Performance Evaluation of YOLOv8-Based Bib Number Detection in Media Streaming Race

Rafael Martínez, Álvaro Llorente, Alberto del Rio, Javier Serrano, and David Jimenez

***Abstract*—The evolution of telecommunication networks unlocks new possibilities for multimedia services, including enriched and personalized experiences. However, ensuring high Quality of Service and Quality of Experience requires intelligent solutions at the edge. This study investigates the real-time detection of race bib numbers using YOLOv8, a state-of-the-art object detection framework, within the context of 5G/6G edge computing. We train (BDBD and SVHN datasets) and analyze various YOLOv8 models (nano to extreme) across two diverse racing datasets (TGCRBNW and RBNR), encompassing varied environmental conditions (daytime and nighttime). Our assessment focuses on key performance metrics, including processing time, efficiency, and accuracy. For instance, on the TGCRBNW dataset, the extreme-sized model shows a noticeable reduction in prediction time when the more powerful GPU is used, with times decreasing from 1,161 to 54 seconds on a desktop computer. Similarly, on the RBNR dataset, the extreme-sized model exhibits a significant reduction in prediction time from 373 to 15 seconds when using the more powerful GPU. In terms of accuracy, we found varying performance across scenarios and datasets. For example, not good enough results are obtained in most scenarios on the TGCRBNW dataset (lower than 50% in all sets and models), while YOLOv8m obtain the high accuracy in several scenarios on the RBNR dataset (almost 80% of accuracy in the best set). Variability in prediction times was observed between different computer architectures, highlighting the importance of selecting appropriate hardware for specific tasks. These results emphasize the importance of aligning computational resources with the demands of real-world tasks to achieve timely and accurate predictions.**

***Index Terms*—YOLO, object detection, bib number detection, cognitive networks, media streaming, broadcasting, edge computing, runner segmentation, image quality.**

I. Introduction

THE EVOLUTION of mobile communication technologies has triggered a significant paradigm shift in multimedia services, ushering in a new era of enriched and personalized offerings tailored to individual preferences [1], [2]. This transformation is underscored by the increasing softwarization of mobile core network functions, which is driving the evolution of the mobile network architecture itself. In its fifth generation (5G) and beyond, mobile networks have transitioned towards a service provider/consumer framework, facilitated by service-based interfaces [3], [4].

The capabilities inherent in 5G networks, including enhanced bandwidth, reduced latency, and improved reliability, hold immense significance for the delivery of audiovisual media services [5], [6]. These capabilities enable the seamless transmission of high-quality video content and support emerging technologies such as augmented reality (AR) and virtual reality (VR), thereby revolutionizing the landscape of media consumption [7], [8]. Among the many applications that have surfaced in this transformation, the integration of multimedia broadcast enrichment through cognitive services has emerged as a frontier [9], [10].

Central to this evolution is the proliferation of edge computing capabilities, which play a pivotal role in real-time multimedia content processing while ensuring compliance with Quality of Service (QoS) and Quality of Experience (QoE) standards. Moreover, the integration of edge computing (MEC) is crucial for fully leveraging the potential of 5G networks to enrich and tailor media services [11], [12]. The convergence of edge computing and 5G/6G networks underscores the need for an infrastructure that can seamlessly handle the increased data load and complexity associated with sophisticated media services, thereby enhancing the overall user experience [13].

Enhanced capabilities of MEC environments become critical enablers for deploying Artificial Intelligence (AI) frameworks in real-time applications. MEC's proximity to the end-users ensures minimal latency and high computational efficiency, which are essential for the effective implementation of AI-driven solutions such as object detection. Within this context, state-of-the-art object detection frameworks like YOLO (You Only Look Once) [14] have demonstrated exceptional performance in various tasks (classification, detection, segmentation…) along different versions, which currently is provided on version 8.[1]

Manuscript received 17 February 2024; revised 27 May 2024; accepted 29 May 2024. This work was supported in part by the Horizon Europe CODECO Project under Grant 101092696; in part by the Horizon Europe NEMO Project under Grant 101070118; and in part by the UNICO-5G I+D TSI063000-2021-79 (B5GEMINI-AIUC) Project funded by the Ministry of Economic Affairs and Digital Transformation of the Spanish Government and the NextGenerationEU [Recovery, Transformation and Resilience Plan (PRTR)]. *(Corresponding author: Alberto del Rio.)*

Rafael Martínez, Álvaro Llorente, and Alberto del Rio are with the Signals, Systems and Radiocommunications Department, Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT), Universidad Poliécnica de Madrid, 28040 Madrid, Spain (e-mail: a.delriop@upm.es).

Javier Serrano is with the Informatic Systems Department, Escuela Técnica Superior de Ingeniería de Sistemas Informáticos (ETSISI), Universidad Politécnica de Madrid, 28031 Madrid, Spain.

David Jimenez is with the Physical Electronics, Electrical Engineering and Applied Physics Department, Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT), Universidad Politécnica de Madrid, 28040 Madrid, Spain.

Digital Object Identifier 10.1109/TBC.2024.3414656

[1] https://docs.ultralytics.com/models/yolov8/

We intend to leverage its capabilities for a specific and practical application: race bib detection. The integration of YOLO in our proposed system is motivated by its proven ability to outperform other tools in object detection capabilities, and the integration of several tasks in the same framework, ensuring accuracy and reliability. Understanding the different YOLO model sizes, from nano to extreme, and configuring them for varied inference scenarios is crucial. This approach allows us to tailor the system's performance to meet specific needs, balancing between processing speed and accuracy, and effectively understanding the potential baseline of MEC configuration to deliver optimal results in diverse conditions.

A. Research Challenges

At the heart of this transformation is the integration of Artificial Intelligence (AI) into media services, enabling intelligent content processing to deliver valuable insights to both content providers and end-users. However, challenges arise when these edge services demand excessive resources or fail to deliver accurate results, emphasizing the importance of predicting real-time performance and configuring robust service architectures [15].

To address these challenges, a specific approach to managing the various layers of networking and computing infrastructure is essential. This requires an overall intelligence framework capable of orchestrating data, control, and service layers to optimize performance and ensure seamless delivery of media content. Furthermore, the validation of models and analysis of their accuracy are critical aspects of this evolving landscape. Rigorous testing and evaluation frameworks are needed to assess the adaptability and robustness of cognitive services, particularly in meeting the stringent performance requirements of multimedia applications [16]. Racing events, characterized by their diverse and dynamic sequences, serve as a canvas for the analysis and application of cognitive services, especially object detection [17], [18].

In this field, companies and developers face significant challenges [19] when trying to implement object detection solutions. One of the main problems is the difficulty in selecting the most suitable model [20] for their specific use case. This selection process is restricted by the lack of complete documentation and detailed specifications, which are essential for making informed decisions. Understanding the specifics of each model, such as its performance capabilities, processing speed and suitability for various tasks, is crucial to optimizing your applications.

For example, when it comes to real-time processing [21], detection in high-resolution images [22] or object identification in high-definition videos [23], the absence of detailed information can lead to suboptimal choices. Users need clarity on which model is most suitable for their particular requirements, taking into account factors such as image characteristics and available computational resources, whether GPU or CPU. Without this data, implementing effective and efficient object detection solutions becomes a complicated task.

In the context of YOLO models, although there is technical documentation and comparisons of YOLO with state-of-the-art datasets, the challenge lies in making these results more accessible and understandable to users. There is a need to make clear these data so that users can easier determine which variant is most appropriate for their specific needs. The lack of easily interpretable information on the speed and processing time of each model complicates the selection process, as these factors are essential for implementing effective solutions.

B. Objective and Contributions

The above problems show the importance of our research objective, which is to perform an analysis of different models and sizes of the last version of YOLO (YOLOv8), focusing on its performance in detecting runner bibs in different race datasets and environmental conditions.

In particular, at the forefront of our research is the real-time object detection system known as YOLO. By offering a spectrum of models with varying parameters such as size, inference speed, and specific task-oriented adaptability, YOLO provides a versatile toolkit. In our case, the main goal is to leverage the capabilities of YOLOv8[2] to detect and decipher the bib numbers worn by runners in different scenarios, extracting several tests and results.

Our work focuses on the optimization of object detection systems for specific tasks in racing events. For that, we examine the efficiency, accuracy and processing time of the YOLOv8 framework in different scenarios, ranging from daytime clarity to nighttime challenges. Furthermore, this study aims to determine the most effective conditions for each YOLOv8 model size, providing guidance for improving detection performance in real-time racing scenarios.

The use of open source datasets for both training and evaluation purpose, together with the wide range of running scenarios in several environmental conditions, ensures that our results can be validated and extended to other studies and research.

