Experimental Evaluation of Road-Crossing Decisions by Autonomous Wheelchairs Against Environmental Factors

Franca Corradini[1(✉)], Carlo Grigioni[1], Alessandro Antonucci[1], Jérôme Guzzi[1], and Francesco Flammini[1,2]

[1] University of Applied Sciences and Arts of Southern Switzerland, IDSIA, Lugano, Switzerland
franca.corradini@supsi.ch

[2] University of Florence, Department of Mathematics and Computer Science "Ulisse DiniâĂİ, Florence, Italy

Abstract. Safe road crossing by autonomous wheelchairs can be affected by several environmental factors such as adverse weather conditions influencing the accuracy of sensors based on artificial vision. Previous studies have addressed experimental evaluation of multi-sensor information fusion to support road-crossing decisions in autonomous wheelchairs. In this study, we focus on the experimental evaluation of its tracking performance against outdoor environmental factors such as fog, rain, darkness, etc. It is rather intuitive that those factors can negatively affect the tracking performance; therefore our aim is to quantify through a set of metrics how the performance of the single sensors and their information fusion changes when such external factors are present. This is a first step in designing warning strategies in a novel framework based on the MAPE-k feedback loop established for the sensor system. System reconfiguration to reduce the reputation of less accurate sensors can then be set, thus improving overall safety. The problem is analysed within the context of the European project REXASI-PRO which aims to design a trustworthy autonomous wheelchairs supported by drones in which security, safety, ethics, and explainability are entangled to improve autonomy for people with reduced mobility. Results have been achieved by using an available laboratory dataset realised for a simplified framework in a road-crossing scenario and by applying appropriate software filters to simulate different environmental conditions.

Keywords: Artificial Vision · Simulation and Modeling · Vehicle Safety Systems · Swarm Systems · Machine Learning

1 Introduction

Recent advances in AI have implied an increasing use of *Machine Learning* (ML) models in sensing systems for autonomous driving [11]. Images from multiple

A. Kocian et al. (Eds.): INTSYS 2024, LNICST 608, pp. 363–380, 2025.
https://doi.org/10.1007/978-3-031-86370-7_22

cameras can be processed through ML models to identify one or more obstacle, enabling obstacle avoidance strategies. Despite efforts to regulate these models, detailed procedures for their validation are still lacking, due to the difficulty of achieving adequate safety levels. The use of multiple sensors and cameras should increase the performance and robustness of the overall system providing additional safety.

In this context, *Autonomous Wheelchairs* (AWs) represent an interesting case of study. As the system is intended for people with severe physical or mental disabilities who have little or no ability to intervene, it must achieve higher levels of safety and be sufficiently robust against possible external disturbances. The research presented in this paper has been performed in the context of an international research project named REXASI-PRO (REliable eXplAinable Swarm Intelligence for People with Reduced mObility),[1] aimed at introducing an innovative engineering framework for safe navigation of AWs using trustworthy AI with the aid of drones. These provide a second point of view to the AW, providing additional information about obstacles or areas where it has partial or no vision.

Safe road crossing is a paramount concern for AW technology, particularly in the face of environmental challenges like adverse weather conditions. In the previous study [5] we explored the fusion of multi-sensor information to support road-crossing decisions in cooperative robotic systems made of AWs and flying drones. In such a context we proposed a mock-up system to simulate the scenario and collected a novel dataset for our tests. We used a set of sensors including cameras, supported by *Convolutional Neural Network* (CNN) models, to extract information on approaching obstacles. However, a structured approach for the safety level of the detection system and an in-depth study of the impact of environmental factors, such as weather conditions, on the performance of ML obstacle tracking was lacking.

In this study, we establish a novel framework for the sensor system based on the *Monitor, Analyse, Plan and Execute* over a shared *Knowledge* (MAPE-K) feedback loop [20] and aim to address the impact of environmental noises by focusing on the quantification of obstacle tracking performance and evaluating its robustness against outdoor environmental factors such as fog, rain, and darkness. By quantifying these effects, we seek to identify thresholds where tracking accuracy falls below acceptable levels, facilitating timely warnings and sensor reconfigurations that can be integrated in the MAPE-K framework. As a result, early warnings and sensor reconfigurations can be developed and integrated into the MAPE-K framework to improve overall safety, as required by international standards [1,4]. Through our research, we aim to contribute improving the trustworthiness [3] of AW navigation systems, ultimately empowering individuals with mobility impairments to navigate their environments with greater confidence and independence.

In order to evaluate performance for both single sensors and their information fusion, we apply appropriate environmental filters to the output coming from

[1] https://rexasi-pro.spindoxlabs.com.

the cameras, before applying a CNN model fine-tuned to our specific case. This analysis is a first step in the realisation of a model for the quantification of the uncertainty associated with one or more sensors.

The main original contributions of this paper are summarised as follows:

1) We experimentally evaluate the performance in video tracking and estimation of road crossing danger function after simulating adverse environmental conditions. Those are generated by applying appropriate software filters to a video dataset generated in a laboratory environment, which is equipped with wheeled robots, sensors, and optical tracking infrastructure [5].
2) We fine tune artificial vision performance compared to previous studies in order to improve video tracking performance by combining diverse modules and training datasets.
3) We set the specific contribution in the paradigm of autonomic computing through a general architectural framework for the sensing system based on the MAPE-K feedback loop supporting self-adaptation.

The remaining of this paper is structured as follows. In Sect. 2, related works are reviewed and summarised. Section 3 provides background information about the road crossing scenario with the related danger function, the laboratory dataset used for the experimentation, and the sensor fusion approach. In Sect. 4, the MAPE-K based approach for the sensing system is presented. Section 5 describes the tools and the implementation strategy for the simulation of the environmental factors and for the obstacle recognition. Section 6 provides and discusses the results of the experimentation summarising the main findings. Finally, Sect. 7 provides conclusions and hints for future developments.

2 Related Works

Fusing data from multiple and diverse sensors can overcome inherent limitations of single-sensor perception in object detection for autonomous driving [13,19]. However, severe weather conditions may compromise the sensors' ability to perform their original functions due to attenuation of signal strength and noise disturbances. Adverse weather condition represents for autonomous vehicles one of the main issue to gain level 4 of SAE standard [16] or higher autonomy for a long time [9].

