \tau]$
end
while im_{RGB} *and start* **do**
  $Robot \leftarrow stand\_position$
  eval($TERRAIN\_CHARACTERIZATION \leftarrow im_{[RGB]}$)
  eval($PROB\_TERRAIN\_ \leftarrow im_{[Depth]}, [terrain - obstacles]$)
  if $Dynamic\_Matrix$ *not null* **then**
    eval ($FOOTHOLD\_SELECTION \leftarrow Dynamic\_Matrix$)
    eval ($AXIS\_ADJUSTMENT \leftarrow Foothold\_Point_{[X,Y,Z]}$)
    if *Obstacles in terrain not null* **then**
      eval ($GAIT\_PATTERN \leftarrow Foothold\_Adjusted_{[X,Y,Z]}$)
    end
    $Robot_{Controller} \leftarrow [q_{d_{[1-12]}}, \dot{q}_{d_{[1-12]}}, \tau_{[1-12]}]$
  end
  $Robot \leftarrow stand\_position$
end



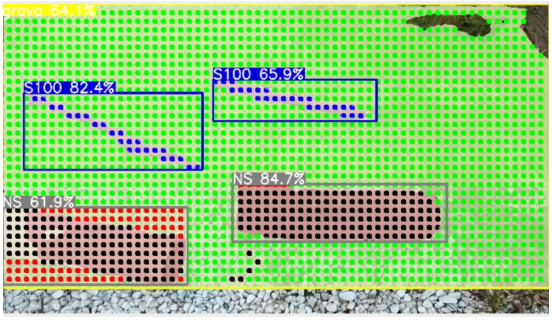
(a) Initial statement for points definition in the probabilistic terrain.



(b) Area close to the first obstacles.



(c) Intermediate step through obstacles.



(d) Passing through an area of insurmountable obstacles.

Figure 4.17: Selection of foothold at different stages of the field advancement process. Results available in link.

4.2.4 Traversability based-on Elevation Maps and Probabilistic Terrains

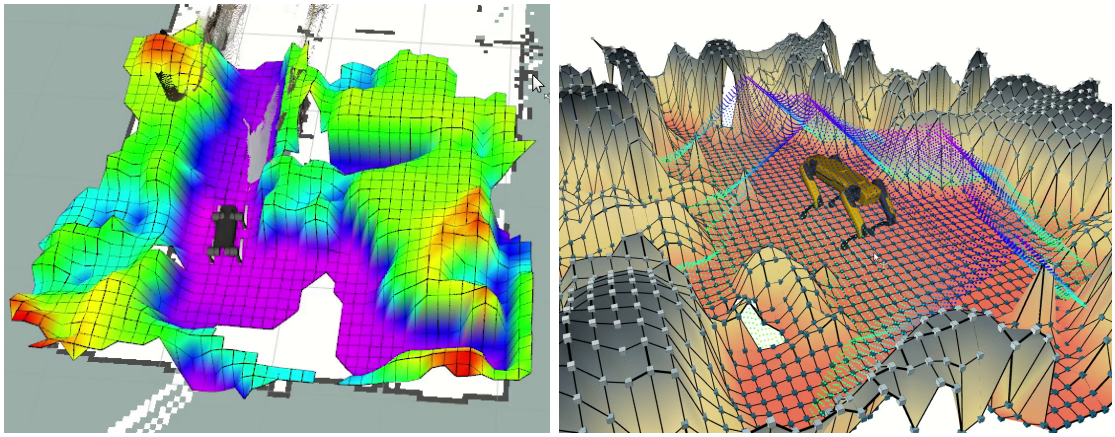
The probabilistic information regarding terrains and support points can be associated with existing frameworks in navigation systems for unstructured environments, enhancing their information. This can improve the current navigation methods based on height (captured from sources such as pointclouds), initially evaluated as rigid zones.

Figure 4.18 illustrates the integration of the elevation maps framework (following the repository `https://github.com/anybotics/elevation_mapping`) as a first step for the general integration between the proposed method. The first case involves integration with the ARTUR robot, while the second case is relative to the SPOT robot.

4.2.5 Discussion

The improved terrain identification achieved through CNN enables support point selection based not only on the robot's kinematic characteristics and constraints but also on the prior environmental classification. Table 4.9 summarizes the aforementioned points.

In the current state of the art, conventional methods for providing the stability control of quadrupedal robots are based on reactive approaches that endeavour to restore equilibrium when it appears compromised. These methods evaluate the terrain as a rigid body to define the robot's trajectory but do not consider the potential instability of these footholds. The implemented method probabilistically assesses the terrain and its elements to select those



(a) ARTUR elevation Map generated. (b) SPOT elevation Map generated.

Figure 4.18: Elevation maps based-on pointclouds.

Table 4.9: Comparison of works related to support point selection for quadrupedal robots.

Work	**Identification and Classification of Terrain**		**Selection of Support Point**	
Analysis	Kinematic	Machine Learning	Kinematic	Machine Learning
[81]	✓	X	✓	X
[276]	X	X	✓	X
[111]	✓	X	✓	X
[148]	X	X	✓	X
[96]	✓	X	✓	X
[299]	X	X	✓	X
[64]	X	X	✓	X
[172]	✓	X	✓	X
[138]	X	X	✓	X
Proposed	X	✓	✓	✓

that offer higher stability assurance while concurrently promoting forward progression towards the desired destination in the robot's gait.

Using terrain mapping methods via elevation maps to identify the most stable foothold appears effective. Nevertheless, these analyses lack information regarding the condition or material composition of the terrain. In this context, the implemented method provides an alternative parallel to terrain mapping, demonstrating exceptionally high efficacy in terrain interpretation and classification.

Furthermore, the advantage of this method over those developed using elevation / traversability maps is that the approach presented in this document does not rely on robot-specific parameters. It simply analyzes the environment from a perspective entirely independent of the robot. This ensures a more objective perception of the surroundings, resulting in a representation closer to reality. Additionally, in line with this approach, this method has the potential to be universally applicable to all types of quadrupedal robots. In contrast, elevation/traversability maps, being specific to each robot, lack such versatility.

4.3 Strategy for Ground Elements Removal

In the context of this doctoral thesis, the discussion progresses to focus on sensory processing, furthering comprehension of the environment and the decision-making process. This subsection delves into an additional phase beyond the initial two phases. Specifically, it explores the capacity of a legged-manipulator robot to undertake an environment-clearing task involving identifying and removing elements that impede mobility. These impediments may include objects tending to **generate instability** during walking or items that could suppose a risk to the robot and its human operator, particularly in cases involving **explosive materials**.

While legged-manipulator robots will be addressed more comprehensively in Section 7.2 of this thesis, this particular section, which delves into strategies for navigating rough terrain, concentrates on the algorithms for sensory processing. The primary objective is to achieve robust identification and characterization of elements that can be removed, distinguishing them from those that cannot be removed due to their size or attachment to the ground.

The proposed method for this development involves processing pointclouds and images through CNN for identification. Additionally, as an essential element for environmental grasping and clearance, the robot SPOT, equipped with a manipulator, was employed, along with the Software Development Kit (SDK) provided by Boston Dynamics for object grasping. This project was undertaken during the International Doctoral Stay at the **Czech Technical University (CTU) in Prague** within the Faculty of Electrical Engineering, Department of Cybernetics - **Center for Machine Perception**.

The comprehensive repository for this development can be found at: `https://github.com/ctu-vras/spot_collector`. The implemented method comprises two main stages, as shown in Figure 4.19:

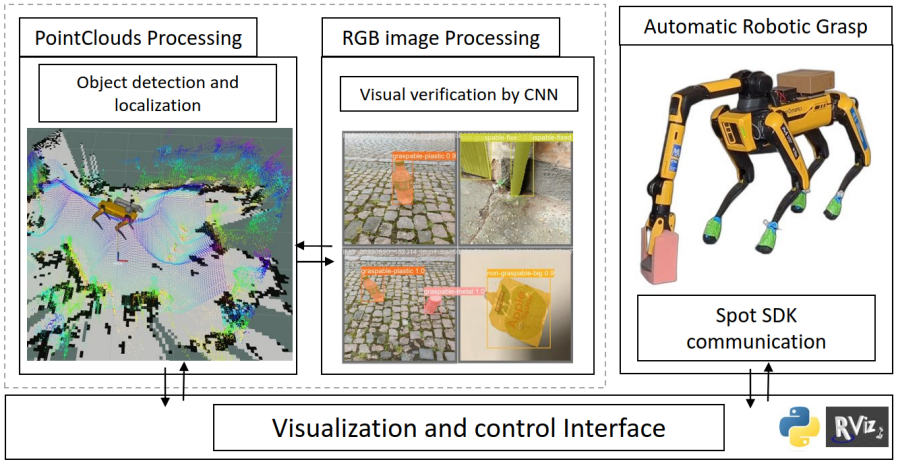


Figure 4.19: Arrangement of interconnections among subsystems in the proposed approach.

1. **Object Detection and Localization**: In this initial stage, the system robustly detects and locates objects to be picked up. It leverages both pointclouds for precise environmental cluster identification and RGB images processed by a trained CNN to assess the graspability of the detected clusters.
2. **Robot Control and Communication**: The second stage utilizes the Spot SDK to command the robot, enabling it to execute the collection actions efficiently.

These two stages interact through a structured system divided into three distinct blocks. The first block centred around perception. The second block employs the Spot SDK for robot control, while the third block acts as an interface that integrates and facilitates communication between the first two blocks. This interface also offers the operator visual verification during execution and supports decision-making. The system's overall structure and interaction are depicted in Figure 4.19.

4.3.1 Sensory Processing and Object Localization

Spot robot has five sets of stereo cameras; depth images from these cameras are published as `sensor_msgs/Image` on topics /spot/depth/cameraname/image. Using a single point-cloud to detect graspable objects is advantageous because one object could be placed at the edge of two depth scans.

The individual depth images can be stitched together in a common tf frame body. The lidar_filter node concatenates the converted pointclouds into one. The new Poincloud combined topic is published as `sensor_msgs/PointCloud2`, and is shown in grey.

Central Interface

A central interface (Figure 4.20) manages the process information and decision-making of the entire system. The developed system is encapsulated within a single ROS (Robot Operating System) package comprising multiple nodes. The source code is accessible at `https://github.com/ctu-vras/spot_collector`.