II. Related Works

A. Significance in the Context of 5G/6G Multicast/Broadcast Services

The new emerging multimedia services and applications differ from the traditional ones by offering an increasingly immersive experience. 4K and 8K video streaming, virtual reality, augmented reality and 360 omnidirectional video applications have popularized new scenarios and media use cases [24], [25]. Audiovisual content providers and broadcasters are highly motivated to use IP-based, mobile, and cellular distribution technologies to deliver to the end-users their media services, for being a broadly accessible and unified distribution platform [26].

5G mobile networks has brought a great revolution in the communications field. High bitrates, low latency, security and improved reliability are fulfilled by 5G technologies, enabling success in multimedia streaming where is critically important to guarantee the stability of the transmission [27]. 5G networks with the new video compression standards, the evolution of the technology and the availability of UHD portable consumer devices provide the infrastructure for "anywhere anytime"

[2]https://github.com/ultralytics/ultralytics

access to real-time broadcast media for new emerging video services. In the current era of information explosion, applications such as 5G autonomous driving, UHD video, 4K video, 8K video, 360 video, gamming and holographic metaverse applications bring massive data increments, imposing more stringent requirements on the performance of 5G wireless communications networks and seeing the need for a leap to 6G networks [28], [29].

Mobile networks are characterized by frequent changes in latency and bandwidth conditions, which might result in an unstable and poor video streaming [30]. For that, assure the QoE and QoS of the applications and services in challenging network scenarios (e.g., live streaming or video on demand) are one of the main objectives of the 5G networks [31] to satisfy the final perceived quality by the end-user through intelligent network management [32], [33].

The incoming specifications of 3GPP with the Release 17[3] [34] include the specifications for 5G Multicast-Broadcast Services (5G MBS) [35], a regulation for multicast and broadcast delivery over 5G networks [36]. During these years there has been a continuous evolution of new broadcast and multicast technology in 5G networks [37] due to the versatility, flexibility and efficiency of the technology, and the easy integration with the deployed mobile communication networks [38]. While 3GPP offers a set of specifications for the Media Industry and for the distribution of TV services to mobile devices, 5G Media Action Group (5G-MAG) has undertaken the task to develop open-source implementations of 3GPP specifications.[4]

B. Multimedia Applications on Edge Computing

The advent of cloud computing and virtualization paradigms created new market gaps for multimedia applications, driving new opportunities for the multimedia content and entertainment industries [39]. The application of Network Functions Virtualisation (NFV) and serverless paradigms for multimedia applications over 5G, has been widely analysed [40] with the use of open-source Function-as-a-Service (FaaS) enablers, such as Openwhisk,[5] for multimedia services. The EU H2020 5G-PPP 5G-MEDIA project [41] developed a transparent Service Virtualisation Platform (SVP), where the vertical service provider can deploy its virtualized service from an application-level perspective [42]. These platforms, already proven for multimedia content, provide in some cases a complete 5G infrastructure for testing verticals [43]. Another platform, in this case focused on immersive multimedia content, is the one proposed by the EU H2020 5G-PPP 5G-Xcast project [8].

From the perspective of the vertical service provider, the simplification of service deployment procedures is a key factor in terms of cost reduction [44]. This simplification reduces the time required to deploy the service [45]. Simplification and automation techniques facilitate the deployment, execution and analysis of vertical services [46]. In the case of multi-site virtualized architectures, it is possible to deploy monitoring systems that allow the analysis of virtualized services in multiple geographical locations [47]. This makes it possible to create procedures for the use of 5G-enabled end-to-end platforms for the creation and performance analysis of vertical services [48].

C. Object Detection Framework

The object detection field has been a hot topic in recent years, driven by advances in artificial intelligence [49] and the growing need for automated solutions [50] in various applications. Many studies have focused on object detection to provide solutions in diverse areas such as surveillance [51], [52], autonomous driving [53], [54], medicine [55], and many others. Object detection involves identifying and classifying objects in an image or video, and has proven to be crucial in the digital transformation of numerous industries.

Text detection is an important subcategory within object detection, which has significant applications in document scanning [56], translating text into images [57] and assisting the visually impaired [58]. Although text detection is an specific topic, it faces similar challenges [59] as general object detection. This need has led to the development of specialized tools for text detection and recognition.

Traditionally, object detection tools such as Tesseract and EasyOCR have been widely used. These tools have proven to be effective in certain contexts, such as food identification and tracking [60] and handwritten character extraction [61]. However, the primary use of these tools is for character recognition on car license plates [62], [63].

Despite their usefulness, these tools have significant limitations in terms of accuracy and the ability to handle objects in complex environments with high variability. Tesseract, for example, can struggle to perform its task when encountering low-quality images [64], as well as having fairly complex setup and configuration for non-technical users [65], which limits its accessibility. EasyOCR, while improving on some aspects of ease of use and configuration, also faces similar challenges. Its ability to handle multiple languages and diverse fonts sometimes results in decreased accuracy [66] when faced with non-standard text or less controlled situations.

To overcome these difficulties in object detection in general, more complex neural network models have been developed, which significantly improve object detection results. A prime example of such models is YOLO, which has revolutionized the computer vision field. YOLO is based on a convolutional neural network architecture that enables real-time object detection by analyzing an entire image in a single pass [67]. Since its introduction, YOLO has evolved through eight versions, each improving in terms of accuracy, speed and versatility [68]. The latest version of YOLO includes several models and sizes, allowing it to be adapted to different needs and hardware constraints.

The use of YOLO has been wide and varied in multiple applications. In passenger detection and counting, its implementation makes it possible to optimize the accuracy and efficiency of the Automation Passenger Counting (APC)

[3]https://www.3gpp.org/specifications-technologies/releases/release-17
[4]https://www.5g-mag.com/explainers
[5]https://openwhisk.apache.org/

system [69]. In the automotive industry, YOLO has been used for various tasks. For example, a novel lightweight vehicle detection method called MA-YOLO (MobileNet Attention YOLO) has been proposed [70]. This tool reduces the number of parameters by almost half compared to YOLOv8, while maintaining similar accuracy.

In addition, in the field of license plate detection and authentication, YOLO-V4 and YOLO-V5 have been used to solve specific problems. In one study [71], YOLOv4 was employed for license plate detection, while YOLOv5 was utilized for license plate class identification for authentication purposes. Similarly, in autonomous driving research [72], the YOLO algorithm has been applied to detect and classify various objects on the road using bounding boxes.

In the case of bib detection, the dynamism and variability in bib position and appearance during a competition present significant challenges. Similar studies, such as one on bib number recognition in running competitions [73], have addressed these issues. This system, which faces variability in bib appearance, size, and deformations, improves recognition accuracy using facial detectors and stroke width transforms (SWT).

Another study has presented modifications to SWT to improve its performance in detecting bib numbers in racing competition images [74]. These modifications, such as hue channel similarity testing and stroke length limitation, have been shown to significantly improve bib detection in assorted images.

Finally, a multimodal technique has been presented that combines biometric and textual features to detect and recognize bib numbers in natural images of marathons and sports competitions [75]. This technique uses face and skin features to identify candidate text regions, improving the accuracy and performance of bib recognition.

III. Methodology

A. Experimental Setup

Our experimental setup aimed to analyze the performance of the YOLOv8 models under various computational constraints, while evaluating their accuracy and efficiency in race bib detection. We employed two different hardware configurations.

The neural networks for the bibs and numbers detection have been trained from scratch. For this training, we used a high-performance desktop computer equipped with an Intel Core i9-10900 CPU and a powerful NVIDIA GeForce RTX 3090 GPU (10,496 CUDA cores, 328 Tensor cores, 24GB RAM) graphic card. This robust system efficiently managed the intense computation required for training. Specifically, the training process exclusively used the GPU's parallel processing capabilities for significant speed optimization, reducing training time to approximately 3 hours per model for bib detection, and higher time for number detection (from half a day, to two days for the extreme model). Our training datasets included around 600 sequences featuring diverse bib sizes and angles, and almost 100,000 digits in different real-world scenarios with variations in object appearance.

For the inference phase, in addition to the desktop setup, we adopted a more portable setup, using a laptop equipped with an Intel Core i5-7200U CPU and an NVIDIA GeForce MX150 GPU (384 CUDA cores, 2GB RAM). This configuration met the minimum inference requirements while offering lower computational power. This strategic choice allowed us to evaluate the feasibility of deploying trained models on resource-constrained edge devices, paving the way for potential real-world implementations in resource-limited environments.