With AI model implemented in the sensing system, new strategies have been studied to increase the level of safety and robustness in presence of adverse weather condition. The main efforts were made in building datasets and simulators to drive and validate ML models. In fact, most of the datasets commonly used for training do not contain many conditions other than clear weather [17]. Furthermore, datasets are limited by the meteorological conditions common to the area where they are collected. Therefore, simulators capable of recreating various weather conditions with different intensities are now of great interest to test the performance of autonomous systems.

In [14] is presented a novel full-scale rain simulation system, developed to quantifying Advanced Driver Assistance Systems sensor performance when driving in the rain. The simulator can recreate a wide range of dynamic rain intensity experienced by the vehicle at different driving speeds, along with the corresponding droplet size distributions.

A Polarized Object Detection Benchmark is introduced in [15], consisting of a benchmark dataset that incorporates the physical dimension of optical polarisation imaging to tackle object detection challenges in complex road scenes under adverse weather conditions.

Reference [17] introduce a new large-scale simulation dataset which is generated by an automated pipeline from a high realism video game with the focus on simulating weather conditions. While in [10], a novel dataset of U.S. road markings is used for the training of an ensemble deep learning model. The collection is realised also applying data augmentation to simulate several atmospheric condition, resulting in the achievement of a more robust model against weather disturbances. We decide to adopt this strategy with the dataset previously recorded and presented in [5]. From the dataset recorded in optimal condition, we apply data augmentation through the library Automold,[2] created specifically to introduce various real-world scenarios for road images that pose challenges for the training of autonomous vehicle neural networks. More details are given in Sect. 5.

3 Background

In urban vehicular traffic environment, pedestrian and wheelchair users are among the most exposed subjects [6**?**]. The road-crossing scenario is therefore an essential situation to analyse in terms of pedestrian-vehicle conflict to prevent accidents [2]. In this context, wheelchair users may encounter further disadvantages, due to a possible lack of reaction, or inability to analyse the danger.

In reference [5], we proposed a danger evaluation approach for road crossing using a sensor system based on multiple, diverse, and redundant components. Information fusion was applied at different levels of information processing and to test the method we created a novel dataset recorded in our laboratory using a simplified scenario, where laboratory devices were used to simulate the crossing scenario. The results highlighted the advantages of using diverse sensors to take safer road crossing decisions, especially when information fusion is applied at the lowest level of data processing.

In this section we provide a brief description of the danger evaluation approach and the laboratory setup of the existing work.

3.1 A Danger Function to Support Road-Crossing Decisions

According to the scope of the project where this experimentation has been performed, the reference system for the road-crossing scenario is composed of an AW and a drone, as depicted in Fig. 1.

[2] https://github.com/UjjwalSaxena/Automold–Road-Augmentation-Library.

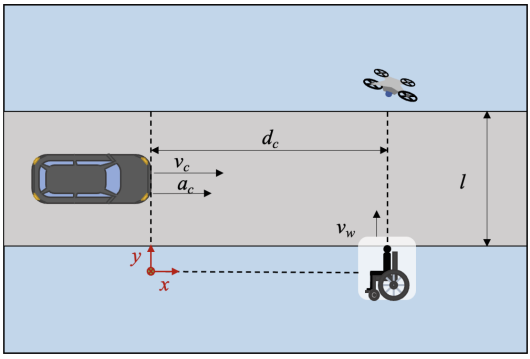


Fig. 1. Diagram illustrating a road crossing scenario with a car, a drone, and a person in a wheelchair. The car is on the left, moving right with velocity (v_c) and acceleration (a_c). The drone is above, and the wheelchair is on the right, moving upward with velocity (v_w). The relative distance between the car and the wheelchair is labeled (d_c), and the road width is labeled (l). Coordinate axes (x) and (y) are shown at the bottom left.

The *Time To Collision* (TTC) is a well established approach to quantify the danger of collision in driving and road-crossing contexts, described in the ISO 23376:2021 as the seconds that will take to two vehicle to collide if their relative speed is constant. In the road-crossing situation represented in Fig. 1 it is defined as $TTC = d_c/v_c$. For our analysis we wanted to include more information related to the kinematics of the car including its acceleration, and to significantly increase the danger value for short distance d_c. We designed a continuous function g, called *Danger Function* (DF) reflecting a kinematic analysis of the road-crossing scenario at a particular instant of time t. For decision-making, the crossing is considered dangerous when the value $g(t)$ exceeds a given threshold $g^* = 1$. The DF and safety condition are defined as:

$$g(d_c, v_c, a_c) := \frac{k_v \cdot h(v_c) + k_a \cdot f(a_c)}{\log(d_c + \epsilon)}, \tag{1}$$

$$g(d_c, v_c, a_c) < 1, \tag{2}$$

where d_c, v_c and a_c are respectively the distance, relative speed and relative acceleration between the vehicle and the pedestrian at instant t. k_v, k_a and ϵ are set following heuristics related to the geometry and kinematics of the scenario, whereas h and f are linear transformations with threshold applied on the components v_c and a_c.

The linear transformation of the speed is

$$h(v_c) := \begin{cases} 0 & \text{if } v_c \leq \underline{v}_c, \\ \frac{v_c - \underline{v}_c}{\overline{v}_c - \underline{v}_c} & \text{if } \underline{v}_c < v_c \leq \overline{v}_c, \\ 1 & \text{if } v_c > \overline{v}_c, \end{cases} \tag{3}$$

and, for the acceleration

$$f(a_c) := \begin{cases} -1 & \text{if } a_c \leq -\overline{a}_c \\ \frac{a_c + \underline{a}_c}{\overline{a}_c - \underline{a}_c} & \text{if } -\overline{a}_c < a_c \leq -\underline{a}_c \\ 0 & \text{if } -\underline{a}_c < a_c \leq \underline{a}_c \\ \frac{a_c - \underline{a}}{\overline{a}_c - \underline{a}_c} & \text{if } \underline{a}_c < a_c \leq \overline{a}_c \\ 1 & \text{if } a_c > \overline{a}_c. \end{cases} \tag{4}$$

Such transformations have been chosen to prevent contribution by low speeds (i.e., $v_c \leq \underline{v}_c$) and low accelerations (i.e., $|a_c| \leq \underline{a}_c$) to the DF, and at the same time places a normalised upper limit on the contributions of high speeds ($v_c \geq \overline{v}_c$) and accelerations ($|a_c| \geq \overline{a}_c$). Furthermore, only positive speed values are included since vehicles are not expect to go backwards and the acceleration contribution is significantly reduced. In our experience, these limitations improved the results of the danger analysis in part to reduce the effects of noisy sensors measures. Detailed information on the coefficients and parameters design can be found in the above-mentioned work [5].