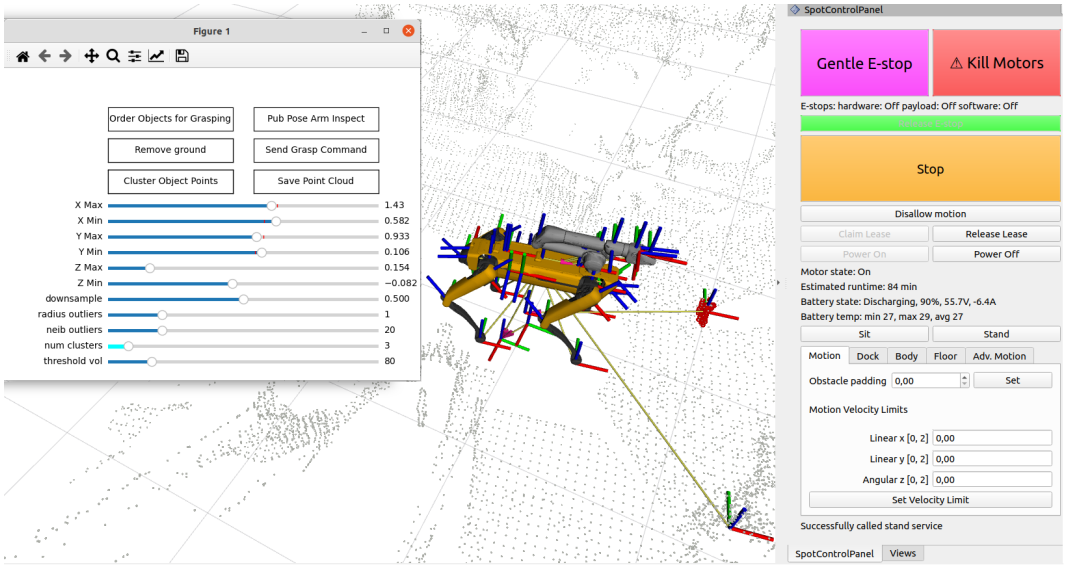


Figure 4.20: Implemented Interface for PointCloud Parameter Control (left) and provided RVIZ control events for SPOT setup (left).

The interface shown in Figure 4.20 has been established to manage the adjustment of pointcloud ($P_{[XYZ]}$) variables to suit the environment. It features a series of sliders for manually modifying the maximum and minimum values within the operational range $[Xmax - Xmin; Ymax - Ymin; Zmax - Zmin]$, outliers reduction following equation 4.22.

$$Pro(P_i) = 1 - \frac{1}{k} \sum_{q_j \in K(p_i)} exp(\frac{-dist(P_i, Q_i)}{d_i}) \tag{4.22}$$

Where P_i is the analyzed point and, Q_i is the neighbourhood, k is the number of neighbour points, this method is based on local density [218] and defines the probability that the point is an outlier.

The interface includes different actions related to each button:

1. Remove ground: execute the function to delete the ground plane.
2. Cluster Object Points: execute the DBSCAN function to obtain the clusters already filtered by the threshold.
3. Order Objects for Grasping: minimizing the distance of the trajectory travelled.
4. Pub Pose Arm Inspect: executes the function that positions the arm with a perspective towards each cluster to inspect it by visual image.
5. Send Grasp Command: calls the multigraph service to collect all detected items.
6. Save Point Cloud: save the current pointcloud.

PointClouds items detection

This development uses the `sensor_msgs/PointCloud2` topic filtered, which is converted to vector3d format using the function **o3d.geometry.PointCloud()**. The operations applied to this new Pointcloud consist mainly of eliminating statistical outliers, downsample and delimitation of the field of work (x,y,z) according to the specific requirement.

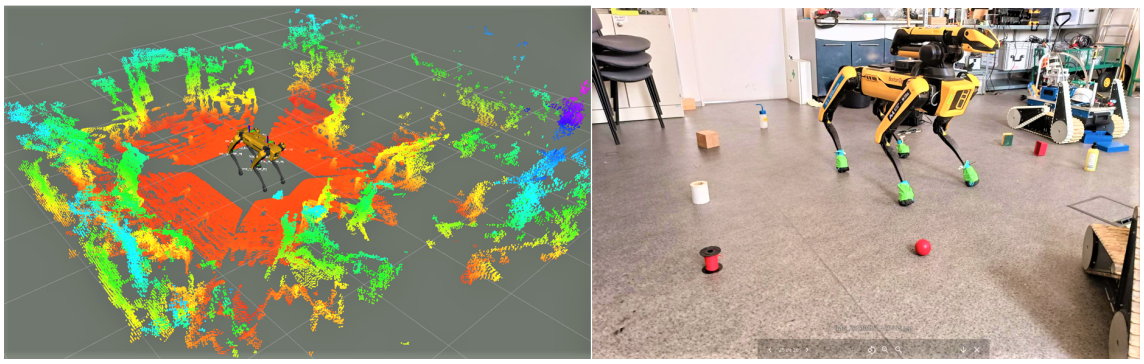
Figure 4.21-a shows the robot and the original pointcloud in false colour (referenced to the vertical axis) concerning its centre. Once the work area is defined (Figure 4.21-b shows a restricted area of 4x2 meters) and to search for the elements (clusters) around it, it is necessary to eliminate the ground component through the segment_plane function or directly restrict the height range in z.

After removing elements from the pointcloud, the clustering process is initiated using the DBSCAN algorithm [266]. Subsequently, the pointclouds whose size exceeds the parameter set by the user through the sliders are filtered out. For these new clusters, corresponding transformation frames (TFs) are generated (Figure 4.21-c), resulting in the relative positions of potential candidate objects, whose position is defined following the relationships established by homogeneous transformation matrices (equation 4.23).

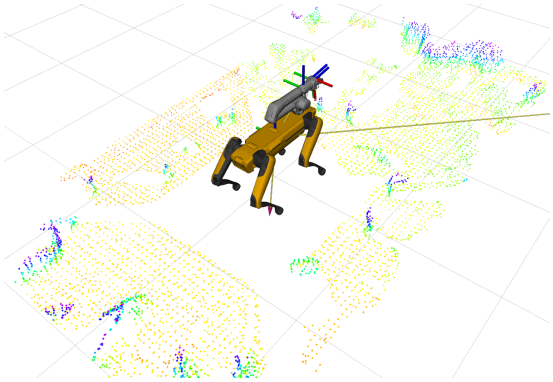
$$Grasp\_OBJ\_List[] = [^{item=1}\mathrm{T}_{Spot}, ..., ^{item=n}\mathrm{T}_{Spot}] \tag{4.23}$$

4.3.2 Verification Elements Phase

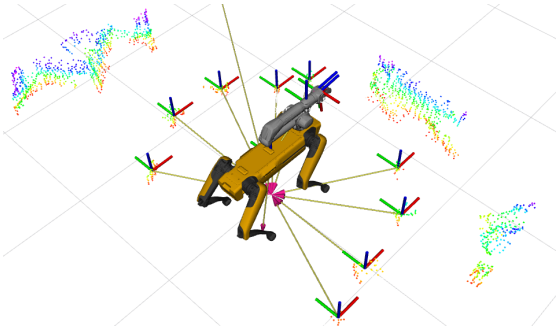
After establishing the relative positions of objects to be picked up concerning the robot, a verification stage becomes essential for each item. Multiple factors could render grasping



(a) (Left) Combined Pointclouds from the five depth cameras of Spot. (Right) Scenario with various elements.



(b) Bounded Pointcloud Processing within a 4x2 Meter Region.



(c) Detected Object Positions.

Figure 4.21: Sensorial Processing and Identification of Potential Objects of Interest.

impossible, primarily when objects are fixed, or their weight exceeds the robot's payload capacity. This critical information is often challenging to deduce from the pointcloud data, and non-graspable objects should be removed from the initial list of items.

A verification phase has been integrated for each cluster detected in the preceding phase to address this challenge. This entails orienting the robot's gripper toward the object and processing the captured RGB image.

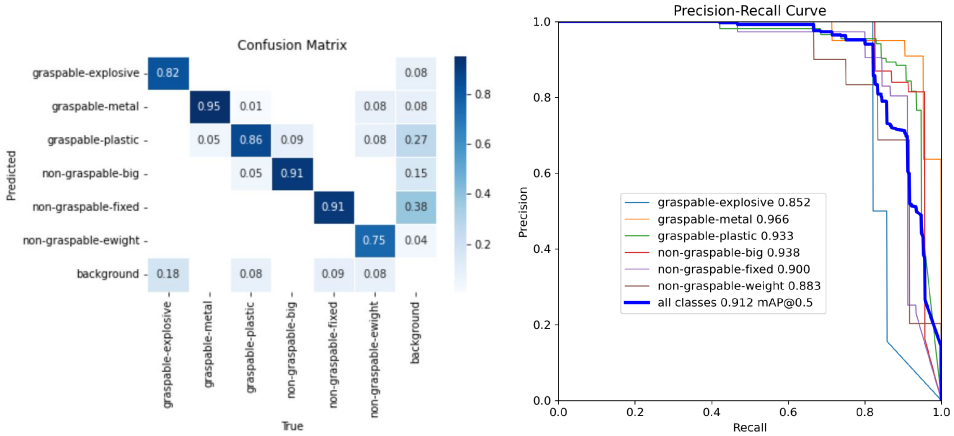
This image processing involves computer vision techniques and a CNN (YOLOv8) explicitly trained for item detection according to defined criteria. The CNN training involved generating a dataset of images and another partially used from publicly available repositories: Domestic Trash [168], Garbage Classification [46], Drinking Waste Classification [265].

The total number of images was 831. Data augmentation, involving approximately ±20% rotation and ±15% brightness modification, was applied to enhance detection robustness. This dataset was divided into training (87%), validation (8%), and testing (4%) subsets, with detailed information provided in Repository `http://ptak.felk.cvut.cz/vras/spot_collector/dataset/`. Labels were assigned to distinguish "graspable" objects and specify their characteristics as "plastic," "explosive," "metallic" (potentially useful for recycling), or "non-graspable." Non-graspable objects were identified based on factors like "size," "weight" (per the authors' criteria), or being "fixed" to the ground.

The neural network was trained over 200 epochs on a high-power computer, with compre-

hensive training results available in Repository `http://ptak.felk.cvut.cz/vras/spot_collector/train_results/`. In Figure 4.22-a, the model's performance on the test data is shown, as indicated by a confusion matrix with a central diagonal reflecting classification values exceeding 81%, denoting a high degree of confidence in object grasping decisions.

The best-detected classes, such as graspable-metal and non-graspable-big/fixed, exhibit precision rates exceeding 90%. This underscores the robot's object selection aligns with operator criteria, mitigating potential damage during execution. Figure 4.22-b illustrates the model's consistent performance, with precision rates around 90%, even as the detection threshold is adjusted.

