AI-RIDE: A Multi-Camera System for the Evaluation of Motorcycle Driving Test

Giuseppe Riccardo Leone*, Marco Righi*§, Davide Moroni*, Anthony Baiamonte*†, Davide Bulotta*† and Francesco Paolucci‡
*Institute of Information Science and Technologies, National Research Council of Italy, Pisa, Italy
e-mail: *name.surname*@isti.cnr.it
†University of Pisa, Pisa, Italy
e-mail: {d.bulotta, a.baiamonte}@studenti.unipi.it
‡Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Pisa, Italy
e-mail: francesco.paolucci@cnit.it
§e-mail corresponding author: marco.righi@isti.cnr.it

***Abstract*—The AI-RIDE project proposes adopting an accelerated, online, and embedded Artificial Intelligence framework in motorcycle rider training, mainly targeting the Practical Driving Courses (PDC) and Driving License Exam (DLE) sessions verification tools. The project targets a disruptive innovation step in the context of driving learning techniques, significantly going beyond the state of the art of the current instruments used in the PDC and DLE ecosystem. This work presents last year's activities with the promising results obtained with the first working prototype.**

***Index Terms*—Camera-based systems, edge computing, trajectory analysis, computer vision, artificial intelligence, motorcycle driving test**

I. Introduction

The evolving state-of-the-art IoT sensors for driving/riding, driven by the automotive ecosystem and autonomous driving applications, generate vast amounts of data for trajectory and behaviour predictions. However, these instruments are unsuitable for providing instructive feedback during human learning and training phases. Interacting with humans in driving school scenarios requires a different approach. Currently, no available technology in the market or scientific literature exists for automating motorbike training and testing procedures. One year ago, we presented a multi-camera system [1]that could solve this problem. From that idea, the Artificial Intelligence-driven RIding Distributed Eye (AI-RIDE) project was born, and with this work, we aim to present the very first working prototype. The AI-RIDE project focuses on Practical Driving Courses (PDC) and Driving License Exams (DLE) for motorbikes. Factors such as performance time, trajectory precision, speed management, driver's posture, and motorbike position influence the exam outcome. Video cameras along the circuit track provide data for computing these factors. However, cognitive features and specific human-interpretable feedback are crucial in improving performance.

Such a framework would also provide instructors with interactive teaching tools and improve learning sessions. It also facilitates automatic verification and verifiability during license exams, offering objective performance evaluation parameters for AI-computed exam scores. This innovative toolkit would benefit the entire drive/ride license ecosystem.

Fig. 1. The motorcycle driving licence test takes place on predefined paths delimited with traffic cones: one path is smaller with close passages and low speed (Low Speed Balance - LSB) and the other is larger and requires higher speeds (High Speed Agility - HSA). On both of them, there is a slalom at the beginning, then a curve, and finally, a straight section. The driver must stop the vehicle in a specific space on the long track. Everything must terminate respecting time constraints. The picture shows the two tracks at the testbed site, the AUG of Pontedera.

The AI-RIDE solution utilizes data fusion and AI processing to improve motorbike riding performance: it combines information from all the cameras on the track circuit, providing details on body posture, vehicle position, speed/trajectory, and penalty events.

Additionally, the AI-RIDE system records critical performance parameters and provides online feedback and test scores. It offers objective indicators to exam committees for evaluating test performance. Automatic performance evaluation, scores, and outcomes are provided remotely during exams thanks to a distributed camera-triggered video pro-

