Journal of Advances in Computing and Engineering

ACADEMY Publishing Center

Volume 3, Issue 2, December 2023 - ISSN 2735 - 5977

Academy Publishing Center
Journal of Advances in Computing and Engineering (ACE) First edition 2021

Arab Academy for Science, Technology, and Maritime Transport, AASTMT Abu Kir Campus, Alexandria, EGYPT
P.O. Box: Miami 1029
Tel: (+203) 5622366/88 – EXT 1069 and (+203) 5611818
Fax: (+203) 5611818
Web Site: http://apc.aast.edu

Volume 3, Issue 2, December 2023
ISSN: 2735-5977 Print Version
ISSN: 2735-5985 Online Version

ACE
Journal of Advances in Computing and Engineering

Journal of Advances in Computing and Engineering (ACE) is a peer-reviewed, open access, interdisciplinary journal with two issues yearly. The focus of the journal is on theories, methods, and applications in computing and engineering. The journal covers all areas of computing, engineering, and technology, including interdisciplinary topics.
All submitted articles should report original, previously unpublished research results, experimental or theoretical. Articles submitted to the journal should meet these criteria and must not be under consideration for publication elsewhere. The manuscript should follow the style specified by the journal.
ACE also publishes survey and review articles in the scope. The scope of ACE includes but not limited to the following topics: Computer science theory; Algorithms; Intelligent computing; Bioinformatics; Health informatics; Deep learning; Networks; Wireless communication systems; Signal processing; Robotics; Optical design engineering; Sensors.

The journal is open-access with a liberal Creative Commons Attribution-NonCommercial 4.0 International License which preserves the copyrights of published materials to the authors and protects it from unauthorized commercial use.
The ACE journal does not collect Articles Processing Charges (APCs) or submission fees, and is free of charge for authors and readers, and operates an online submission with the peer review system allowing authors to submit articles online and track their progress via its web interface.
The journal is financially supported by the Arab Academy for Science, Technology, and Maritime Transport AASTMT in order to maintain quality open-access source of research papers on Computing and Engineering.

ACE has an outstanding editorial and advisory board of eminent scientists, researchers and engineers who contribute and enrich the journal with their vast experience in different fields of interest to the journal.

Advisory Board

TABLE OF CONTENT

A FRAMEWORK FOR DEVELOPING USER-BASED ADAPTIVE GAMIFIED SYSTEMS

Wael El Gammal[1], Nada Sherief[2] and Walid Abdelmoez[3]

[1] College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt.
[2,3] Software Engineering department, College of Computing and Information Technology, Alexandria, Egypt.

Emails: { waelgammal@gmail.com, nada.sherief@aast.edu, Walid.abdelmoez@aast.edu}

Received on, 13 June 2023 - Accepted on, 16 September 2023 - Published on, 20 October 2023

ABSTRACT

Gamification, which is the use of game elements in a non-game context, has become a trend in many industries. It is used as a means for motivating and engaging users and influencing their behaviors. Thus, during the last few years, many researchers have attempted to find a way to design and implement gamified systems that are adaptively personalized based on user types. This paper proposes a framework that aims to aid software engineers in systematically designing adaptively personalized gamification applications of any context type by adapting the appearance of gamification elements based on each user type. The framework introduces a method that provides a systematic means to modify the system at runtime (i.e., while the system is in use) based on the user's preferences and behavior by changing the existing elements based on the usage of each user. The researchers validate the proposed framework using a case study conducted with employees of a large-scale software development company. By applying all their practical comments, the methods of the proposed framework were enhanced. Moreover, to formalize the proposed framework the researchers develop an ontology that implements the mappings between game elements and user types by defining rules that govern their usage. In this paper, an ontology validation is presented including the use of Reasoner, Instances, SPARQL queries, and requirements model implementation. Also, a sample prototype is presented on one of the most used applications and show how the researchers apply this framework to it.

Index Terms: Gamification; Adaptive Gamification; Software Development; User-based Adaptation; Game elements.

I. INTRODUCTION

In 2010, the term "gamification" was coined in the software industry, and it was quickly recognized as one of the most important trends in software engineering [1]. The implementation of game design features in non-gaming environments is known as gamification. [2]

There are various advantages and benefits to incorporating gamification into software systems. It improves end-user happiness and participation. It also inspires people to finish daily and tedious tasks with zeal. Many businesses and organizations are gamifying their systems to give their staff and users a better experience.

Not all users have the same types and approaches for completing tasks/goals in different circumstances. In other words, in the software development field, the "One size fits all" approach of typical gamification no longer works because different people are motivated in different ways utilizing distinct mechanics and dynamics tailored to their requirements and personalities [3] [4].

Personalization of software systems based on user types is one of the most investigated subjects in gamification literature nowadays [5] [6] [7] [8] for a variety of disciplines (e.g. health [9], learning [10] [11], crowdsourcing [1] [9] and Enterprise Information Systems [1]). Traditional software design requirements can no longer be used to create gamified software applications since gamification must search for what motivates the software's intended users. [2] [12].

Most of the researchers applied their frameworks with the concept of "One size fits all" by using the same elements for all types of users. This way does not suit the design of a fully personalized system which is not motivating because users do not have the same thinking and same behavior, especially regarding the gamification elements [3].

The "Design principles for designing gamified software" outlined in the prior literature was one of the attempts made and described in [2]. The offered ideas, on the other hand, did not integrate software customization notions in a way that allows software engineers to create an adaptable gamification system depending on user types.

In addition, numerous studies in the literature have attempted to personalize the gamification elements, such as in [5] [6] [7] [8]. However, they only consider how to adjust game mechanics and dynamics such as the value of the points a user may earn, the suggestions the system can make to each user, and the reward process. There is no mention, however, of how the software can regulate the elements that show to different user types.

Also, the adaptive gamified system is one of the top research topics nowadays. Lots of the researches created frameworks for very specific kinds of systems which cannot be used as a general idea for the different fields and contexts.

Some researchers applied their proposed framework in real-life work environments from the user perspective and not from the software engineers perspective as it lacks comprehensive details and clear guidelines on how they can use it to design a gamified software system and how to make it adaptive on runtime neither how it can adapt the gamification elements based on the user types.

Thus, there is still a lack of systematic methods that can be used by software engineers to design dynamic adaptive software that can customize the elements that appear to the users based on the different user types and how to measure the rules of adaptivity.

Ontology makes the computer understand the language or logic as much as the human does by having definitions of basic concepts in the domain and relationships

http://dx.doi.org/10.21622/ACE.2023.03.2.024

among them. The reasons to have Ontology are to share the same understanding of concepts, to be available for new users to learn those concepts, and to define the data and structure to be used in the future as domain knowledge. The ontology is simply a model of reality [13].

The existing ontologies in gamification were implemented to a limited basic level as they missed lots of relations between the concepts and missed the rules that govern the software engineers how they use those concepts and how they apply the mappings between user types with the game elements [14].

This paper is organized as follows: Section two represents the "Literature Review" and discusses the gamification applications, elements, user types, gamification frameworks, adaptive gamification, and existing gamification ontologies. Then, section three explains one of the user-based adaptive gamified frameworks that the researchers adopted with an explanation of the main extended blocks. Section four explains the interviews with actual engineers in the industry to enhance the proposed framework and test its validity when applied in practice. Section five represents the results of the interviews and represents the enhanced framework according to the interviewees' comments. Section six represents a demonstrated Proof of Concept (POC) application resulting from the study and evaluation. Section seven explains in detail the ontology implementation and evaluation using instances, Reasoner, and the SPARQL queries which were developed to demonstrate the usefulness of the implementation and demonstrating how it can extract knowledge that can aid software engineers in making more informed design decisions. Finally, section eight concludes the research validation, and discusses its contributions and possible future directions.

II. LITERATURE REVIEW

This section provides a background on the concepts of "Gamification", "Gamification elements and User types", "Gamification Frameworks" and "Adaptive Gamification". Also, the related works of the previous "Gamification Frameworks" and previous "Gamification Ontologies" research related to the proposed framework are presented and give an idea of the efforts of previous researchers and what the gapes in their work are.

A. GAMIFICATION DEFINITIONS AND APPLICATIONS

Gamification is the process of turning non-game environments into games to increase people's participation in a variety of fields [1]. Feedback, themes, leaderboards, challenges, badges, and points are all utilized in gamification to change undesirable behaviors, enhance motivation, and reward good behavior and productivity [1] [15].

Gamification is used in many different domains and applications as shown in lots of research in education in [16] where it studies the development of Pedagogical Agents enriched with Gamification for an e-Learning system, while in [17], the authors discussed the smart feedback while using Gamification in math application in a primary school and in [18] the authors validate the gamification mechanics and player types in an e-learning environment. Gamification mechanics and player types are discussed below in this section. Others include, but are not limited to, researchers who use gamification in health-related applications [19], and in crowdsourcing [20].

B. GAMIFICATION ELEMENTS

Gamification elements have been divided into many categories. For example, some researchers divide the elements into two categories (*Mechanics and Dynamics*) [21].

The Mechanics parts are the functional components that offer the actions and controls mechanism (for example, points, leaderboards, levels, challenges, badges, onboarding, etc.). The Dynamics elements, on the other hand, indicate the reactions that occur when the user interacts with the mechanics' elements (e.g. rewards, competition, status, and achievements).

Marczewski further categorized the elements in [22] that divide the gamification elements into six types ***(Feedback, Schedule, Emotion, Element, Dynamic, and Mechanic).*** ***Schedules*** are the factors that indicate when something may occur; ***Emotion*** is a component that represents a user's feelings, such as interest and the fear of losing something. The user receives ***Feedback*** in the form of indications or messages from the system. Narrative/story and themes are examples of ***Elements*** that can assist the user in using the system [23].

Other researches classify the elements through the MDA model ***(Mechanics, Dynamics, and Aesthetics)***, where ***Aesthetics*** is the emotional result of the interaction of the users' dynamic experience, or DMC (Dynamics, Mechanics, and Components) [24].

In addition to this, Self Determination Theory of Motivation (SDT) intrinsic motivation activities are those that fulfill people by their interests, pleasure, and lack of conditions and are based on their psychological requirements. Those activities are based on the psychological and social demands listed below ***(Autonomy, Competency, and Relatedness)*** [25]. ***Autonomy*** refers to a user's ability to do tasks depending on his interests and the ability to choose and make judgments without having to follow directions (For example, profiles, avatars, and a customizable UI). ***Competency*** is defined as a user's desire to feel efficient and competent as a result of learning information and skills or receiving good feedback (Feedback, challenges, progressive information, points, levels, as well as leaderboard, for example) ***Relatedness***: when a user is socially linked and related to others (e.g. groups, social networks, and blogs).

C. CLASSIFICATION OF USER TYPES

Several research studies have recommended that gamified systems should be personalized based on the personalities of the users. Many studies have provided frameworks for investigating user and player-type models. According to [26] the Big Five Personality Traits "OCEAN", the user's personalities are divided into ***Openness:*** known for being curious and open to new ideas; ***Conscientiousness:*** known as ordered and systematic; ***Extraversion*** is characterized by outgoing behavior and a desire to interact with others; Tolerance and trustworthiness are traits of ***Agreeableness***; and Anxiety and irritability are symptoms of ***Neuroticism***.

Players are characterized as Dominant, Objectivist, Humanists, Inquisitive, and Creative by Ferro et al. [27]. ***Dominant*** users enjoy being seen in public. They may be assertive, aggressive, confident, egotistical, and self-driven; an ***Objectivist*** 's attention is on oneself before others, but they are not selfish; they may be assertive, aggressive, confident, egotistical, and self-driven. ***Humanists*** prefer to work in groups; ***Inquisitive*** users enjoy trying new things and discovering; and ***Creative*** users enjoy creating and developing things while learning through experimentation.

In [22], Marczewski proposed the "Four Keys of Fun," which are: ***People fun (friendship)*** when they engage in activities such as competition and cooperation. ***Easy Fun (Novelty):*** enjoys exploration, role play, and invention; ***Hard Fun (Challenge):*** Favors spectacular victory over accomplishing a difficult goal; ***Serious Fun (Meaning):*** Enjoys altering the player's environment.

According to [15] [28] [22], Gamification player types are ***Philanthropists, Socializers, Free Spirits, Achievers, Players, and; Disruptors***. ***Philanthropists are humanitarians and altruists who enjoy assisting others without expecting anything in return.*** Users who are ***Socializers*** interact with others, form social bonds, and prefer to be social. ***Free Spirits*** like creating and exploring, and they value their independence. ***Achievers*** prefer to conquer hurdles, difficulties, and challenges, and they relish the opportunity to learn new things and grow. ***Players*** user types will do whatever is required of them to obtain benefits from a system. ***Disruptors*** like obstructing, interfering, and sabotaging activities. They intend to cause havoc in any system, either directly or through other users, by introducing positive or negative changes.

Bartle Player Types include the MUD (Multi-User Dungeon) games which are classified into four types: ***Achievers, Explorers, Socializers, and Killers*** [29]. ***Achievers*** have a point-gathering goal as well as a level-rising goal.; ***Explorers*** are driven by a desire to learn more about the game's inner workings. They test out new acts in the wild, search for distinctive features, and try to find out how things function.; People, communication with other people, and what they have to say are all things that ***Socializers*** are interested in. Empathizing, sympathizing, joking, and listening are all things that they value. Relationships play an important role in their development, and emotions are rewarding for them; ***Killers*** get pleasure in not only inflicting misery on others but also in imposing their will on them. Each type can be divided into two types and is called "Eights Types Model" as described in [30].

Finally, the "Five Domain of Play" was discussed in [30] whic***h are*** (***Novelty, Challenge, Stimulation, Harmony, and Threat). Novelty:*** distinguishes open, imaginative experiences from repeating, conventional ones. ***Challenge:*** deals with how much effort and/or self-control the player is expected to use. ***Simulation:*** deals with the stimulation level and social engagement of play. ***Harmony:*** reflects the rules of player-to-player interactions. ***Threat:*** reflects the game's capacity to trigger negative emotions in the player.

D. GAMIFICATION FRAMEWORKS

This section presents a review of related gamification frameworks. Several attempts were made to design gamification frameworks that can be used while implementing gamification in different context systems using systematic ways. The researchers needed to review the previous works in this part and define what the gaps that are resolved are in their proposed framework.

For example, in [31], the authors presented a framework to guide the process of project management in the work environment. They also support the framework by designing an ontology for their work. However, this research did not mention how the proposed framework can be used to design a gamified software system.

In [2], Morselheuser et al. provided a method for engineering and developing gamified software using a list of design principles. However, this research did not provide any details on how to design systems that can be adaptive on runtime and how they can adapt the gamification elements based on the user types.

Martin et al. have provided a design framework to be used while designing adaptive Gamification applications [9]. But, it only works with a small number of gamification elements ***(Feedback, Points, Level Difficulty, Customized Challenges, Competition)***. Also, there are no clear guidelines that can be used by the Software engineers to design a dynamic adaptive software that can customize the elements that appear to the users based on the different user types. For Example, there is no clarification on how to adapt the system for the different Player Types and Personality Types and there is no clarification on how to personalize the contents, adapt the navigation paths, or adapt the user interface.

E. ADAPTIVE GAMIFICATION

Researchers have attempted to develop a personalized user-centered adaptive mechanism to dynamically re-engage users to achieve the goal of adaptive gamification. This is because they discovered that the efficiency of gamification varies depending on the situation and individual. This means that various users are motivated in different ways, and the same users are motivated in different ways in different settings. In addition, continual monitoring has been used to take into account the system's deteriorating engagement and loss of interest to adapt it [2]. This poses numerous issues and questions regarding user kinds, gamification mechanics, and dynamics, as well as what factors influence the design of such an adaptive system [32].

Adaptive gamification is achieved by tracking how gamification features are used and how they affect each user type, then tailoring gamification mechanics and dynamics accordingly. This adaptive technique is used to improve the efficiency of constructing and designing information systems, as well as to incorporate adaptive gamification to encourage user acceptance and assist businesses through long-term user engagements [3].

Codish and Ravid proposed a framework to take into consideration the playfulness of personalities while taking into consideration the contexts and gamification analytics. Then Ferro et al. [27] explore the relationship between the personality types and the player types [33].

Specht et al. proposed a classification scheme for adaptive methods: "What is adapted?" "Why?" "How?" "To which feature?" This schema served as the foundation for the final classification of adaptive gamification, which was divided into four categories [11]. Purpose of adaptivity: [5] presented an adaptive reward mechanism, Adaptivity criteria: [6] proposes customizable challenges, Adaptive game mechanics and dynamics: dynamically modify the awards points proposed by [7], Adaptive intervention: tailored articles suggestions proposed by [8].

In [3], adaptive gamification research has been categorized into three thematic areas:

1. ***ADAPTIVE ENVIRONMENTS:*** which use gamification to support the adaptive functionalities. The adaptive part represents the user needs and interests while the gamification part works as a support for user engagements through instant feedback and multiple navigation paths or adapts the user interface based on the satisfaction level of each user.

2. ***SUPPORTING CONTRIBUTIONS:*** which work toward adaptive gamification-like frameworks and approaches and research focusing on the relationship between user types and gamification mechanics and dynamics.