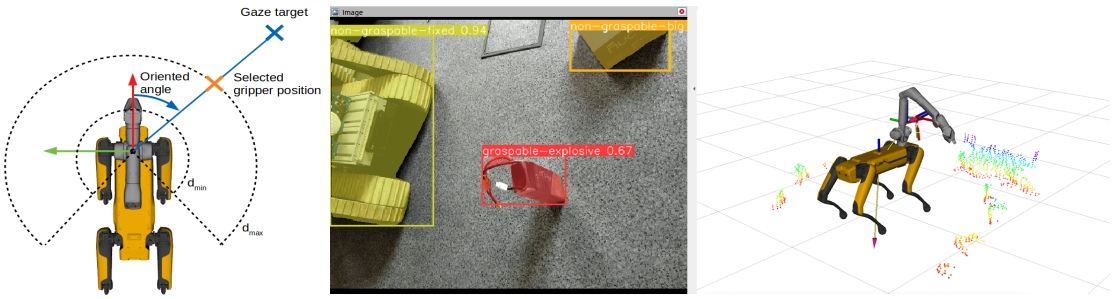


(a) Confusion Matrix.

(b) Precision-Recall Curve

Figure 4.22: Assessment of the Trained CNN Model.

The verification phase entails positioning the robotic arm within a range around the robot, using the */spot/arm_gaze* service to specify 3D coordinates of the gaze target. The gripper is then positioned in a circle with the origin at the axis of the first joint. The radius of the circle is set based on the distance of the object from the robot (Figure 4.23-a); the arm is rotated in such a way that it points towards the object (arm is rotated from the position in front of the robot by the computed angle).



(a) Selection of gripper position in gaze control.

(b) Detection of a Graspable Explosive Object.

Figure 4.23: Items Verification using CNN.

Figure 4.23-b illustrates the identification of an explosive object (a battery) within the identified cluster. Additionally, other objects of interest are presented, including a robot identified as stationary.

After characterizing all elements, the next step involves excluding those objects from

the list *Grasp_OBJ_List*[] that cannot be lifted due to their size or ground anchoring condition. With the positions of the objects known, the need to optimize the route for their collection arises. To address this, a genetic algorithm inspired by the Traveling Salesman Problem (TSP) [97] was employed. The objective is to minimize the travel distance (as expressed in equation 4.24), consequently reducing the execution time of the process.

$$TotalDist = \sum_{i=1}^{n-1} \text{dist}(\text{Grasp\_obj}[i], \text{Grasp\_obj}[i+1]) + \text{dist}(\text{Grasp\_obj}[n], \text{Grasp\_obj}[1]) \quad (4.24)$$

Figure 4.24 on the left displays the optimized sequential route for collecting twelve objects based on distance. On the right, the evolution curve of distance optimization is presented over epochs. It is observed that after epoch 50, the solution found does not exhibit significant further improvements, and as such, it is considered valid.

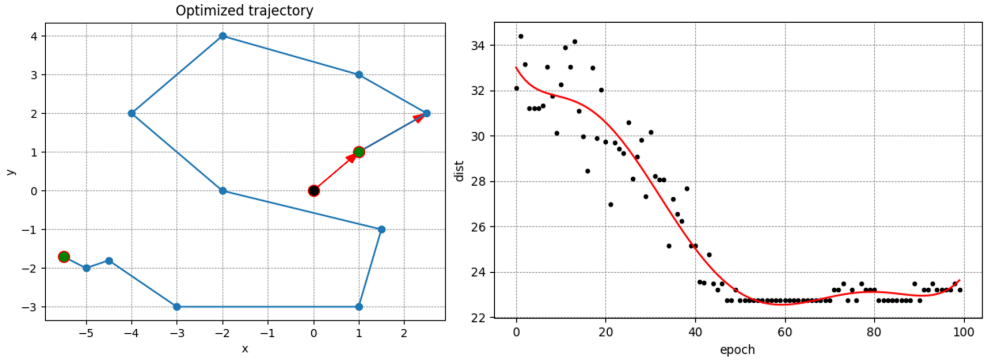


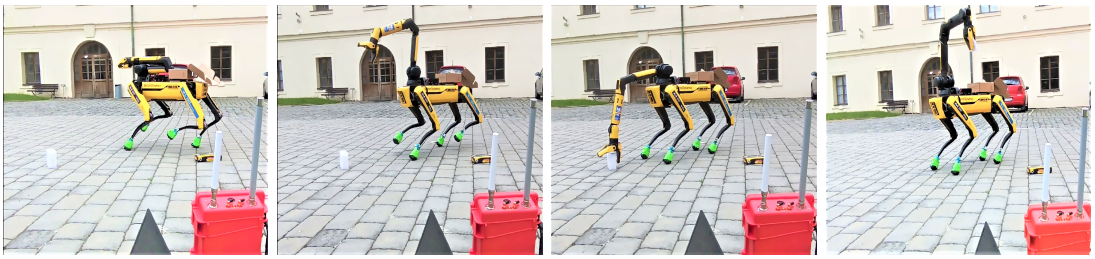
Figure 4.24: Optimized Path for twelve items and Evolution Curve of Distance vs. Epochs.

4.3.3 Automatic Grasping Process

The functional structure of the entire implemented algorithm is synthesized in Pseudocode 3, while the complete video of the experimental and developmental phase is available at the following link: `https://www.youtube.com/watch?v=E-phzgwn7Fw`

The proposed algorithm was designed to detect and grasp multiple objects. A ROS service (/multi_pick_and_place) that accepts a list of objects (3D position) to be grasped (defined previously) was created to simplify the grasping procedure. The grasping process involves four steps, illustrated in Figure 4.25:

1. Walk - robot automatically walks to a good grasping position; this means that Spot is standing close to the object and facing it with its front cameras.
2. Gaze - pose of the gripper is set so that the object is in the image of the gripper camera. Once the gripper reaches the position, a grasp search is performed. Spot searches for a pose of the gripper that would result in a good grasp. It is possible to specify some desired pose of the gripper and tolerance in the grasp_params field of the request; Spot tries to use this provided pose. If the search is successful, the robot proceeds to the next step. If no solution is found, the grasping procedure is aborted, and feedback stating no solution exists is returned.



(a) Approach. (b) Visual Inspection. (c) Grasping. (d) Verification.

Figure 4.25: Four Phases of Object Grasping.

Algorithm 3: Modular System for Multigrasp

Data:
$im_{[RGB]} \leftarrow$ RGB arm image [640x480]
$PC_{XYZ} \leftarrow$ Depth-Scan PointCloud
$Spot_{POSE} \leftarrow$ XYZ Position
Result:
$grasp[list]$
Function *GET_CLUSTERS* $(PC_{FILT-XYZ})()$ ▷ *Clusters extraction*
  $grasp[list] \leftarrow DBSCAN[PC_{FILT-XYZ}]$
  $grasp[list] \leftarrow threshold\_limit$
  $grasp[list].RelativePose \leftarrow Spot_{POSE}$
  return $grasp[list]$
end
Function *VERIFY_GRASP* $(im_{[RGB]})$ ▷ *CNN vision verification*
  $Move\_Arm\_Service \leftarrow pose(grasp[list])$
  $CNN_{based-algorithm} \leftarrow im_{[RGB]}$
  return $[graspable_{TYPE}, non\_graspable_{TYPE}]$
end
while *management interface* **do**
  $PC_{FILT-XYZ} = limits[X, Y, Z, outliers] \rightarrow PC_{XYZ}$
  if *Extract Soil Button* **then**
    $PC_{FILT-XYZ} \leftarrow$ Ground Plane Extraction
  end
  if *Get Clusters Button* **then**
    eval$(GET_CLUSTERS \leftarrow PC_{FILT-XYZ})$
  end
  if *Arm Inspect Button* **then**
    eval$(VERIFY_GRASP \leftarrow im_{[RGB]}, grasp[list])$
    if *non_graspable* **then**
      $grasp[list] \leftarrow$ delete object
    end
  end
  if *Send Grasp Button* **then**
    MultiGrasp Service $\leftarrow grasp[list]$
  end
end

3. Grasp - Joint trajectory for the manipulator is found and executed. At the end of the trajectory, the gripper is closed. After the feedback for the grasp command is returned, the arm remains at the grasp position; it is not lifted to the carry position.

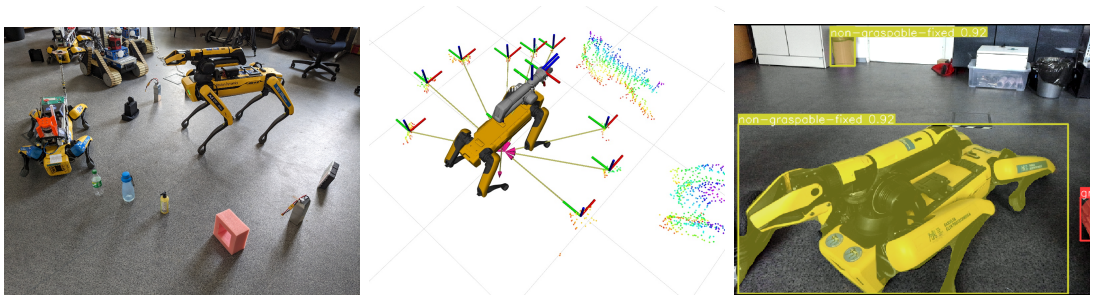
4. Grasp verification is part of the new service implemented and was added to the existing service for PickObject (grasp at 3D coordinates). Verification is done after the grasp is completed and the arm is lifted to the carry position. The gripper is not empty if the torque exceeds a preset threshold (0.5 Nm).

4.3.4 Test and Discussion

Two sets of tests were conducted to assess the effectiveness of the proposed system. The initial phase evaluated the perception system, which involved detecting and localising objects within the pointcloud and classifying these detected objects using the gripper camera. Subsequently, the second phase assessed the robot's capability to retrieve the identified objects.

Environment Elements Characterization

Multiple objects of different sizes and shapes (such as drills, plastic bottles, boxes, batteries, etc.) were placed around the robot. Results of these experiments are shown in Table 4.10. The environment could also contain large objects that should not be grasped (other robots, furniture). These objects should not be detected as an item for grasping by the PointCloud system; to test this, another robot was placed inside the range of the cameras. The desired result is that even though the robot is visible in the pointcloud, it is considered a non-graspable object because of its size and is, therefore, undetected. Figure 4.26-b shows the result of this experiment.



(a) Placement of the objects around the robot, including another Spot.

(b) The second Spot was correctly excluded from the detected objects because it is too big.

(c) Inspection object correctly identified non-graspable-fixed.

Figure 4.26: Test of object detection and localization.

Visual verification was performed as another step of the experiments described in Section 4.3.2. After the objects were localized, a phase for arm gaze was issued for each. Performance was evaluated as the ability of the CNN to distinguish between non-graspable and graspable objects. Results of these tests are shown in the Table 4.10. Figures 4.26-c and 4.23-b show detection performed during experiments; fixed objects that cannot be grasped are detected with high precision on all those and explosive objects.