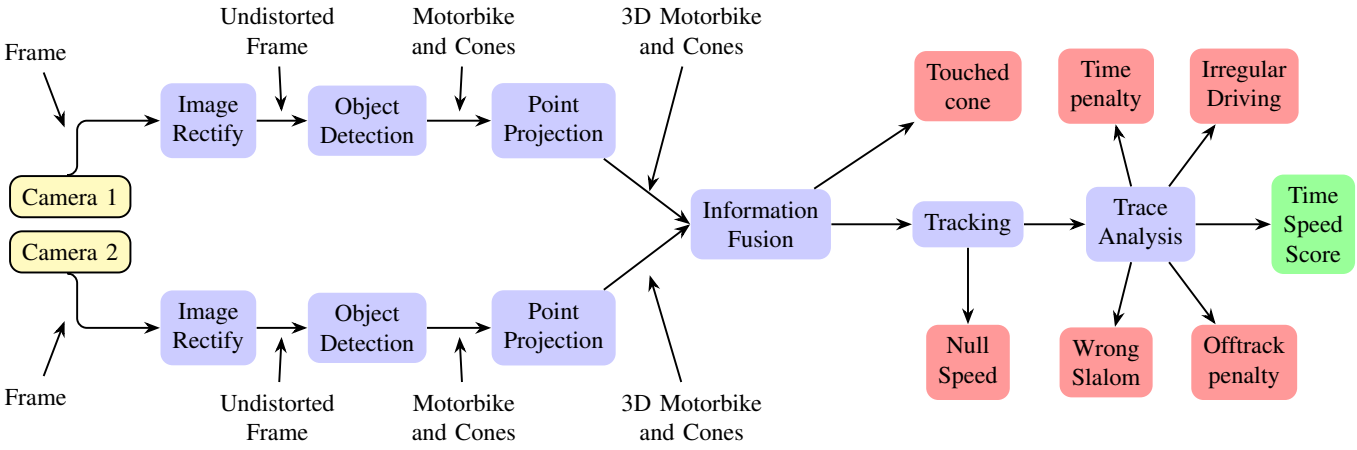

Fig. 2. AI-RIDE system architecture: the violet borderless shape are the image processing steps and the red rectangles indicate the penalties. In case of positive result the system gives information about the execution time, the average speed and an experimental score with respect to a reference path executed by the instructor. This example is the instance of the Low Speed Balance (LSB) track which uses two cameras, but it is possible to work with any number of visual sensors.

cessing system, achieving objective audibility without exam commissioners needing physical attendance.

In summary, AI-RIDE provides a complex and complete video analytics system, leveraging data fusion, AI processing, and augmented reality to enhance motorbike riding performance, facilitate training, and automate exam evaluation. As such, AI-RIDE uses advanced methods of computer vision and deep learning, relying on a vast corpus of related works, as already described in [1]. Moreover, it addresses the need for tailored feedback in driving training by adapting IoT sensors and data analytics tools to enhance human learning and improve performance in motorbike riding scenarios. In the following sections, we present last year's activities: Section II shows the system design and implementation, Section III describes the testbed setup, and the evaluation results and Section IV concludes the paper discussing about the future improvements.

II. System design and Methodology

According to the Italian law [2], the first part of the motorcycle driving licence test takes place on predefined paths delimited with traffic cones: one path is smaller with close passages and low speed (Low Speed Balance - LSB) and the other is larger and requires higher speeds (High Speed Agility - HSA). To achieve a positive result, the candidate must conclude the test without committing any of the following penalties: touching one or more cones; skipping a cone during the slalom phase; exiting the course; demonstrating irregular driving; putting a foot on the ground; failing the time constraints. The AI-RIDE system architecture is scalable and designed to process streams of multiple cameras and fuse the resulting information with the primary goal of discovering the penalties that could occur (the red shape in 5). In case of a positive result, the system gives information about the execution time, the average speed, and an experimental score concerning a reference path executed by the instructor. The workflow depicted in fig.2 is based on two cameras, the instance we use for the LSB track. Let's describe in detail the processing steps.

A. Image rectification

Pinhole cameras introduce significant distortions in the grabbed images. Typically straight line are not straight anymore. A pixel of coordinate $(x_{\text{distorted}}, y_{\text{distorted}})$ is in general in the the wrong position. Image rectification allows to calculate the correct position (x, y) [3]. The main distortion is the *radial distortion*, which depends on the lens curvature and it can be represented by the following equations:

$$x_{\text{distorted}} = x \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) \quad (1)$$

$$y_{\text{distorted}} = y \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) \quad (2)$$

The farther the point (x, y) is away from the centre of the image (distance r), the more the distortion grows. k_1, k_2, k_3 are called the *coefficient of radial distortion*.

Another significant error depends on the *tangential distortion*, which occurs because the lens is not aligned with the sensor image plane. This distortion can be represented in the following way:

$$x_{\text{distorted}} = x + [(2p_1 \cdot x \cdot y) + p_2 \cdot (r^2 + 2x^2)] \quad (3)$$

$$y_{\text{distorted}} = y + [p_1 \cdot (r^2 + 2y^2) + (2p_2 \cdot x \cdot y)] \quad (4)$$

where (x, y) is the same point in the rectify image space, $r^2 = x^2 + y^2$ and p_1, p_2 are the *coefficients of tangential distortion*.

Using standard camera calibration technique [4], which involves a set of pictures of an object with known dimensions, typically a calibration grid, it is possible to find a 5-dimensional vector $\mathbf{p}$ of distortion parameters:

$$\mathbf{p} = (k_1, \quad k_2, \quad p_1, \quad p_2, \quad k_3) \quad (5)$$

and use it to map the grabbed frame into an undistorted image, which is fundamental to performing exact measurements of distance and speed.