Like in [34] adaptive gamification applied this concept by extending the MDA (***Mechanics, Dynamics, and Aesthetics***) category framework with user demographic data like age and gender, however, this only helps to categorize the elements with their effect on different

http://dx.doi.org/10.21622/ACE.2023.03.2.024

classes of individuals but not by personalities. On the other hand, Bartle in [23] provides the *(Achiever, Socializer, Explorer, and Killer)* types while Marczewski and Tondello in [15] provide the *(Socialisers, Achiever, Philanthropist, Disruptor, Free Spirits, and Player)*. Ferro et al. [27] examined the relationships between personalities and player types. But still, these researches lack how to effectively select the right gamification elements that motivate each user type adaptively.

3. ***ADAPTIVE GAMIFICATION APPROACHES:*** which try to find meaning between users and their activities like customized challenges, adaptive paths functionality, and personalized feedback. This area also was divided into Partial and Ful approaches.

The partial approach uses extrinsic rewards to prevent the lack of intrinsic motivation like providing personalized suggestions and rewarding users with free choices and new categories [35].

Also, personal recommendations can be applied using a pedagogical agent [16]. Shi and Cristea [36] use SDT (Autonomy, Competency, and Relatedness) for social adaptive e-learning by using feedback and flexible choices in Autonomy, goals, and tasks with levels of difficulties in Competence and status visualization and contributions and interactions in Relatedness.

a. For the Ful approach, González et al. [37] try to adapt the user interface and the gamification elements of an Intelligent Tutoring System (ITS) based on user needs. Also, Andrade in [38] proposes to the ITS system a way to avoid negative impacts and overuse. Other research tries to investigate specific gamification elements and how they can affect the different user types [3].

F. GAMIFICATION ONTOLOGIES

Ontologies are designed to capture information about a particular topic and provide a machine-interpretable representation that can be reused and shared by a variety of applications and groups [39].

Researches try to investigate specific gamification elements and how they can affect the different user types [3]. Some of the researches try to implement ontologies for gamification, like in [1], named ***OntoGamif*** (Ontology of Gamification), which implements lots of classes and subclasses for concepts like target users, ethical issues, organizational structures, and psychological factors but without mapping it with game elements.

Some other researchers try to implement ontologies for specific areas like in [31] as they built a framework named ***GOAL*** (Gamification focused On Application Lifecycle Management) to be used in the Software Engineering area and they implement an ontology, especially in the areas of requirement gathering management, project management process and the testing phase.

Another area like the intelligent knowledge exchange was enriched with gamification methodology where the authors in [40] have built the ***ONARM+*** ontology which is used as a knowledge discovery technique that helps the user to get his optimal decision path to achieve his objectives funnily and they applied it into tourism area to help the user get the types of places and interests he/she likes when traveling using his/her social networking for common interests.

In [41], researchers implement an ontology for the learning area of Software Modeling to increase learner engagement. They implement it in two specific areas which are the learning UML and the learning of SQL.

Also, for the area of Intelligent Tutoring Systems (ITSs), an ontology has been implemented in [42] and named GaTo (Gamification Tutoring Ontology) to perform adaptive tutoring to learners using artificial intelligence techniques taking into consideration their knowledge into a specific domain.

G. REQUIREMENTS ENGINEERING MODELS

System modeling is an abstraction of software systems, and it helps to identify and list the features and requirements of the system. This modeling facilitates communication with stakeholders. In system analysis documentation, the system modeling can be done through the traceability matrix which provides the mapping between requirements, design, and test cases. The traceability matrix needs to be up-to-date all the time by updating it after each change request because it allows one to know the impact of any changes on the system and where exactly this change can affect the researchers' proposed system [43].

In other words, the requirements engineering models are an adequate representation of the real-life that maps all the required features to the system. If this model does not exist, this will have potential consequences while trying to make an extent of a feature [43]. That is why the feature model is very important for software engineers because it models all the system features and the relations between them. This helps software engineers capture the requirement of a model shape and link all its details and any related things to it. In this research, one of the links that need to be modeled is the link of the features with the gamification elements. For example, if a change request wants to make a change in one of the elements, then, the software engineers need to know all the features that are linked to this element and will be affected by this change. That is why the feature model is critical for maintenance, impact analysis, managing change requests, and requirements traceability.

IV. INITIAL PROPOSED METHOD FOR USER-BASED ADAPTIVE SOFTWARE DEVELOPMENT FOR GAMIFIED SYSTEMS

As presented above, all the research and the four groups of adaptive gamification have not provided a way to dynamically adapt the gamification elements' appearance at runtime according to user types. This gap was the main motivation for the proposed framework [44] .

This research focuses on the relation between the elements and user characteristics and proposes a framework that focuses on finding the solution to implement an adaptive gamification environment in software development based on user types and their mapped elements. The framework works to change the gamification elements themselves based on user types and not only to adapt the game mechanics and dynamics implementation by, for example, changing the feedback mechanism (warning messages) or using the points to create a suitable degree of Level Difficulty for each user types or even customize the challenges and competitions [9]. The framework works to manage the choices of elements for each user based on his/her type and the elements that fit him/her to get the most user' engagement, satisfaction, and performance while using the developed software. The below subsections give an overview of this study.

The proposed model in [44] adopts the "Design principles for engineering gamified software" proposed in [2] by adding some components and providing guidelines and steps for Software Engineers to follow while designing a gamified software to make the gamification items adaptive based on each user type by showing to each of the users only his preferred gamification items. Also, the research in [44] proposed a new extension on the implemented ontology by [1] named "OntoGamif" following the seven steps process provided in [14]. The added components to the framework are explained in the subsections below:

http://dx.doi.org/10.21622/ACE.2023.03.2.024

A. THE ONTOLOGY COMPONENT

The "User-Centered Gamification Ontology" in [44] components defines the concepts of the "Gamification Elements", "Elements Categories", "User Types" and the mapping between the "User Types" and "Gamification Elements". This was implemented by declaring the ontology classes and relations to define a formal definition of the rules to be utilized by the Software Engineers while having systematic automated reasoning for those concepts.

1. CLASS HIERARCHY

In [44] the Ontology Class Hierarchy has been divided into three main classes (Elements, Elements_Categories, and User_Types). First, the ***Elements Class*** includes a list of 36 gamification elements [23]. Second, the ***Elements_Categories Class*** includes sub-classes representing the different researches elements categorizations like "Mechanic_Dynamic_Model" [21], "MDA_Model [24], "Mechanics_Elements_Model" [22], and "Self_Determination_Theory_of_Motivation_SDT" [25]. Third, the ***User_Types Class*** includes sub-classes representing the different researches user types categories like "BigFive_PersonalityTraits_OCEAN" [26], "Ferro_Players_Classification" [27], "Five_Domain_of_Play" [45], and "Four_Keys_of_Fun" [22].

- "Marczewski_User_Types" which includes two subclasses ("Hexad" and "Initial_Motivators") [28] [22] [15] .
- "Bartle_Player_Types" class which includes also two subclasses ("Four_Types_Model" and "Eights_Types_Model") [29].

2. OBJECTS PROPERTIES

In [44], the authors classify the mapping relationships into some relation types as the examples shown in Figures 1 and 2 which show ***Direct*** and ***Indirect*** relations between the User Types and Gamification Elements while the ***Indirect*** is divided into ***Partial*** and ***Total.***

The mapping linkages between the user type and the gamification elements can be ***direct***. This means that, like in the case of the "User Types Hexad" with the gamification elements, each User Type includes a list of Elements that are directly linked and defined to it [15] [46]. Alternatively, one can go the ***indirect*** route by mapping one user type classification to another, which is then linked directly to gamification features. The ***indirect*** mapping can be ***partial,*** meaning that each user can be linked to two additional user types in a different category, each of which is directly mapped to the elements. For example, the mapping between the "Big Five Personality Traits" and the "User Types Hexad" [28] or ***total,*** implying that each user type can only be associated with one type in another category, like in the case of the "User Types Hexad" and their "Initial Motivators" [28] [22] [46].

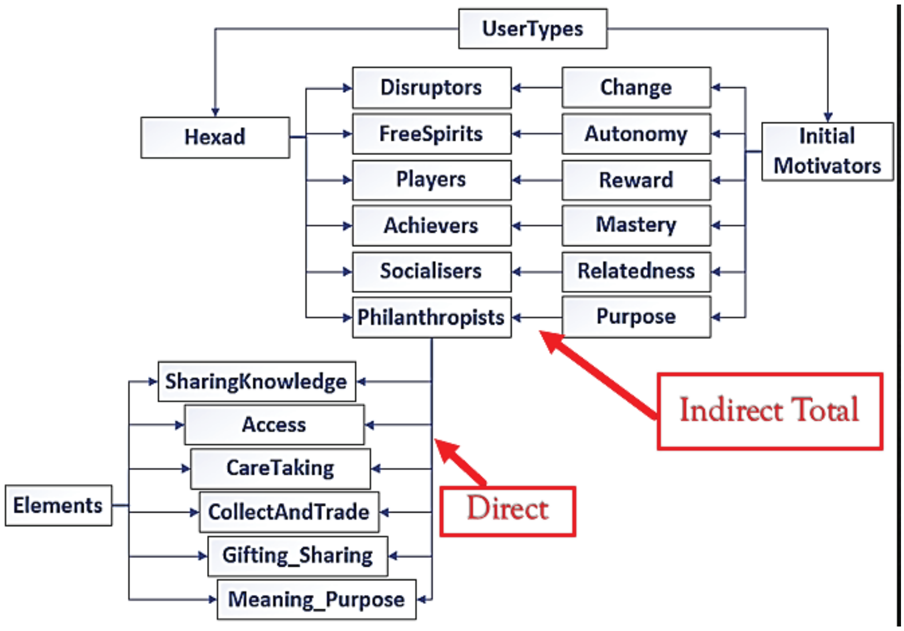

Fig. 1. Graph representation of mapping between the initial motivators and Hexad user types

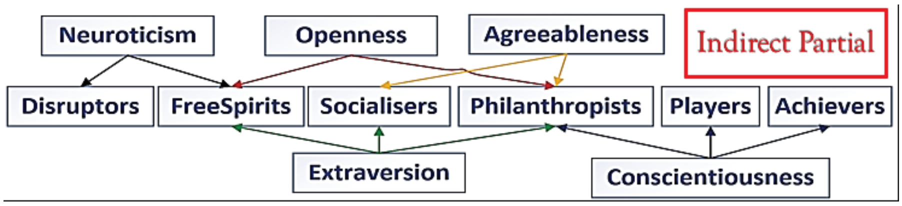

Fig. 2. Graph representation of mapping between "Big Five" and the "Hexad" User types

The above relations have been represented in the ontology as object properties like "MAP_Element_ElementCategory", as shown in Figure 3, which represent the relation between the Element and its categories.

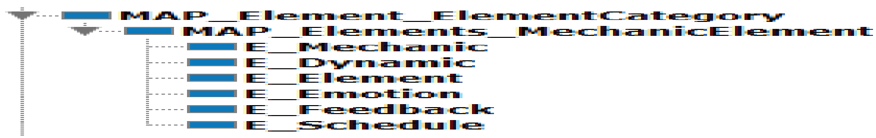

Fig. 3. "Element" to "Element Category" objects properties [44]

Then the "MAP_Element_UserCategory" shown in Figure 4 is divided into "DirectMapping" and "IndirectMapping" which in turn is divided into "TotalMapping and PartialMapping" like the examples shown in Figures 1 and 2. Also, each of the Objects' Properties Domains, and Ranges were configured as in the example in Figure 5. Table I shows a sample of the created object properties with their domains and ranges.

http://dx.doi.org/10.21622/ACE.2023.03.2.024

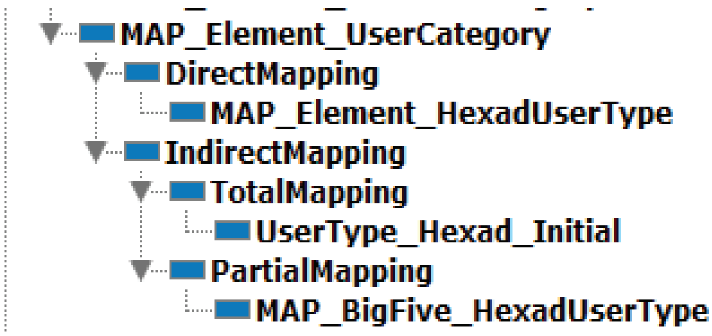

Fig. 4 “DirectMapping” and “Indirect Mapping” sub-properties [44]

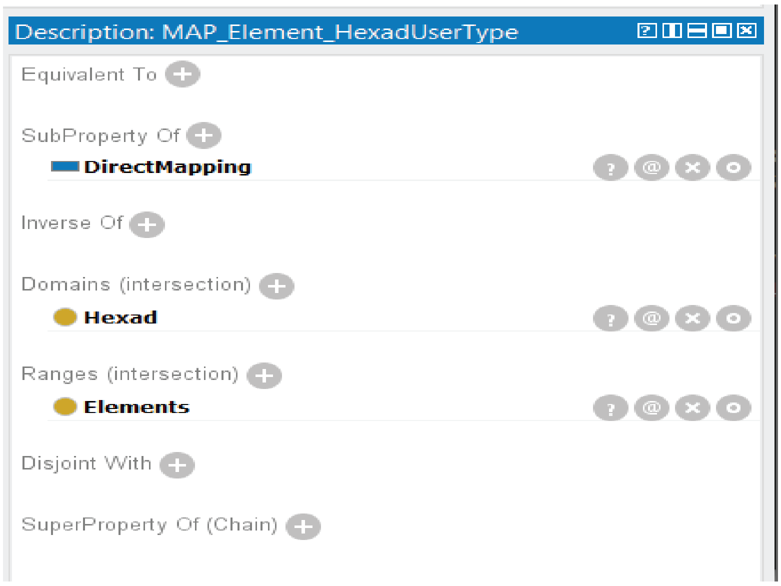

Fig. 5. Domain and range of the “MAP_Element_Hexad UserType” object property

TABLE I: EXAMPLE OF THE CREATED OBJECT PROPERTIES WITH ITS DOMAINS AND RANGES

Mapping type	Object Property Parent Group	Object Property Example	Domain	Range
Elements with Elements categories	MAP_Element_ ElementCategory	MAP_Elements_ MechanicElement	Mechanics_ Elements_ Model	Elements
Elements with Users categories Direct Mapping	MAP_Element_ UserCategory / DirectMapping	MAP_Element_ HexadUserType	Hexad	Elements
Elements with Users categories Total Indirect Mapping	MAP_Element_UserCategory / IndirectMapping / TotalMapping	UserType_ Hexad_ Initial	Hexad	Initial Motivator
Elements with Users categories Partial Indirect Mapping	MAP_Element_ UserCategory / IndirectMapping / PartialMapping	MAP_BigFive_ HexadUserType	BigFive_ PersonalityTrait_ OCEAN	Hexad

3. CLASS RULES

After describing the class and object property hierarchies, this section demonstrates how to use them to define the rules that govern class usage.

Figure 6 shows a sample of the rules which is the “Feedback” Subclass of the “Mechanics_Elements_Model” and the list of the only elements assigned to it as described in [22]. It shows the list of elements which is considered Feedback elements. Those elements are (Badges_Achievements, Certificates, Leader boards_ Ladders, Levels_Progression, Lottery_Game Of Chance, Physical Rewards_Prizes, Points_Experience Points XP, Social Status, Unlockable_Rare Content, Virtual Economy). Table II shows a list of implemented rules of the subclasses with their filters.