Table 4.10: Results of the Perception Experiments

Test	Repetitions	Graspable Obj PCL	Recall [%]	Precision [%]	Graspable Obj CNN	CNN recall [%]
1	5	10	100	96.4	8	87.5
2	5	10	92	94.2	8	100

Grasppping Environment Elements Test

The proposed solution relies on the Spot's ability to grasp objects autonomously. If the robot can not grasp the object, it will never be collected. The most common cause of

failure is not finding a suitable grasping position. The only solution would be to plan a custom manipulator trajectory that leads to grasping the object, but this was not the focus of this project.

Six tests with graspable objects placed around the robot were performed; in each experiment, the number of successfully collected objects and the run time were measured. The results of these tests are in the table 4.11.

The last three experiments were performed with more challenging objects. The proposed detection algorithm successfully detected all of them, but Spot's autonomous grasping failed on one of the objects. The experiments were stopped after the first failure. In experiment 5, the grasp of the second object failed, and therefore, only one object was grasped successfully.

Table 4.11: Results of the grasping experiments.

Test	N Objects	PCL+CNN Graspable objects	Success rate [%]	Time [*seconds*]
1	3	2	100	76
2	6	4	100	125
3	5	4	100	135
4	7	6	66	240
5	9	7	14.3	70
6	8	5	80	201

Comparisson with Related Works

According to the main developments in the state-of-the-art, the prototypes carried out, although they show great effort in the development at the Hardware level (collection mechanisms, etc.), lack robust environment elements detection systems; neither have studies been carried out on the efficiency (collection times and garbage-free zones) of the process. Another limitation that these prototypes present is the size of objects and specific geometries that they are limited to pick up, as well as the vast majority of developments are teleoperated. Table 4.12 summarizes these works, mainly highlighting different aspects of the collection process, such as the maximum size of objects to be collected and the elements detection method.

Table 4.12: Related Works on Robotic Items Collection in the State of the Art.

Work	Robot	Max. objects size	Detection method	Autonomy
[189]	wheeled-manipulator	4 [cm]	Visual inspection	teleoperated
[239]	car-like + gripper	4.5 [cm]	Ultrasonic proxim	teleopetared
[11]	wheeled + ramp	up to 200 [g]	Visual inspection	teleopetared
[4]	floating boat	not specified	Visual inspection	teleopetared
[221]	wheeled + ramp	30 [cm]	Ultrasonic proxim	not specified
[133]	wheeled	not specified	Ultrasonic proxim	flow sequence
[139]	floating boat	up to 192 [g]	Visual inspection	teleopetared
[48]	omnidirectional+arm	not specified	CNN Classifier	flow sequence
[14]	3DoF Manipulator	up to 200 [g]	CNN Detector	autonomous
[147]	wheeled-manipulators	not specified	CNN Detector	flow sequence
[87]	social DustCart	not specified	non detection	autonomous
Proposed	**Legged-manipulator**	**up to 2.5Kg**	**PCL+Image-CNN**	**autonomous**

4.3.5 Discussion

While the robotic system provided by Boston Dynamics exhibits a high level of autonomy in item collection, it is constrained by the operator's manual identification and selection

process through a touch panel. The system implemented in this section leverages the data captured by the robotic perception system to perform this process autonomously. It identifies, discriminates and grap elements within the environment, a capability that can aid in clearing the ground by removing potentially explosive or stability-disrupting items during navigation.

Implementing a modular system for robotic systems to retrieve objects autonomously has provided a mechanism for centralized data flow management. This innovation has the dual advantage of affording operators improved visualization and control capabilities through the RVIZ interface, thereby influencing their decision-making processes.

The ROS services provided for communication with the robot enable the modular integration of the decision-making system implemented in this study. The system for detecting and precisely locating graspable elements operates in a two-stage paradigm involving validation through neural networks. Remarkably, this system has demonstrated an exceptional efficiency rate, surpassing 90%, in its ability to discriminate between items amenable to grasping and those not. This capability is pivotal in mitigating the potential risk of damage to the robot arm or its immediate environment during the critical grasping phase.

Chapter 5

Victims Identification based-on Vision Systems

"If you want different results, do not do the same things."

- Albert Einstein.

The early identification of victims using the infrared range of light invisible to the human eye provides relevant information about the environment. It allows capturing situations that are imperceptible to a first-responder. The precarious conditions of post-disaster environments hinder the easy and rapid identification of victims. Through different vision sensors, this chapter aims to establish a victim detection system utilizing ARTU-R as an element for exploring the environment.

Scientific publications related to this chapter [1, 2, 3, 4, 5, 6].

Preamble

One of the primary objectives of a search and rescue mission is to identify, locate, provide assistance to, and, to the extent possible, evacuate victims from an emergency area; aligned with this philosophy, the present chapter aims to explore alternative methods for the early identification of victims, using various ranges of the light spectrum emitted by bodies and elements in the environment as input data. This information is captured by specific sensors—cameras with sensitive optical elements capable of acquiring environmental data—whose function is to process it as digital images. Indeed, to fulfil this mission, it is necessary to integrate these sensors onto a mobile platform. To achieve this, ARTU-R has been equipped with different cameras through adaptations, including mechanical couplings, power systems, and communication systems for data acquisition.

Specifically, the experimental phase will focus on evaluating three specific ranges of the light spectrum, namely RGB $[400nm - 700nm]$, thermal infrared $[8\mu m - 14\mu m]$, and multispectral (Red Edge $[690nm - 730nm]$ and Near-Infrared [NIR] $[750nm - 2.5\mu m]$). Figure 5.1 exemplifies these spectral ranges. It is evident that the visible light spectrum range for the human eye, i.e., the RGB range, is minimal compared to the infrared or ultraviolet ranges, which are perceptible by animals such as snakes or insects, respectively. Among these ranges, perhaps the multispectral range is the most atypical in victim detection, given its extensive application in agriculture. These ranges have been selected based on future research directions in various state-of-the-art studies and following commercially available data acquisition instruments (images) that are portable by quadruped robots (weighing less than 10 kg).

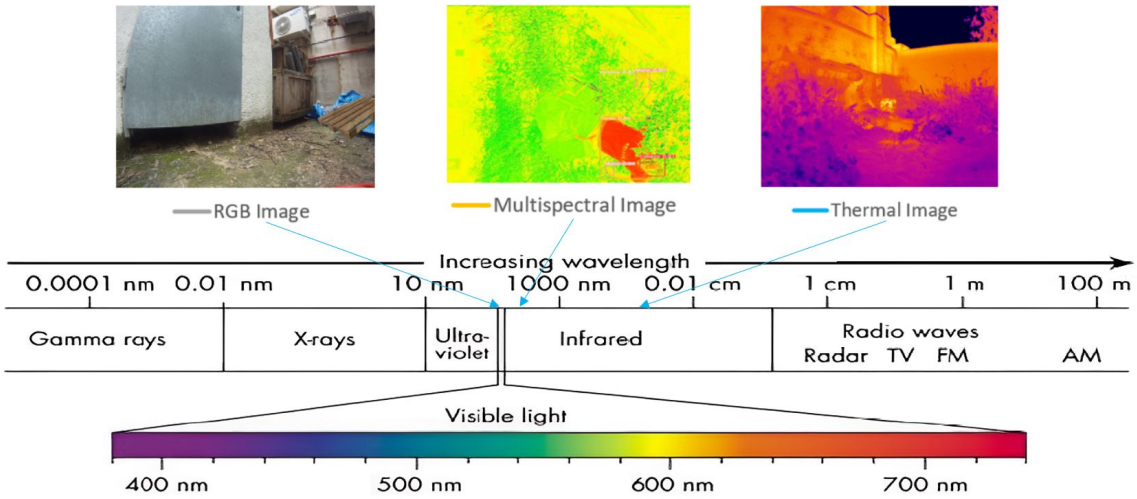


Figure 5.1: Ranges of light in the visible and non-visible spectrum.

In this way, the robot's mission in the environment is focused on exploratory movement for data acquisition, with some instances involving teleoperation and others being autonomous. The last aspect related to the robot's autonomy capacity for environmental exploration will be the subject of study in the upcoming doctoral thesis Chapter.

Furthermore, additional software challenges exist due to the inherent diversity, quantity, and fluctuation of data and elements to be controlled from the acquisition and processing phase onwards; this is always aimed at achieving real-time processing to the extent possible, aiming to optimize processing times and detections as the robot explores an environment; for this challenge, a modular system based on ROS has been proposed, allowing

for the integrating of all process phases. It should also be noted that for the three proposed methods of victim detection, the same modular approach will be followed to perform tests in scenarios within the Center for Automation and Robotics facilities.

Figure 5.2 illustrates the schematization and interconnection of the subsystems involved. Two groups are highlighted in the diagram. The first group comprises a command station with a high-performance computer, where computationally intensive calculations are directed, and it also serves as the user interface for visualization and high-level decision-making for robot movement. On the other hand, the second group corresponds to the field robot, which is equipped with an embedded platform and is responsible for on-board operations related to sensory readings of the environment (image acquisition), environment mapping, route planning, and low-level robot control, as well as the transmission of data and positions to the command station. The data flow between the command station and the field robot is executed through wireless communication (Wireless 5G) and the ROS communication protocol.

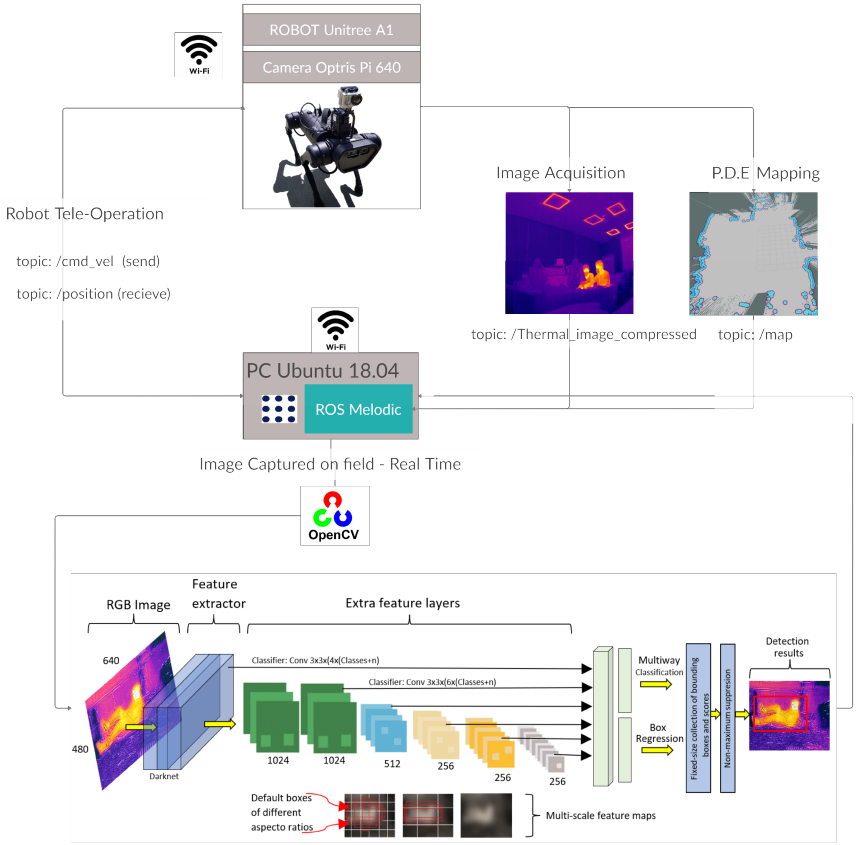


Figure 5.2: Subsystems integration for the detection of victims in real-time.