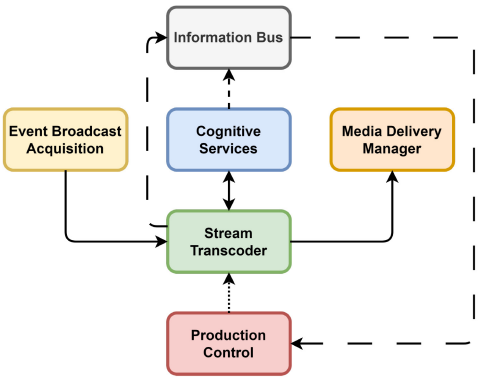

Fig. 1. General architecture proposal.

B. Media Architecture

The research presented in this work is closely related to the Cognitive Service module of a general multimedia broadcasting architecture. Although the overall goal of the architecture is to capture and enrich the User-Generated Content (UGC), it is vital to contextualize the YOLO's performance within the broader architecture. The other components serve to illustrate the composition of the real scenario, demonstrating how the Cognitive Service module can operate within a dynamic environment.

Illustrated in Figure 1, our architecture is an interconnected set of components. First, event stream acquisition ensures that the infrastructure manages access to the broadcast stream. Next, the stream transcoder optimizes the media formats, ensuring compatibility, and also adapts the bitrate of the stream. The Cognitive Services module, the main component of this work, assumes the fundamental role of enriching multimedia content, providing intelligent capabilities to enhance the overall viewing experience. In this scenario, the objective is to segment and identify the runners' bibs.

The information bus is responsible for real-time communication and coordination between the different components, acting as a data exchange channel. Finally, the production control supervises the orchestration, processing, and rendering of the content; while the media delivery manager is in charge of distributing the selected content to the different channels for end-users.

In this research manuscript, we focus on a singular test case, based on the Cognitive Services module. Our objective is to evaluate the performance of different YOLO models, focusing on the detection and prediction aspects to enrich the multimedia content, as shown in Figure 2. The process started with training YOLOv8, using data coming from the BDBD

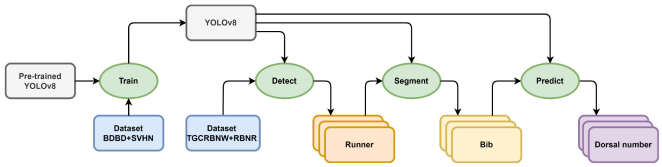

Fig. 2. Workflow of the YOLOv8 model performance analysis.

Fig. 3. Pictures samples from BDBD dataset.

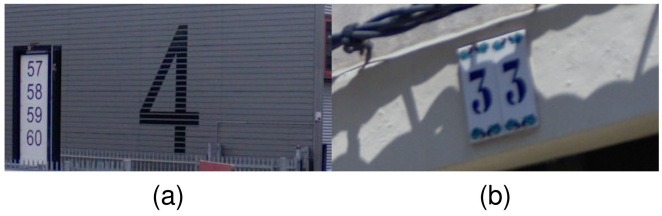

Fig. 4. Pictures samples from SVHN dataset.

(Bib Detection Big Data) dataset [76] (examples in Figure 3) for physical bibs detection and the SVHN (Street View House Numbers) dataset [77] (additional examples in Figure 4) for the identification of numbers within those bibs.

To guarantee the relevance and robustness of our findings, we chose to analyze two distinct datasets, in addition to those employed for training. After training our different sized models, we conducted the prediction process using images extracted from the TGCRBNW (Trans Gran Canaria Race Bib Number in the Wild) dataset [78] and RBNR (Racing Bib Number Recognition) dataset [79]. The selection of these datasets is deliberate, as they encompass images of runners in a variety of scenarios, thus presenting intriguing challenges for evaluating the robustness of our neural networks under different conditions. These conditions include scenarios with a single runner, multiple runners, daytime conditions and nighttime conditions (check Figure 5 for visual references on daytime, nighttime, and a crowded environment).

In this framework, an initially pre-trained YOLOv8 neural network [80] is deployed to detect each individual runner present in the image. Next, a detection process is performed within each identified runner to discern the physical paper simulating the bib. Subsequently, an additional detection is executed within the bib to determine the number associated with each runner. This stepwise filtering process, ranging from the overall image to the individual, the bib and the number, allows our tool to mitigate errors associated with the detection of extraneous elements, such as background signs or irrelevant objects, particularly in challenging scenarios.

Visual analysis of our two validation datasets reveals a rich variety of conditions and scenarios, allowing us to evaluate the YOLOv8 detection system. We begin with daytime races under ideal lighting conditions (Figure 5(a)), which serve as a baseline due to their inherent ease of detection. Subsequently, we investigate nighttime races (Figure 5(b)), where low light, shadows, and artificial illumination pose significant challenges. Finally, we investigated variations in crowd density (Figure 5(c)), evaluating how well each YOLOv8 model adapts to handle congested environments and complex interactions between objects.

C. Cognitive Services

After having analyzed the evolution of YOLO in general way, it is necessary to emphasize that, at this point, we have chosen to focus on YOLOv8. The reason for this decision lies in the advances and improvements introduced in YOLOv8 with respect to its predecessors [82]. To go into the details, Ultralytics, the developers of YOLOv8, introduced several configuration sizes, each tailored to specific needs, as illustrated in Table I.

This table presents performance metrics for different versions of YOLOv8, including nano, small, medium, large, and extreme. The metrics include size (in pixels), mean Average Precision (mAP) over the range 50-95 [83], speed in CPU, speed in TensorRT, number of parameters (Params), and the number of floating-point operations (FLOPs). These variations in model sizes allow users to choose a specific YOLOv8 configuration that fits their needs, whether prioritizing speed, accuracy, or a balance between both. For example, the nano version is optimized for speed, with lower parameters and FLOPs, while the extreme version provides higher accuracy at the cost of higher computational complexity.

In addition to the different model configurations, YOLOv8 provides users with a set of hyperparameters [84] that can be adjusted to further optimize the model performance based on specific use cases. It is often necessary to experiment with different combinations and values to find the optimal configuration for a given use case. Some of the key hyperparameters in YOLOv8 are as follows:

- Learning Rate. A crucial parameter that determines the step size during the optimization process. Proper adjustment of the learning rate is essential to achieve a balance between fast convergence and avoiding overfitting.
- Batch Size. The number of training samples used in an iteration. Adjusting the batch size can impact the convergence speed and memory requirements during training.

(a) Excellent lighting during the day (b) Reduced lighting (c) Crowd of runners at night

Fig. 5. Test cases scenarios on different conditions from TGCRBNW dataset.

TABLE I
YOLOv8 MODEL PERFORMANCE METRICS FOR COCO DATASET DETECTION [81]

Model	Size (pixels)	mAP (50-95)	Speed on CPU (ms)	Speed on TensorRT (ms)	Params (M)	FLOPs (B)
nano	640	37.3	80.4	0.99	3.2	8.7
small	640	44.9	128.4	1.20	11.2	28.6
medium	640	50.2	234.7	1.83	25.9	78.9
large	640	52.9	375.2	2.39	43.7	165.2
extreme	640	53.9	479.1	3.53	68.2	257.8

- Input Size. The resolution of the input images. Smaller pictures can make things quicker but might mean the model isn't as accurate.
- Epochs. The number of times the entire training dataset is processed during training. Modifying the number of epochs can significantly influence the learning and convergence of the model. In the context of YOLO, a common practice is to terminate the training process after a predefined number of epochs without observing any improvement in learning. This strategy, known as early stopping, serves to mitigate the risk of overfitting.

Our methodology employs a set of evaluation metrics to capture various aspects of its functionality and effectiveness. These metrics serve as key benchmarks to evaluate the system performance in different models and sequences.

- Time efficiency. Measure the speed at which each YOLOv8 model processes and analyzes image sequences.
- Accuracy. Evaluate the system's ability to accurately identify and locate bib numbers on runners in different scenarios.
- Real-world applicability. Evaluate the adaptability of the YOLOv8 system to various real-world racing environments.

D. Test Cases and Scenarios

The datasets used in this research consist of four different sources. Two of these datasets are intended for training the neural network for person and bib number detection. The third and fourth datasets are intended to evaluate the performance of the model under different conditions and image qualities.

The first dataset employed for bib detection (BDBD) contains photos of runners participating in various races. Each photo captures a runner wearing a race number on their clothing, providing data to train and test the bib detection model. Table II describes the details of the BDBD dataset, indicating 440 images for training, 30 for testing, and 130 for validation. To optimize the performance of our model, we have chosen to combine the training and testing sets for joint training, reserving the validation set for post-training evaluation.