We would like to point out that such DF is a simplification of a real road crossing scenario, where other factors as the danger perception of the individuals or environmental elements might play an important role. Some of these factors might be considered by varying the values g^*, however, some situations may require a more complex model. Nevertheless, such representation results to be quite straightforward to use in the context of information fusion since just kinematic properties are required without any evaluation or hypothesis of the conductor capabilities.

3.2 A Laboratory Dataset for Simulated Road-Crossing

In order to build a dataset for the reference scenario, we used three wheeled ground robots named *RoboMasters*[3] (RMs) equipped with cameras and distance sensors. This allows to evaluate system performance in the simplified experimental setup depicted in Fig. 2. A dataset has been recorded in the *IDSIA Autonomous Robotics Laboratory*.[4] We collected ground-truth poses from a very accurate motion tracking system, which were used to evaluate the performance of the system. We shared the dataset with the scientific community by making it available on a publicly accessible repository.[5]

It should be noted that we did not use a real drone in the simulated environment, as for the proof-of-concept we only needed to acquire an additional view of the scene, a bit higher and from a different perspective. In future experiments, we plan to use a more realistic setting with a real drone.

For each experimental run, we recorded data from two cameras and a set of range sensors located on the two RMs acting as the AW ($\mathrm{RM_w}$) and the drone ($\mathrm{RM_d}$), together with the information of the motion tracker. In conclusion, we

[3] https://www.dji.com/ch/robomaster-ep.
[4] https://idsia-robotics.github.io.
[5] https://huggingface.co/datasets/carlogrigioni/safe-road-crossing-aw-dataset.

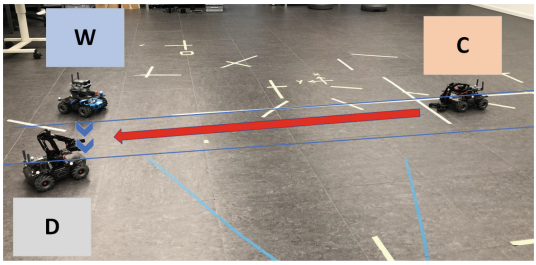


Fig. 2. Example of data collection scenario. Three RoboMasters devices are positioned on the laboratory floor where the dataset has been recorded. The device on the left acting as the wheelchair is labeled "W," the one on the right representing the obstacle is labeled "C," and the one at the bottom labeled "D" represents the drone. A red arrow runs horizontally from the left to the right show the trajectory of the obstacle. Blue chevrons pointing left near the bottom vehicle are equivalent to the pedestrian crossing. Blue lines run parallel to the red arrow, indicating the boundaries of the obstacle trajectory. (Color figure online)

collected three main streams of information from the sensors (i.e. two from the cameras and one from the range sensors block) that were used to derive the distance of the RM acting as the approaching vehicle (RM_c). No particular transformation is required for the range sensors measurements, whereas each frame of the recorded videos were elaborated through a ML model to obtain the RM_c distance.

Overall, twelve runs were recorded with some variations of the experiments setup and the kinematics of the RM_c. In particular, two different laboratory setups were adopted: scenario A (for the experiments $1-6$) consisted in shorter tests with RM_c running along a brief path. On the other hand, scenario B consisted of a longer run. Furthermore, the background of scenario B was a plain white wall, instead in scenario A there were multiple objects as shelves, devices and boxes. This aspect proved to be relevant for the analysis of the results in Sect. 6.

Additional information on the devices, the laboratory and the experimental setup can be found in reference [5], and in the associated repository.[6]

3.3 Sensor Fusion

When sensors data are combined, among the various aspects to consider, an important one is to define the default value when the obstacle is not detected. This may be due either to the actual absence of the obstacle or to the inability of the sensor to perceive it. In the work [5] only a partial discussion was made, considering the contribution of each sensor absent until the first obstacle detection. At this point, when the sensor no longer sees the obstacle, the last value detected is kept constant. This choice proved to be inadequate for the data coming from the cameras, which do not have an automatic reset procedure, such as the range sensors.

[6] https://github.com/IDSIA/rexasi-pro/tree/main/DECSoS_Workshop.

Information fusion from sensors and cameras can be done at different levels. We can directly average the distances measured by the sensors or perform the fusion only after the evaluation of the DF for each sensor or even at the level of the binary road-crossing decisions (e.g., by a voting procedure). Here we focus on the combination of the distances, which, according to [5], is the one providing best performances when a trivial average of the sensors data is applied without considering whether the sensor detects an obstacle or not, as described in the Subsect. 3.2.

4 Sensing System Framework

To improve safety and enable future certification against relevant standards, we propose a multi-agent, multi-modal, self-adaptive sensing system to achieve reliable event detection, where sensor outputs are combined to provide a common result for a specific task [1]. Multi-modal fusion has become a paramount task in the context of autonomous driving systems [13], and can be achieved through the use of multiple sensors diverse by software, hardware or position. Self-adaptation is achieved by combining a *Managed System*, consisting of the swarm system under consideration (i.e. AW and drones), with a *Managing System* based on the MAPE-K feedback loop, as depicted in Fig. 3. The *Managing System* elaborates the strategy for the autonomous safety logic over the *Managed System*. This is monitored together with the environment through the sensing system and implements the actions decided by the managing component.