Similarly, Convolutional Neural Networks (CNNs) have been standardized to optimise the victim detection phase in all three explored methods, as in Section 4.1.3. This process is directed towards the command station due to the significant computational resources required for real-time operation, as summarized in the diagram shown in Figure 5.2.

5.1 Victims Detection in RGB Imagery

Exploring a first approach for detecting victims in post-disaster contexts involves an RGB image analysis framework. For this approach, three models of CNN will be generated from the training of three datasets: real, virtual and mixed (real + virtual), the last one in particular generated through virtual environments, enabling the generation of CNN models to identify victims in regions worldwide, utilizing satellite data containing relief and texture information for creating the dataset. What distinguishes this approach is the attention dedicated to the functional validation of the models trained using synthetic data, applied to images taken in real scenarios (through cross-validation). The overarching objective is to ascertain that the resultant models and algorithms can adeptly discern and localize victims in veritable circumstances. Through this rigorous validation regimen, the approach seeks to establish the practical viability and efficacy of the developed models, thus conferring a tangible solution for exigent situations.

5.1.1 Test scenarios and environment generation

The experimental phase encompassed controlled investigations conducted within diverse environmental contexts, comprising both indoor and outdoor settings. Specifically, real environment datasets generation was carried out at the ETSII of the Polytechnic University of Madrid (depicted in Figure 5.3-a), using an indoor environment as the reference scene. Additionally, outdoor scenarios for dataset generation were conducted at the EII of Malaga University, focusing on a tunnel-like area scenario (depicted in Figure 5.3-e) during the simulation of the "XVI Jornadas Internacionales de la Universidad de Málaga sobre Seguridad, Emergencias y Catástrofes". The reconstruction of the distinct scenarios, encompassing indoor and outdoor environments, as depicted in Figures 5.3-b-f, has been executed by implementing the Unity engine.

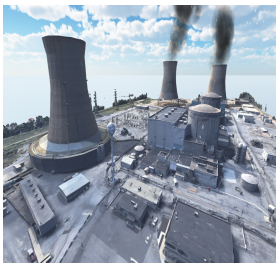
For the virtual dataset generation, the Unity framework was employed to meticulously craft virtual representations that faithfully mirrored the intricate attributes of the scenarios under scrutiny. This encompassed the recreation of the spatial configurations, architectural nuances, and environmental context specific to each scenario. Furthermore, satellite-derived data from Google Earth was integrated to enrich the authenticity of these virtual reconstructions. Using Google Earth's satellite data endowed the virtual reconstructions with geospatial accuracy and realism, ensuring that the simulated environments faithfully embodied their real-world counterparts' topographical and geographical attributes. By synergistically harnessing the capabilities of the Unity engine and the geospatial data from Google Earth, the reconstructions attained a heightened level of fidelity, facilitating a nuanced representation of the scenarios for rigorous experimentation and analysis. This innovative fusion of technology and data sources fostered an integral base for generating virtual datasets since the environments could be modified, recreating post-disaster conditions that commonly cannot be had.

An aggregate of five distinct environmental settings was reconstructed to encapsulate scenarios featuring fire and landslide contingencies, concomitant with the introduction of simulated human subjects representing victims. A comprehensive array of frames capturing diverse perspectives was systematically documented within these synthesized environments. This was done through virtual cameras integrated within the simulated environments, thus enabling the acquisition of an expansive dataset capturing visual information.

(a) ETSII-UPM Frontal View.

(b) ETSII-UPM Virtual recreation Frontal View.

(c) Three mile island nuclear accident - real plant.



(d) Three mile island nuclear accident - recreation in Unity.

(e) EII-UMA test environment. Tunnel - post-disaster scenario.

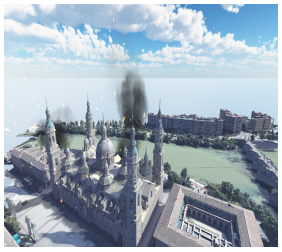
(f) Virtual test environment - global view tunnel.

(g) ARTU-R robot at one end of the EII-UMA tunnel.

(h) Buenos Aires's Obelisk.

(i) Obelisk of Buenos Aires, fire scene recreation in Unity.

(j) Basilica of Zaragoza.



(k) Basilica of Zaragoza, recreation in Unity.

Figure 5.3: Environments in which the tests were conducted, comprising both real and virtual settings.

- Escuela Técnica Superior de Ingenieros Industiales - UPM (Figure 5.3-b).
- Three mile island, this for the history of the 1979 nuclear accident (Figure 5.3-d).
- The EII scenario - University of Malaga (Figure 5.3-f).
- Obelisk in Buenos Aires (Figure 5.3-i).
- Basilica del Pilar in Zaragoza (Figure 5.3-k).

In parallel, the endeavour extended to real-world scenarios, where data acquisition was realized through an RGB-D camera adeptly on the robotic platform. The virtual reconstruction and real-world acquisition engendered a comprehensive and dynamic dataset for the CNN models training.

5.1.2 Datasets generation and training of CNN models

A methodical approach was adopted to construct comprehensive and versatile datasets. The images were meticulously captured to ensure uniformity and fidelity, adhering to a fixed resolution of 1280x720 pixels. The real-world data was meticulously curated through the acquisition of images from two distinct locations: the Escuela Técnica Superior de Ingenieros Industriales (ETSII-UPM) and the Escuela de Ingenierías Industriales (EII-UMA). These sources, visualized in Figures 5.3-a and 5.3-e, were selected to encompass a range of environmental nuances, thereby endowing the dataset with contextual diversity. The synthetic data has been obtained from their corresponding virtual reconstructions, including the environments of Three Mile Island, the Obelisk, and the Basilica del Pilar.

Integral to dataset robustness is the application of data augmentation. This strategic process involved the introduction of controlled disturbances to the captured images. These augmentations aimed to amplify the dataset's resilience. Specifically, modifications encompassed a 10% enhancement in both brightness and contrast, thereby enhancing adaptability to diverse lighting conditions. Furthermore, augmentations encompassed rotations spanning thirty degrees. In this way, the datasets are distributed as follows:

- Model 1: Real Dataset (1000 images)
- Model 2: Virtual Dataset (1200 images)
- Model 3: Combined Dataset (2200 images)

Figure 5.4 shows the partitioning scheme for each dataset, delineating the allocation proportions for training (82%), validation (12%), and test (6%) subsets. Another phase preceding training involves labelling the images using bounding boxes assigned to their respective classes, which have been defined based on criteria governing the positioning of victims within the environment. It is crucial to acknowledge the likelihood that victims might be concealed by rubble, resulting in only extremities being visible. Similarly, distinguishing first-responders from victims is essential to prevent false detections. This necessitates the inclusion of labels for first-responder (rescuers) as well. The labels used were victim, victim-head, victim-torso, victim-arm, victim-leg, rescuer and rescuer-legs, exemplified in Figure 5.4.

The training regimen has been executed on a computationally potent system characterized by an MSI Nvidia 1660Ti GTX graphics processing unit (GPU) and an Intel Core i7 10th generation. The training protocol has been staged by integrating TensorFlow libraries and the Darknet framework, configured to accommodate the YOLOv5 version.

In adherence to established best practices, a suite of hyper-parameters has been calibrated for optimal training efficacy. The Learning Rate Schedule is parameterized by a burn-in value of 1000, a step configuration of 400,000, scale values of 0.1, and an image dimension of 1280 pixels. The Initial Learning Rate is at 0.001, fostering an adaptive learning process. The Batch size, set at 16, informs the granularity of weight updates during each optimization step. The training duration, encapsulated by Training Epochs, spans 80 epochs, striking a balance between model convergence and computational efficiency.

The assessment of the trained models' performance will be conducted through the use of confusion matrices. These matrices offer a comprehensive representation of predictive