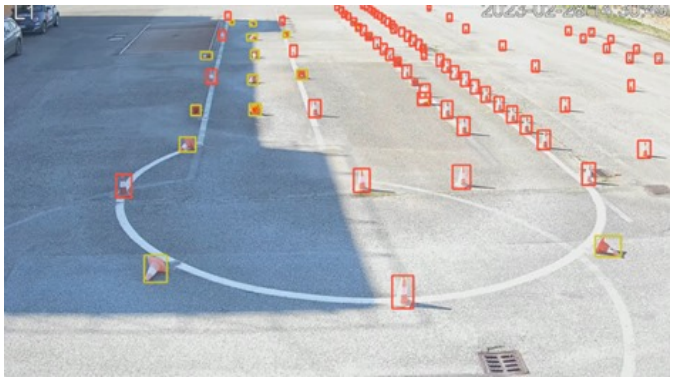

Fig. 3. Differential cone detection has been performed to distinguish between untouched cones (red) and cones on the ground (yellow). This detection is crucial for identifying exam mistakes.

B. Object detection

The object detection module is applied to all the frames to identify the motorbike, the pilot and the cones of the track. The accuracy of the outputs is vital to achieve an overall good result for the system: the exact position of the motorbike is essential to calculate the trajectory and analyse its goodness, while the cones are used to determine the border of the track. In the proposed approach, object detection is achieved by resorting to background subtraction methods [5] for identifying moving blobs of interest as well as on recent methods based on deep learning belonging to the so-called category of one-stage object detector [6]. In more detail, differential cone detection has been performed to distinguish between untouched cones and cones on the ground, see fig.3 with a bounding box of different colours. This detection is crucial to detect exam mistakes (cones touched are critical, cones moved and fallen are more critical).

For the detection of the motorbike, only cropped portions of the whole frame are used: the Foreground blob coming from the Background subtraction technique identifies the regions of the frame that are changed with respect to the previous one; this improves the accuracy and the speed of the detection of the motorbike and enables the system to know where are the cones that can be occluded or touched.

C. Points projection

In order to work with real-world coordinates, a perspective projection is a necessary step. The aim is to transform the image coordinates (fig.4-D) into coordinates of a plan view (fig.4-E) that can be a common reference system for all the different cameras. This is done with standard homography calculation [7]. The best points to project are those that lay on the ground plane. In the case of cones, the middle point of the bottom side of the bounding box is generally a good choice since cones do not move and they are relatively small

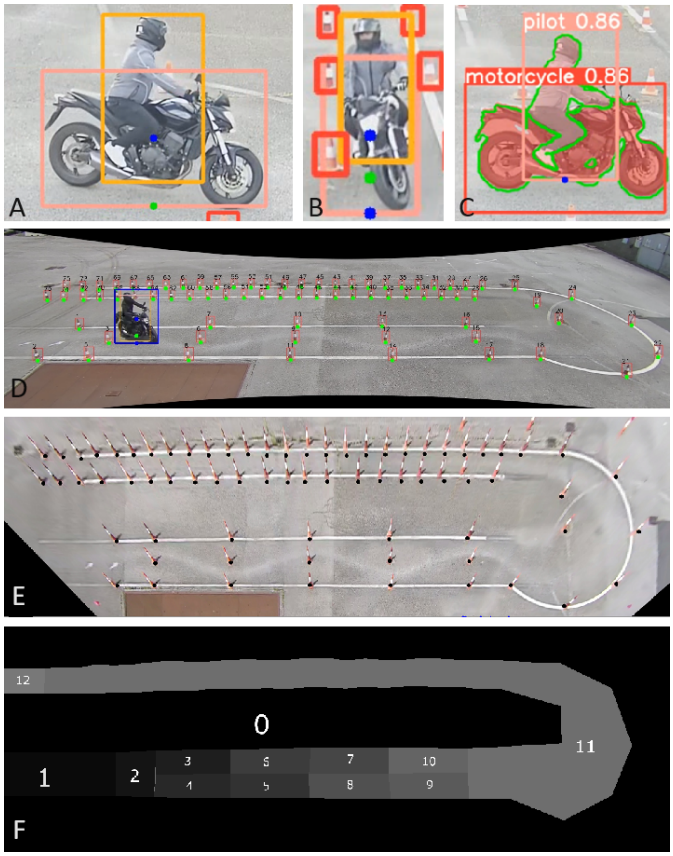

Fig. 4. Point projection: the representative point of the item should lay on the ground. A) For the motorcycle the ground point depends on the angle of view and the naive choice, like the green point, could lead to mistakes; B) Simple interpolation is a fair choice but C) more elaborate processing with semantic segmentation is the optimal. D) The middle point of the bottom side of the bounding box is a good choice for cones because they are relatively small objects; E) The projection of the LSB track with the black dots representing the cones; F) using these points the track is divided into virtual zones to understand if the path follows the right sequence or if the motorcycle enters in any forbidden area (skipped cone or exit the course

objects. In the case of motorcycles, we need more accurate processing because the ground point depends on the angle of view and the same choice of the cone could introduce big mistake (the green dot in see fig.4-A). With a simple linear interpolation between the centre of mass and the bottom of the bounding box (see the blue dots in fig.4-B) it is possible to obtain a better result (the green dot in fig.4-B. To enable finer detections, an additional semantic segmentation model has been trained for motorbike and pilot, to obtain not only the bounding box, but also the precise object profile at the pixel level; this feature is useful to identifying a much better motorbike reference point (blue dot in in fig. 4-C).