The swarm system may consist of one or several autonomous systems working together as subsystems of the *Managed System* to achieve a given purpose. In general, autonomous vehicles can be expressed through logical or functional blocks defined by the information flow and processing steps performed. We adopt the architecture presented in [12], where the main functional blocks identified are: *Perception*, *Planning and Decision*, *System Supervision* and *Motion and Vehicle Control*. Information regarding the environment and the system itself are collected through the *Perception* block composed of the sensors on-board the autonomous system and the pre-processing elaboration of their raw data. Elaborated data together with actions outlined by the *Managing System* are processed by the *Planning and Decision* block to evaluate the navigation plan for the vehicle. This is finally implemented by the *Motion and Vehicle Control* by means of the propellers. The *System Supervision* monitors the overall functioning of each components and detects any possible errors.

For the entire system, we identify as 'Sensing System' the sum of all the *Perception* blocks of the single autonomous systems with the MAPE-K loop of the *Managing System*. In the monitoring phase, the sensor data are further processed, applying alignment, resampling or other operations as required [18]. In the monitoring and analysis phases, data information is analysed and combined appropriately, taking into account any discrepancies between sensors and the degree of reliability of each one. All relevant information is then extracted in order to make appropriate planning choices. Finally, the overall strategy for the swarm system is planned and actions are sent to the managed subsystems.

In the context of the road crossing carried out by a swarm system composed of an AW and an autonomous flying drone, these two devices represent the subsystems of the *Managed System*. The *Managing System* might be a separated elaboration unit on-board the AW able to communicate with other systems, such as multiple autonomous drone, AWs or smart systems around the city. The data from the sensors on-board the two autonomous systems are combined in the monitoring phase of the *Managing System* and a danger level for the road crossing is calculated in the analysis phase. In the planning phase, the behaviour to be performed by the swarm system is decided, i.e. whether to cross the road or not according to predefined requirements. Finally, the single actions are sent to the two systems to be executed.

In the remaining part of the paper we analyse by means of metrics how the performance of individual sensors on-board the *Managed Subsystems* and their information fusion varies in response to weather disturbances. This is a first approach to define appropriate strategies to be applied at the *Managing System* level in the *Monitoring* and *Analysis* phase.

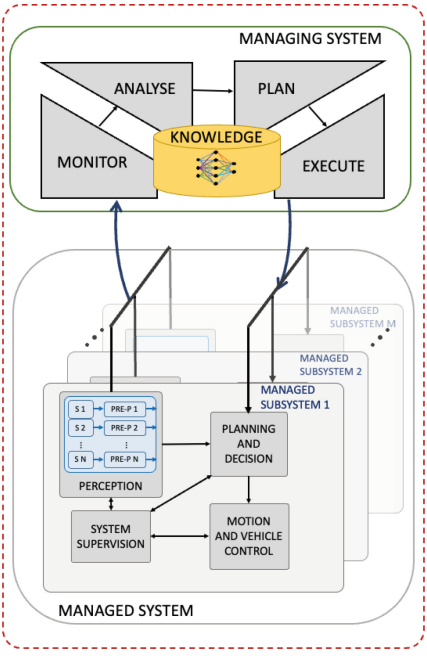


Fig. 3. Diagram illustrating the reference architecture of the self-adaptive swarm autonomous system framework composed by a managing system and a managed system. The managing system is based on the Monitor, Analyse, Plan and Execute over a shared Knowledge (MAPE-K) feedback loop. The managed system consists of multiple subsystems, with Managed Subsystem 1 highlighted. This includes the blocks Perception, Planning and Decision, System Supervision, and Motion and Vehicle Control. Arrows indicate the flow of information between the managing and managed systems, emphasising the interaction between knowledge processing and subsystem management.

5 Simulation and Obstacle Recognition

This section provides information about the approach used to simulate specific environmental condition, to implement the obstacle detection and for the information fusion.

To study how environmental (i.e., exogenous) factors affect road-crossing decisions, we simulate relevant outdoor and environmental disturbances by means of appropriate *filters*. Those filters, applied to the frames recorded by the video cameras, can mimic the presence of adverse conditions such as darkness, rain and fog.

Raw data from cameras are split into frames using the OpenCV library.[7] The environmental factors are simulated at the frame level by the Automold library for data augmentation. We use the library to simulate four different effects with different intensity that can be set through coefficients:

- *fog* with coefficient 0.3, 0.5 and 0.7 (Fig. 4a);
- *rain* of type *drizzle*, *heavy*, and *torrential* (Fig. 4b);
- *bright* with coefficient 0.3, 0.5, 0.7, and 0.9 (Fig. 4c);
- *dark* with coefficient 0.3, 0.5, 0.7, 0.9 (Fig. 4d).

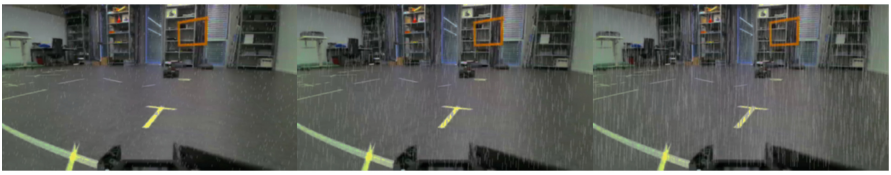
The application of most filters on a frame of our dataset is shown in Fig. 4. We refer to the unfiltered images and sequences as the *original* ones.

For obstacle detection, original and augmented frames are processed by YOLO (*You Only Look Once*).[8] YOLO is a state-of-the-art object detection architecture, widely used for its real-time performances, generalization and adaptability capabilities. YOLO's pre-trained deep neural networks are based on the COCO image dataset [8], consisting of 80 categories including person, bicycle, car, motorcycle, bus, train, and truck. Together with the recognition of the object class in the frame under elaboration, YOLO provides the pixel coordinates of the bounding box of the recognised object. If the focal length of the camera is known, the distance between the object and the camera can be derived.