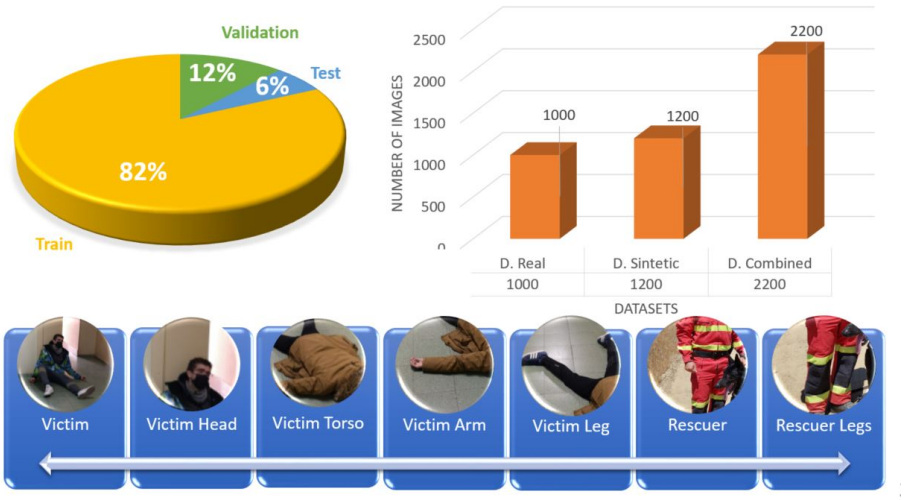


Figure 5.4: Percentage distribution of the dataset and classes

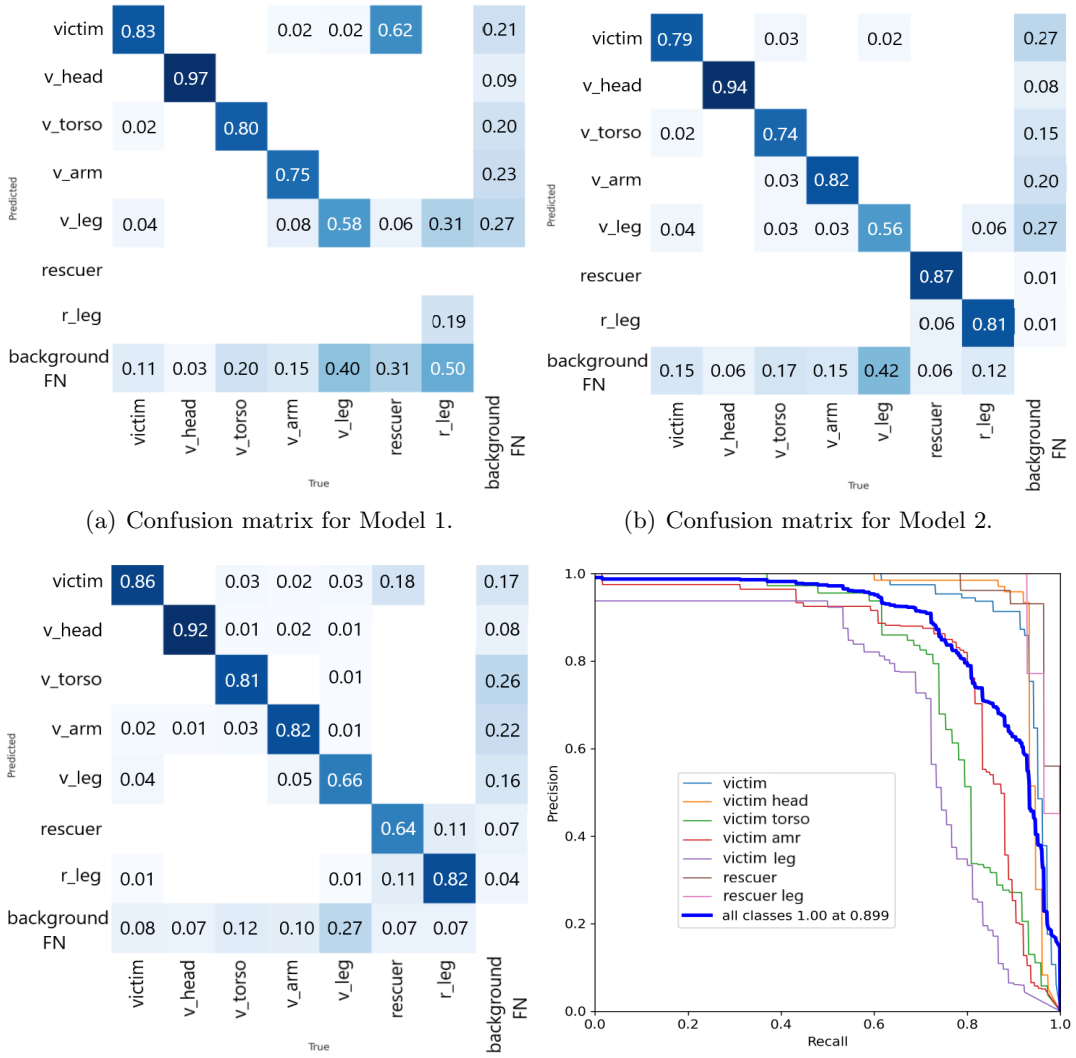
outcomes. Specifically, the Main diagonal of these matrices effectively quantifies the percentage of accurate predictions, denoted as True Positives (TP), manifesting within the range of $[0 - 100\%]$. Simultaneously, attention is directed towards the off-diagonal elements, where distinct scenarios come into focus. False Positives (FP), positioned within the matrix's rows, depict instances in which predictions have been assigned to a class not congruent with the actuality. Conversely, False Negatives (FN), positioned in the columns, represent events where classes have been erroneously assigned as negative despite their actual classification. This analysis encapsulates the nuanced interplay of predictive precision and highlights the model's ability to delineate the intricacies inherent within the dataset.

5.1.3 Model detection results

After the completion of model training, a comprehensive evaluation was conducted using the designated Test dataset, constituting 6% of the overall dataset. This evaluation indicates essential results in the obtained confusion matrices, as shown in Figure 5.5 a-b-c. The main objective of these matrices is to facilitate an in-depth assessment of the network's performance. Furthermore, leveraging the outcomes from this evaluation, establishing the precision-recall curve (Figure 5.5-d), an essential analytical tool, ensued.

The confusion matrices for the three models exhibit notably elevated values along the principal diagonal. Notably, the classes about victims, victim heads, and victim's arms garner detection precision accuracies exceeding 75%. Among these instances, it is paramount to underline that Model 3, in particular, attains remarkably robust precision outcomes. Nonetheless, the efficacy of this model will be subjected to a meticulous evaluation in the subsequent phase, employing cross-validation against new real-world data. This upcoming evaluation aims to gauge the models' performance more rigorously.

The classes posing the greatest detection challenge encompass Victim Legs and Rescuer, boasting an average accuracy of 60%. The Precision-Recall curve, depicted in Figure 5.5-d, unveils a degradation in precision around mean values of 80%. In this type of graph, while the curve tends more towards the upper right corner, it represents a greater efficiency for detection in the corresponding class. The classes demonstrating optimal model performance are victim and victim head.



(a) Confusion matrix for Model 1.

(b) Confusion matrix for Model 2.

(c) Confusion matrix for Model 3.

(d) Precision-Recall curve of model 2.

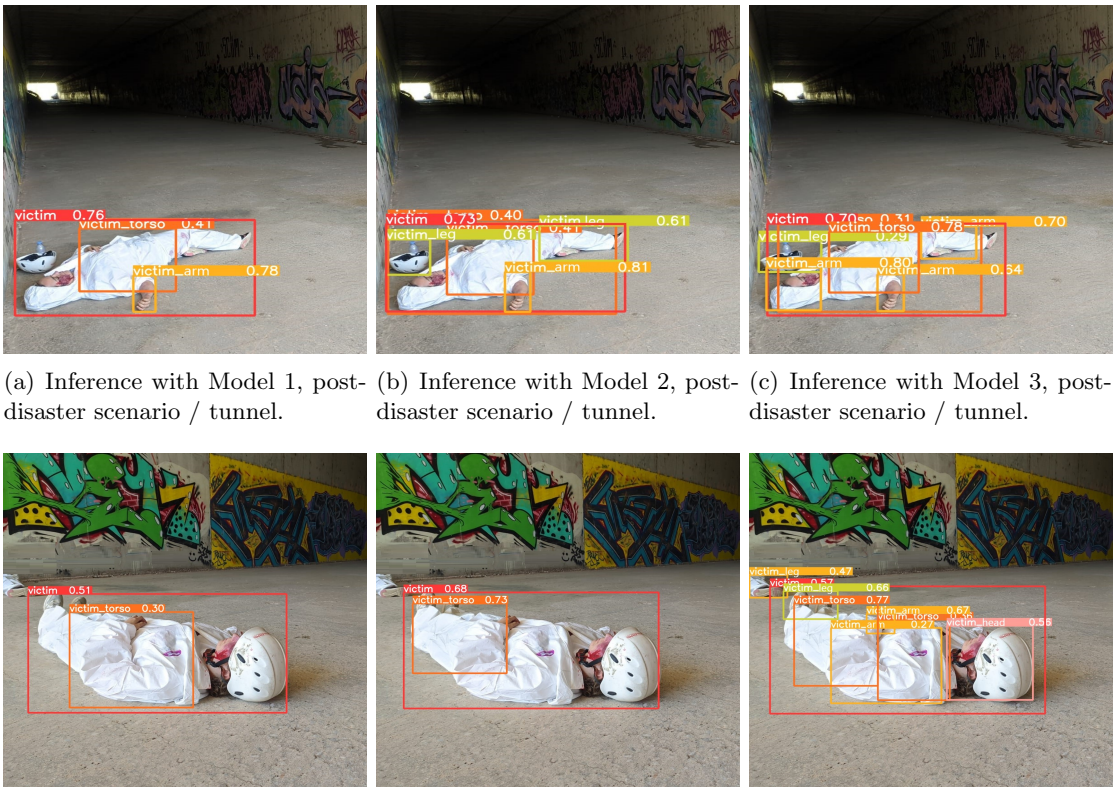
Figure 5.5: Performance metrics of trained models.

5.1.4 Cross validation experiments and results

This analysis phase mainly determines if the CNN model trained from synthetic data can detect victims under realistic conditions. For which new images containing a victim are evaluated with the trained models.

Illustrated in Figure5.6 are the detections, encompassing bounding boxes, corresponding classes, and the associated success percentage in detecting victims, rendered by the three CNN-trained models. Notably, Figures 5.6-a-b-c, respectively, elucidate the outcomes of inference generated by the three meticulously trained models. Present within the image is a subject personifying a victim, attired in white clothing and a helmet, to lend an air of realism to the exercise.

Foremost, it is noteworthy that the Model 2 has exhibited remarkable efficiency and success in its detection capabilities. This model, exclusively trained with synthetic data, has demonstrated a pronounced acumen in accurately detecting victims when confronted with real data, as illustrated in Figures 5.6-b and 5.6-e. This accomplishment remains relevant



(a) Inference with Model 1, post-disaster scenario / tunnel.

(b) Inference with Model 2, post-disaster scenario / tunnel.

(c) Inference with Model 3, post-disaster scenario / tunnel.

(d) Inference with Model 1, post-disaster scenario / tunnel.

(e) Inference with Model 2, post-disaster scenario / tunnel.

(f) Inference with Model 3, post-disaster scenario / tunnel.

Figure 5.6: Cross-evaluation of the trained models with real new data.

to the central objective of this particular section of the study. In the context of the doctoral thesis, while it is acknowledged that not all classes (specifically legs and head) have been identified, the inherent redundancy entailed by multiple classes assumes a strategic advantage within the system. This advantage materializes in the system's ability to trigger a detection signal upon identifying any of the classes corresponding to the victim, thus underscoring the robustness of the system's performance.

Depicted in Figure 5.7 are the attained mean values, expressed in percentages, about the detections executed within the scenarios delineated in Figure 5.6. Particularly pronounced are the varying associated with the detection outcomes for individual classes. Barring an exception regarding detecting a victim's leg, the average precision for the detected classes for the victim remains consistently poised at approximately 80%.

5.1.5 Conclusion

The present study introduces a methodological paradigm for automated victim recognition, training three convolutional neural network models from RGB images acquired in real and virtual reconstructed post-disaster environments. The ensuing conclusions are discerned as follows:

Firstly, the viability of extrapolating models engendered within virtual environments for application to real-world images is a significant finding. Notably, this extrapolation demonstrates commendable efficacy in victim detection within post-disaster landscapes, thereby

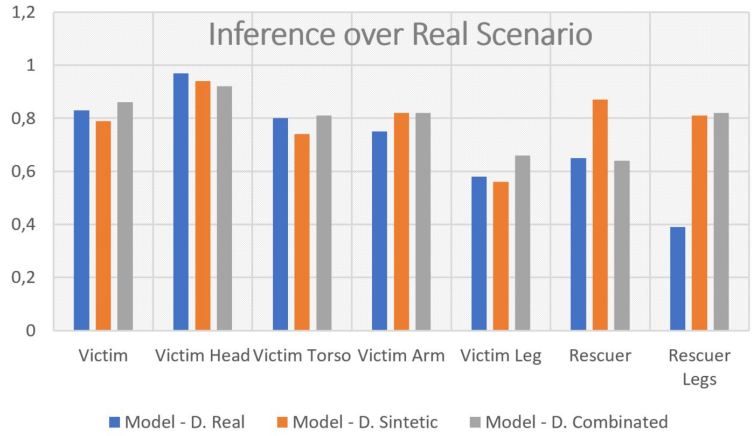


Figure 5.7: Precision obtained in the cross-validation of the models on the images of the real environments.

heralding an innovative avenue for CNN model development tailored to locales impeded by physical inaccessibility while still catering to hypothetical post-disaster conditions.