D. Information fusion

The fusion process permits to get the best performance by the system by using two or more cameras. The developed system uses two cameras installed as shown in figure 5-A. The best performance obtained in this context depends strictly on the resolution in pixels of the image. Let's consider cameras

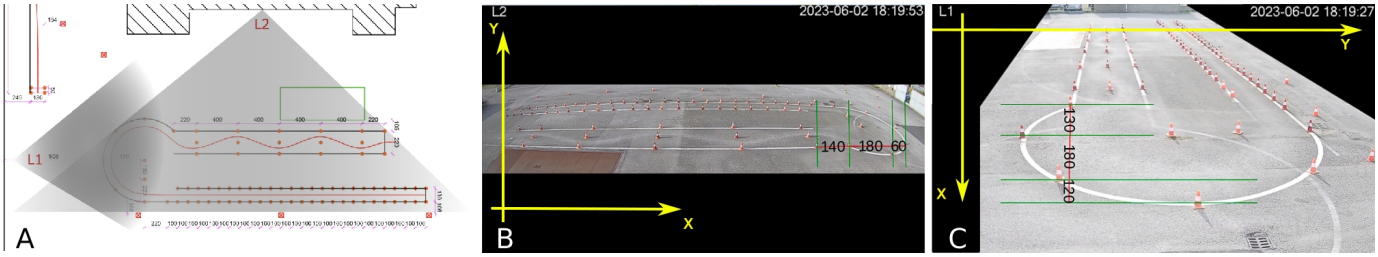

Fig. 5. For the LSB we are using 2 cameras. A) the position and the field of view of the cameras: they are placed orthogonal to each other and in a way that every camera is aligned with one dimension of the track; B) the lateral cam L2 is placed in the middle of the X dimension of the track; C) the frontal cam L1 is placed aligned with the central cones of the slalom initial part with respect to the Y dimension of the track;

without image aberration. Intuitively, the video resolution of an area next to a camera is better than that of a far area. For this purpose, we computed the resolutions of some particular areas (see 5-B and 5-C) and we noticed that, with respect to the X axis, the cam L2 has the best spatial resolution on almost all the track, except for the area closest to the cam L1. Here, L1 gets a more excellent resolution, and its spatial resolution became predominant with respect to L2. The figures 5-B and 5-C show that the closest part of the track to L1 measures the same spatial distance of L2 using 120 pixels instead of 60. After this observation, the X axis has been divided into two zones: for X values previously shown, the best choice is the L1 cam, in the other cases, the best choice is L2. Regarding the depth of the track, corresponding to Y axis, the best choice is L1 for all the tracks. In fig.4-E, the black dots represents the positions of the cones obtained after this fusion algorithm.

E. Tracking

A standard procedure connected with the tracking step is using the Kalman Filter (KF) [8]. The KF operates through a two-step process involving prediction and update steps, where it continuously refines state estimates. Its adaptability and effectiveness in real-time applications have made it a cornerstone in modern control theory and estimation theory [9]. The output of the KF is a vector which indicates the position and the speed of the vehicle. It is, therefore, straightforward to signal when the speed goes down to zero. With a high probability, this case is connected with the foot-on-the-ground penalty.

F. Trace analysis

With the output of the KF tracker, we have a good estimation of the positions and the speed of the vehicle in every moment of the recorded video. It is, therefore, an easy task to calculate the principal information of the driving test, which are average, maximum and minimum speed and the completion time. If this last value fails to complain about the constraint, the system signs a time penalty. In order to work with mathematical functions instead of linear segments, the KF points are used to calculate a set of mono-dimensional splines [10]. These functions offer a smooth visual path and can be used for integral or derivative calculation.

1) Path checking: There are two types of penalties related to the correctness of the path: if the motorbike exits the course or if one cone is skipped during the slalom phase. The system is able to discover both penalties using a lookup table that is built dynamically based on the positions of the cones calculated at the beginning of the video. In fig. 4-F, you can see the lookup table which divides the track into different virtual zones: the correct path is given by a sequence of positions that starts from value 1 and remains on that for a while and then increase by 1 passing to the consecutive number. In other words, the correct sequence of the virtual zone is 1-2-3-4-5-6-7-8-9-10-11-12. The virtual zone 0 means "exit the course" penalty. Any other sequence raises the "skipped cone" error message.