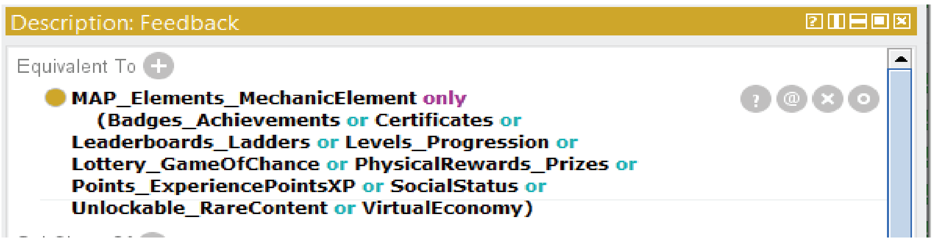

Fig. 6. Rules of the “Feedback” subclass of the “Mechanics_Elements_Model”

TABLE II: LIST OF IMPLEMENTED RULES OF THE SUBCLASSES

Subclass	Restricted property	Restriction type	Restriction filter
Achievers	MAP_Element_ Hexad User Type	Only	(BossBattles, Certificates, Challenges, Learning_NewSkills, Levels_Progression or Quests)
Disruptors	MAP_Element_ Hexad User Type	Only	(Anarchy, Anonymity, Development Tools, Innovation Platform or LightTouch or Voting_Voice)
Free spirits	MAP_Element_ Hexad User Type	Only	(Branching Choices, Creativity Tools, Customization, Easter Eggs, Exploration or Unlockable_Rare Content)
Philanthropists	MAP_Element_ Hexad User Type	Only	(Access or CareTaking or Collect And Trade or Gifting_Sharing or Meaning_Purpose or SharingKnowledge)
Players	MAP_Element_ Hexad User Type	only	(Badges_Achievements, Leaderboards_Ladders or Lottery_ Game Of Chance, PhysicalRewards_Prizes, Points_Experience 85% Points XP or Virtual Economy)
Socialisers	MAP_Element_ Hexad User Type	only	(Competition or Guilds_Teams, Social Discovery, Social Network, Social Pressure or Social Status)
Achievers	-	Equivalent To	Mastery
Disruptors	-	Equivalent To	Change
Free Spirits	-	Equivalent To	Autonomy
Philanthropists	-	Equivalent To	Purpose
Players	-	Equivalent To	Reward
Socialisers	-	Equivalent To	Relatedness
Mechanic	MAP_Elements_ Mechanic Element	only	(Access, Boss Battles, Branching Choices, Challenges, Collect And Trade, Competition or Creativity Tools, Customization or Development Tools, Easter Eggs, Exploration, Gifting_Sharing, Innovation Platform, Learning_NewSkills, Physical Rewards_ Prizes, Quests, Sharing Knowledge, Unlockable_Rare Content, Virtual Economy or Voting_Voice)

Subclass	Restricted property	Restriction type	Restriction filter
Dynamic	MAP_Elements_ Mechanic Element	only	(Anarchy, Anonymity, Boss Battles, Care Taking, Collect And Trade, Creativity Tools, Customization, Development Tools, Exploration, Gifting_Sharing, Innovation Platform, Leader boards_Ladders, Learning_New Skills, Light Touch, Sharing Knowledge, Social Network, Unlockable_Rare Content, Virtual Economy or Voting_Voice)
Element	MAP_Elements_ Mechanic Element	only	(Boss Battles, Challenges, Competition, Creativity Tools, Customization, Guilds_Teams, Innovation Platform, Learning_ New Skills, Levels_Progression, Quest, Social Discovery or Social Network)
Emotion	MAP_Elements_ Mechanic Element	only	(Care Taking, Competition, Meaning_Purpose, Social Pressure or Social Status)
Feedback	MAP_Elements_ Mechanic Element	only	(Badges_Achievements, Certificates, Leader boards_Ladders, Levels_Progression, Lottery_Game Of Chance, Physical Rewards_Prizes, Points_Experience Points XP, Social Status, Unlockable_Rare Content or Virtual Economy)
Schedule	MAP_Elements_ Mechanic Element	only	Lottery_Game Of Chance

4. THE WEIGHTING MODULE COMPONENT

A method was proposed in [44] as a guideline for Software Engineers to apply the concept of the adaptive customized Gamification Elements for each user depending on his/her user type, preferred elements, and system usage. The algorithm is mainly dependent on weighting scores and values that are given to each of the available gamification elements in the system which are changed based on the user feedback and usage. The system then customizes the shown elements to each user based on the elements' scores and on the mapping of those elements which is formalized in the ontology. This will be further elaborated before and after evaluation in section 4.

5. THE EXTENDED PHASES

This component describes the extended phases that need to be added to the design principles for engineering gamified software [2]. The first part is that the "Monitoring" component is changed to "Monitoring and Runtime Evaluation" to permit the system to monitor the users' preferences though continuously capturing their feedback on the gamification features in the system and monitoring their features usage. To achieve that, the discussed weighting system shown in Figure 9 is utilized to show the steps upon which the adaptation decisions are made. Accordingly, these decisions are realized in the following added phase named "Adaptation". A feedback arrow is also added between the "Adaptation" phase and the "Monitoring and Evaluation" phase to show the continuous loop (an ongoing task) of capturing feedback and personalized adaption at runtime.

VI. EVALUATING THE PROPOSED FRAMEWORK

This Section explains the evaluation process of the proposed framework. The below sections provide details on the recruitment and the participants and details about the company the researchers used for recruiting. Then, the next section gives some details on the introductory session and what was included in it. The Immersion scenario used is then explained in detail as well as the software that was employed and the reason to use this software in addition to the supporting documents and the Interviews structure. After that, the interviews results are discussed and categorized to finally apply the recommended modification to the applied framework.

A. RESEARCH METHODOLOGY

In this research, the ***qualitative approach*** was chosen to help fulfill the research objective, to validate the proposed framework and to get the experts' opinions to enhance it. Due to the exploratory nature of this research, after identifying the gaps in the previous frameworks which are missing the adaptivity of the different gamification elements based on the different user types classifications is explained. This research provided a suggestion to solve this problem. This suggestion needed to be validated by actual software engineers by exploring the idea providing their feedback and collecting their concerns and ideas to enhance the proposed framework based on their experience.

Direct feedback from software engineers is needed for exploring ideas and evaluating the design of this research. Interviews are a traditional way to apply this kind of research. There are many kinds and ways for interviews [14].

1- Structured interviews: need a prerequisite of a good understanding of the topic from all sides and a well developing questionnaire. This kind of interview can be handled even face-to-face or through telephone.

2- Semi-structured interviews: this is a kind of formal interview. It is handled by having some qualitative open-ended questions and points that need to be covered during the interviews. Those points can be re-ordered, or a little bit changed based on the situation of the interviews but without getting away from the initial target points that need to be covered. This way helps in expressing their views in a freedom way.

During the interview, the interviewers mainly wrote notes because of the open-ended questions. However, it was difficult in some cases to write notes and to discuss points with interviewees at the same time. To solve this problem tape-recordings can be used to be able to focus on conducting the interviews.

3- Unstructured interviews: are only limited with a plan in mind regarding the goal and focus of the interviews and let the discussion be open-up and let both parties of the interview talk in their ways.

4- Informal interviews: In this type of interview, the interviewer has a casual informal conversation with the participant without any structure guide and he/she can take small notes.

In this research, semi-structured interviews were chosen due to their flexibility, discovery nature, and ability to go deep freely with the interviewees' concerns and detailed responses, which results in ambiguities and incomplete answers being cleared and filled up. Recordings of the sessions were applied in addition to taking notes.

However, there are also disadvantages to interviews, which are: 1) time-consuming: planning, setting up, recruiting, interviewing, transcribing, analyzing, feedback, and reporting; 2) they can be costly: participants can cancel or change the meeting place at the last minute; 3) different interviewers and different interpretations [47].

The ***data analysis technique*** that was used in this research was the coding technique. In the beginning, the "Open Coding" technique was used to list the comments of the participants from the resulting documents notes, and recordings. The data collected is represented into codes by mapping each comment from each participant and representing it with one code-named "Open Code". Then all the codes with similar meanings and concepts across all the participants were grouped and merged into groups. These groups represent the "Axial Codes" which is simply a grouping by meanings and concepts of the "Open Codes". Then finally, the researchers grouped the "Axial Codes" into bigger categories named "Selective Codes" representing the main ideas of the comments of the participants [48].

B. PARTICIPANTS RECRUITED

Interviews were applied to employees in a Saudi Arabian software company working on many artificial intelligence projects with many customers in the Middle East. This working field was one of the biggest reasons to choose this company. Software engineers of such a company focus on the details of the idea, the user experience (UX) of the users who will use an adaptive gamified system, and the steps they will follow when they try to design this kind of system. The company develops applications using recent technologies in Artificial Intelligence, Machine Learning, and Business Intelligence. The demographic analysis of the interviewee is shown in Table III.

The projects the company works on are: ***Robotics software***: Android applications to manage Humanoid Robots; ***Robots management systems:*** Systems to manage many types of Robots using one interface; ***Chatbots and Voice bots***: To make digital interaction faster and more human; ***Intelligent Travel Assistants:*** To help users get the best travel offers without human interference; ***Intelligent Insurance Assistants:*** To help users to compare the best insurance offers; ***Robotic Process Automation (RPA):*** To handle repetitive tasks that do not need human thinking effort. Table III gives an overview of the demographic of the participants (Gender, job position and years of experience)

TABLE III: THE DEMOGRAPHIC ANALYSIS OF THE INTERVIEWEE

Interviewee number	Gender	Job position	Highest level of education	Years of experience
1	Female	Database developer	Bachelor degree	10
2	Female	Senior software engineering	PhD	15
3	Male	UX designer	Bachelor degree	14
4	Female	Junior business analyst	Bachelor degree	2
5	Male	Junior business analyst and bot implementer	Bachelor degree	1
6	Male	UI designer	Bachelor degree	3
7	Female	AI developer	Bachelor degree	2
8	Male	UX-UI team leader	Bachelor degree	15
9	Male	System architect consultant	PhD	22
10	Male	AI team leader	Master's degree	11
11	Male	Senior system architect	Bachelor degree	18

C. INTRODUCTORY SESSION

The evaluation methodology started with an Introductory session to explain the topic through a presentation. Around three introductory sessions were handled with one hour each. The presentation introduces gamification and how gamification can help people get more engaged and enhance their motivation to do tasks. Then, Gamification Elements were shown and described in general with some examples of the elements and how they can be used. Gamification Elements categories are described after that with some examples and the difference between them. User

Types are then discussed and shown samples of their categories. After that, a basic introduction to Ontology, what it is, and what the steps to make an Ontology are.

Then after that, the purpose and need for this research were given. Besides, the research problem was clarified, and a quick overview of the literature work. Finally, a description of the proposed framework and a description of each block was given.

After the introductory session, an evaluation session was conducted with each participant through a semi-structured interview which included a set of predefined questions to get their opinion/suggestions to modify/enhance the proposed framework and the weighting module design. Those semi-structured interviews started with an immersion scenario that is discussed in the next section.

D. IMMERSION SCENARIO

To better engage the participants in the interview and get their focus to evaluate and enhance the proposed framework, a fictional scenario was used to help the participants apply the steps they are following while immersing themselves in situations similar to what they do in real life [49].

The software employed in the immersion scenario was "Samsung Health" which is a personal health application that can be used on users' mobile devices and can be personalized based on each user's needs. The user for example can choose all the exercises that he/she is interested in and he/she can also customize his/her home screen. [50]

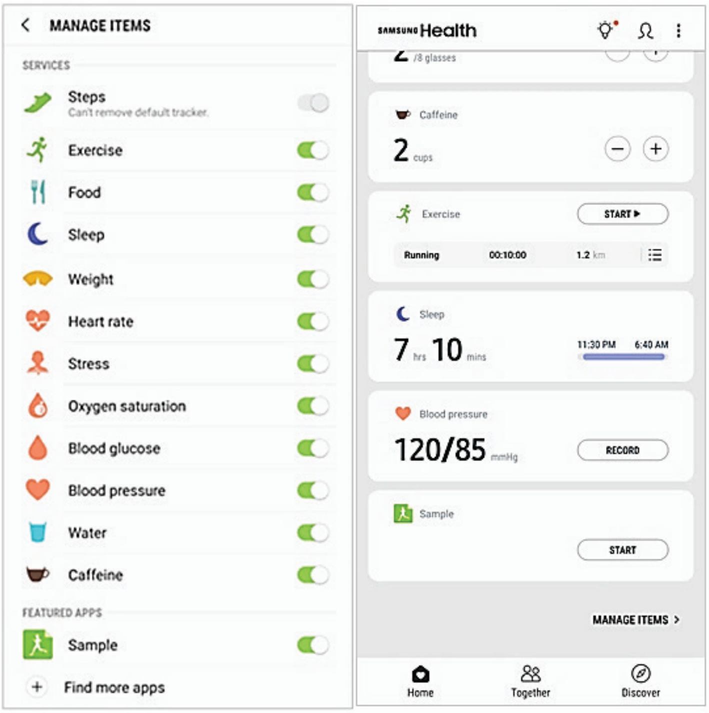

Fig. 7. Screenshot from the immersion scenario of Samsung Health [51]

A small introduction was given to "Samsung Health" software in addition to some of its capabilities and features as shown in Figure 7.

The immersion scenario software features were discussed and taken into consideration during the discussion. For example, how the list of displayed features can be customized manually for each user on his home screen (Landing Page). Some

http://dx.doi.org/10.21622/ACE.2023.03.2.024

of the discussed features are "Wellness" features where the user can track his workout activities, set up goals, and track the progress of calories, quality of sleep, and water intake. Also, the "Learning from others" feature and how to get other people's experiences by viewing videos and reading about their stories and their suggested workouts. The "Challenging yourself" feature can be used by getting some fitness motivation while challenging friends and other people and comparing your progress with their progress.

After that, the participants were asked to try to enhance this software and add gamification elements to it while trying to make the homepage features and gamification elements appear adaptively for each user based on his/her personality and based on his/her usage. They were also asked to follow the proposed steps to do so.

E. SUPPORTING MATERIALS UTILIZED IN THE INTERVIEWS

Before each session, a printout of some supporting documents was provided in addition to some points that have been declared and discussed with the participants to immerse them into the mode of designing and enhancing the "Samsung Health" application and adding some game elements and making them adaptive based on the end-user types. The points and supporting documents include:

1- The proposed framework diagram was given to the participants to be visualized and to take their comments on it.

2- Screenshots overview of the developed Ontology.

3- Immersion scenario description:

    - What is the tool that has been chosen?
    - Why choosing this tool?
    - What is the feature that is currently implemented in "Samsung Health"?
    - What is the feasibility of adding gamification elements to the chosen tool?
    - What are the types of users who will use this immersion scenario?

4- Gamification elements Periodic Table [22]

5- Gamification Elements Descriptions [22].

6- One chosen list of user type categories with their descriptions [15] [28] [22].

7- Elements mapping with selected user types category [22].

8- Mapping Table (Elements – User types category mapping – Elements category mapping)

9- Design principles for engineering gamified software [2].

10- Elements Relations

11- Weighting Algorithm Activity Diagram

12- Simulation table for simulating the weights of elements for each user to be used in the Immersion Scenario

13- Simulation table for the appearing Initial list of elements used in the Immersion Scenario tracing

The participants immersed in that fictional scenario previously prepared as if they were designing a system using the proposed framework (The supporting documents can be found at this link https://bit.ly/3m9gjfq). This helps during the interview session to walk through the proposed framework with the participants and to provide more valuable results and more proper testing of the framework.

F. INTERVIEW'S STRUCTURE

An interview was conducted with each of the recruited participants. The interviews took about 14 hours of the total time of discussions with an average of 1 hour and 15 min for each interview. All the interviews were voice-recorded after getting approval from each participant. Those records were then transcribed later to be used in addition to the notes in the interviews' analysis. The interview was divided into two parts:

- The first part is walk-through testing on the whole framework when using it to design a Health application with adaptive gamified elements based on the users' types. This part was mainly to get the interviewee's suggestions and opinions on the weighting module.

- The second part of the interview was a discussion about the overall proposed framework. After that, the Software Engineers (the interviews participants) walked-through the proposed idea components, this second part was to put all these components of the framework (Process, Ontology, Weighting Module, …) into practice while designing a gamified application based on the given fictional scenario and given requirements. This part helps to identify the proposed framework's strengths and weaknesses. Also, it helped to collect their feedback to enhance the framework while applying their suggested changes and solving the issues.

1. EVALUATION RESULTS

The comments of the interviews were analyzed and categorized into three types:

Comments on the "***Design principles for engineering gamified software"*** phases→ *(Process Evaluation)*

Comments on the linkage between the "***Design principles for engineering gamified software"*** phases and the proposed framework modules → *(Components Links Evaluation)*

Comments on the proposed framework structure, modules, and their relations → *(Modules Evaluation)*

A. PROCESS EVALUATION

In the beginning, the participants understood the immersion scenario and agreed that it would be feasible and profitable to add gamification elements to it and that it could enhance user engagement for such applications. Then, they started applying the proposed framework with the immersion scenario and comments on each part while simulating the real process.

The first group of comments of the interviewees commented on the design principles for engineering gamified software phases and how they need to be modified to cope with the proposed framework.

Some Interviewees commented on the *Project Preparation and Ideation Phases* when analyzing the immersion scenario and trying to integrate gamification elements into

http://dx.doi.org/10.21622/ACE.2023.03.2.024

it. They wanted to move the ideation phase after the preparation phase before the user and context analysis phases. Other participants see that the ideation cannot be done before analyzing the user and context and knowing exactly the target users and the target context. Project preparation is done through some searches, data collection, and scoping while studying the gamification applicability and how to implement it. On the other hand, ideation is done through brainstorming and focus group sessions directly after the preparation phase. Also, some users think about how the researchers can get the statistics to define the project objectives in case they do not have a history, or it is very poor in the case of the adaptive gamification data which may be similar to the recommendation engines which depend on the history and mining of the data of other previous users.

All interview comments are uploaded on the drive and can be found at this link https://bit.ly/3m9gjfq).

Changes to the framework (Process: Project Preparation and Ideation Phases)
• Leave the Project Preparation and Ideation Phases as is because the ideation cannot be done without knowing the target users and context.

Other Interviewees focused on the Analysis Phase, even ***Context or User analysis*** or both. Some users see that both of them can work in parallel and not sequentially which means that they are not dependent on each other, and some other opinions want to switch the context analysis before the users' analysis because the elements defined during the context analysis can be changed for the same user from domain to another. Others see that the Context/Field/Domain of the project needs to be defined then the researchers define which users in this domain will focus on as users for the system and that is why the analysis is conducted. Also, the researchers can swap the user analysis with the project preparation as user analysis happens through research and interviews then the researchers start the project preparation based on the users and market research results. Another comment was that phases should be repeated per context and integration and that they should have a phase to define gamification elements that will be used for each context level. Also, the researchers need a phase that structures the Hierarchy of the Context levels (in case of sequential or parallel levels or after a specific period). For example, Fitness/Exercise is parallel to Food/Diet in the case of health applications.

Changes to the framework (Process: *Context or User analysis*)
• Leave both phases of analysis to be parallel as there is no dependency between the two of them.