Secondly, noteworthy proficiency is exhibited in recognizing victim and head victim categories. The uniformity in outcomes across the three models accentuates the potential of training a model adeptly attuned to practical exigencies by leveraging synthetic imagery as a training resource.

Lastly, the CNN model from the synthetic image dataset substantiates its prowess in real-image victim inference. This is acutely pronounced within models trained on synthetic data, exemplified by the EII-UMA context, underscoring synthetic data's competence in bolstering actual-world inference capabilities.

Collectively, these discernments reaffirm the method's potency, shedding light on the transferability of virtual models to authentic scenarios and validating the utility of synthetic data-propelled training in pragmatic victim identification within post-disaster domains.

5.2 Victims Detection in Thermal Imagery

This subsection aims to assess a second method for victim detection using thermal images, which are captured within the spectral range of light between $[8\mu m - 14\mu m]$. This assessment is based on a thermographic study and analysis of the information obtained from these images. The system's robustness will be evaluated under various lighting and environmental conditions. Main developments in the thermographic area and computer vision focus applications of people detection, surveillance [231], face recognition [128] or low-resolution thermal image analysis [98], but efforts in victim detection are limited in the state-of-the-art.

5.2.1 Thermographic analysis context

One of the key advantages of working with thermal images lies in the principle of thermography, which is a technique that enables the remote determination of temperatures of living organisms or objects without establishing physical contact. This is achieved by using thermographic cameras to capture the infrared radiation emitted by all bodies within the electromagnetic spectrum [129].

According to Planck's law, which describes the electromagnetic radiation emitted by a blackbody in thermal equilibrium at a given temperature, any object above absolute zero emits infrared radiation [256]. This principle is the basis for estimating the temperatures of objects and representing them in thermal images. Thermal images show the energy emitted, transmitted, and reflected by an object or subject (as described in Figure 5.8). The emitted energy (ϵ) originates from the subject itself, the transmitted energy (τ) is the energy that passes through the subject from another remote thermal source, and the reflected energy (ρ) is the energy reflected by the subject's surface from a remote thermal source. The thermal camera captures the sum of these three energies.

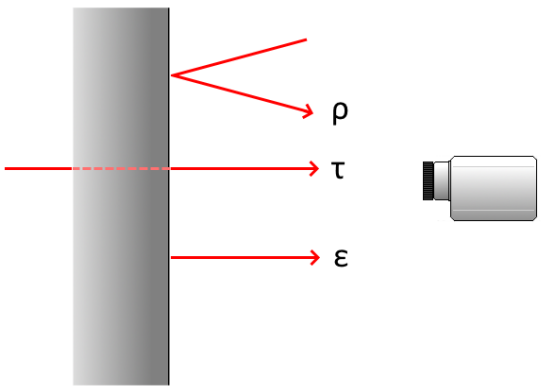


Figure 5.8: Energy emitted ϵ, transmitted τ, and reflected ρ by an object.

The interrelationships involving the energy emissions (ε), transmissions (τ), and reflections (ρ) of an object are dictated by the principles of energy conservation and the laws governing radiation [282]. These relationships are involved in the Energy Conservation Equation, which postulates that the overall incident energy (1) directed at an object can be disaggregated into constituent portions: ε, τ, and ρ, defined by the equation 5.1.

$$\varepsilon + \tau + \rho = 1 \tag{5.1}$$

Objects with high, medium, and low emissivity are established according to this context:

- High emissivity ($1 < \epsilon < 0.8$).
- Medium emissivity ($0.8 < \epsilon < 0.6$): Temperature can be accurately measured with ease.
- Low emissivity ($0.6 < \epsilon < 0$): Temperature is challenging to measure accurately.

Furthermore, certain materials exhibit varying emissivities depending on the wavelength and, consequently, temperature. These materials are called coloured materials; some examples include most metallic materials.

Determining a material's emissivity under specific ambient conditions can be accomplished through three distinct approaches. Firstly, emissivity tables (as shown in Table 5.1) serve as a valuable resource, offering emissivity values for commonly encountered materials across diverse settings. A second method involves comparing and adjusting measurements using a contact thermometer, ensuring accurate emissivity determination. Lastly, another effective technique involves comparing the material with an object of established emissivity.

Table 5.1: Emissivity of materials.

Material	**Temp. (° C)**	ε
Aluminum	170	0,04
Asphalt	20	0.93
Concrete	25	0.93
Lead, rusted	20	0.28
Ice	0	0.97
Iron, shiny	150	0.13
Iron, rusted	20	0.85
Soil	20	0.66
Glass	90	0.94
Silver 20	20	0.02
Wood 70	70	0.94
Plastic (PE, PP, PVC)	20	0.94

Organic materials such as paper, wood, and others typically exhibit high emissivity, simplifying their temperature measurement, as their emissivity falls around 0.95 in the 8 to 14 μm spectral range. Thermal cameras are often preset to this emissivity value due to the prevalence of materials closely approximating this standard, thus mitigating measurement errors arising from incorrect adjustments. Furthermore, water, ice, and snow also possess elevated emissivities, ranging from 0.83 to 0.98. However, common materials can manifest diverse emissivities in urban, construction, and industrial environments. For instance, construction materials like concrete, brick, glass, and asphalt exhibit high emissivities, facilitating precise temperature measurements. Conversely, metallic materials display a wide spectrum of emissivities contingent upon various factors, including the metal type, surface condition, and coatings.

In the case of plastics, thickness becomes a critical factor. Thicker plastics exhibit high emissivity, typically ranging from 0.86 to 0.95. However, a challenge arises when dealing with thin plastic films. These films have high transmissivity, meaning that when measuring temperature with a thermal camera, what is recorded is not the temperature of the film itself but rather that of the objects behind it, as illustrated in Figure 5.9. This presents an advantage for our study, as it allows us to contemplate situations to identify victims even when they are covered.

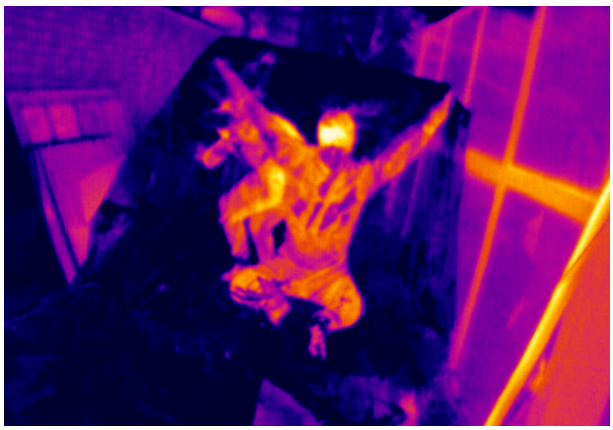


Figure 5.9: Thermal image of thin-film transmissivity example.

5.2.2 Environmental conditions and Dataset generation.

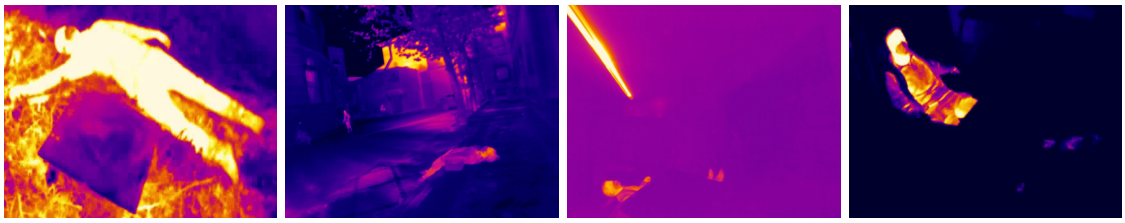
In addition to understanding the behaviour of the materials that could compose the setting, it is important to consider the environmental factors that may influence the measurement. The awareness of external environmental conditions that can introduce variations and potential errors in the measurements is equally crucial. These environmental factors encompass a range of parameters, including ambient temperature, humidity levels, solar radiation, air currents, pollution, and lighting conditions, which are described in detail in Table 5.2.

Table 5.2: Environmental factors influence during thermal measurement.

Env. Factor	**Impact on Thermal Measurement**
Ambient Temperature	Influences materials with low emissivity. Affects detection in thermal images if similar to the object's temperature.
Solar Radiation	Infrared radiation from the sky and the sun can interfere, preferable to work in cloudy outdoor conditions.
Humidity	Water, ice, and snow have high emissivity. Humidity can cause condensation on the camera lens.
Air Currents	Can change the temperature of the air near the object, affecting measurements.
Pollution	Particles in suspension like smoke, soot, or dust can interfere with infrared radiation.
Light	Light sources such as incandescent bulbs or sunlight can generate infrared radiation that affects measurements.

One of the most important aspects being evaluated in this study revolves around examining the impact of thermal image contrast in detecting victims. This contrast is the

temperature difference captured by the thermal camera between the person to be detected and the surrounding environment. The predominant factor in this equation is the ambient temperature. When the ambient temperature aligns with the average temperature of individuals in outdoor settings, typically between 20 and 25 °C, the contrast between the person and their surroundings diminishes significantly.



(a) Outdoors sunny-day scene. (b) Outdoors night-day scene. (c) Indoors covered victim. (d) Outdoors partially covered victim.

Figure 5.10: Elements of the dataset employed for training neural networks.

Figure 5.10 shows several images from the generated datasets. While in specific scenarios, such as Figure 5.10-d (nighttime environment), the identification task can be simplified as segmentation through computer vision filters, this approach becomes non-trivial for the remaining scenarios. Particularly, it poses challenges for outdoor scenarios during the summer and indoor settings containing heat sources. Hence, Convolutional Neural Networks are being assessed to detect victims automatically.

Three CNN models were trained using three distinct datasets to evaluate victim detection and the influence of materials and ambient factors. These datasets encompass daytime, nighttime, and combined scenarios, encompassing situations both indoors and outdoors. As an illustrative example, Figure 5.10 shows a subset of one of these datasets, comprising an impressive total of 354, 518, and 872 images, respectively. The complete dataset is available at the provided link (`https://mega.nz/folder/T3BBHSqa#QJkfC2yYp5zPx0OCDP2XqA`) for more in-depth analysis.