2) Overall score computation: The score computation takes place by comparing the loss under examination with the optimal path. The optimal route is evaluated and estimated by the driving instructors. The idea behind the analysis consists of the difference between the path established by the instructors and the path followed by the candidate. Each path is characterised by three main parameters: the execution time, speed and route. Very good characteristics are the average speed and the difference between the area enclosed in the expert's path and the candidate's defined by the route (see fig.6; focusing on what can be really different in irregular driving, we think that maximum and minimum speed are good values to watch and for the spatial part the correlations along the X and Y axes give significant information too. Therefore, the following features are examined:

- s_1 path execution time
- s_2 motorcycle average speed
- s_3 motorcycle maximum speed
- s_4 motorcycle minimum speed
- s_5 area defined by the route
- s_6 correlation of the X coordinates
- s_7 correlation of the Y coordinates

The candidate's driving scores, from to s_1 to s_5, are calculated using eq. 6 (r=instructor reference value, c=candidate value)

$$score = \frac{|r|}{|r| + |r - c|} \tag{6}$$

The s_6 and s_7 ratings involve the computation of the correlation coefficients as described in eq. 7 and 8. The

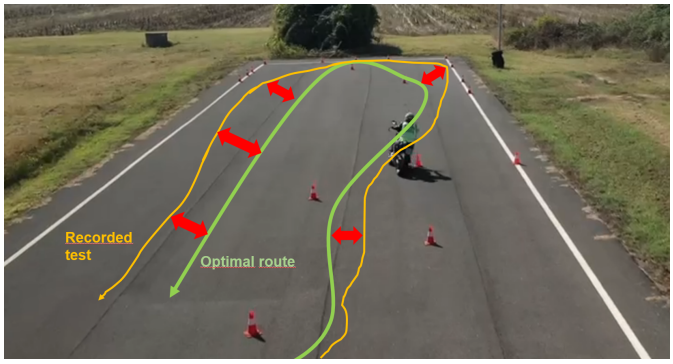

Fig. 6. Irregular driving: to determine if the driving of the test has been irregular, average speed, time and path are compared to an optimal route done by an expert instructor. In particular, the system considers the area enclosed between the recorded test and the optimal one (the less, the better).

correlation coefficients are computed for x and y coordinates.

$$\text{corr\_coeff}(x_r, x_c) = \frac{\mathbf{cov}(x_r, x_c)}{\sigma(x_r) \cdot \sigma(x_c)} \tag{7}$$

$$\text{corr\_coeff}(y_r, y_c) = \frac{\mathbf{cov}(y_r, y_c)}{\sigma(y_r) \cdot \sigma(y_c)} \tag{8}$$

These parameters are used to provide a rank of the candidate test drive (see eq. 9).

$$\text{rank} = \left(\frac{1}{n} \cdot \Sigma_{i=1}^{n} \phi_i \cdot s_i\right) \cdot \left(\prod_{j=1}^{m} e_j\right) \tag{9}$$

where $n = 7$ and each parameter s_i has its own coefficient ϕ_i because not all the events have the same importance. e_j is a binary value that becomes 0 if the corresponding penalty is detected. If some fault happens, the corresponding e_j gets the value 0, and rank values go to zero. The computed fault events are the ones we already discussed in II: 1) hitting a cone, 2) exiting the course, 3) failing time constraints, 4) skipping a cone during the slalom phase, 5) putting a foot on the ground.

III. Experimental activities

A. Testbed setup

The very first activity of the project was a detailed recognition of the track testbed Facility in Pontedera (AUG track) in order to understand the requirements and the constraints of the location. To identify the sensor characteristics and their position in the track to successfully cover all the use cases, we acquired preliminary videos at different resolutions and framerates. For this task we used a GoPro Hero 10 [11] with up 2704x1520 resolution at 240fps and 3840x2160 resolution (4K) at 120fps; the camera was mounted on an extendable telescopic stand to reach the altitude of 5 meters from the ground. After preliminary processing, considering precision and execution time, we concluded that the video requirements should be the Full HD resolution (1920x1080,2MPixel, 1080p) at 50 f.p.s which is fast enough for the maximum speed