Some attendees focused on the ***System Design*** Phase. For Example, their idea was to add an Onboarding Component to be able to give the users an idea of the system controls and how the weighting module will adapt to their needs.

Changes to the framework (Process: *Context or User analysis*)
• Leave both phases of analysis to be parallel as there is no dependency between the two of them.

Some of them see that the ***Evaluation and Monitoring*** practice needs to be modified by for example applying it after each phase. Others want to add an ***Evaluation*** phase called "***Usability Testing***" after the ***Ideation or Design*** phases. Others see that after the "Evaluation" Phase the researchers need to add a ***Feedback Loop***.

Another idea was to add ***KPIs*** to the evaluation phase to be able to set the targets of the evaluation process.

Changes to the framework (Process: Evaluation and Monitoring)
• The researchers need to have two types of evaluations; one for the evaluation of the software development phases and results and the other one will be for the evaluation on runtime for the system adaptations based on the users' feedback and usage.

B. COMPONENTS LINKS EVALUATION

The second group of comments from the interviewee focuses on the relations between the design principles for engineering gamified software phases and the proposed modules/components. In addition, they commented on the relations between the proposed modules themselves.

Some of them say that instead of working with the *Proposed Framework in parallel during the phases* of (User analysis, context analysis, ideation, design, and implementation) the researchers must remove the implementation. Others see that the researchers should work with the framework only during the (Context analysis and ideation) phases.

Some of the interviewees see that they need to *Link each phase of the Design Principles to its related component in the proposed framework*. So, for example, the User Analysis phase is to be extended by the User Characteristics/Types.

Changes to the framework (Components Links Evaluation)
• Link each phase of the design principles to its related component in the proposed framework to be as an extension to it (e.g. user analysis will be linked to the "user type" component, context analysis to be linked to the "Feature Model" Component).

C. MODULES EVALUATION

The third group of comments from the interviewee focuses on the proposed framework components themselves (Gamification Mappings, System Design, Elements Relations, Weighting Module, and Elements Adaptation).

1. ONTOLOGY MODULE

Some comments were related to the *Ontology section*. Moreover, the *other comments* category were added as well in the database. Some comments mention that the researchers need to *add subtypes or tasks or features under each element* because the users may like an element for a specific feature/task and do not like it in another feature/task. Also, tasks may differ from one context to another. They want also to add this mapping between the elements and the tasks to the Ontology using Feature Models. Again, some users are against this because they do not want to complicate the model calculations.

Changes to the framework (Process: Modules Evaluation - Ontology section)
• Divide the Ontology sub-modules to include a part for the user type, a part for the relations and categories of the elements and their features, and a part for the mappings. • Add Tasks/Feature Model based on the context into the ontology module.

2. WEIGHTING MODULE

Some comments were related to the weighting module and how it works and most of them try to enhance the framework and add some other criteria to get the best results.

Some of the comments were trying to ***enhance the weighting system by not only depending on the clicks of the users but also on adding a feedback section*** to get the user's input through the Like/Dislike functionality AND/OR adding a star rating on each element level. However, other users were against this because they wanted to measure the behavior of the user and not to ask him/her. Some other comments wanted to ***add a component to limit the stress on the user*** as some users may put loads on themselves to reach the targets or to gain points.

Some interviewees think that this ***framework may face the bubble issue*** which appears a lot in the recommendations systems of the search engines. The bubble issue mainly happens when a user searches for the stuff of the same category for a long time. Then, most of the recommendation engines, in this case, recommend stuff from the same category. Other opinions provided a solution for this issue by enabling other elements from the nearest category even if they were not in the top high weights. Other opinions want to use the 80/20 algorithm used in some recommendation engines to solve the same issue by providing 80% of the same user recommendation bubble and 20% of any random other categories. Another idea was to hide some of the top high weights at some points and replace them with other elements with low weights and enable the user to try other elements and be away from the bubble issue.

Other interviewees want to ***make the weighting module more personalized*** by asking the user if he/she likes each element or not. Some other ideas include adding more intelligence to the weighting module and wanted to add some AI and machine learning algorithms like Random Forest to better enhance the results for the users, but the problem that the researchers can face will be the data history/Dataset to train the model to be more intelligent. Another opinion was to integrate social media to build clusters based on the similarities between the related people. Another idea was to add a threshold to get more confidence level before taking the decision and to be sure that the user is interested in this element.

Some of the interviewees commented on the ***weighting module activity diagram as it needs more details for software engineers*** to understand the actions in each branching condition.

Changes to the framework (Process: Modules Evaluation - Weighting_section)
• Change the feedback to include other types like (Like/Dislike, Star Rating, and Direct Questions). • Add a User Load Stress Control component in the weighting module. • Add a component to handle the threshold for confidence level. • Add a component to handle the ability to pin an element if he/she likes it and does not want to change it. • Add more explanation on how to manage the mapping of the elements and the user types with all its types of mappings (direct, indirect, Total, and Partial) to the activity diagram.

At the end of each interview, three main questions were asked to each of the participants to know their overall opinion on the idea of the proposed framework.

http://dx.doi.org/10.21622/ACE.2023.03.2.024

The first question was "*Do you think this framework shall enhance and increase the motivation of systems users?*" They had to choose one choice of three (Too Much, The right amount, Not really) and almost all of them chose "Too Much" but they insisted on having an Onboarding strategy when designing the system to help the end-users understand how the systems will be adaptive based on their feedback and that the system will interact with those feedbacks and with their usage.

The second question was "*Is it easy to integrate this framework with your software process?*", some of them replied with "yes" it is easy to integrate it and some others said that it needs first to have new design patterns to work with this new kind of adaptive software and that it will have some resistance from some of the software engineers to adapt the process they are following now.

The third question was "*How clear are the framework diagrams of the proposed framework and the sequence diagram?*", most of them said that it was extremely clear for them to understand and to read it.

3. FINALIZED FRAMEWORK MODIFICATIONS

After studying and analyzing all the comments and collecting all the biggest and most feasible comments as shown in the "Changes to the Model" blocks of the above section. A redesign of the whole framework was handled taking into consideration the participants' comments. Figure 8 shows the updated gamified adaptive framework after applying the changes.

Also, the weighting module activity diagram has been enhanced as shown in Figure 9 by adding more details and conditions to fulfill all the cases of the provided Ontology rules that are explained later in this research. Finally, Algorithm 1 shows the weighting system pseudocode to facilitate the work of the software engineers to follow the steps while designing adaptive gamified systems based on user types.

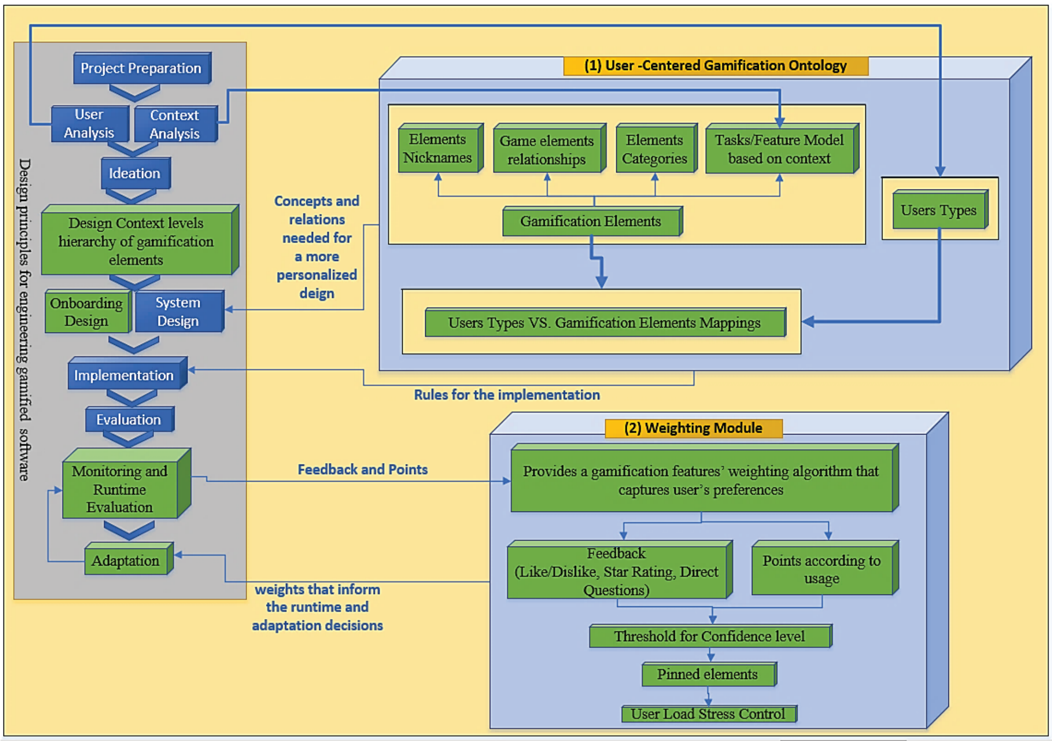

Fig. 8. Updated gamified adaptive framework

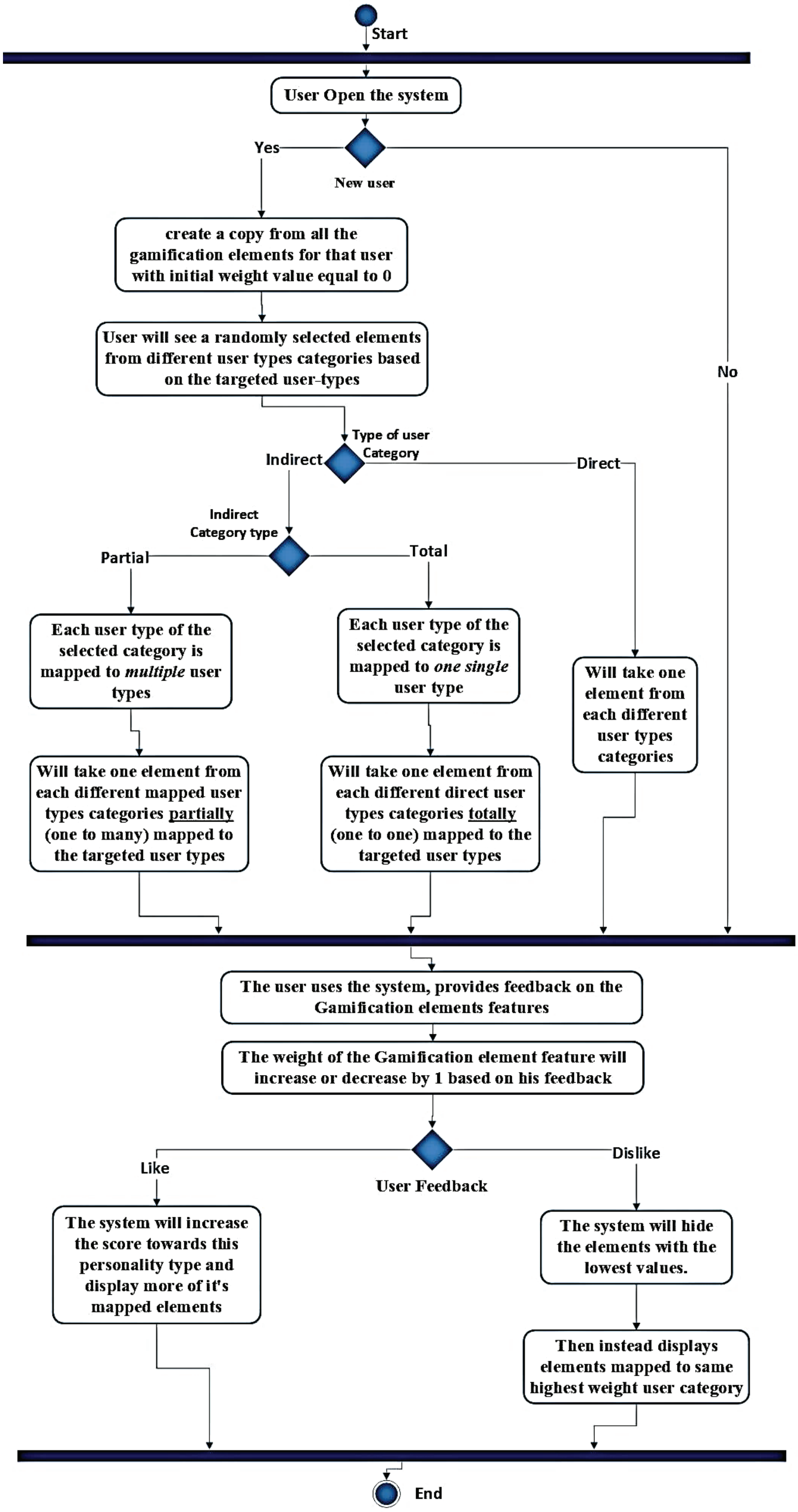

Fig. 9. Activity diagram of the weighting system after modification

Algorithm 1: Weighting System Pseudocode

LOGIN to the system
FOR EACH User
IF New User
CREATE a copy from all the gamification elements for that user with initial weight value equal to 0;
DETERMINE RANDOMLY some selected elements from different user types categories based on the targeted user-types;

IN CASE OF choosing user types that are mapped directly to the elements "*DIRECT*":
Will take one element from each different user types categories;
END "*DIRECT*" **CASE**

IN CASE OF choosing user types that are mapped indirectly to the elements "*INDIRECT*":
IN CASE OF One-to-One mapping "*TOTAL*":
Each user type of the selected category is mapped to one *single direct* user type; Will take one element from each different direct user types categories *totally* mapped to the targeted user types;
END "*TOTAL*" **CASE**

IN CASE OF One-to-Many mapping "*PARTIAL*":
Each user type of the selected category is mapped to *multiple direct* user types; Will take one element from each different user types indirectly mapped elements minus the other directly related user types;
END "*PARTIAL*" **CASE**
END "*INDIRECT*" **CASE**
END RANDOM DETERMINATION

SET the randomly selected elements to be initially visible for the user;
END New User

COLLECT feedback from the user on the Gamification Elements Features
IF user Feedback is Like
INCREASE by 1 The weight of the Gamification element feature
ELSE IF user Feedback is Dislike
DECREASE by 1 The weight of the Gamification element feature
END User Feedback
IN CASE OF choosing user types that are mapped directly to the elements "*DIRECT*":
INCREASE / DECREASE the score towards this personality type based on the feedback
DISPLAY more of its directly mapped elements in case of highest user types values
HIDE the elements with the lowest user types values
DISPLAY instead of it the elements mapped to same highest weight user categories
END "*DIRECT*" **CASE**

IN CASE OF choosing user types that are mapped indirectly to the elements "*INDIRECT*":
IN CASE OF One-to-One mapping "*TOTAL*":
INCREASE / DECREASE the score towards the targeted user types categories totally mapped to the direct user types based on the feedback
DISPLAY more of its indirectly mapped elements in case of highest user types values
HIDE the elements with the lowest user types values
DISPLAY instead of it the elements mapped to same highest weight user categories
END "*TOTAL*" **CASE**

http://dx.doi.org/10.21622/ACE.2023.03.2.024

IN CASE OF One-to-Many mapping "*PARTIAL*":

INCREASE / DECREASE the score towards the targeted user types categories partially mapped to some of the direct user types based on the feedback

DISPLAY more of its indirect partially mapped elements in case of highest user types values

HIDE the elements with the lowest user types values

DISPLAY instead of it the elements mapped to same highest weight user categories

END "*PARTIAL*" **CASE**

END "*INDIRECT*" **CASE**

END User Repeat

Result: Write here the result

Initialization;

while *While condition* **do**

Instructions;

if *condition* **then**

instructions1;

Instructions2;

else

Instructions3;

end

End

4. THREATS TO VALIDITY

1- During the interviews, while explaining the fictional immersion scenario, some examples of the features and how the researchers can apply gamification to them were given to the participants. This could have influenced the thinking of the participants, especially in providing new ideas. To minimize this effect, the interviewer pushes the interviewees to give more and different examples and to think out of the box. Also, the interviewer always challenges the interviewees with the issues that can happen to the system while they provide ideas to let them think in all directions to get the most profit from the interview and to enhance the framework.

2- Another threat in the interviews was the lack of experience of the interviewees in creating new frameworks. This was taken into consideration by encouraging them to think from a high-level view and to look for general ideas that can work with a different type of software and not to think about how this will be applied technically. Besides, the use of the supporting documents and the initial proposed framework helped to give them an idea of how to create new frameworks that can be used by other software developers in different domains.

3- All the recruited interviewees were from the same company which might produce a population bias because they may have been working on the same projects and have current near-thinking ways despite their experience levels.

A common threat to validity might be the lack of knowledge of the participants on the ontology concepts and why it is used as a barrier to understanding the rules and what the benefit of creating them is. This was solved by giving them an idea of the need for ontology while providing the basic concepts with some examples.

V. PROOF-OF-CONCEPT APPLICATION

While running the interviews and discussing the immersion scenario of the Samsung Health application, the interviewees provided some ideas on how to add more elements that adapt to the different types of users. Below are some samples of some elements that can be added to the Samsung Health Application to cover all types of users and to be an evolution for the application. This will help the system to adapt the elements that are shown based on the user preferences and user type by following the steps of the proposed framework in this research.