TABLE II
SPECIFICATIONS OF THE BDBD AND SVHN DATASETS

ID	Set	Resolution	* imgs
[76]	Train	Variable	440
	Test	Variable	30
	Valid	Variable	130
[77]	Train	Variable	73,257
	Test	Variable	26,032
	Extra	Variable	531,131
Total [76]			600
Total [77]			99,289

Continuing with the next dataset responsible for training the neural network to detect digits within each bib number, we highlight SVHN. This is a dataset designed to develop machine learning algorithms similar to MNIST [85] but incorporating an order of magnitude more labeled data. Upon closer examination of its specifications, which are described in Table II, it becomes evident that the digits designated for training and testing exhibit a noticeably higher level of complexity compared to their supplementary counterparts. This observation justifies our decision to exclusively use the training and testing digits for two main reasons: their higher complexity in terms of discernibility and the consequent computational overhead associated with the integration of additional digits, due to their substantially larger volume.

In addition, the SVHN dataset offers flexibility in download formats, with two viable options: first, the entire image corpus in PNG format and, second, a format akin to MNIST where all digits are uniformly resized to a fixed resolution of 32-by-32 pixels. We opted for the first format, the entire image corpus in PNG format, for several reasons, but mostly because by retaining the original image resolution, we allow for more nuanced feature extraction and preserves finer details that could be crucial for our tasks.

TABLE III
SPECIFICATIONS OF THE TGCRBNW AND RBNR DATASETS

ID	Set	Resolution	* imgs	* RBNs	*imgs subset
[78]	1	1,920x1,080	1,033	1,523	204
	2	1,920x1,080	637	707	97
	3	1,920x1,080	407	479	151
	4	1,920x1,080	209	242	153
	5	1,920x1,080	244	286	58
[79]	1	480x720	92	100	92
	2	850x1,260	67	77	67
	3	768x1,024	58	113	58
Total [78]			2,530	3,237	719
Total [79]			217	290	217

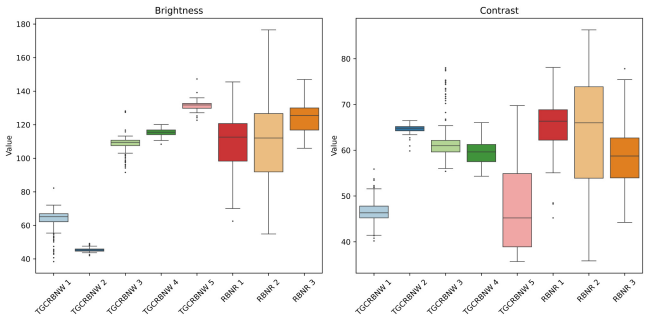

Fig. 6. Image datasets analysis.

Finally, we proceed to explain the datasets used to test and validate the trained models, named TGCRBNW and RBNR. This first dataset, TGCRBNW, comprises over 3,000 samples from more than 400 different individuals and provides a diverse set of samples, reflecting a wide range of conditions and scenarios (Table III). Upon further investigation, the provided dataset is divided into 5 folders simulating different race scenarios.

Set 1 collects images of nighttime runs in which the camera is strategically positioned at the end of a slope. This location ensures that the runners are captured in a frontal orientation. On the other hand, set 2 presents nighttime races, but with the camera situated along a curve, which complicates the task of runner detection due to their oblique position with respect to the camera's field of view.

Set 3 depicts daytime races that start in shadow environments and gradually transition to sunlight. Camera placement in this scenario is skillfully chosen to provide clear frontal views of the runners' bibs, facilitating identification. Set 4 collects races under direct sunlight, with runners facing directly at the camera, optimizing visibility and detection accuracy.

Finally, set 5 depicts races during the twilight hours, starting with ample illumination but culminating in dimmer conditions. The camera angle in this scenario is noticeably skewed, capturing the runners in an almost profile orientation as they approach, posing a challenge for detection algorithms.

Moreover, in addition to the other test dataset explained, we have the RBNR dataset, as detailed in Table III. This dataset consists of 217 color images, each annotated with ground truth Race Bib Numbers (RBNs) per image. The dataset is divided into three sets, each derived from a different race. The first and second sets exhibit similar compositions of runners within the images, although the latter demonstrates greater variability in terms of brightness and contrast. Lastly, the third set encompasses images with a substantial number of runners, potentially posing challenges for our neural network's detection capabilities.

E. Picture Analysis

The evolution of AI detection and recognition tasks has been remarkable, but significant challenges persist when exposed to real-world conditions. Factors such as brightness and contrast of an image, or the positioning of the camera, are critical and have a real impact on the performance of such systems. Several image enhancement techniques modifying these factors have been developed to increase both the quality of images and the efficiency of image processing-based applications [86], [87], [88], [89], [90].

In low light environments, such as at night or in dark scenarios, captured images often have characteristics of low brightness, low contrast and limited visibility to the human eyes. On the other hand, a high level of contrast is usually associated with good visual quality [91], [92], [93].

Figure 6 shows the brightness and contrast values for the images of the different sets used in the validation of our work and described in Table III. In a grayscale image, we represent the brightness as the mean luminance value and contrast as the variance of the luminance values. The brightness information can be used to characterize the type of scene. In that figure, both sets from TGCRBNW, 1 and 2, include dark images in a night scenario, while the rest contain daytime images. In these two sets, the brightness value is below 70 on a scale from 0 to 255, where 0 indicates pure black and 255 indicates a pure white. In addition, sets 1 and 5, also from TGCRBNW, have the lowest contrast set of images.

This variety of scenarios and conditions under different lighting conditions will allow us to evaluate the robustness of our YOLOv8-based bib detection system. Results of this evaluation are presented in Section IV.

IV. RESULTS

In this section, we will present the outcomes achieved through analysis and experiments, representing the results of our research.

A. Training Time YOLO

In our study, we begin by examining the training durations necessary for various iterations of YOLOv8 applied to both the BDBD dataset and the SVHN dataset. In this context, the measured time encompasses the entire duration of an execution, from its initiation to completion, including periods of process blocking such as during input/output (I/O) operations or when other processes are active. The data presented in Table IV provides information on the training duration of the different versions of YOLOv8 on the two datasets.

Analyzing the results, it is evident that the training times not vary significantly between different versions of the YOLOv8 model. For instance, in the BDBD model, the nano

TABLE IV
TRAINING TIME IN MINUTES

Model	BDBD	SVHN
nano	158	800
small	152	632
medium	172	1,048
large	172	1,408
extreme	175	2,640

version requires 158 minutes of training, while the extreme version requires much more time, 175 minutes. This trend is repeated across all versions, with larger models systematically requiring more training time.

When we consider the SVHN dataset, the training times increase significantly across all model sizes. For example, while the nano version of the SVHN-trained model took 800 minutes to train, the extreme version extended this duration to 2,640 minutes. This highlights the impact of dataset complexity on training time, with more complex datasets requiring proportionally more training time, regardless of model size.

B. Comparative Performance of YOLO Models During the Training

As the YOLOv8 neural network completes its training, it generates a set of metrics to evaluate its performance on the given dataset and its predictive accuracy. These metrics serve as an initial indicator of how well each model version aligns with the characteristics of the dataset and its intended task. They provide information about the model's precision, recall, and overall performance, which are essential for understanding its suitability for real-world applications.

In the context of this study, these metrics are applied to the validation set of the BDBD dataset and the SVHN dataset. Before delving into the specific results for each dataset, it is essential to provide a general explanation of the metrics used to evaluate the performance of the YOLOv8 models. These metrics include precision, recall, and mean average precision (mAP) at different Intersection over Union (IoU) thresholds [94]. Precision measures the accuracy of positive predictions, recall quantifies the model's ability to detect all relevant instances, and mAP provides an assessment of the model's object detection capabilities across various IoU thresholds.

When analyzing the performance of YOLOv8 models on the BDBD dataset, as shown in Table V, it is evident that the effectiveness of the model is similar across different size categories. Across all size categories (nano, small, medium, large, and extreme), the model consistently demonstrates high precision, with scores ranging from 91.2% to 93.8%. Similarly, recall rates remain robust, with values ranging from 89.5% to 93.2%. In particular, the medium-sized model presents the highest precision and recall rates among the different sizes. Moreover, the evaluation of the Mean average precision (Map) with thresholds of 0.5 and 0.95 reveals the effectiveness of the model in detecting objects of different scales within the BDBD dataset, with Map50 scores ranging from 95.1% to 96.4% and Map0.95 scores ranging from 68.7% to 72.1%.

TABLE V
METRICS OBTAINED ON THE BDBD AND SVHN TEST SET

ID	Model	Precision	Recall	Map50	Map0.95
[76]	nano	0.935	0.895	0.951	0.687
	small	0.912	0.932	0.964	0.712
	medium	0.938	0.906	0.958	0.717
	large	0.914	0.927	0.958	0.720
	extreme	0.928	0.916	0.957	0.721
[77]	nano	0.950	0.957	0.967	0.535
	small	0.951	0.964	0.968	0.541
	medium	0.952	0.964	0.968	0.541
	large	0.952	0.966	0.970	0.545
	extreme	0.954	0.965	0.970	0.547

Fig. 7. Yolo real application on non-training dataset.