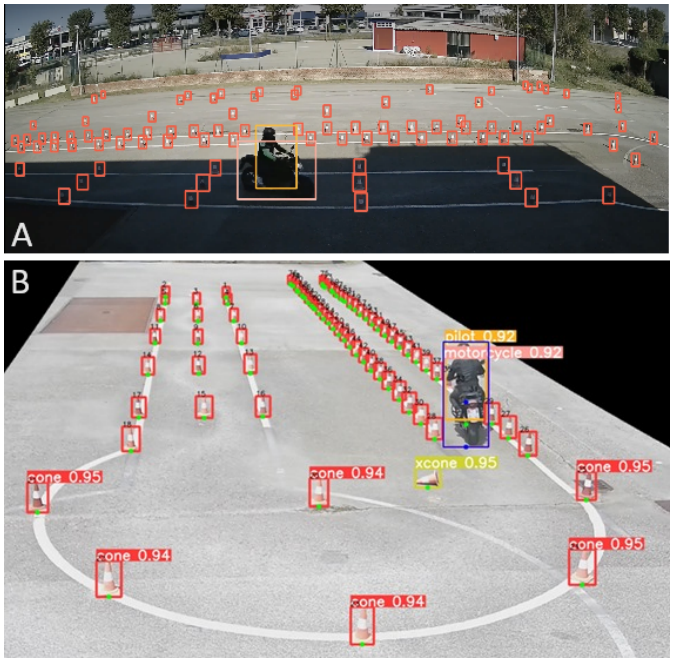

Fig. 7. Object detection of cones, motorbike and pilot: the YOLOv5 custom model has been trained with more than 4000 images, and the system is able to recognise the classes also in very different light conditions like the intense shadow you can see in the picture above.

recorded on the tracks. In addition to the previous requirements, the camera should be able to operate outdoors and have wired ethernet connectivity. The choice fell on the Dahua IPC-HFW5541E-ZE [12], an Outdoor IP camera with the advantage of a configurable focal length ranging from 2.7mm up to 13.5mm. This vari-focal feature allows the use of the same sensor as a wide-angle or low-angle, depending on the particular position. The optimal camera number is 19: these would include 2 wide-angle (6 meters height using dedicated poles) and 4 low-angle cameras for LSB track, and 2 wide-angle (6 meters height) and 9 low-angle cameras for HSA track. The need of low-angle cameras would be required to perform foot-on-the-ground detection along the whole track. This type of recognition is particularly challenging and requires detailed motorbike lateral detection. The first section of the tracks is the most critical due to the slalom trajectory. The instructors confirmed that the foot-on-the-gound event is more likely to occur in this zone. Given the budget constraints, we have identified a sub-optimal (final setup) possible camera placement: a total of 8 cameras allowing the foot-on-the-gound feature detection covering 50% of the two tracks (from one side only). In particular, we used 6 cameras for the HSA track and only 2 of them for the LSB track. The last one is the working prototype we refer to from now on. You can see the position and the field of view of the cameras in fig.5-A: they are placed orthogonal to each other and in a way that every camera is aligned with one dimension of the track; with respect to the driver L2 is called the lateral cam and L1 is called the frontal cam; fig.5-B and fig.5-C show the actual frames that

are processed in the workflow pipeline, where the mask step with the black area is done by the camera itself.

B. YOLOv5 Custom model

The detection module utilizes artificial intelligence libraries, particularly the YOLO version 5 [13], [14], trained using more than 4000 images manually annotated with Roboflow [15]. Multiple training sessions have been employed to improve the detection, facing several issues, such as the rotation of the cone, the invariance of the detection with respect to lightning and contrast image (the track is subject to different lighting conditions and shadows due to the near presence of buildings and the different sun positions during the day. This time consuming model training activity has been done using the High Performance Computer NVidia DGXA100 [16] which relies on 8 A100 G.P.U. for a total of 320GB dedicated memory. The multiple training allowed to achieve excellent detection results in extremely different light conditions, as you can see in fig.7-A. Fig.7-B shows the confidence of the four classes: it is about 0.95 for cones and between 0.9 and 0.92 for the motorbike and the pilot.

C. Evaluation results

The working prototype that we used for these early results is related to the LSB track. All the motorcycle tracking is done using only the side camera, which has a better view of all the track. The frontal camera information is used to better estimate the positions of the cones. Figure 8 shows three examples of the system's output: it consists of a top-down view of the LSB track where the green dots indicate the positions of the cones at the beginning of the test, and the black path is the one computed with the Kalman Filter. In the info-bar at the bottom, information is reported about the time and the average velocity, the outcome of the test and the experimental score that is calculated according to II-F2. The most common mistake is the "cone penalty". Fig.8-A is an example of this case: the blue dot indicates what cone is missing from the standard configuration and the red dot shows where the system sees a cone that should not be there. The yellow square around the red dot means that the system "sees" that the cone lies on the ground. A more difficult case to understand is when the cone is touched but it remains in the stand-up position: the system can understand this event if there is at least 8 cm of displacement. Other major penalties which correspond to a negative result are the exit-of-course and the skipped cone: these penalties are detected very well using the look up table that we explains in paragraph II-F1: a red dot is placed in the location where the system detects the fault; in these cases the error is connected to the precision of the position of the motorbike which is relative to its middle point; a downside of this approach is that if only a little part of the front wheel exits the border of the track the system is not able to signal it.