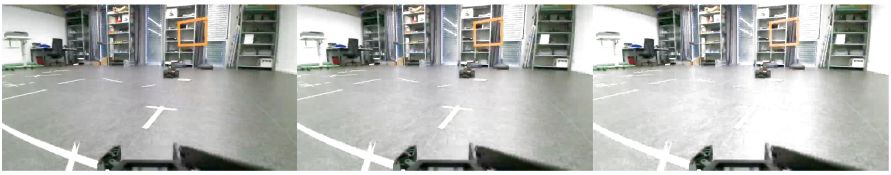
In [5], YOLO v5—the latest version available at that time—is adopted for the obstacle detection elaboration from the cameras videos. Such model does not have a dedicated class for the RM, however, it recognises the device always under the *motorcycle* class. In conclusion, unfortunately this version did not provide high performance for distances longer than a couple of metres. As an alternative to this pre-trained model, we implement the fine-tuning of the new YOLO v8 architecture by means of a dataset of RM images[9] whose results can be seen in Fig. 5. This model demonstrates to have better performance for long distances of the obstacle RM_c. Instead, for short distances we encounter similar or worst results with respect to YOLO v5. Furthermore, the new fine-tuned model provides enlarged bounding boxes, therefore the focal length has to be modified by a constant factor to overcome this distortion. In the following, we refer to the pre-trained architecture as Y5 and to the fine-tuned one as Y8.

[7] https://docs.opencv.org/4.x/index.html.
[8] https://github.com/ultralytics/yolov5.
[9] https://universe.roboflow.com/godwyll-aikins/robomaster-i5ydd.

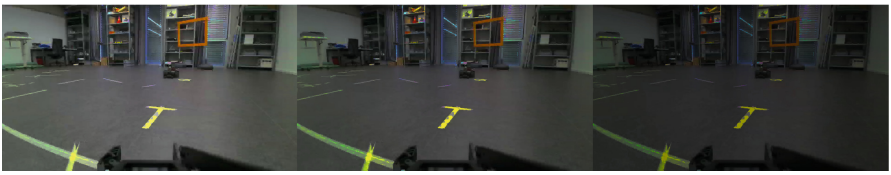
(a) Fog filter at level 0.3, 0.5 and 0.7.



(b) Rain filter at level drizzle, heavy and torrential.



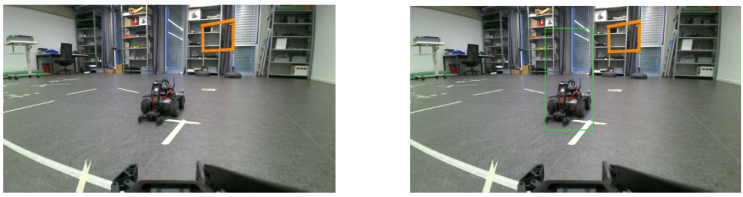
(c) Brightness filter at level 0.3, 0.5 and 0.7.



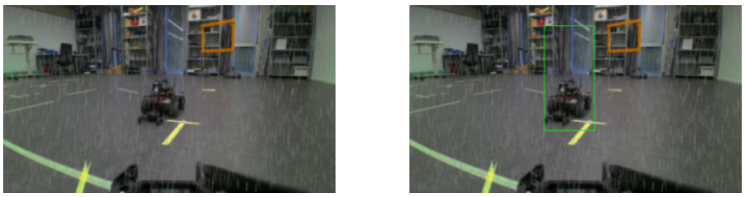
(d) Darkness filter at level 0.3, 0.5 and 0.7.

Fig. 4. A frame of the recorded dataset processed with multiple filters simulating diverse environmental conditions. Each row demonstrates the impact of the respective filter on the visibility of the obstacle. 1. Top row: Fog filter at levels 0.3, 0.5, and 0.7, progressively increasing fog density. 2. Second row: Rain filter with effects of drizzle, heavy rain, and torrential rain. 3. Third row: Brightness filter at levels 0.3, 0.5, and 0.7, showing increasing brightness. 4. Bottom row: Darkness filter at levels 0.3, 0.5, and 0.7, showing increasing darkness.

To guarantee redundancy and diversity during the information fusion, we consider both Y5 and Y8. Both models are therefore applied to all the frames of all the experiments to identify the bounding boxes of the obstacle (identified as a *motorcycle* by Y5 and as a *RM* by Y8). Therefore, a total of four data flows for the information fusion can be defined: two processed from the $\mathrm{RM_d}$ camera



(a) Original frame without and with the bounding box.



(b) Frame with filter rain without and with the bounding box.

Fig. 5. Examples of environmental filters applied on a frame of our dataset before and after the application of the bounding boxes elaborated by the fine-tuned model of YOLO version 8. (a) Top row: On the left the original frame and on the right the same frame with the green bounding box generated by the model around the obstacle. (b) Bottom row: On the left the frame with a rain filter and on the right the same frame with the rain filter and the green bounding box generated by the model around the obstacle generated by the model.

using the two different YOLO models, and, processed in the same way, another two from the RM_w camera. It can be noted that in this approach redundancy and sensory diversity are obtained through software thanks to the use of different ML models. The distance sensors introduced in Subsect. 3.2, are not included since artificial noises can not be applied to reproduce the above-mentioned environmental factors. The distance of the obstacle is computed from the bounding boxes obtained by the YOLO models [7].

For each sequence, we have four signals generated by the two cameras. We can combine these by a trivial sensor fusion approach using the arithmetic average of the position measurements for a given frame as discussed in Subsect. 3.3. We denote such a procedure as *A-Fusion*. As an alternative approach, denoted in the following as *W-Fusion*, we consider an arithmetic fusion of the distance measurements that removes the signal that are not based on actual measurements but on imputations. In case of missing data, due to the absence of obstacles or sensor malfunction, a *reference* maximum distance value is imputed. This equates to the maximum detectable distance of interest, which in the case of our tests is 4 m. This strategy correctly identifies situations where the sensor has never seen an obstacle or it is no longer visible in the sensor field of view. If all the four measures are based on an imputation, we just consider the *reference* maximum distance.

6 Experimental Results

In this section, we present the results of our experimental analysis. As discussed in the previous section, we apply relevant filters to the video sequences to simulate environmental conditions. The original, unperturbed, video sequences are also considered for comparison.