To make the evolution of the system without changing the whole user interface and user experience, the interviewees suggest some added icons that represent new gamification features mapping to different user types. So, the application in this case will cover the needs of the different user types, permit the integration of the new proposed framework, and be able to adapt the gamification elements based on the usage and preference of the users on runtime.

The Home screen of the Samsung Health Application is composed of the items that the user prefers to manage the most as shown in Figure 10 (Number of steps per day, Active time, Caffeine, Water, Sleep time, Weight management, Blood pressure, …). Each user can manage the items that appear on his/her home screen from the "Manage Items" screen as shown in Figure 11. As a POC, the interviewees have chosen one of Samsung's features to show how the researchers can add Gamification Elements to it that can cover all the different user types' needs. The chosen feature is called "Together" as shown in Figure 12 which is used to track one's steps activity with regards to other people. One can also challenge one of his/her friends or participate in the global challenges as shown in Figure 13.

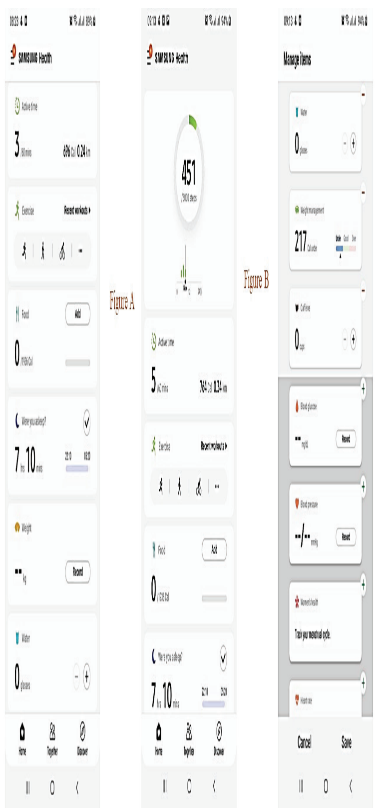

Fig. 10. Samsung Health Home screen

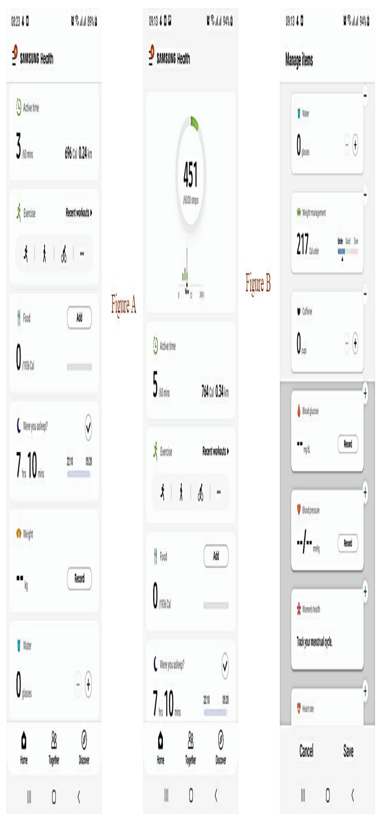

Fig. 11. Manage Items

http://dx.doi.org/10.21622/ACE.2023.03.2.024

Fig. 12. Together Icon

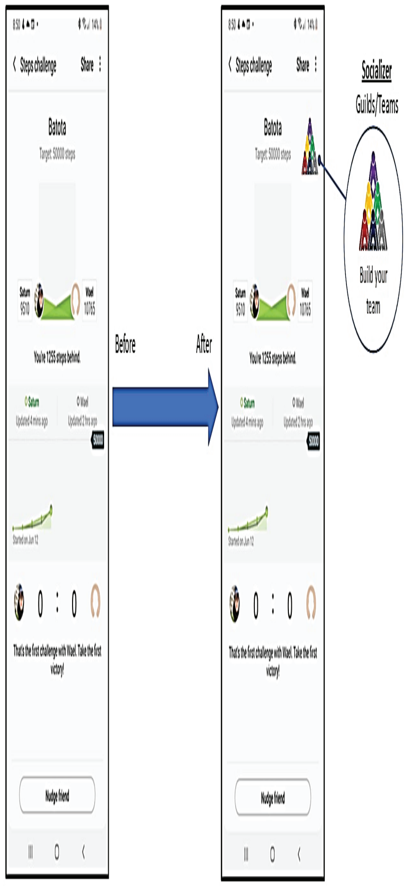

Fig. 13. Steps Challenge

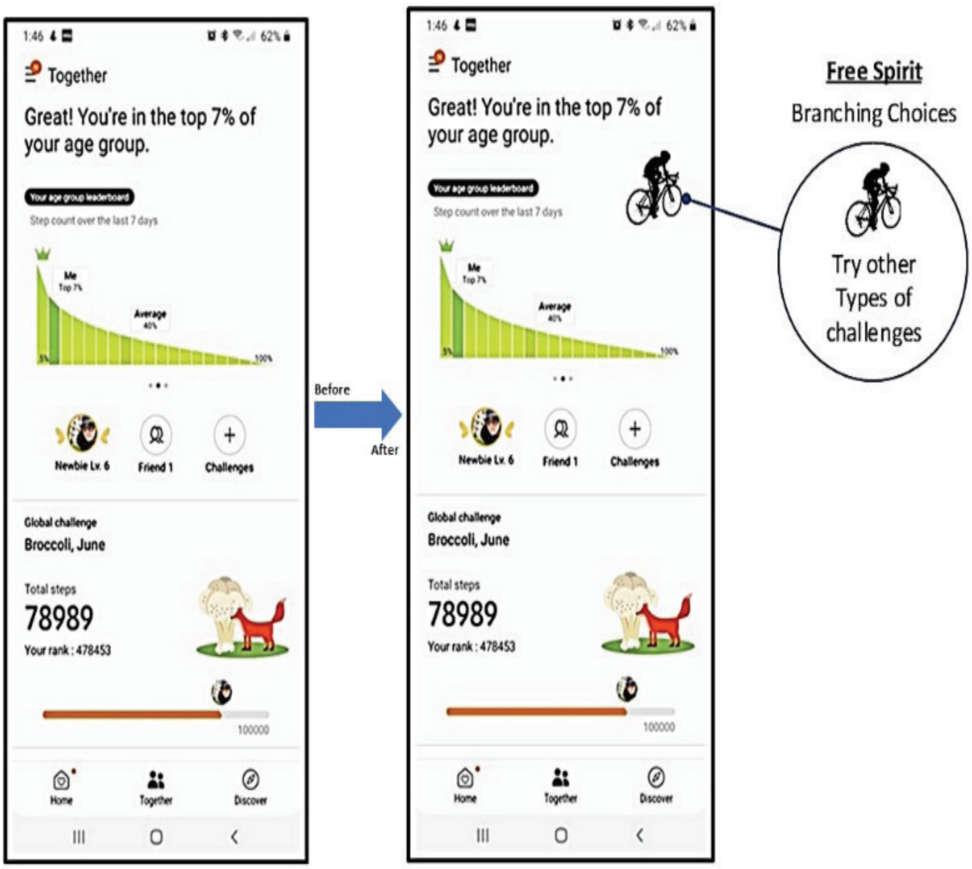

Fig. 14. Branching Choices

http://dx.doi.org/10.21622/ACE.2023.03.2.024

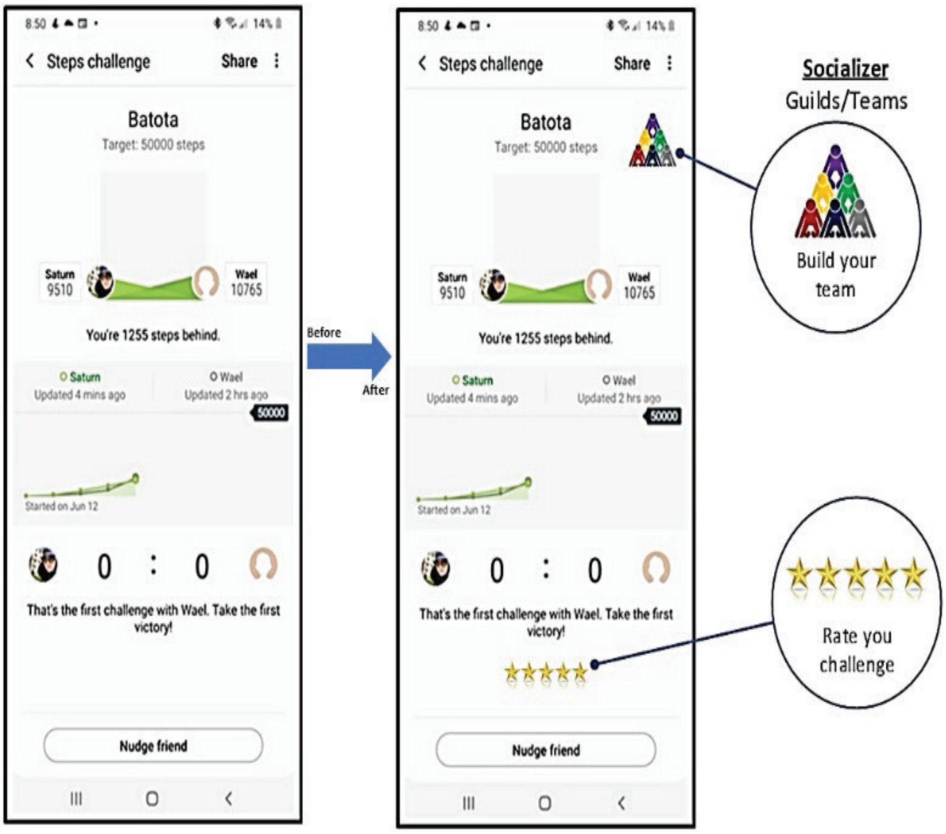

Fig. 15. Guilds/Teams and Rating

Another suggested example is to add a new icon called "Build your teams" as shown in Figure 15 to help the socializer user type to have a place where he/she can use the "Guilds/Teams" gamification element to engage and attract him/her to the system and fulfill his/her needs. A star rating can be added to take the feedback of the user on each feature too.

To this point, everything is fine but the problem is that this feature does not cover all the gamification elements that help in the engagement of the different user types. Some of the interviewees suggested some changes to the "Together" screen while conducting the evaluation study with the Software engineers. For example, the application can provide different types of challenges other than the "Number of Steps" challenges by adding a new icon called "Try other Challenges" like in Figure 14 to help in the engagement of the Free Spirit user type using the "Branching Choices" element and give them different branches to make their choices.

Another example is to add Cloud on the top of the challenge map as shown in Figure 16 and let the Free Spirit user enjoy the exploration of the full map step by step by having the "Exploration" gamification element. Also, a (Like/Dislike) icon can be added to take the feedback from the users in different ways on the different features.

Figure 17 shows the "Personal Bests and Achievements" of the user. Besides, other interviewees suggested having an icon called "Share your achievement" which can be used for the Socializer users to make them able to have the gamification elements that suit them like the "Social Status" element. Also, the "Share and Give reward" icon can be added for the Philanthropist users to enable to them the "Gifting/Sharing" gamification element.

http://dx.doi.org/10.21622/ACE.2023.03.2.024

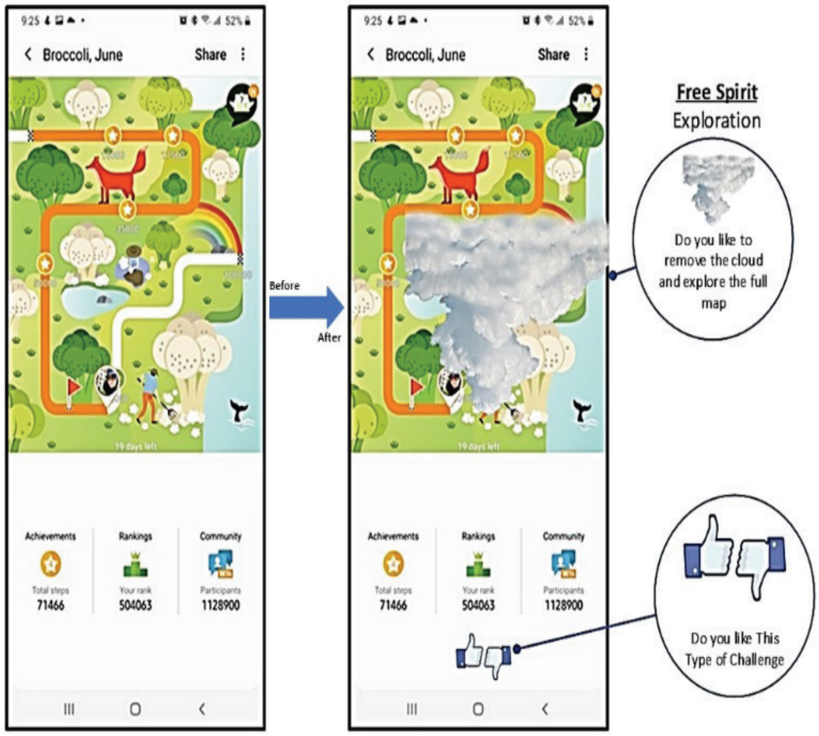

Fig. 16. Feedback and exploration

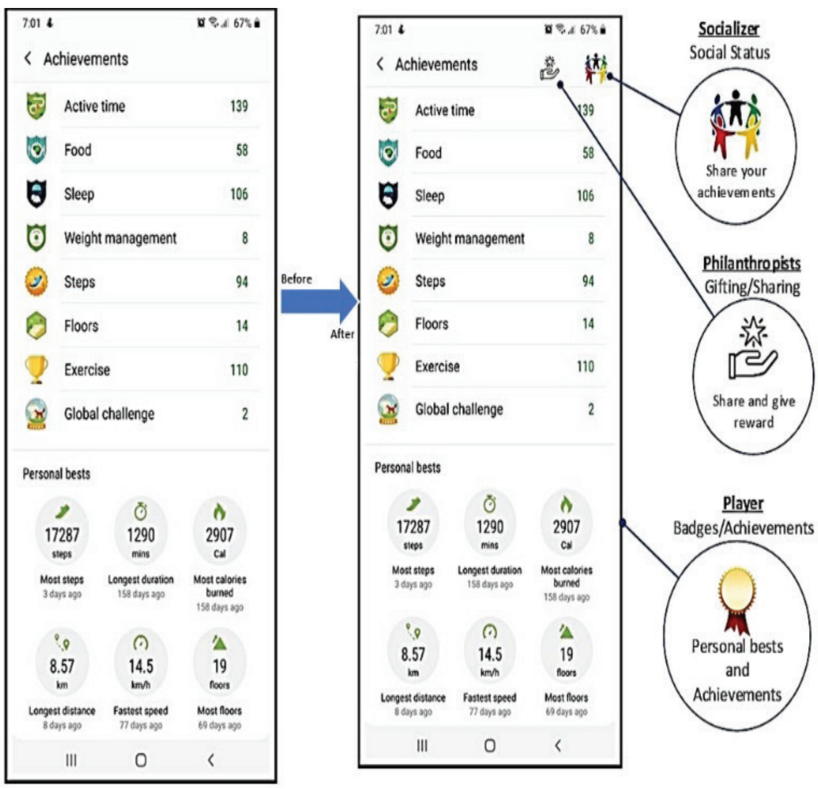

Fig. 17. Social Status, Gifting/Sharing, and Badges/Achievements

The interviewees want to enhance the functionality of the "Promotions" icon in the side menu of the Samsung Health application shown in Figure 18 to give the user different types of promotions and gifts like the Physical Rewards and Prizes which help the users of type "Player" to be more satisfied. A menu item called "Help us with your ideas" can be added if the Disruptors users have ideas for enhancement

http://dx.doi.org/10.21622/ACE.2023.03.2.024

that use the "Innovation Platform" gamification element to engage them more in the application.

Figure 19 shows how the application can satisfy the users of type Disruptor by enabling them to masquerade their data for other users using the "Anonymity" gamification element. Also, the application can satisfy the needs of the users of Free Spirit by enabling them to customize their avatars using the "Customization" gamification elements.