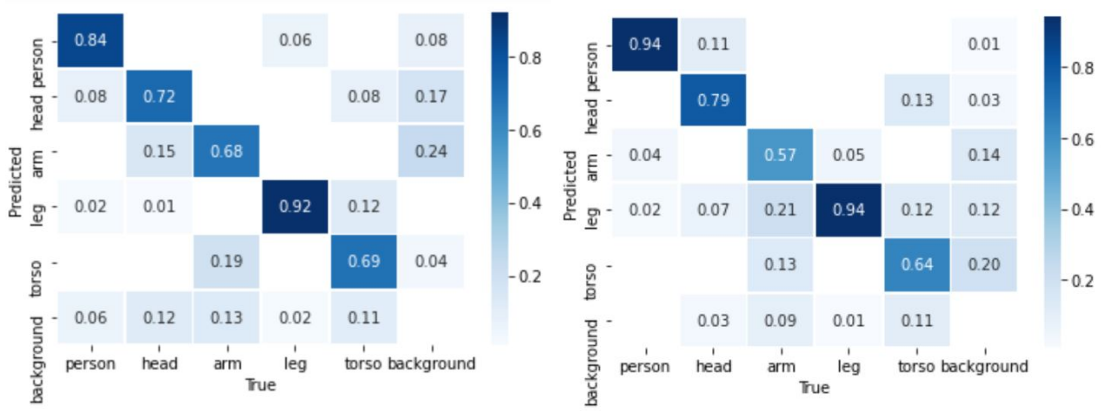
Similarly to the method described in Section 5.1, labels corresponding to body parts such as leg, arm, head, torso, and person have been employed. In this labelling phase, only the "person" class has been introduced, encompassing both individuals in a supine position (potential victims) and standing first-responders, as opposed to the previous method (RGB). A post-analysis will be conducted to differentiate between victims and first-responders.

5.2.3 Automatic Victims Detection using CNN.

For the CNN training, the following hyperparameters were used: An initial learning rate set at 0.001, a learning rate schedule with parameters (burn_in=1000, steps=400000, scales=0.1), a batch size of 16, and a total of 100 training epochs [61].

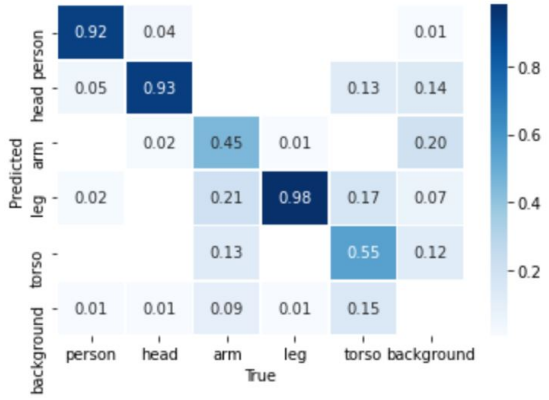
Figure 5.11 presents the evaluation of model training for each of the datasets. Figures 5.11-a-b-c correspond to the confusion matrices, where in all three cases, the main diagonal exhibits relatively high values for the detection of each class. Notably, the best results are achieved for the "person" and "leg" classes, while the lowest detection rates are observed for the "arm" and "torso" classes.

On the one hand, although the values in all three confusion matrices indicate favourable outcomes for the three models, particular attention is drawn to the network trained with the combined dataset. This model holds particular significance due to its applicability in environments characterized by variable conditions and differing luminosity levels.



(a) Confusion matrix for nigth dataset - Model 1.

(b) Confusion matrix for day dataset - Model 2.



(c) Connfusion matrix for combined dataset - Model 3.

Figure 5.11: Evaluation metrics for trained CNN models.

In this method, a specific approach based on geometric proportions has been introduced to differentiate victims from first-responders, in contrast to the first one (RGB method) presented. While the labelling process has exclusively incorporated the 'person' class, despite the primary objective of this section being the identification of 'victims,' distinguishing between a first-responder and a victim proves to be an intricate task. Although labelling individuals as first-responders and victims in the datasets appears to be the most logical solution, this strategy entails a substantial margin of error due to the resemblances between both categories.

A more efficient and straightforward approach depends on analysing bounding box shapes. Accident victims typically assume a horizontal position on the ground or similar postures, manifesting in a length-to-width (l/w) ratio of less than 0.75 within the bounding boxes (obtained experimentally), as shown in Figure 5.12. Contrarily, individuals in a standing or sitting position usually exhibit bounding boxes with a significantly greater height than width. This relationship (0.75) has been validated by examining images containing pedestrians and victims.

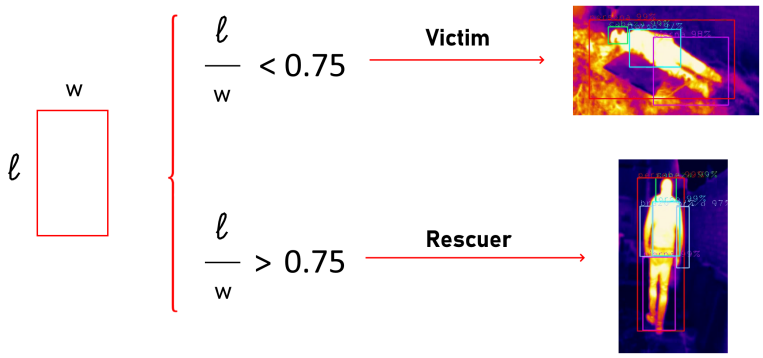


Figure 5.12: Utilizing the Length-to-Width Ratio as a Parameter for Victim Detection.

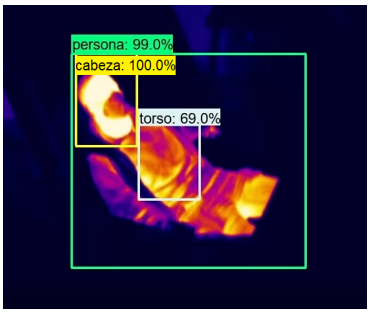
5.2.4 System evaluation with different environment conditions

In the preceding section, it was determined that among the three trained CNN models, the preliminary detection results were comparable. However, the third model (combined) proved effective for both indoor and outdoor detections. Therefore, this section will assess victim detection using this CNN model.

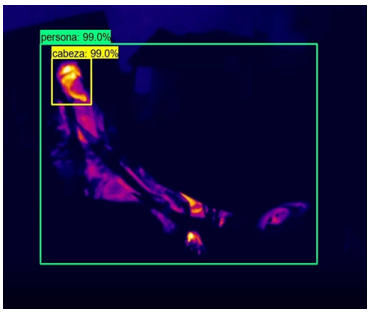
Figure 5.13 presents the results of victim detection in different scenarios. The first two Figures 5.13-a-b, respectively, depict the evaluation results for detection in scenarios with a victim exposed to the environment and one covered by plastic material. In both cases, the system can easily detect the victim, with notable success in detecting the completely covered victim. In this scenario, an RGB-based method would not be precise. While it is true that not all classes are detected, detecting at least one of them is sufficient to trigger an alert.

The scene shown in Figure 5.13-d features a partially covered victim with a pallet on the upper body, which obstructs the capture of emissivity from objects on the victim's posterior side. In this case, the system can detect the lower extremities. Another challenging scenario is presented in Figure 5.13-e, where the environment corresponds to a summer setting and has been significantly heated. Nonetheless, the system can detect the victim due to its robustness.

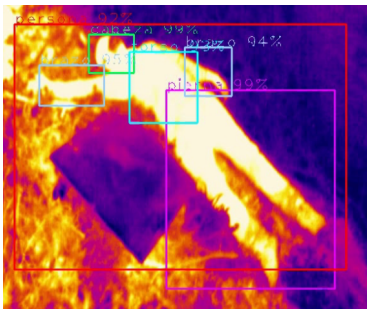
Finally, Figure 5.13-f corresponds to the advancement of the ARTU-R robot inside the tunnel where the simulation took place at "Universidad de Malaga". Meanwhile, Figure 5.13-g represents its thermal footprint in a later stage of advancement with two first-responders. Notably, the thermal image of the Spot robot, which also participated in the exploration mission, is visible in the background. However, the system correctly does not detect it as a person or victim. Meanwhile, the first-responders involved in the mission are effectively detected.



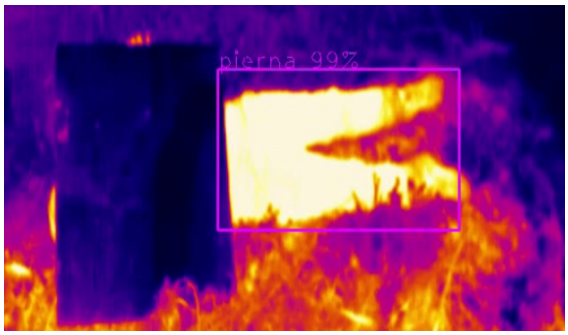
(a) Victim detection in outdoors - night conditions.



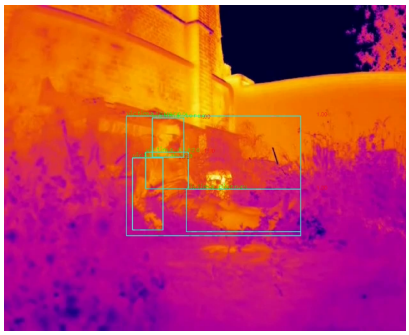
(b) Covered victim detection in outdoors - night conditions.



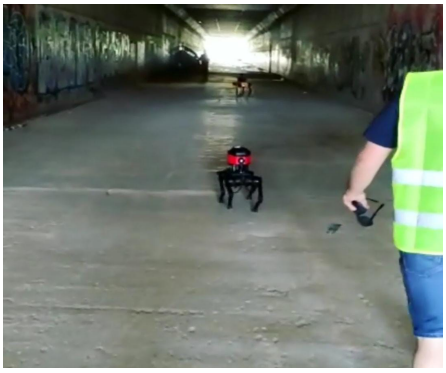
(c) Victim detection in outdoors - sunny day conditions.



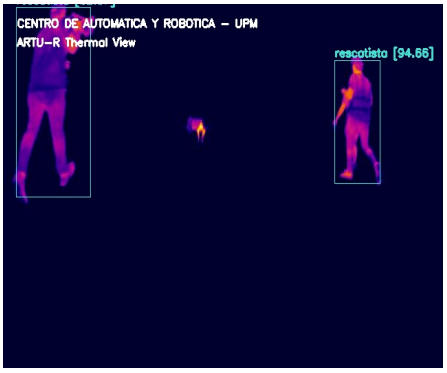
(d) Partially covered victim detection in outdoors - day conditions.



(e) Victim detection in outdoors - summer conditions.



(f) RGB image scene of SAR mission during robots advance.



(g) First-responder detection during field test, corresponding to previous subfigure-f.

Figure 5.13: Evaluation of the system basde on CNN model 3 trainned for victims and first-responders detection.