As performance descriptors, we consider the percentage of frames where the target has been recognised by YOLO (Table 1), the *Root Mean Square Error* (RMSE) of the DF based on the sensors measurements, or the sensors fusion procedures, compared against the ground-truth measurements provided by the tracker (Table 2). We similarly proceed for the road-crossing decisions, based on the DF value compared against the threshold levels described in Eq. 2, by reporting accuracies (Table 3) and F1 measures (Fig. 6). As quantitative descriptors we provide mean values over the different sequences and, in parentheses, the standard deviations. The only exception is the F1 score. For the sake of simplicity, we prefer to present the aggregated results of the two scenarios as box plots. These data are reported separately for the two datasets scenario as a large discrepancy in performance was found. This is probably due to the difference in backgrounds between scenarios A and B, described in the Subsect. 3.2, which can influence the obstacle detection of the two YOLO models.

Regarding the frame recognition rates in Table 1, the filters simulating adverse weather conditions typically induce rates that are lower than those in the original sequences. In extreme conditions (i.e., fog with coefficient 0.7 and heavy rain) those rates are close to zero, resulting in very low performance. The corresponding results are not reported for the sake of space.

As expected, the Y8 model results to work better with respect to the Y5 model, especially for highly adverse conditions (e.g., *heavy rain*). However, there is a clear difference in performance between the data from $\mathrm{RM_d}$ and $\mathrm{RM_w}$. Indeed, the best performance is achieved using the videos recorded by the $\mathrm{RM_d}$, probably due to the fact that in these shots the $\mathrm{RM_c}$ has poses similar to those of the dataset used to fine tune the Y8 model.

Unexpectedly, the brightness and darkness filters with low coefficient levels (i.e., 0.3 and 0.5) induced higher recognition rates. In practice, such filters act as an image preprocessing improving the performance of obstacle recognition. Probably this is due to the blurring of background elements, so that the $\mathrm{RM_c}$ is better recognisable from the YOLO models. For *bright* and *dark* filters, the rates are quite similar for the different values of the coefficients, and we therefore report only the data for a single coefficient for each filter.

Regarding the RMSE and accuracy values in Table 2 and Table 3, similar observations to Table 1 can be obtained. Adverse environmental filters generally lead, with a few minor anomalies, to a deterioration in performance with the exception of the *bright* and *dark* filters, which keep them constant or even improve them. The two fusion *W-Fusion* and *A-Fusion* do not differ significantly in terms of RMSE. On the other hand, *W-Fusion* is definitely better considering the accuracy values in Table 3. This means that although the two fusion methods

Table 1. Means and deviations of the frame recognition rates of YOLO pre-trained (Y5) and fine-tuned (Y8) for the two videocameras of RM_d and RM_w.

Sc.	Cam	YOLO	Original	Fog		Rain		Bright	Dark
				0.3	0.5	drizzle	heavy	0.5	0.5
A	RM_d	Y5	20.52 (4.04)	21.11 (4.94)	7.61 (4.31)	27.33 (4.03)	0.82 (0.72)	18.10 (3.97)	17.28 (4.06)
		Y8	**73.41** (9.21)	**52.30** (10.37)	**28.16** (9.25)	**73.11** (9.52)	**69.00** (7.99)	**75.32** (8.49)	**76.10** (7.95)
	RM_w	Y5	35.96 (18.16)	31.53 (15.96)	6.44 (6.34)	25.82 (13.24)	0.20 (0.50)	31.73 (16.56)	36.61 (5.87)
		Y8	41.63 (5.45)	8.12 (9.92)	6.13 (1.57)	29.30 (14.07)	24.83 (13.80)	46.56 (11.28)	52.31 (10.90)
B	RM_d	Y5	38.03 (12.22)	35.59 (10.75)	11.12 (4.33)	38.27 (10.80)	0.16 (0.39)	42.23 (12.69)	43.42 (12.74)
		Y8	**47.39** (8.87)	**52.02** (4.66)	**19.09** (6.22)	**40.53** (3.56)	**26.16** (2.61)	**81.42** (6.95)	**81.17** (6.10)
	RM_w	Y5	31.86 (10.03)	27.10 (14.29)	18.53 (10.54)	15.53 (5.63)	0.20 (0.32)	27.41 (11.31)	26.66 (11.44)
		Y8	25.51 (5.45)	12.44 (4.31)	4.59 (1.62)	15.02 (5.31)	14.16 (5.12)	58.31 (5.95)	64.70 (9.63)

Table 2. Means and deviations of the RMSE for the danger function of YOLO pre-trained (Y5) and fine-tuned (Y8) for the two video cameras of RM_d and RM_w and for the two fusion strategies *A-Fusion* and *W-Fusion*.

Sc.	Cam	YOLO	Original	Fog		Rain		Bright	Dark
				0.3	0.5	drizzle	heavy	0.5	0.5
A	RM_d	Y5	1.44 (0.40)	1.39 (0.38)	1.52 (0.45)	1.46 (0.38)	1.55 (0.47)	1.40 (0.41)	1.38 (0.34)
		Y8	1.15 (0.45)	1.33 (0.39)	1.54 (0.37)	**1.13** (0.42)	1.26 (0.46)	1.15 (0.42)	1.15 (0.43)
	RM_w	Y5	1.35 (0.35)	1.40 (0.38)	1.55 (0.45)	1.44 (0.39)	1.55 (0.47)	1.37 (0.36)	1.35 (0.35)
		Y8	1.43 (0.44)	1.53 (0.44)	1.54 (0.46)	1.46 (0.40)	1.49 (0.42)	1.41 (0.43)	1.50 (0.42)
	A-Fusion		1.22 (0.46)	1.29 (0.44)	**1.40** (0.44)	1.23 (0.41)	1.32 (0.43)	1.20 (0.45)	1.25 (0.43)
	W-Fusion		**1.09** (0.41)	**1.15** (0.41)	1.49 (0.37)	**1.17** (0.40)	**1.22** (0.46)	**1.07** (0.41)	**1.07** (0.35)
B	RM_d	Y5	1.26 (0.32)	1.25 (0.29)	1.16 (0.30)	1.28 (0.35)	1.13 (0.34)	1.27 (0.35)	1.28 (0.27)
		Y8	1.06 (0.29)	1.03 (0.19)	1.39 (0.28)	1.02 (0.27)	1.10 (0.33)	0.90 (0.16)	1.11 (0.29)
	RM_w	Y5	1.08 (0.22)	1.20 (0.19)	1.23 (0.22)	1.11 (0.31)	1.13 (0.34)	1.17 (0.25)	1.07 (0.24)
		Y8	1.01 (0.29)	1.07 (0.31)	1.12 (0.33)	1.05 (0.30)	1.06 (0.30)	0.93 (0.28)	1.09 (0.31)
	A-Fusion		**0.88** (0.27)	**0.88** (0.28)	**1.06** (0.34)	**0.93** (0.29)	1.01 (0.31)	**0.82** (0.28)	**0.97** (0.29)
	W-Fusion		1.00 (0.13)	1.12 (0.18)	1.36 (0.22)	1.13 (0.24)	**1.01** (0.29)	1.01 (0.23)	1.12 (0.16)