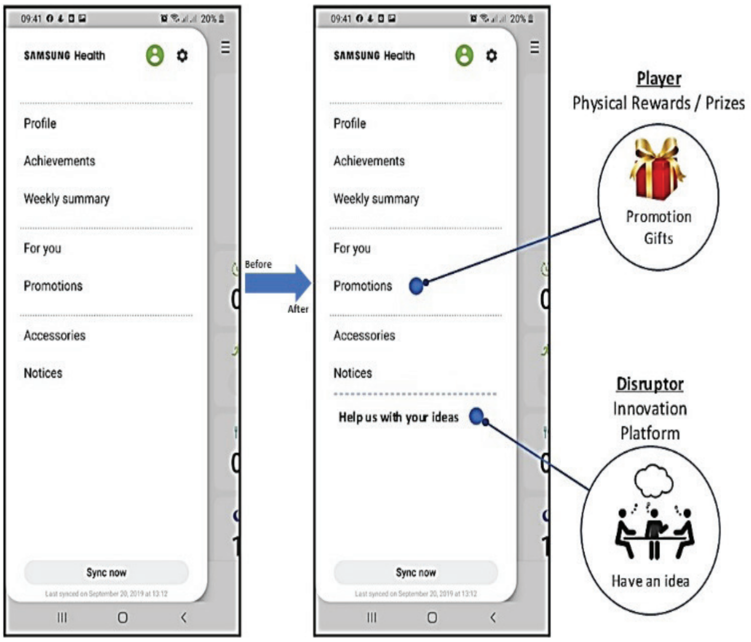

Fig. 18. Physical rewards and innovation platform

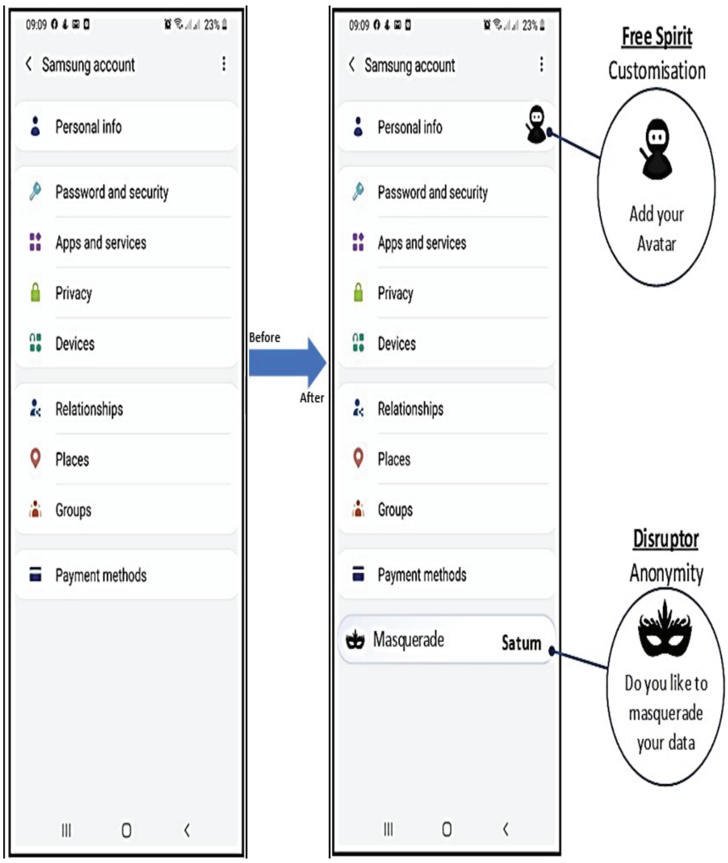

Fig. 19. Customization and anonymity

http://dx.doi.org/10.21622/ACE.2023.03.2.024

Figure 20 shows how the onboarding that was suggested by some of the Software engineers during the interviews can be applied for the user to know the system works and help him/her get the full benefits from the system. Also, this will help the user to understand that the system will be customized based on his interaction and based on his preferences.

The above-suggested changes will permit the different users to access the elements that suit their user types, engage them more with the system, and fulfill and satisfy their needs. On the other hand, the system will record their usage and their preferences to show and hide the elements while following the steps of the proposed framework.

VI. EXTENDED ONTOLOGY IMPLEMENTATION

This section discusses the evaluation of the design and structure of the ontology of the model. The research of [44] shows the main classes of ontology and the object properties hierarchy which contains the relationships between the classes in ontology. In the below subsections, the ontology validation is presented using reasoned, Instances, and SPARQL queries to test the ontology and evaluate its utilization.

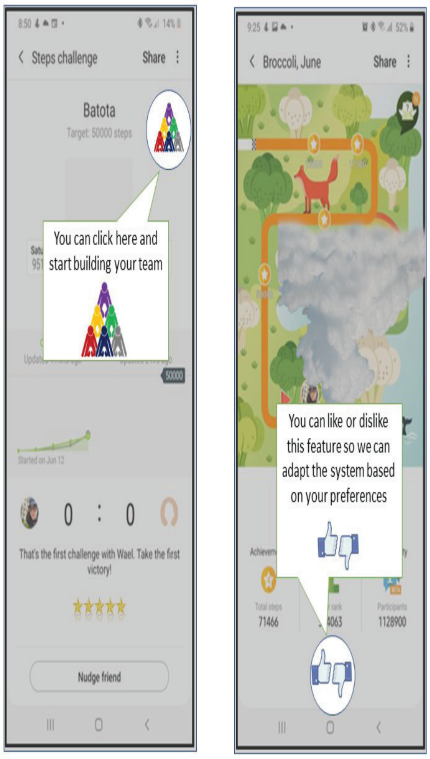

Figure 20: Onboarding

VII. REASONER AND ONTOLOGY VALIDATION

One of the important stages of the ontology that helps to guarantee that the implemented structure is following the common best practices is the Reasoners. Reasoner automatically detects any inconsistency in the ontology while checking the instances and the equivalence. Also, the reasoner checks on the properties and their hierarchy and ranges. In this research, a Reasoner was added to validate the ontology structure and rules of the proposed framework. This step of validation is used to validate the inner structure of a model [52].

http://dx.doi.org/10.21622/ACE.2023.03.2.024

In this research, the researchers used HermiT reasoner [53] to validate the inner structure of the ontology which has an easy user interface that works with Web Ontology Language (OWL) that can be integrated with Protégé and supports object properties, classes, and rules with OWL 2 standards. This is a parallel task that helps in enhancing and structuring the ontology.

VIII. INSTANCES

Instances were created and used with classes and object properties. This helps to validate the created ontology structure by linking the classes, testing the rules, and the object properties, and defining if the listed rules and classes are enough and sufficient for the target model.

Figure 21 shows a sample with a list of instances created for the User type Hexad, Motivators, and Big Five personality traits. The instances have been created with the same naming convention on the subclasses.

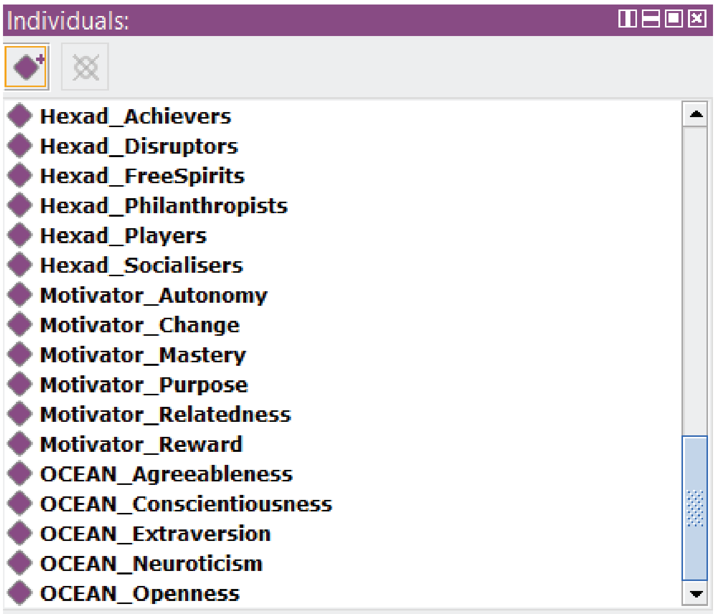

Fig. 21. List of instances created for the User Type Hexad, Motivators, and Big Five personality traits

Figure 22 shows a sample of the Instances of Hexad user Types. Each of them is linked to its mapped subclass type to be used in the validation of the ontology. Figure 23 shows a sample of the Hexad Instances property assertions to apply the Object properties created.

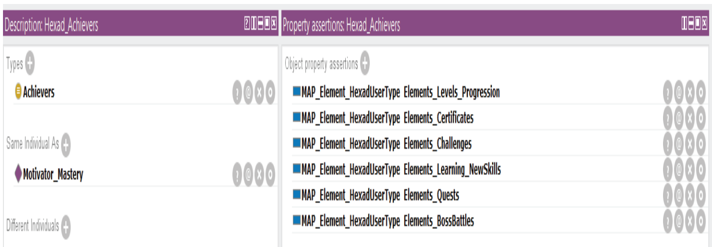

Fig. 22. Sample on the Instances of Hexad User Types

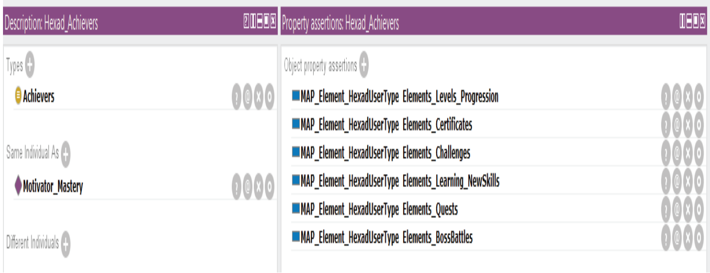

Fig. 23. Sample on the Hexad Instances property assertions

Figure 24 shows some of the list of instances created for the Elements. Each of them is linked to its elements subclass to be mapped to the needed user types and to configure whether the rules of the mappings (Direct, Indirect, Partial, Total) are implemented in the right way.

Each of the Elements Instances configuration is linked to its mapped subclass type. In addition, each of the Instances of Motivators is linked to its mapped subclass type. The OCEAN Instances configuration is linked to its mapped subclass type to be used in the validation of the ontology as well all found at this link https://bit.ly/3m9gjfq

Figure 25 shows a sample of the OCEAN Instances property assertions to apply the Object properties created.

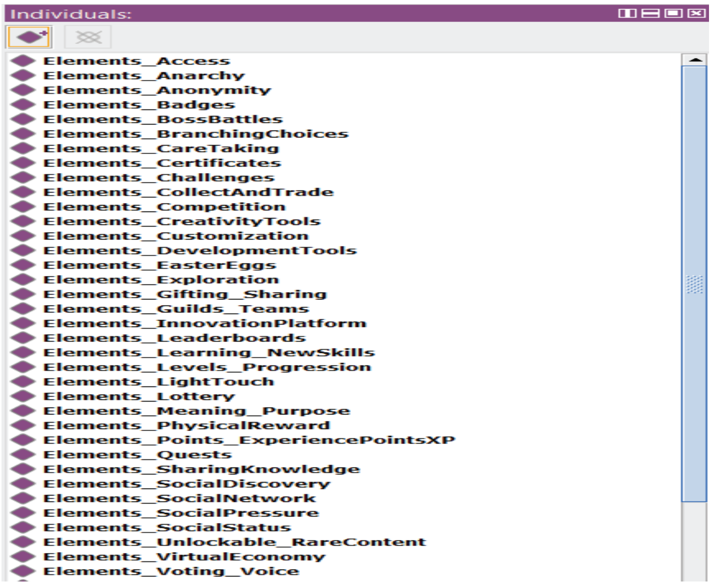

Fig. 24. A sample of the list of instances created for the Elements

Fig. 25. Sample on the OCEAN Instances property assertions

SPARQL QUERIES

After ontology implementation, enhancement, and fixing the bugs that resulted from the reasoner comes the need for queries that extract all the needed information from the ontology and test the result too. Those queries prove that the software engineer will be able to extract the needed information from the implemented ontology and he/she can use it as a knowledge base of structure and rules of the model and be able to enhance the process of designing an adaptive gamified software based on each user type.

The Query shown in Figure 26 is designed to get all the available mapping between the Hexad player types and its Motivators and their related elements for each of them representing the direct mapping that is explained in the Weighting module previously discussed. The result of this query can be used by the software engineers to select the random elements and to adapt the system elements based on these mappings. Mapping between the OCEAN user types and Hexad player types is available as well. Finally, the ontology is designed to get all the available mapping between the OCEAN user types and their related elements for each of them represents the indirect mapping that is explained in the Weighting module of the "Proposed framework" using SPARQL queries. The result of this query can be used by the software engineers to select the random elements and to adapt the system elements based on these mappings.

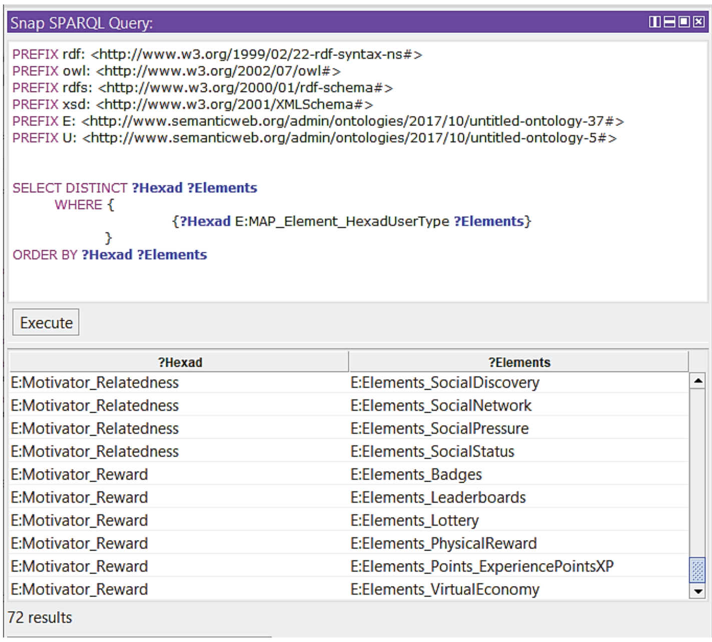

Fig 26. Screenshot for SPARQL Query to get elements of each Hexad player type and its motivators

IX. FEATURE MODEL IMPLEMENTATION

According to the comments resulting from the interviews in the section of "Evaluation Results", it was mentioned that it is very important for software engineers to know each feature in the system is linked to which game elements. So, to add this part, a tool is needed to help in modeling and listing the features. This can be done through the ontology by adding the list of features in a dedicated class for Features Model which can carry any of the features. Then, creating object properties to allow the different types of linkage between the features and the elements. Finally, creating instances was implemented to simulate the link between the feature model and the elements already created and to validate the rules of the Feature Model.

http://dx.doi.org/10.21622/ACE.2023.03.2.024

In this section, a small example representation of the Healthcare Feature Specification (Feature Model) is represented as an example of the feature-to-elements mapping.

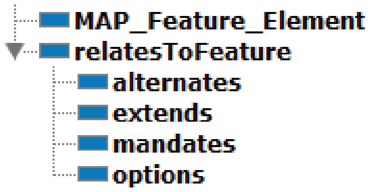

Fig. 27. Feature Model Object Property List for sharing feature

First, a class for the Features Model was created and named "FM_Features" to carry any of the features. After that, Feature Model Object Properties were created. The "relates To Feature" object property was created with the four types of applicable relations (alternates, extends, mandates, and options). Also, the "MAP_Feature_Element" is created to relate the features with the elements. Figure 27 shows the created object properties.

Then, to test and validate the created Feature Model class and the Object properties, sample instances for the features were created. Assuming that the researchers have a sharing feature that has two alternatives (ShareWithFriend and ShareWithOther). The ShareWithFriend feature has three options (Facebook, Instagram, and Twitter). Figure 28 shows the list of created instances.

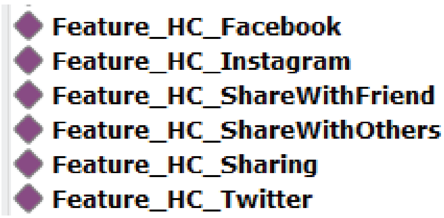

Fig. 28. Feature Model Instances List for sharing feature

Figure 29 shows a representative graph of the list of features and the relations between each other and the elements based on the object properties.

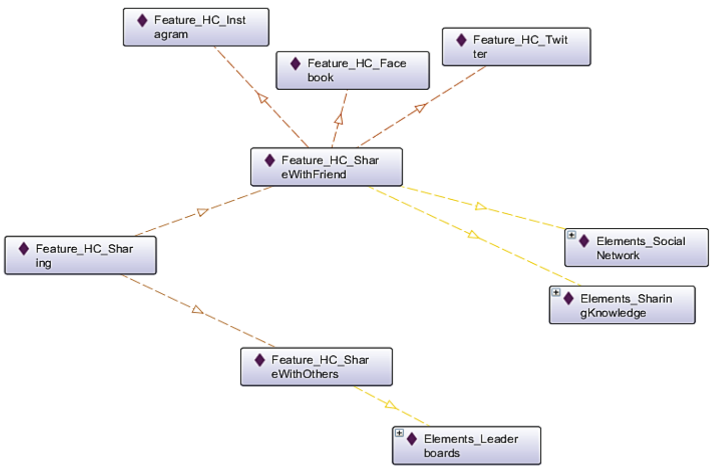

Fig. 29. Representation graph on the Features Instances property assertions relations

http://dx.doi.org/10.21622/ACE.2023.03.2.024

Figure 30 shows the Feature_HC_Sharing Instance configuration of a type Feature class and with two object properties assertions with "alternates" relation (Feature_HC_ShareWithFriend and Feature_HC_ShareWithOthers) to identify the two types of sharing.

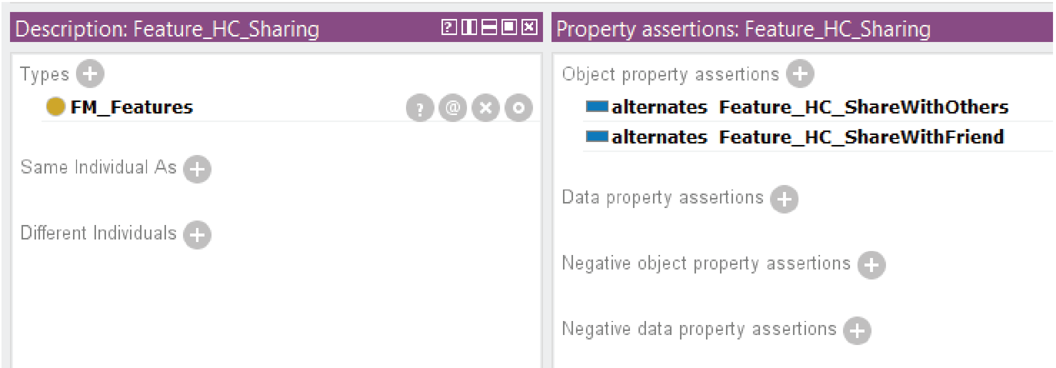

Fig. 30. Feature_HC_Sharing Instance configuration and property assertions

Figure 31 shows the "Feature_HC_ShareWithFriend" Instance configuration of a type Feature class and with three object properties assertions with "options" relation (Feature_HC_Facebook, Feature_HC_Twitter, Feature_HC_Instagram). Figure 32 represents the options relations, respectively.