Interestingly, the disparity between the Map scores at these thresholds suggests that while the model performs well in detecting objects with a higher confidence threshold (0.95), it encounters challenges in maintaining precision at this threshold, potentially due to increased false negatives or decreased recall rates.

On the other hand, the evaluation on the SVHN dataset reveals exceptional performance on all versions of the YOLOv8 model as shown in Table V. The models consistently achieve high precision and recall scores, indicating their effectiveness on digit recognition tasks. Interestingly, the variation in performance across different model sizes is minimal, suggesting that smaller models are equally effective in this context. The elevated mean average precision scores provide additional confirmation of the models' precision in recognizing digits in the SVHN dataset, underscoring the adaptability and applicability of the YOLOv8 framework across various datasets and tasks.

C. Application on Real-World Scenarios

In this subsection, we detail the practical application of our trained models in real-world scenarios. After successfully training our models to detect people, race bibs, and numbers, we proceeded to evaluate their performance on two different real-life datasets: TGCRBNW and RBNR. The workflow of our application process is illustrated in Figure 7, which shows the whole process. First, the whole image is processed, employing segmentation to isolate individual runners. Next, each detected runner is cropped and another neural network, trained to detect bibs, is employed to locate and extract the bibs. Subsequently, the identical procedure is iterated to identify and predict the numbers within numerical values present in the bibs. The cropping of the identified items helps minimize detection errors, like incorrectly recognizing advertisements or other forms that resemble numbers, thus improving the precision of our models.

TABLE VI
PREDICTION TIME IN SECONDS

Device	Model	CPU								GPU							
		TGCRBNW					RBNR			TGCRBNW					RBNR		
		1	2	3	4	5	1	2	3	1	2	3	4	5	1	2	3
Desktop	nano	101	46	39	66	18	35	38	31	38	13	14	28	6	8	9	7
	small	192	89	62	156	32	72	78	63	36	12	13	27	6	9	9	8
	medium	453	190	119	396	65	147	166	142	39	15	15	32	7	10	12	10
	large	754	311	188	671	113	267	281	235	43	16	15	35	8	12	13	11
	extreme	1,161	477	263	1,018	160	373	404	306	54	21	17	42	10	15	17	13
Laptop	nano	336	137	110	183	44	127	110	85	111	51	43	75	19	34	35	29
	small	588	260	161	490	88	229	277	201	171	69	51	123	26	57	60	48
	medium	1,376	570	344	1,298	194	471	522	427	349	139	100	301	52	109	123	99
	large	2,374	918	626	2,147	414	864	824	701	530	217	134	463	86	178	181	153
	extreme	3,674	1,434	811	3,369	528	1,440	1,348	936	826	313	189	653	116	239	263	196

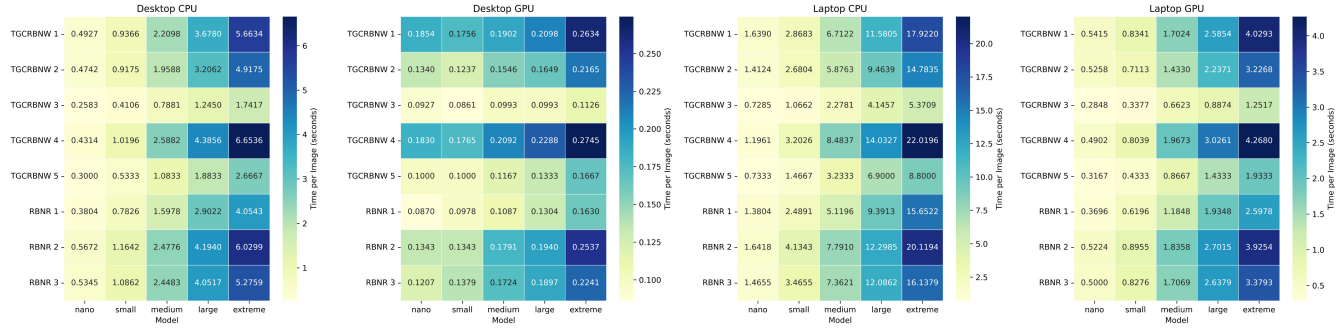

Fig. 8. Prediction time per image.

To analyze the prediction times for each scenario and dataset, we refer to Table VI. The two computers used in this analysis are equipped with both CPU and GPU resources, one being more powerful than the other. The flexibility of the YOLO tool allowed us to choose between CPU and GPU for inference, facilitating the extraction and comparison of inference speeds on models of different sizes. Across all models and datasets, using the more powerful GPU results in reduced prediction times compared to CPU-only processing.

For instance, on the TGCRBNW dataset, the extreme-sized model shows a noticeable reduction in prediction time when the more powerful GPU is used, with times decreasing from 373 to 160 seconds. Similarly, on the RBNR dataset, the extreme-sized model exhibits a significant reduction in prediction time from 42 to 17 seconds when using the more powerful GPU. When comparing the prediction times between datasets, it is evident that the TGCRBNW dataset typically requires more computational time than the RBNR dataset, likely due to the number of images and the complexity of them. These findings highlight the importance of resource optimization and indicate that leveraging more powerful hardware can substantially improve the efficiency of our models in real-world applications.

With respect to the approximate prediction time per image in each of the scenarios depending on the model used, it is should be noted that they vary greatly depending on the computer used and the device, whether it is GPU or CPU, as can be seen in Figure 8. Thus, with respect to the desktop computer, we can observe a quite significant variability of the results depending on whether we use GPU or CPU, since for example in small models it is hardly appreciable because there is a magnitude of tenths, while if we move on to larger models, the magnitude has to do with several seconds of difference. Moving on to the case of the laptop, whose computational resources are lower, we can see that the times increase significantly with respect to the other computer, and something similar occurs in the CPU-GPU relationship. However, here we can already see that for CPU, we reach values of approximately 22 seconds to perform a detection on an image.

As for the accuracy of the different versions of YOLOv8 in each scenario, we have performed evaluations using also the TGCRBNW and RBNR datasets. The accuracy results for each model version in both scenarios are summarized in Figure 9. These tables provide insight into the performance of each version of YOLOv8 in different scenarios. It is clear that the accuracy varies significantly depending on the model version and the dataset. For instance, in the TGCRBNW dataset, YOLOv8n shows the highest accuracy in most scenarios, while YOLOv8x consistently exhibits lower accuracy. However, in the RBNR dataset, the performance differs, with YOLOv8m obtaining the highest accuracy in several scenarios.

V. ANALYSIS AND DISCUSSION

In examining the results obtained, an observation concerns the relationship between training times and performance metrics. It is evident that as the model size increases, so does the training time, with the exception of the YOLOv8s model, which converges faster than YOLOv8n. This discrepancy can be attributed to the early stopping mechanism, wherein the YOLOv8s model stops training earlier as it achieves convergence sooner. In addition, the model trained with SVHN

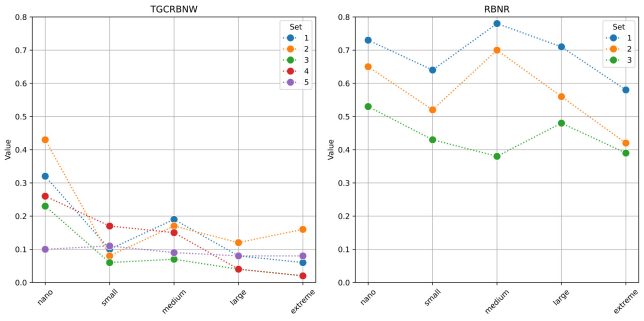

Fig. 9. Accuracy results.

data requires much more time to train than the BDBD model due to its larger dataset size, although both models return moderately similar and generally favorable metrics.

Moreover, performing predictions solely on a CPU is impractical due to long processing times. On the other hand, processing times on GPU vary depending on hardware specifications. Furthermore, the choice of model version significantly influences the processing time, as larger models require more time due to the larger number of layers through which the input must traverse. However, it is essential to note that larger models do not always produce better results, as they may require larger and more diverse datasets to efficiently learn features.

A more detailed exploration of accuracy metrics for different test datasets uncovers interesting trends. For example, the difference in performance between the TGCRBNW and RBNR datasets can be attributed to multiple factors, like camera position (angles, distance to the runners...), general image quality (brightness and contrast level) or even image resolution differences (lower quality images on TGCRBNW). In addition, challenges arise in the detection of subjects that are distant or oriented to the side. Moreover, the presence of shadows in the dorsal area significantly complicates the detection. In particular, if a bib number is imperceptible to the human eye, machine detection becomes equally challenging. In general, the time taken for detection exceeds that for non-detection, and larger models exacerbate this discrepancy.