do not differ much from each other in terms of risk evaluation, the *W-Fusion* method is more precise for road-crossings decisions. For a deeper analysis of the quality of the road-crossing decisions provided by our system, we also consider the F1 performance. The boxplots in Fig. 6 for the original sequences and for two filters clearly show the advantages of the *W-Fusion* procedure we proposed with respect to the trivial arithmetic average *A-Fusion*. Notably, the performance of our fusion procedure is superior to that of individual sensors, with few exceptions, advocating the advantages of adding redundancy and diversity to the sensor system equipment. The cases where this is not observed are special situations where one sensor outperforms the others. Since we use a majority voting system, in this particular case, the fusion tends to worsen the results. As a solution to this problem, performance indicators can be imputed to each sensor and then combined by averaging their contributions appropriately.

Table 3. Means and deviations of the accuracy for road-crossing decision of YOLO pre-trained (Y5) and fine-tuned (Y8) for the two video cameras of RM_d and RM_w and for the two fusion strategies *A-Fusion* and *W-Fusion*.

Sc.	Cam	YOLO	Original	Fog		Rain		Bright	Dark
				0.3	0.5	drizzle	heavy	0.5	0.5
A	RM_d	Y5	44.64 (7.98)	48.45 (6.35)	36.63 (7.28)	46.04 (8.21)	35.55 (8.15)	44.74 (7.04)	49.75 (5.47)
		Y8	69.55 (4.55)	59.06 (13.94)	46.23 (13.90)	**69.56** (5.59)	**63.76** (4.73)	**73.92** (6.12)	68.90 (4.86)
	RM_w	Y5	57.29 (6.94)	51.85 (7.29)	36.65 (8.43)	49.10 (5.37)	35.55 (8.15)	54.29 (7.05)	58.04 (6.72)
		Y8	48.61 (2.66)	37.77 (7.24)	37.08 (7.75)	46.01 (8.91)	41.59 (4.29)	48.06 (3.92)	43.26 (5.51)
	A-Fusion		43.43 (3.92)	43.58 (5.12)	35.55 (8.15)	42.06 (5.10)	35.55 (8.15)	44.53 (4.25)	44.17 (4.52)
	W-Fusion		**69.37** (5.10)	**63.53** (10.09)	**53.44** (13.28)	**64.39** (5.20)	**62.19** (5.36)	**71.39** (4.95)	**70.09** (2.53)
B	RM_d	Y5	67.95 (11.02)	68.75 (9.60)	62.62 (18.12)	68.04 (9.43)	62.34 (18.89)	70.94 (6.69)	66.01 (11.26)
		Y8	63.22 (15.09)	65.23 (12.42)	62.26 (16.10)	66.45 (12.49)	63.09 (17.41)	75.17 (5.25)	62.90 (16.70)
	RM_w	Y5	68.49 (11.52)	65.61 (14.55)	63.16 (15.88)	64.50 (16.78)	62.34 (18.89)	64.01 (13.72)	68.65 (12.12)
		Y8	72.18 (10.20)	66.26 (15.51)	62.73 (18.68)	67.73 (13.11)	66.99 (14.75)	71.90 (8.53)	65.19 (15.54)
	A-Fusion		64.49 (14.93)	66.32 (14.17)	62.35 (18.53)	64.02 (16.41)	62.34 (18.89)	66.84 (12.71)	64.07 (15.74)
	W-Fusion		**72.30** (5.54)	**70.98** (5.89)	**63.60** (13.97)	**72.16** (6.67)	**68.58** (11.49)	**74.49** (4.48)	**70.03** (9.22)

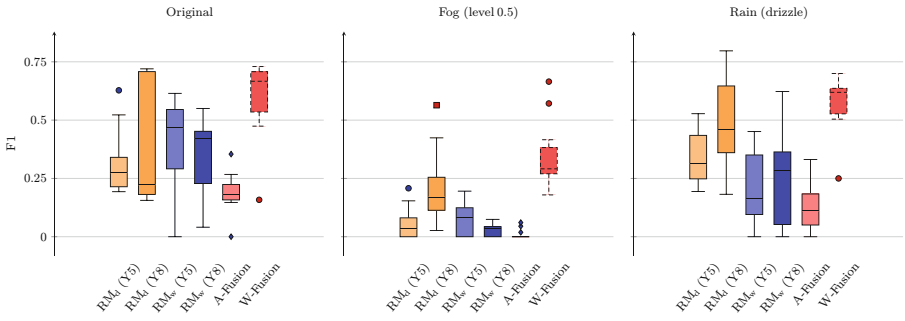


Fig. 6. The image displays three box plots comparing F1 scores across different conditions: Original, Fog (level 0.5), and Rain (drizzle). The plots illustrate the impact of weather conditions on model performance for the two video cameras of RM_d and RM_w processed with the two models YOLO version 5 and the fine tuned YOLO version 8 and for the two fusion strategies *A-Fusion* and *W-Fusion*.

Both generic Y5 and fine-tuned Y8 are negatively influenced by adverse weather as expected, with the exception of the *bright* and *dark* filter. However, Y8 appears to be more robust against environmental disturbances. The new *W-Fusion* outperforms the simpler *A-Fusion* and all single sensors in terms of accuracy and F1 performance for the road-crossing decision.