Also, in Figure 31, the relation between the "Feature_HC_ShareWithFriend" instance and the gamification elements of (Elements_SharingKnowledge and Elements_SocialNetwork) is shown.

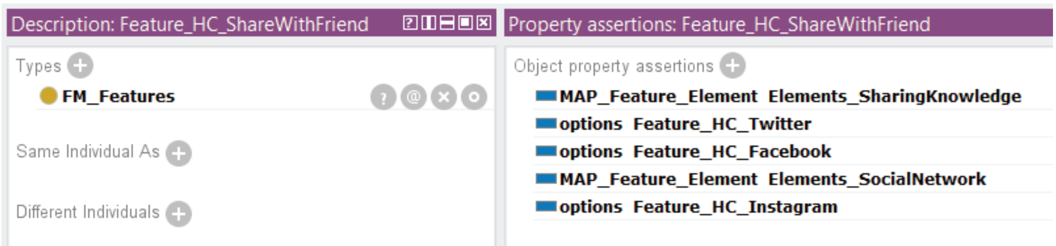

Fig. 31. Feature_HC_ShareWithFriend Instance configuration and property assertions

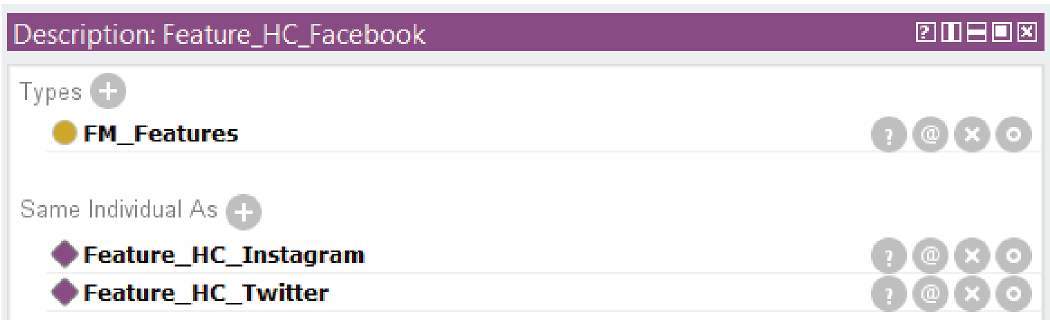

Fig. 32. Sample on the Feature_HC_Facebook Instance configuration

Figure 33 shows the "Feature_HC_ShareWithOthers" instance configuration of a type Feature class and its relation to the gamification elements of "Elements_Leaderboards".

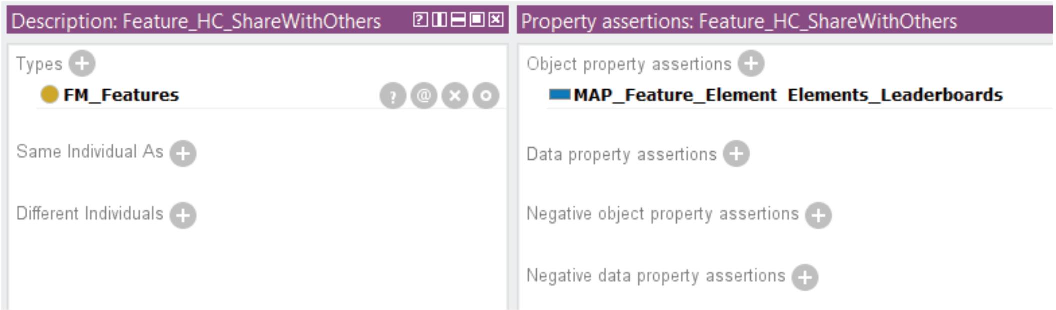

Fig. 33. Feature_HC_ShareWithOthers instance configuration and property assertions

X. CONCLUSION

Adaptive personalized gamified systems are one of the top research scopes for software engineers. This paper focuses on evaluating a previously introduced adaptive personalized gamified framework in the actual context, i.e., evaluating software when users are using it in practice. By access to a broader and different set of users and contexts of use that were unpredictable by analysts, this approach allows users to act as the actual validators of the system and give feedback and it informs the software development process, e.g., by introducing a more formalized structure for concepts and their relationships to provide useful and meaningful information to accurately accomplish the design and adaptation tasks. The enhanced framework and weighting module is also presented after applying the experts' comments.

Also, the paper includes validation testing of the previously introduced Ontology for the adaptive personalized gamified framework using reasoned, Instances, and SPARQL queries.

For future work, extending the framework to include a recommender module which will be an asset to the design framework by integrating social media, using historical data for initial weights, identifying the probability that the user can see the nearest elements, clustering the elements, use teaser popups and suggestions for the users.

REFERENCES

[1] R. Bouzidi, A. De Nicola, F. Nader, and R. Chalal, "OntoGamif: A modular ontology for integrated gamification," *Appl Ontol*, vol. 14, no. 3, 2019, doi: 10.3233/AO-190212.

[2] B. Morschheuser, L. Hassan, K. Werder, and J. Hamari, "How to design gamification? A method for engineering gamified software," *Inf Softw Technol*, vol. 95, 2018, doi: 10.1016/j.infsof.2017.10.015.

[3] M. Böckle, J. Novak, and M. Bick, "Towards adaptive gamification: A synthesis of current developments," in *Proceedings of the 25th European Conference on Information Systems, ECIS 2017*, 2017.

[4] S. A. Elshorbagy, N. Sherief, and W. Abdelmoez, "A framework for utilizing unexplored game elements in designing learning systems," *Advances in Computing and Engineering*, vol. 2, no. 2, 2022, doi: 10.21622/ace.2022.02.2.096.

[5] R. Cheng and J. Vassileva, "Adaptive Reward Mechanism for Sustainable Online Learning Community," in *Frontiers in Artificial Intelligence and Applications*, 2005.

[6] A. Miloff, A. Marklund, and P. Carlbring, "The challenger app for social anxiety disorder: New advances in mobile psychological treatment," *Internet Inter-*

http://dx.doi.org/10.21622/ACE.2023.03.2.024

ventions, vol. 2, no. 4. 2015. doi: 10.1016/j.invent.2015.08.001.

[7] R. Filipcik and M. Bielikova, "Motivating learners by dynamic score and personalized activity stream," in *Proceedings - 9th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2014*, 2014. doi: 10.1109/SMAP.2014.25.

[8] C. Sotirakou and C. Mourlas, "Designing a gamified news reader for mobile devices," in *Proceedings of 2015 International Conference on Interactive Mobile Communication Technologies and Learning, IMCL 2015*, 2015. doi: 10.1109/IMCTL.2015.7359614.

[9] M. Böckle, I. Micheel, M. Bick, and J. Novak, "A design framework for adaptive gamification applications," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2018. doi: 10.24251/hicss.2018.151.

[10] A. Colosimo, "e-Learning kit: Gaming," McGill University. Accessed: Jan. 03, 2023. [Online]. Available: http://libraryguides.mcgill.ca/eLkit/gamingstrategies

[11] M. a. B. D. Specht, "Modeling Adaptive Eductional Methods with IMS Learning Design," *Journal of Interactive Media in Education*, 2007.

[12] J. H. Nannan Xi, "Does gamification satisfy needs? A study on the relationship between gamification features and intrinsic need satisfaction," *Int J Inf Manage*, Jun. 2019.

[13] Takahiro Hasunuma, "Semantic Search Based on Domain Ontology Using Apache Spark and Jena," Hackernoon. Accessed: Dec. 10, 2019. [Online]. Available: https://hackernoon.com/semantic-search-based-on-domain-ontology-using-apache-spark-jena-e00b8838a658

[14] N. H. H. Sherief, "A framework for modelling and utilization of users' feedback for software systems evolution," Doctoral, Bournemouth University, 2017. [Online]. Available: https://eprints.bournemouth.ac.uk/30114/

[15] G. F. Tondello, A. Mora, A. Marczewski, and L. E. Nacke, "Empirical validation of the Gamification User Types Hexad scale in English and Spanish," *International Journal of Human Computer Studies*, vol. 127, 2019, doi: 10.1016/j.ijhcs.2018.10.002.

[16] A. Y. Utomo and H. B. Santoso, "Development of gamification-enriched pedagogical agent for e-learning system based on community of inquiry," in *ACM International Conference Proceeding Series*, 2015. doi: 10.1145/2742032.2742033.

[17] M. D. Kickmeier-Rust, E. C. Hillemann, and D. Albert, "Gamification and smart feedback: Experiences with a primary school level math app," *International Journal of Game-Based Learning*, vol. 4, no. 3, 2014, doi: 10.4018/ijgbl.2014070104.

[18] B. Gil, I. Cantador, and A. Marczewski, "Validating gamification mechanics and player types in an E-learning environment," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015. doi: 10.1007/978-3-319-24258-3_61.

[19] P. Pereira, E. Duarte, F. Rebelo, and P. Noriega, "A review of gamification for health-related contexts," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014. doi: 10.1007/978-3-319-07626-3_70.

[20] B. Morschheuser, J. Hamari, and J. Koivisto, "Gamification in crowdsourcing: A review," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016. doi: 10.1109/HICSS.2016.543.

[21] S. Thiebes, S. Lins, and D. Basten, "Gamifying information systems - A syn-

thesis of gamification mechanics and dynamics," in *ECIS 2014 Proceedings - 22nd European Conference on Information Systems*, 2014.

[22] A. Marczewski, "Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Desig," in *Gamified UK*, 2015.

[23] R. Bartle, "HEARTS, CLUBS, DIAMONDS, SPADES: PLAYERS WHO SUIT MUDS," *Journal of MUD Research*, vol. 1, no. 1, 1996.

[24] B. Monterrat, M. Desmarais, É. Lavoué, and S. George, "A Player model for adaptive gamification in learning environments," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015. doi: 10.1007/978-3-319-19773-9_30.

[25] A. F. Aparicio, F. L. G. Vela, J. L. G. Sánchez, and J. L. I. Montes, "Analysis and application of gamification," in *ACM International Conference Proceeding Series*, 2012. doi: 10.1145/2379636.2379653.

[26] V. Pornsakulvanich *et al.*, "An Analysis of Personality Traits and Learning Styles as Predictors of Academic Performance," *ABAC Journal*, vol. 32, no. 3, 2012.

[27] L. S. Ferro, S. P. Walz, and S. Greuter, "Towards personalised, gamified systems: An investigation into game design, personality and player typologies," in *ACM International Conference Proceeding Series,* 2013. doi: 10.1145/2513002.2513024.

[28] Ö. E. Akgün and M. Topal, "Adaptation of the Gamification User Types Hexad Scale into Turkish," *International Journal of Assessment Tools in Education,* vol. 5, no. 3, 2018.

[29] B. Kim, "Designing Gamification in the right way, in Understanding Gamification," *Libr Technol Rep,* 2015.

[30] J. A. Ratliff, *Integrating video game research and practice in library and information science*. 2015. doi: 10.4018/978-1-4666-8175-0.

[31] F. García, O. Pedreira, M. Piattini, A. Cerdeira-Pena, and M. Penabad, "A framework for gamification in software engineering," *Journal of Systems and Software,* vol. 132, 2017, doi: 10.1016/j.jss.2017.06.021.

[32] K. Seaborn and D. I. Fels, "Gamification in theory and action: A survey," *International Journal of Human Computer Studies*, vol. 74, 2015, doi: 10.1016/j.ijhcs.2014.09.006.

[33] D. Codish and G. Ravid, "Detecting playfulness in educational gamification through behavior patterns," *IBM J Res Dev,* vol. 59, no. 6, 2015, doi: 10.1147/JRD.2015.2459651.

[34] D. Codish and G. Ravid, "Adaptive approach for gamification optimization," in *Proceedings - 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC 2014*, 2014. doi: 10.1109/UCC.2014.94.

[35] L. Shi and A. I. Cristea, "Motivational gamification strategies rooted in self-determination theory for social adaptive e-learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* 2016. doi: 10.1007/978-3-319-39583-8_32.

[36] C. S. González, P. Toledo, and V. Muñoz, "Enhancing the engagement of intelligent tutorial systems through personalization of gamification," in *International Journal of Engineering Education*, 2016.

[37] F. R. H. Andrade, R. Mizoguchi, and S. Isotani, "The bright and dark sides of gamification," in *Lecture Notes in Computer Science (including subseries Lec-*

ture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016. doi: 10.1007/978-3-319-39583-8_17.

[38] H. Abbes, S. Boukettaya, and F. Gargouri, "Learning ontology from Big Data through MongoDB database," in *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, 2016. doi: 10.1109/AICCSA.2015.7507166.

[39] C. Tatsiopoulos and D. M. Katsanta, "On ontologies and knowledge associations in gamified environments," in *2018 9th International Conference on Information, Intelligence, Systems and Applications, IISA 2018*, 2019. doi: 10.1109/IISA.2018.8633589.

[40] Valerio Cosentino, "Modeling gamification and using it to improve the learning of software modeling," *Modeling Language*, 2017. [Online]. Available: https://modeling-languages.com/modeling-gamification/

[41] D. Dermeval, J. Albuquerque, I. I. Bittencourt, S. Isotani, A. P. Silva, and J. Vassileva, "GaTO: An Ontological Model to Apply Gamification in Intelligent Tutoring Systems," *Front Artif Intell*, vol. 2, 2019, doi: 10.3389/frai.2019.00013.

[42] G. O'Regan, *Concise Guide to Software Engineering From Fundamentals to Application Methods*. 2017.

[43] W. El Gammal, N. Sherief, and W. Abdelmoez, "User-based Adaptive Software Development for Gamified Systems," in *ACM International Conference Proceeding Series*, 2020. doi: 10.1145/3397056.3397088.

[44] J. Vandenberghe, "The 5 Domains of Play: Applying Psychology's Big 5 Motivation Domains to Games," in *Proceedings of the Game Developers Conference 2012*, 2012.

[45] G. F. Tondello, R. R. Wehbe, L. Diamond, M. Busch, A. Marczewski, and L. E. Nacke, "The gamification user types Hexad scale," in *CHI PLAY 2016 - Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, 2016. doi: 10.1145/2967934.2968082.

[46] Syed Muhammad Sajjad Kabir, "Methods of data collection," in *Basic Guidelines for Research: An Introductory Approach for All Disciplines*, Jahur, Ed., Bangladesh: Book Zone Publication, 2016.

[47] A. Strauss and J. Corbin, *Basics of qualitative research: Grounded theory procedures and techniques (2nd ed)*. 1998.

[48] H. Kip *et al.*, "Identifying the added value of virtual reality for treatment in forensic mental health: A scenario-based, qualitative approach," *Front Psychol*, vol. 10, no. FEB, 2019, doi: 10.3389/fpsyg.2019.00406.

[49] L. Samsung Electronics CO, "Feel good with Samsung Health."

[50] Samsung Electronics Co. LTD, "Samsung Health," Samsung Electronics Co. LTD. Accessed: Sep. 02, 2023. [Online]. Available: https://www.samsung.com/global/galaxy/apps/samsung-health/

[51] D. D. Kehagias, I. Papadimitriou, J. Hois, D. Tzovaras, and J. Bateman, "A methodological approach for ontology evaluation and refinement," in *ASK-IT Final Conference. June.(Cit. on p.)*, 2008.