The time penalty is the case of fig.8-B where the end time is lower than the minimum for this track. The outcome of the test is negative and the score is zero like every case where

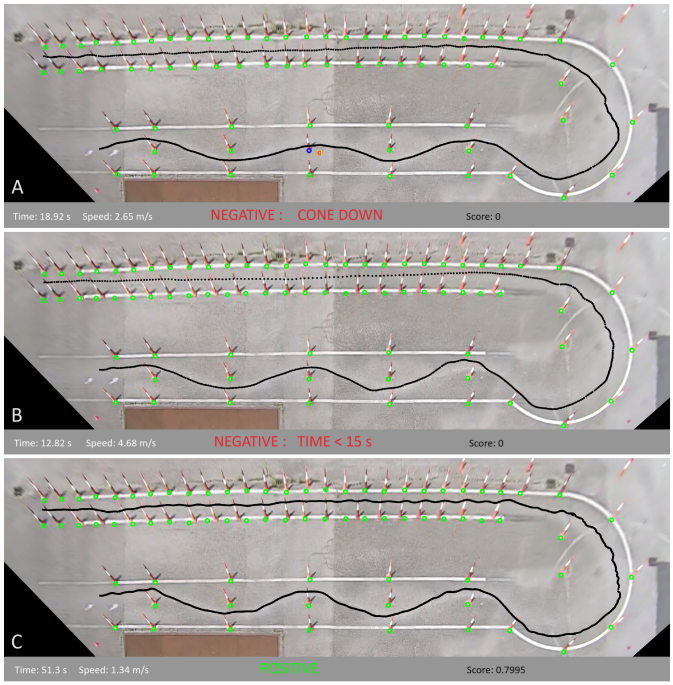

Fig. 8. Three examples of the system's outcome: A) A cone has been touched and the result is negative with an overall score of zero: the blue dot indicates the original location of the missing cone, while the red dot shows the final position; the yellow square means that the cone is laying down; B) the motorcycle has done a clean path but the end time was under the limit of 15 sec and the test is negative; C) this test is positive because there is no major penalty, however the score is low because the end time is very high compared to the optimal

there is a major penalty. The foot-on-the-ground penalty is very hard to recognised: as for now we can only check if the speed of the vehicle goes down to zero during the test and that is an error. Unfortunately very often there is a very small touch of the foot just in case of loosing the balance; we have to work on this involving the analysis of the posture of the pilot. If there is no penalty the system response is positive and an overall score is calculated: in fig.8-C the end time is almost a minute, which is not a penalty, but it is more than the double of what is expected by a good test; according to this the score of 0.79 indicates a sufficient, but not good result. Table I offers other examples of this scoring system based on the formulas shown in paragraph II-F2: the first gray column shows the reference values relative to an expert driver and the following four columns represent different test-drives. "Good" is a positive outcome with parameters very similar to the reference and the overall score is 0.945, which is higher than the "Very slow" we already discussed. It is interesting to notice the difference between the two negative outcomes in the central gray columns: both have an overall null score, but looking at the average, the example "Too Fast" is a low 0.745 while the "Out of bound" is 0.901 because the end time is more similar to the "Reference" one. We'd like to remind that this is a very experimental scoring system that needs further improvements.

TABLE I

EXAMPLES OF THE EXPERIMENTAL SCORING SYSTEM: THE PARAMETERS OF THE TESTS ARE COMPARED TO THE "REFERENCE" PERFORMED BY THE INSTRUCTOR. "GOOD" IS A POSITIVE OUTCOME WITH PARAMETERS VERY SIMILAR; "VERY SLOW" IS A POSITIVE OUTCOME TOO, BUT THE SCORE IS LOWER BECAUSE THE END TIME IS MORE THAN DOUBLE WITH RESPECT TO THE REFERENCES. "OUT OF BOUND" AND "TOO FAST" ARE EXAMPLES OF NEGATIVE RESULTS BECAUSE MAJOR PENALTIES HAS OCCURRED DURING THE TEST-DRIVE AND THEREFORE THE OVERALL SCORE IS ZERO.