Furthermore, it is noteworthy that the accuracy achieved on a laptop GPU is comparable to that of a desktop CPU, which mainly affects processing time. Consequently, the availability of a GPU is preferred due to its ability to significantly reduce processing time while maintaining identical metrics.

VI. Conclusion

The work presented in this paper performs a comparative study of YOLOv8-based bib number detection in several races media datasets, comparing not only the performance due to the different dataset features, but the training performance in terms of time and accuracy depending on the YOLOv8 model and the hardware used for training and prediction.

Understanding the suitability of different YOLOv8 models under varying circumstances is crucial for real-time applications. Factors such as image resolution (e.g., High Definition or Ultra High Definition) or video frame rates (e.g., 30 fps or 60 fps), deployment environments (edge vs. cloud), and computational resources availability influence the choice of model.

A set of datasets were carefully selected and analysed, studying their characteristics in terms of size, quality and real circumstances variability, looking to have the highest generality of image conditions for training.

The results demonstrated the significant impact of hardware selection on prediction times and accuracy in object detection tasks. For instance, on the TGCRBNW dataset, the extreme-sized model shows a significant reduction in prediction time from 1,161 seconds (5.66 seconds per image) to 54 seconds (0.26 seconds per image) when using a more powerful GPU on a desktop computer. Similarly, the RBNR dataset exhibits a reduction from 373 seconds (4.05 seconds per image) to 15 seconds (0.16 seconds per image) for the same model. For the laptop case, the difference in prediction time between TGCRBNW for GPU and CPU is most noticeable in the extreme model, decreasing from 3,674 (17.92 seconds per image) to 826 seconds (4.02 seconds per image). For RBNR, the difference is also significantly reduced in the extreme model, from 1,440 (15.65 seconds per image) to 239 (2.59 seconds per image).

After studying the prediction phase into the two type of hardware architectures, it becomes clear that the time needed for prediction is much higher (on average in a factor by 3) in the case of the laptop. One of the main reasons for this difference lies in the hardware used to perform the predictions. While CPUs are generally more versatile and efficient in handling a wide variety of tasks, GPUs tend to excel in parallelizable tasks. However, this advantage does not come without its own implications. Unlike CPUs, GPUs tend to be more expensive and consume more power, which can be a limiting factor in resource-constrained environments such as edge devices or virtualization systems.

This notable difference in prediction time between GPU and CPU can have important implications on the feasibility of real-time implementations. For example, while smaller models, such as nano- or medium-sized models, exhibited higher accuracy than large and extreme versions, this factor should not be considered solely from an accuracy perspective. It is also crucial to consider the prediction time associated with each model. Considering that prediction in these models takes 1 second or less, it is possible to consider that these models are suitable for real-time (GPU-enabled) or near real-time (CPU-enabled) object detection in a multimedia streaming use case. However, if the prediction time is increased by even just one second more, it could compromise the system's responsiveness, which is critical for the quality of user experience and the ability to process a high number of images efficiently.

However, accuracy varied across scenarios and datasets. On the TGCRBNW dataset, results were generally below 50% across all sets and models, whereas the YOLOv8m model achieved nearly 80% accuracy on the RBNR dataset in the best scenario. It is important to note that these values remained unchanged when extracting metrics regardless of the hardware configuration selected (GPU versus CPU or between laptop and desktop).

Our study also revealed several key insights that could have been incorporated into our methodology, such as image augmentation techniques [95], dataset division based on lighting conditions, and the integration of explainable AI to

improve model robustness and interpretability [96]. Ensuring image quality throughout the audiovisual transmission chain is essential to guarantee the correct operation of our system. A future analysis on how brightness, contrast and sharpness affect the accuracy of the YOLOv8-based bib detection system and apply image enhancement techniques in future work to improve detection accuracy.

The application of the trained neural networks can be extended beyond their initial tasks. For instance, the neural network trained specifically for bib detection can be used in other similar events, such as marathons or cycling races, to identify participants by their bib numbers. Additionally, this neural network could be adapted for detecting other types of identifications in various contexts, such as vehicle identifications in toll systems or product identifications in production lines.

Similarly, the neural network trained for number detection can be valuable in diverse scenarios, such as OCR in printed or digital documents, vehicle license plate recognition, or barcode reading. When these two neural networks are combined (bib and number detection), the range of applications expands even further. For example, in sporting events, the network trained for bib detection could work alongside the network trained for number detection to identify participants and automatically record their times. In commercial environments, the combination of both networks could facilitate automated inventory tracking by reading barcodes and product identification numbers.

Future research can converge into developing guidelines or frameworks for selecting the most appropriate YOLOv8 model based on specific application requirements and deployment constraints. It could enable to dynamically adapt and optimize object detection algorithms based on contextual factors such as network conditions, user preferences, and environmental constraints.

These new emerging trends and opportunities may enable researchers to contribute to the advancement of the object detection field and its integration in different areas, such as smart cities, autonomous vehicles, healthcare systems and video surveillance. This interdisciplinary nature will offer the possibility to exploit the potential of YOLOv8 and similar algorithms, driving innovation and addressing challenges in the audiovisual sector.

REFERENCES

[1] S. Alraih et al., "Revolution or evolution? Technical requirements and considerations towards 6G mobile communications," *Sensors*, vol. 22, no. 3, p. 762, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/762

[2] P. S. R. Henrique and R. Prasad, "The road for 6G multimedia applications," in *Proc. 23rd Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, 2020, pp. 1–6.

[3] S. Robitzsch et al., "Prospects on the adoption of a microservice-based architecture in 5G systems and beyond," *Comput. Netw.*, vol. 237, Dec. 2023, Art. no. 110058. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128623005030

[4] G. Darzanos, C. Kalogiros, G. D. Stamoulis, H. K. Hallingby, and Z. Frias, "Business models for 5G experimentation as a service: 5G testbeds and beyond," in *Proc. 25th Conf. Innov. Clouds, Internet Netw. (ICIN)*, 2022, pp. 169–174.

[5] D. Gomez-Barquero et al., "IEEE transactions on broadcasting special issue on: Convergence of broadcast and broadband in the 5G era," *IEEE Trans. Broadcast.*, vol. 66, no. 2, pp. 383–389, Jun. 2020.

[6] D. Gomez-Barquero et al., "IEEE transactions on broadcasting special issue on: 5G for broadband multimedia systems and broadcasting," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 351–355, Jun. 2019.

[7] D. Gomez-Barquero, J. J. Gimenez, G.-M. Muntean, Y. Xu, and Y. Wu, "IEEE transactions on broadcasting special issue on: 5G media production, contribution, and distribution," *IEEE Trans. Broadcast.*, vol. 68, no. 2, pp. 415–421, Jun. 2022.

[8] D. Mi et al., "Demonstrating immersive media delivery on 5G broadcast and multicast testing networks," *IEEE Trans. Broadcast.*, vol. 66, no. 2, pp. 555–570, Jun. 2020.

[9] T. B. Iliev, E. P. Ivanova, I. S. Stoyanov, G. Y. Mihaylov, and I. H. Beloev, "Artificial intelligence in wireless communications—Evolution towards 6G mobile networks," in *Proc. 44th Int. Conv. Inf., Commun. Electron. Technol. (MIPRO)*, 2021, pp. 432–437.

[10] A. Khedkar, S. Musale, G. Padalkar, R. Suryawanshi, and S. Sahare, "An overview of 5G and 6G networks from the perspective of AI applications," *J. Inst. Eng. (India) Series B*, vol. 104, no. 6, pp. 1329–1341, Dec. 2023. [Online]. Available: https://doi.org/10.1007/s40031-023-00928-6

[11] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[12] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[13] A. Doumanoglou et al., "Quality of experience for 3-D immersive media streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 379–391, Jun. 2018.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

[15] S. Rizou et al., "A service platform architecture enabling programmable edge-to-cloud virtualization for the 5G media industry," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2018, pp. 1–6.

[16] J. Serrano et al., "Design, implementation, and validation of a multi-site gaming streaming service over a 5G-enabled platform," *IEEE Trans. Broadcast.*, vol. 68, no. 2, pp. 464–474, Jun. 2022.

[17] F. Sultana, A. Sufian, and P. Dutta, "A review of object detection models based on convolutional neural network," *Intelligent Computing: Image Processing Based Applications* (Advances in Intelligent Systems and Computing), J. Mandal and S. Banerjee, Eds., Singapore: Springer, 2020, pp. 1–16. [Online]. Available: https://doi.org/10.1007/978-981-15-4288-6_1

[18] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[19] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3388–3415, Oct. 2021.