7 Conclusions

Our study addressed the crucial aspect of road-crossing decisions by autonomous wheelchairs, supported by flying drones, in the face of various environmental factors. By focusing on the refinement of obstacle detection performance

through artificial vision and its evaluation against outdoor conditions like fog, rain, brightness and darkness, we aimed to quantify the impact of these factors on tracking accuracy of both single sensors and information fusion approach. This analysis was conducted in the perspective of a future integration of safety strategies into the novel framework for the sensory system we proposed in this work.

Adverse environmental conditions were found to significantly challenge video tracking performance. However, our approach facilitated a systematic assessment of these effects within the specific operational context. By leveraging available laboratory datasets and employing tailored software filters, we demonstrated the feasibility of evaluating video tracking robustness against environmental variables in reference scenarios. Our findings underscore the importance of proactively addressing environmental challenges in autonomous wheelchair navigation systems. We also established that different types of sensors are affected differently by environmental factors. This highlights the importance of using an information fusion approach between multiple sensors.

In future works, we plan to identify instances where tracking accuracy falls below acceptable thresholds, enabling the issuance of timely warnings and the potential reconfiguration of sensor priorities to enhance overall safety. In addition, the incorporation of supplementary sensors, such as light and rain sensors, will provide additional layers of detection for critical situations.

The approach presented in this study can be generalised and therefore applied to a larger class of cooperative autonomous robots and self-driving vehicles in order to test their robustness against environmental factors. In the future, we plan to extend the experimentation in multiple operating conditions and more realistic scenarios, also including real sensors on-board the autonomous wheelchair that can be affected by different disturbances.

Acknowledgment. This work was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract no. 22.00291 (REXASI-PRO project). The project has been selected within the European Union's Horizon Europe research and innovation programme under grant agreement ID: 101070028 (call HORIZON-CL4-2021-HUMAN-01-01). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the funding agencies, which cannot be held responsible for them.

References

1. Corradini, F., Flammini, F., Antonucci, A.: Probabilistic modelling for trustworthy artificial intelligence in drone-supported autonomous wheelchairs. In: Proceedings of the First International Symposium on Trustworthy Autonomous Systems (TAS 2023), Article no. 52, pp. 1–5. Association for Computing Machinery, New York (2023)
2. El Hadmani, S., Benamar, N., Younis, M.: Pedestrian support in intelligent transportation systems: challenges, solutions and open issues. Transp. Res. Part C: Emerg. Technol. **121**(3), 102856 (2020)

3. Flammini, F., Alcaraz, C., Bellini, E., Marrone, S., Lopez, J., Bondavalli, A.: Towards trustworthy autonomous systems: taxonomies and future perspectives. IEEE Trans. Emerg. Top. Comput. **12**(2), 601–614 (2024)
4. Flammini, F., Marrone, S., Nardone, R., Caporuscio, M., D'Angelo, M.: Safety integrity through self-adaptation for multi-sensor event detection: methodology and case-study. Futur. Gener. Comput. Syst. **112**, 965–981 (2020)
5. Grigioni, C., Corradini, F., Antonucci, A., Guzzi, J., Flammini, F.: Safe road-crossing by autonomous wheelchairs: a novel dataset and its evaluation. In: Computer Safety, Reliability, and Security, SAFECOMP 2024 Workshops: DECSoS, SASSUR, TOASTS, and WAISE, Florence, Italy, 17 September 2024, pp. 47–60. Springer, Heidelberg (2024)
6. Henje, C., Stenberg, G., Lundälv, J., Carlsson, A.: Obstacles and risks in the traffic environment for users of powered wheelchairs in Sweden. Accid. Anal. Prev. **159**, 106259 (2021)
7. Lee, J.M., Hwang, K., Jung, I.H.: Real distance measurement using object detection of artificial intelligence. Turkish J. Comput. Math. Educ. (TURCOMAT) **12**(6), 557–563 (2021)
8. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014)
9. Zhang, Y., Carballo, A., Yang, H., Takeda, K.: Perception and sensing for autonomous vehicles under adverse weather conditions: a survey. ISPRS J. Photogramm. Remote. Sens. **196**, 146–177 (2023)
10. Ng, M., Jagetiya, D., Gao, X., Shi, H., Gao, J., Liu, J.: Real-time detection of objects on roads for autonomous vehicles using deep learning. In: 2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, pp. 73–80 (2022)
11. Sellat, Q., Ramasubramanian, K.: Advanced techniques for perception and localization in autonomous driving systems: a survey. Opt. Mem. Neural Netw. **31**, 123–144 (2022)
12. Velasco-Hernandez, G., Yeong, D.J., Barry, J., Walsh, J.: Autonomous driving architectures, perception and data fusion: a review. In: 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, pp. 315–321 (2020)
13. Wang, Z., Wu, Y., Niu, Q.: Multi-sensor fusion in automated driving: a survey. IEEE Access **8**, 2847–2868 (2020)
14. Li, L., et al.: An investigation of ADAS camera performance degradation using a realistic rain simulation system in wind tunnel. Presented at the WCX SAE World Congress Experience, SAE International (2024)
15. Zhu, Z., Li, X., Zhai, J., Hu, H.: PODB: a learning-based polarimetric object detection benchmark for road scenes in adverse weather conditions. Inf. Fusion **108**, 102385 (2024)
16. SAE International Recommended Practice. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. SAE Standard J3016_202104 (2014)
17. Liu, D., Cui, Y., Cao, Z., Chen, Y.: A large-scale simulation dataset: boost the detection accuracy for special weather conditions. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2020)
18. Khaleghi, B., Khamis, A., Karray, F.O., Razavi, S.N.: Multisensor data fusion: a review of the state-of-the-art. Inf. Fusion **14**(1), 28–44 (2013)

19. Cui, Y., et al.: Deep learning for image and point cloud fusion in autonomous driving: a review. IEEE Trans. Intell. Transp. Syst. **23**(2), 722–739 (2022)
20. Arcaini, P., Riccobene, E., Scandurra, P.: Modeling and analyzing MAPE-K feedback loops for self-adaptation. In: 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, Florence, Italy, pp. 13–23 (2015)