	Reference	Good		Too Fast		Very Slow		Out of bound	
	value	*value*	*score*	*value*	*score*	*value*	*score*	*value*	*score*
Area	202.368 m^2	201.093 m^2	0.99374	201.401m^2	0.995244	204.845 m^2	0.987908	208.725m^2	0.969544
Corr x		0.981505	0.973566	0.995709	0.995709	0.92647	0.92647	0.974724	0.974724
Corr y		0.998595	0.998595	0.996716	0.996716	1	1	0.998567	0.998567
Time	18.5602 s	18.9202 s	0.980972	12.82 s	0.763784	51.30 s	0.361791	19.9003 s	0.932663
Max speed	571.41 cm/s	710.03 cm/s	0.804774	1158.15 cm/s	0.493384	649.81 cm/s	0.879346	718.886 cm/s	0.79486
Min speed	27.10 cm/s	24.7178 cm/s	0.919165	65.79 cm/s	0.411908	21.11 cm/s	0.818896	13.06 cm/s	0.658718
Avg speed	340.20 cm/s	319.125 cm/s	0.941642	468.62 cm/s	0.725976	133.78 cm/s	0.622372	331.367 cm/s	0.974668
Average	1		0.944636		0.764674		0.799539		0.900521
Score	1		0.944636		0		0.799539		0

IV. CONCLUSIONS AND FURTHER WORK

The AI-RIDE prototype successfully faces the problem of assigning an automatic evaluation to a candidate during a motorbike licence on-the-road test, leveraging computer vision and machine learning. The obtained results have demonstrated the system's feasibility by realising a successful proof-of-concept activity. The evaluation process uses only a few cameras (one or almost two) to compute the outcome of the examined test drive. Thus, the acquisition process introduces non-eliminable artefacts due the occlusions in the recording area, the spatial camera resolution, the frame per second of the cameras, the real-time requirements and the computational resource. The challenge is to use the available information from low-cost devices to obtain an acceptable assessment. Future goals are to use only one camera with high spatial/time resolution for the LSB track, to finish the monitoring of the HSA circuit and to extend the analysis including the car driving license examination. Other investigations will concern embedded hardware solutions to enable some edge processing, e.g. to perform the camera image distortion and the object detection step on the camera hardware. The overall goal is to leverage pervasive computing solutions and embedded systems to devise a new range of intelligent camera systems for responding to the needs of smart cities and future internet applications.

ACKNOWLEDGMENT

This work is partially supported by the VEDLIoT project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 957197.

REFERENCES

[1] G. R. Leone, M. Righi, D. Moroni, and F. Paolucci, "Towards Multi-Camera System for the Evaluation of Motorcycle Driving Test," in *PRELUDE 2022-International Workshop on PeRvasive sEnsing and muLtimedia UnDErstanding-In conjunction with SITIS 2022,*, 2022.

[2] Ministero dei Trasporti, "Decreto 26/09/2018 - prove di valutazione per conseguimento patenti a1, a2 e a," https://www.gazzettaufficiale.it/eli/id/2018/10/12/18A06493/sg, 2018, last reviewed October 17, 2023.

[3] A. Distante, C. Distante, W. Distante, and Wheeler, *Handbook of image processing and computer vision*. Springer, 2020.

[4] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov 2000.

[5] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Computer Science Review*, vol. 35, p. 100204, 2020.

[6] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, p. 103514, 2022.

[7] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2014, ch. 11.2.3 "Estimating homography from point correspondences", pp. 558–561.

[8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[9] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep. 95-041, 1995. [Online]. Available: http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html

[10] B. Bojanov, H. Hakopian, and B. Sahakian, *Spline Functions and Multivariate Interpolations*, ser. Mathematics and Its Applications. Springer Netherlands, 2014.

[11] GoPro, "HERO10 Black action camera," https://gopro.com/it/it/shop/cameras/hero10-black/CHDHX-101-master.html, 2022.

[12] Dahua, "IPC-HFW5541E-ZE: 5MP IP Vari-focal Bullet Network Camera," https://www.dahuasecurity.com/products/All-Products/Network-Cameras/WizMind-S-Series/5MP/IPC-HFW5541E-ZE=S3, 2023.

[13] G. J. et al., "Yolov5 classification models, apple m1, reproducibility, clearml and deci.ai integrations," https://zenodo.org/record/7002879#.YwUUqHZByUk, 2022.

[14] Glenn Jocher and Sergiu Waxmann, "Comprehensive guide to ultralytics yolov5," https://docs.ultralytics.com/yolov5/, last reviewed October 17, 2023.

[15] Roboflow Inc., "Roboflow annotation framework," http://www.roboflow.com/, last reviewed October 17, 2023.

[16] Chris Campa, Chris Kawalek, Haiduong Vo and Jacques Bessoudo, "Defining AI Innovation with NVIDIA DGX A100," https://developer.nvidia.com/blog/defining-ai-innovation-with-dgx-a100/, last reviewed October 17, 2023.