[20] A. K. Shetty, I. Saha, R. M. Sanghvi, S. A. Save, and Y. J. Patel, "A review: Object detection models," in *Proc. 6th Int. Conf. Converg. Technol. (I2CT)*, 2021, pp. 1–8.

[21] X. Liu, N. Iftikhar, and X. Xie, "Survey of real-time processing systems for big data," in *Proc. 18th Int. Database Eng. Appl. Symp.*, 2014, pp. 356–361.

[22] Y. Zeng, P. Zhang, J. Zhang, Z. Lin, and H. Lu, "Towards high-resolution salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7234–7243.

[23] L. Jiao et al., "New generation deep learning for video object detection: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3195–3215, Aug. 2022.

[24] J. Montalban et al., "Broadcast core-network: Converging broadcasting with the connected world," *IEEE Trans. Broadcast.*, vol. 67, no. 3, pp. 558–569, Sep. 2021.

[25] C. Colman-Meixner et al., "Deploying a novel 5G-enabled architecture on city infrastructure for ultra-high definition and immersive media production and broadcasting," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 392–403, Jun. 2019.

[26] T. Stockhammer et al., "Media over 5G in action—Target 2023 for 5G-MAG reference tools," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2023, pp. 1–6.

[27] L. Shrama, A. Javali, and S. K. Routray, "An overview of high speed streaming in 5G," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, 2020, pp. 557–562.

[28] J. Zong, Y. Liu, H. Liu, Q. Wang, and P. Chen, "6G cell-free network architecture," in *Proc. IEEE 2nd Int. Conf. Electron. Technol., Commun. Inf. (ICETCI)*, 2022, pp. 421–425.

[29] B. Altman et al., "Bonding functionality for live video streaming over 5G networks," in *Proc. IEEE Int. Mediterr. Conf. Commun. Netw. (MeditCom)*, 2022, pp. 274–279.

[30] T. Boros, P. Zuraniewski, R. Hindriks, N. V. Adrichem, E. Thomas, and L. D'Acunto, "Enabling superior and controllable video streaming QoE with 5G network orchestration," in *Proc. 22nd Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, 2019, pp. 124–129.

[31] C. Ge et al., "QoE-assured live streaming via satellite backhaul in 5G networks," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 381–391, Jun. 2019.

[32] J. Nightingale, P. Salva-Garcia, J. M. A. Calero, and Q. Wang, "5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 621–634, Jun. 2018.

[33] M. Torres Vega, C. Perra, F. De Turck, and A. Liotta, "A review of predictive quality of experience management in video streaming services," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 432–445, Jun. 2018.

[34] A. Rico-Alvariño, I. Bouazizi, M. Griot, P. Kadiri, L. Liu, and T. Stockhammer, "3GPP Rel-17 extensions for 5G media delivery," *IEEE Trans. Broadcast.*, vol. 68, no. 2, pp. 422–438, Jun. 2022.

[35] A. Ibanez, E. Garro, D. Gomez-Barquero, H. Jung, S.-I. Park, and N. Hur, "5G multicast broadcast services performance evaluation," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2021, pp. 1–6.

[36] D. He et al., "Overview of physical layer enhancement for 5G broadcast in release 16," *IEEE Trans. Broadcast.*, vol. 66, no. 2, pp. 471–480, Jun. 2020.

[37] T. Tran et al., "Enabling multicast and broadcast in the 5G core for converged fixed and mobile networks," *IEEE Trans. Broadcast.*, vol. 66, no. 2, pp. 428–439, Jun. 2020.

[38] Q. Zeng, S. Li, and J. Song, "Field test for 5G NR multicast and broadcast services," in *Proc. Int. Conf. Elect. Eng. Photon. (EExPolytech)*, 2021, pp. 129–133.

[39] G. Caruso et al., "Embedding 5G solutions enabling new business scenarios in media and entertainment industry," in *Proc. IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 460–464.

[40] D. Breitgand et al., "Towards serverless NFV for 5G media applications," in *Proc. 11th ACM Int. Syst. Storage Conf.*, 2018, p. 118. [Online]. Available: https://doi.org/10.1145/3211890.3211916

[41] F. Alvarez et al., "An edge-to-cloud virtualized multimedia service platform for 5G networks," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 369–380, Jun. 2019.

[42] S. Rizou et al., "Programmable edge-to-cloud virtualization for 5G media industry: The 5G-MEDIA approach," in *Proc. Artif. Intell. Appl. Innov. IFIP WG 12*, 2020, pp. 95–104, doi: 10.1007/978-3-030-49190-1_9.

[43] M. Gupta et al., "The 5G EVE end-to-end 5G facility for extensive trials," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2019, pp. 1–5.

[44] M. Femminella, M. Pergolesi, and G. Reali, "Simplification of the design, deployment, and testing of 5G vertical services," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, 2020, pp. 1–7.

[45] W. Nakimuli, J. Garcia-Reinoso, B. Nogales, I. Vidal, D. Gomes, and D. Lopez, "Reducing service creation time leveraging on network function virtualization," *IEEE Access*, vol. 8, pp. 155679–155696, 2020.

[46] W. Nakimuli et al., "Automatic deployment, execution and analysis of 5G experiments using the 5G EVE platform," in *Proc. IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 372–377.

[47] R. Perez, J. Garcia-Reinoso, A. Zabala, P. Serrano, and A. Banchs, "A monitoring framework for multi-site 5G platforms," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2020, pp. 52–56.

[48] J. Garcia-Reinoso et al., "The 5G EVE multi-site experimental architecture and experimentation workflow," in *Proc. IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 335–340.

[49] B. Y. Kasula, "Advancements and applications of artificial intelligence: A comprehensive review," *Int. J. Stat. Comput. Simulat.*, vol. 8, no. 1, pp. 1–7, 2016.

[50] F. Bran, D.-A. Bodislav, and M. A. Mitriţă, "The age of automatization and the evolution of globalization," in *Proc. SHS Web Conf.*, vol. 74, 2020, p. 02002.

[51] L. Sachan, P. Katiyar, Y. Kumbhawat, G. K. Rajput, and T. Mehrotra, "Comparative analysis on violence detection using Yolo and ResNet," in *Proc. 12th Int. Conf. Syst. Model. Advance. Res. Trends (SMART)*, 2023, pp. 89–92.

[52] M. J. A. Daasan and M. H. I. B. Ishak, "Enhancing face recognition accuracy through integration of YOLO v8 and deep learning: A custom recognition model approach," in *Proc. Asia Simul. Conf.*, 2023, pp. 242–253.

[53] D. Feng et al., "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[54] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.

[55] M. Lalinia and A. Sahafi, "Colorectal polyp detection in colonoscopy images using YOLO-V8 network," *Signal, Image Video Process.*, vol. 18, no. 3, pp. 2047–2058, 2024.

[56] X. Ding, D. Wen, L. Peng, and C. Liu, "Document digitization technology and its application for digital library in China," in *Proc. 1st Int. Workshop Document Image Anal. Libraries*, 2004, pp. 46–53.

[57] C. Ma et al., "Improving end-to-end text image translation from the auxiliary text translation task," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, 2022, pp. 1664–1670.

[58] T. Khete and A. Bakshi, "Autonomous assistance system for visually impaired using Tesseract OCR & gTTS," in *Proc. J. Phys. Conf. Ser.*, 2022, Art. no. 12065.

[59] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.

[60] S. Čakić, T. Popović, S. Šandi, S. Krčo, and A. Gazivoda, "The use of Tesseract OCR number recognition for food tracking and tracing," in *Proc. 24th Int. Conf. Inf. Technol. (IT)*, 2020, pp. 1–4.

[61] B. K. Pattanayak, A. K. Biswal, S. R. Laha, S. Pattnaik, B. B. Dash, and S. S. Patra, "A novel technique for handwritten text recognition using easy OCR," in *Proc. Int. Conf. Self Sustain. Artif. Intell. Syst. (ICSSAS)*, 2023, pp. 1115–1119.

[62] A. Shanthakumari, R. Kalpana, J. Jayashankari, B. Umamaheswari, and M. Sirija, "Mask RCNN and Tesseract OCR for vehicle plate character recognition," in *Proc. AIP Conf.*, 2022, Art. no. 20135.

[63] S. Dhyani and V. Kumar, "Real-time license plate detection and recognition system using YOLOv7x and EasyOCR," in *Proc. Global Conf. Inf. Technol. Commun. (GCITC)*, 2023, pp. 1–5.

[64] M. Brisinello, R. Grbić, M. Pul, and T. Andelić, "Improving optical character recognition performance for low quality images," in *Proc. Int. Symp. ELMAR*, 2017, pp. 167–171.

[65] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 2, 2007, pp. 629–633.

[66] M. Salehudin et al., "Analysis of optical character recognition using EasyOCR under image degradation," in *Proc. J. Phys. Conf. Ser.*, Nov. 2023, Art. no. 12001. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2641/1/012001

[67] S. A. Magalhães et al., "Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse," *Sensors*, vol. 21, no. 10, p. 3569, 2021.

[68] M. Hussain, "YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, 2023. [Online]. Available: https://www.mdpi.com/2075-1702/11/7/677

[69] M. I. Sari, P. D. Ibnugraha, M. I. Sani, M. F. Rizal, F. H. Hanifa, and A. P. Kurniawan, "Initial estimation passenger number using YOLO-NAS," in *Proc. Int. Conf. Artif. Intell., Blockchain, Cloud Comput., Data Analyt. (ICoABCD)*, 2023, pp. 181–185.

[70] X. Wang, Z. Jing, L. Shi, G. Cheng, Y. Gao, and B. Hao, "MA-YOLO: A lightweight vehicle detection framework based on YOLO," in *Proc. Int. Conf. Artif. Intell. Autom. Control (AIAC)*, 2023, pp. 277–281.

[71] A. Sufiun, M. H. I. Bijoy, N. R. Chakraborty, and M. A. A. K. Akash, "Automatic bengali number plate detection and authentication using YOLO-V4 and YOLO-V5," in *Proc. 26th Int. Conf. Comput. Inf. Technol. (ICCIT)*, 2023, pp. 1–6.

[72] A. Sarda, S. Dixit, and A. Bhan, "Object detection for autonomous driving using YOLO [you only look once] algorithm," in *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV)*, 2021, pp. 1370–1374.

[73] I. Ben-Ami, T. Basha, and S. Avidan, "Racing bib numbers recognition," in *Proc. BMVC*, 2012, pp. 1–10.

[74] W. M. de Jesus and D. L. Borges, "An improved stroke width transform to detect race bib numbers," in *Proc. 10th Mex. Conf. Pattern Recognit.*, 2018, pp. 267–276.
[75] S. Roy, P. Shivakumara, P. Mondal, R. Raghavendra, U. Pal, and T. Lu, "A new multi-modal technique for bib number/text detection in natural images," in *Proc. 16th Pacific-Rim Conf. Multimedia Adv. Multimedia Inf. Process.*, 2015, pp. 483–494.
[76] "HCMUS: Bib detection big data dataset," Jun. 2023. Accessed: 2024-02-12. [Online]. Available: https://universe.roboflow.com/hcmus-3p8wh/bib-detection-big-data
[77] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, pp. 1–9. [Online]. Available: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf
[78] P. Hernández-Carrascosa, A. Penate-Sanchez, J. Lorenzo-Navarro, D. Freire-Obregón, and M. Castrillón-Santana, "TGCRBNW: A dataset for runner bib number detection (and recognition) in the wild," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, 2021, pp. 9445–9451.
[79] I. Ami, T. Basha, and S. Avidan, "Racing bib number recognition," in *Proc. BMCV*, 2012, pp. 1–10.
[80] S. N. Moutsis, K. A. Tsintotas, I. Kansizoglou, S. An, Y. Aloimonos, and A. Gasteratos, "Fall detection paradigm for embedded devices based on YOLOv8," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, 2023, pp. 1–6.
[81] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics." Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
[82] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extract.*, vol. 5, no. 4, pp. 1680–1716, 2023.
[83] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," in *Proc. 13th Asian Conf. Comput. Vis.*, Taipei, Taiwan, 2017, pp. 198–213.
[84] "Ultralytics YOLOV8 hyperparameter tuning documentation." Ultralytics. Accessed: Jan. 12, 2024. [Online]. Available: https://docs.ultralytics.com/guides/hyperparameter-tuning/
[85] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[86] S. A. Priyanka, H.-J. Tung, and Y. K. Wang, "Contrast enhancement of night images," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, vol. 1, 2016, pp. 380–385.
[87] K. T. Kim and E.-Y. Chung, "Brightness and contrast adaptive face recognition system," in *Proc. Int. Tech. Conf. Circuits/Syst., Comput., Commun. (ITC-CSCC)*, 2023, pp. 1–4.
[88] S. Fazilov, E. Urinov, S. Kakharov, and A. Khashimov, "Improving image contrast: Challenges and solutions," in *Proc. Conf. Inf. Sci. Commun. Technol. (ICISCT)*, 2021, pp. 1–5.
[89] M. Alavi and M. Kargari, "A new contrast enhancement method for color dark and low-light images," in *Proc. 9th Iran. Joint Congr. Fuzzy Intell. Syst. (CFIS)*, 2022, pp. 1–7.
[90] H. Tanaka and A. Taguchi, "Brightness preserving generalized histogram equalization," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2020, pp. 1397–1400.
[91] S. Kansal, S. Purwar, and R. K. Tripathi, "Trade-off between mean brightness and contrast in histogram equalization technique for image enhancement," in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, 2017, pp. 195–198.
[92] A. Talun, P. Drozda, Y. Romanyshyn, O. Tehlivets, and S. Yelmanov, "Test images for training convolutional neural networks for image contrast assessment," in *Proc. IEEE 18th Int. Conf. Comput. Sci. Inf. Technol. (CSIT)*, 2023, pp. 1–4.
[93] T. Petrova and Z. Petrov, "Contrast enhancing by applying histogram analysis in image processing," in *Proc. 22nd Int. Symp. Infoteh-Jahorina (INFOTEH)*, 2023, pp. 1–4.
[94] "YOLO performance metrics," Accessed: Jan. 30, 2024. [Online]. Available: https://docs.ultralytics.com/guides/yolo-performance-metrics/
[95] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *J. Med. Imag. Radiat. Oncol.*, vol. 65, no. 5, pp. 545–563, 2021.
[96] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu, "Explainable AI: A brief survey on history, research areas, approaches and challenges," in *Proc. 8th CCF Int. Conf. NLPCC*, Dunhuang, China, Oct. 2019, pp. 563–574.

Rafael Martínez received the bachelor's degree in data engineering and systems and the postgraduate degree from the Polytechnic University of Madrid focused on Algorithmic Trading and Quantitative Systems, where he is currently a member of the Visual Telecommunications Applications Group as an Artificial Intelligence Intern. He received a scholarship with Cambridge Judge Business School related to sustainability.

Álvaro Llorente received the Bachelor of Engineering degree in telecommunication technologies and services and the master's degree in telecommunication engineering from the Universidad Politecnica de Madrid (UPM), Madrid, Spain, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in communication technologies and systems with the Signals, Systems and Radiocommunications (SSR) Department.

Since 2015, he has been working as a Researcher with the SSR Department, UPM, collaborating in different national and international research projects and with the Chair of RTVE in UPM. His professional interests include the broadcast of digital television, accessibility services, interactive TV, new video and audio formats and technologies, and quality of experience in audiovisual signal.

Alberto del Rio graduated in mobile and space communications engineering from the Universidad Carlos III of Madrid. He received the master's degree in signal processing and machine learning for big data from the Universidad Politécnica of Madrid (UPM). He is currently pursuing the Ph.D. degree with UPM and working within the Grupo de Aplicación de Telecomunicaciones Visuales of the UPM in dedicated projects on 5G communications networks with the help of use cases focused on AI. He worked with Deutsche Telekom, Berlin, in the specification of the standard of 5G telecommunications system, Release 16. Specifically in the development of a system framework concept focused on cloud services.

Javier Serrano graduated on sound and image engineering from the Universidad Politécnica de Madrid (UPM). He received the master's degree in signal, image, speech and telecommunications from the Institut Polytechnique de Grenoble, France, and the Ph.D. degree (cum laude) in telecom from UPM in 2023. He is currently working with the Grupo de Aplicación de Telecomunicaciones Visuales of UPM on testing and validation of new UHD and immersive content delivery models in next-generation mobile networks. He is also an Assistant Professor with the Informatics Systems Technical School, UPM, specialized on advanced networks and networks management.

David Jimenez received the Telecom Engineer degree and the Telecom Ph.D. degree (cum laude) from the Universidad Politécnica de Madrid (UPM) in 2004 and 2012, respectively. He joined GATV as a Researcher in 2004, and he is an Assistant Professor within the Department of Physical Electronics, Electrical Engineering and Applied Physics. He is the author and a coauthor of several scientific publications and reviewer for several entities both for research and development certification and scientific publications. His main research activities are focused on 5G communications, media quality and quality of experience assessment, smart energy, and electric vehicles. He is part of the Executive Board of the Chair of RTVE at UPM.