[52] N. Seyff, F. Graf, and N. Maiden, "Using mobile RE tools to give end-users their own voice," in *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE2010*, 2010. doi: 10.1109/RE.2010.15.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

OBJECT DETECTION IN INLAND VESSELS USING COMBINED TRAINED AND PRETRAINED MODELS OF YOLO8

Ahmad A. Goudah[1], Maximilian Jarofka[2], Mohmed El-Habrouk[3], Dieter Schramm[4] and Yasser G. Dessouky[5]

[1,5] Arab Academy for Science, Technology and Maritime Transport

[3] Department of Electrical Engineering, Faculty of Engineering, Alexandria University, Alexandria, Egypt,

[1,2,4] Institute of Mechatronics University Duisburg-Essen D-47057 Duisburg, Germany,

Emails: {ahmad.goudah@aast.edu, maximilian.jarofka@uni-due.de, eepgmme1@yahoo.co.uk, dieter.schramm@uni-due.de, ygd@aast.edu}

Received on, 14 June 2023 - Accepted on, 11 September 2023 - Published on, 20 November 2023

ABSTRACT

One of the central challenges within the domain of computer vision is object detection, encompassing the identification and localization of specific entities within an image. Introducing a pioneering approach, the YOLO (You Only Look Once) algorithm emerged in 2015, executing object detection within a singular neural network. This innovation triggered a profound transformation within object detection, ushering in remarkable advancements beyond the capacities of the preceding decade. Subsequently, YOLO underwent successive iterations, culminating in eight versions that have earned prominent stature among leading object identification algorithms. This recognition is attributed to YOLO's integration of state-of-the-art concepts prevalent in the realm of computer vision research. Particularly noteworthy is the latest iteration, YOLOv8, which demonstrates superior performance in terms of both accuracy and speed when juxtaposed with YOLOv7 and YOLOv5. This study delves into the most recent strides in object detection as an important field of computer vision, which has been seamlessly assimilated into YOLOv5, YOLOv7, YOLOv8, and their antecedents. The introductory section, delineating the foundational importance of object detection, aligns seamlessly with the research's overall narrative. The elucidation of object detection's significance within diverse contexts, such as vehicle identification across varying scales and environments, underscores its multifaceted utility. The refinement process further enhances the discernment of YOLO's progression through its iterations, elucidating the evolution from the preeminent YOLOv1 to the recent apex represented by YOLOv8. Notably, the text now highlights YOLOv8's distinctive advancements in accuracy and speed over YOLOv7 and YOLOv5, lending heightened clarity to the incremental evolution of the algorithm. The augmentation extends to the exploration of YOLOv8's amalgamation with contemporary computer vision concepts. These concepts' incorporation is now underscored, demonstrating how YOLOv8 benefits from the strides made in computer vision research. The final passage captures the thrust of the research, examining the application of the developed object detection models within the specific context of inland waterway vessels. The distinct stages of detection, the addition of new classes, manual annotation, and the process of network training are now presented with greater precision,

http://dx.doi.org/10.21622/ACE.2023.03.2.064

ensuring a lucid understanding of the methodology. Moreover, the description of the combined model's competence to detect all 85 classes with a measure of accuracy enhances the comprehensiveness of the study's contributions.

Index Terms: Object Detection, YOLO, YOLO8, Autonomous, Vehicles, Waterway Inland Vessels, Bounded Boxes, Neural Networks, CNN

I. INTRODUCTION

Object detection is a well-known research topic that has been extensively studied in computer vision systems. Object detection objective in many fields is to identify the location and classify the objects that are of interest on the scene. The analysis of detection methods involves the utilization of different techniques to extract features that are primarily designed for detecting vehicle objects at multiple scales and in various environmental conditions. In this context, the term "object" typically refers to entities that are either human-made or possess a high degree of structure (such as vehicles, buildings, ships, etc.), And which are distinguishable from complex background environments and landscapes. Over the past two decades, the improved accuracy of image interpretation in these applications has allowed them to fulfill the necessary requirements in real world situations, which has in turn greatly advanced the development of Earth observation technologies and object detection methodologies. Many computer vision fields rely on moving object detection as a fundamental research component. Over the last few decades, numerous detection methods have been suggested. Existing surveys have mostly concentrated on the accuracy of detection, but practical detection tasks were not taken into account. However, in various application tasks, the training modes and requirements differ significantly. Moving object detection serves as the initial stage in numerous computer vision processes aimed at identifying moving objects that are not part of a scene, known as the foreground. Afterward, the objects are isolated from the background through segmentation. Several intelligent monitoring tasks rely on moving object detection and foreground segments, including but not limited to target tracking, behavior analysis, traffic monitoring, visual surveillance, and human-machine interaction [1]–[4].

The field of computer vision is currently experiencing a wide spread use of deep learning models, thanks to the development of Deep Convolutional Neural Networks (DC- NNs) and the increasing computational power of GPUs. Object detection aims to identify visual objects belonging to specific classes, such as TV/monitor, books, cats, humans, etc., And determine their location by enclosing them in bounding boxes. Once located, these objects are then classified into their respective categories. Object detection is a task that involves detecting and categorizing a diverse range of objects within an image. It involves identifying the location of an object in an image, drawing a bounding box around it, and then determining the category it belongs to. Object detection also emphasizes the recognition of instances belonging to predefined categories. The advancement of object detection can be divided into two distinct historical phases. The period before 2014 was dominated by traditional methods, whereas the era following 2014 was characterized by the emergence of deep learning-based methods. The architectures of these two phases differ in terms of accuracy, speed, and hardware resources required. When compared to traditional techniques, Convolutional Neural Networks (CNNs) have superior architecture and are significantly more expressive, which contributes to their improved performance [5].

In this paper, different methods of object detection will be -roughly- over viewed. YOLO (You Only Look Once) as a popular object detection method, will be compared -briefly- to other state-of-the-art object detection methods through some relevant studies. The rest of this paper will cover the application of detecting objects for

http://dx.doi.org/10.21622/ACE.2023.03.2.064

inland waterway vessels through three stages and combined trained and pretrained model. The first one, is to detect the objects listed in the ready trained list from the coco dataset that contains 80 classes and includes objects like:(person, bird, boat, … and etc.). The next stage of this system is to add five classes which are more popular in the environment of the inland vessels through their waterway path and start to train and validate the output of a certain set. The selected classes which are added at this step are as follows:

- L Shore
- R Shore
- Bridge Pillar
- Crane
- Buoyer

After doing the steps of manual annotation and labeling, the images set will be divided to two sub-sets: train and validate sets. The training of the YOLO8 custom network is done and labels are generated accordingly. The third stage of the system here is responsible for combining the detected output from the ready pretrained YOLO8 network with the new output generated from the next level of the 5 classes trained network. Combined model will be able to detect all 85 classes with a measures' accuracy as shown on results part of this paper.

II. OBJECT DETECTION METHODS: HISTORY AND STATE OF THE ART

Object detection is a fundamental task in computer vision that involves detecting the presence and location of objects in an image or video. Over the years, various object detection methods have been developed, each with its own strengths and limitations. These methods will be roughly overviewed in the next part of this paper section, and they could be categorized as two main big groups, as [6] states:

A. Traditional Methods.

B. Deep Learning Based Methods.

A. TRADITIONAL METHODS

Traditional methods of object detection involve using computer vision techniques to analyze the image and extract information about the objects within it. These methods typically involve a series of image processing steps, including feature extraction, object classification, and object localization. These steps are processed step by step as follows:

1) Feature extraction: The first step in traditional object detection involves extracting features from the image that can be used to identify and classify objects. Common feature extraction techniques include edge detection, corner detection, and texture analysis.

2) Object classification: Once features have been extracted, the next step is to classify them into different categories. This can be done using a variety of machine learning algorithms, such as support vector machines (SVMs) or decision trees.

3) Object localization: The final step in traditional object detection is to localize the objects within the image. This typically involves identifying the position and size of the object relative to the image frame.

There are several traditional methods of object detection that are commonly used in computer vision research and applications, as shown in figure 1. These methods are including:

http://dx.doi.org/10.21622/ACE.2023.03.2.064

1) Feature-based methods:

- SIFT (Scale-Invariant Feature Transform) features: SIFT features are a feature extraction technique that is commonly used in object detection and recognition. The technique involves identifying distinctive features in an image, such as corners and edges, and then matching those features to a database of known objects. SIFT detects and describes local features that are invariant to scale, rotation, and illumination changes. SIFT works by identifying key points in an image that are both stable and distinctive, and then describing the surrounding region of the image in a way that is invariant to rotation and scale changes. SIFT is quite accurate and robust to noise and illumination changes, but can be slow to compute and is patented, which may limit its use in some contexts [7].

- HOG (Histogram of Oriented Gradients) features: HOG features are a feature extraction technique that involves calculating the gradients of an image and then constructing a histogram of the gradient orientations. HOG computes a histogram of gradient orientations in an image to capture edge information and shape features. This technique has been used to successfully detect objects in a variety of applications, including pedestrian detection and face detection [8].

- SURF (Speeded Up Robust Features): SURF features are similar to SIFT features in that they involve identifying distinctive features in an image. However, SURF features are designed to be faster and more robust than SIFT features, making them well-suited to realtime object detection applications. SURF detects and describes local features that are invariant to scale, rotation, and affine distortion using a modified Hessian matrix. [9].

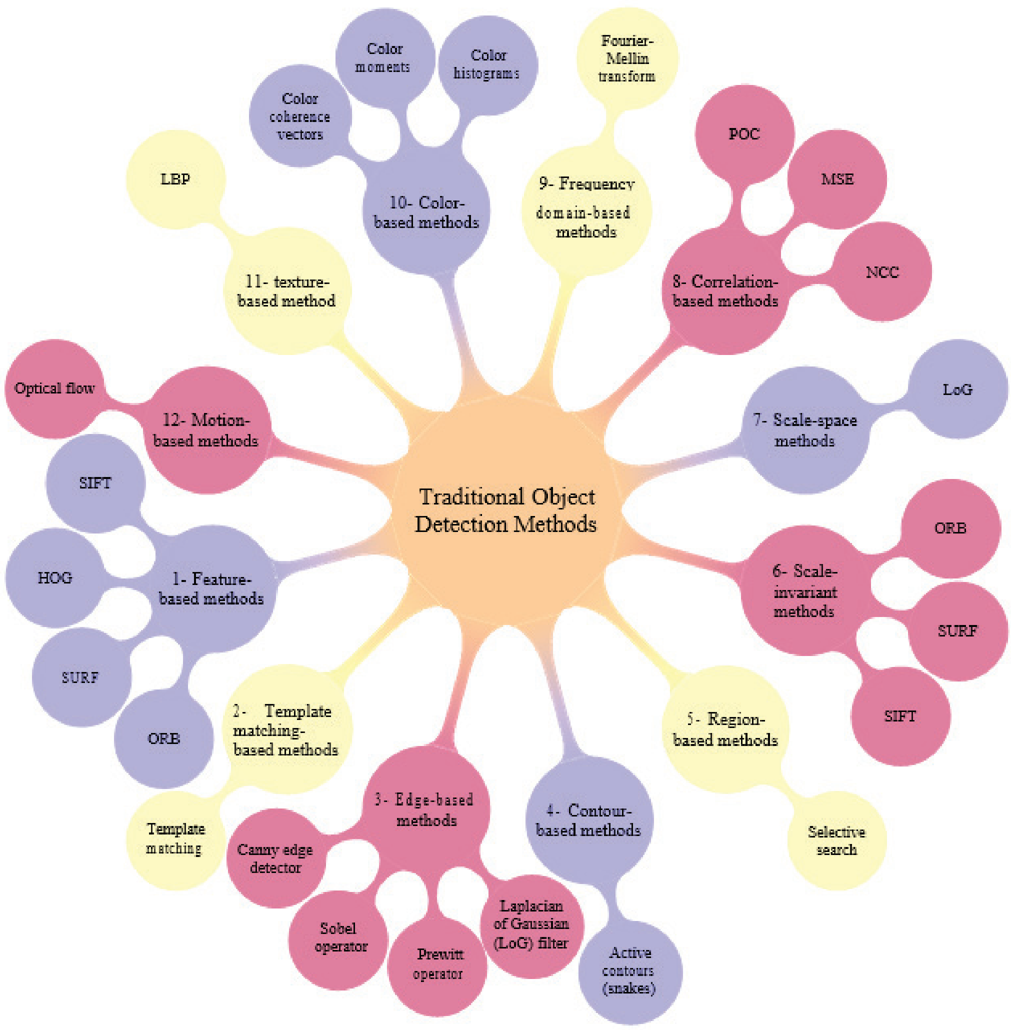

Fig. 1. Traditional Methods Based Object Detection Techniques.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

- ORB (Oriented FAST and Rotated BRIEF): a fusion of the FAST (Features from Accelerated Segment Test) keypoint detector and the BRIEF (Binary Robust Independent Elementary Features) descriptor that is faster and more robust than SIFT and SURF. ORB is less accurate than SIFT, but is more robust to lighting changes and has a lower memory footprint. ORB is also free to use and does not have any patent restrictions. Overall, both SIFT and ORB are effective methods for object detection, but they differ in their trade-offs between accuracy, speed, and robustness. The choice of which method to use will depend on the specific application and the properties of the images being analyzed [10].

Detailed overview of the ORB algorithm and its approach to object detection could be listed as:

a) Feature Detection: ORB starts by detecting features within an image, such as corners or edges, that are likely to be distinctive and repeatable. ORB uses a modified version of the FAST algorithm for feature detection, which is designed to identify features with high-contrast edges.

b) Feature Description: Once features are detected, ORB then extracts a compact binary descriptor for each feature. The descriptors are designed to be robust to changes in lighting and scale, making them suitable for object detection across a wide range of conditions.

c) Feature Matching: After descriptors have been extracted for each image, ORB then matches the descriptors between the target object and the scene image. This is typically done using a nearest neighbor search, where each descriptor in the target object is compared to descriptors in the scene image to find the best match.

d) RANSAC-based Pose Estimation: Once feature matches have been identified, ORB uses a robust estimation algorithm called RANSAC (Random Sample Consensus) to estimate the pose (position and orientation) of the target object within the scene. RANSAC iteratively samples subsets of the feature matches and computes the pose estimate for each subset, selecting the estimate with the highest number of inliers (matches that agree with the estimated pose) as the final estimate.

e) Object Localization: Finally, once the object pose has been estimated, ORB can localize the object within the scene by drawing a bounding box around the object and overlaying it on the scene image.

Overall, the ORB algorithm is an effective and efficient traditional method for object detection, especially in scenarios where deep learning approaches may not be feasible due to limited data or hardware resources. However, it may not perform as well as state-of-the-art deep learning methods in complex and varied scenes or with highly similar objects [11] [12].

2) Template matching-based methods:

- Template matching: Template matching is a simple method of object detection that involves comparing an image to a template image of the object being detected. If the two images match, the object is considered to be present in the image. Template matching compares a template image with a larger search image at different locations and scales to find the best match, using meth- ods such as cross-correlation or normalized correlation [13].

3) Edge-based methods: These methods are often used for real-time applications, as they are computationally efficient and can work well in low light conditions [14].

- Canny edge detector: One of the earliest and most widely used edge detection algorithms which was introduced by John Canny in 1986. The Canny detector applies a series of filters to an image to identify edges, and then uses non-maximum suppression and hysteresis thresholding to produce the final edge map. The algorithm has been shown to produce high-quality edge maps with low false positive rates. It detects sharp changes in intensity or color, which are then used to define the boundaries of objects [15].

- Sobel operator
- Prewitt operator
- Laplacian of Gaussian (LoG) filter

The Sobel and Prewitt operators are simple gradient based edge detectors that compute the gradient magnitude of an image and threshold it to obtain an edge map. The LoG filter is a more sophisticated approach that convolves an image with a Gaussian filter followed by the Laplacian operator to identify edges [16] [17].

4) Contour-based methods:

- Contour-based methods are a type of image processing technique that focus on identifying and analyzing the edges or contours in an image. These methods are commonly used in computer vision, pattern recognition, and object detection. As [18] proposes, new algorithm for contour detection that combines both bottom-up and top-down approaches to achieve more accurate results.

5) Region-based methods:

- Selective search: generates a large set of region proposals in an image based on color, texture, and size similarity, which can then be classified as object or background regions. This algorithm involves identifying and localizing objects within an image by dividing the image into regions and analyzing each region for the presence of an object. These methods are commonly used in computer vision applications such as object recognition and tracking [11].

6) Scale-invariant methods:

- SIFT, SURF, ORB: as described above. Scale-invariant methods are a type of object detection technique that aim to identify objects at different scales within an image. These methods are commonly used in computer vision applications such as object recognition, tracking, and surveillance [19].

7) Scale-space methods: refer to techniques that analyze images at different scales to detect objects of varying sizes. These methods are based on the principle that objects in images can appear at different scales due to their size, distance from the camera, and other factors. By analyzing images at multiple scales, scale-space methods can detect objects regardless of their size or location in the image. One popular scale-space approach is:

- The Laplacian of Gaussian (LoG) operator, which involves convolving the image with a Gaussian filter at different scales and then applying the Laplacian operator to the filtered image. The resulting image highlights regions with high intensity variations at different scales, which can be used to detect objects of different sizes [20].

8) Correlation-based methods: Correlation-based methods are a type of object detection technique that involve computing the correlation between an image and a template to identify objects. Correlation-based methods for object detection describe several variations of methods, including:

- Normalized cross-correlation (NCC): a method for comparing two images by computing their correlation coefficients at each pixel location to find the best match.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

- Mean Squared Error (MSE) correlation
- Phase-only correlation (POC)

These methods are commonly used in computer vision applications such as face recognition, tracking, and surveillance. correlation-based methods can be extended to handle variations in lighting, pose, and occlusion [21].

9) Frequency domain-based methods:

- Fourier-Mellin transform: a technique for matching objects in an image by analyzing their frequency content and phase relationships. Frequency domain- based methods are a type of object detection technique that analyze the frequency content of an image to identify objects. These methods are commonly used in computer vision applications such as image processing, texture analysis, and pattern recognition. Frequency based method for object detection that uses a bank of Gabor filters to analyze the frequency content of an image. The method is able to detect objects at different scales and orientations by analyzing the responses of the Gabor filters across the image. the effectiveness of this method on several object detection shows that it is able to achieve high accuracy with low computational complexity [22].

10) Color-based methods: Color-based methods are a type of object detection technique that use color information to identify objects in an image. These methods are commonly used in computer vision applications such as traffic monitoring, object tracking, and image retrieval.

- Color histograms: Represent the distribution of color values in an image using a histogram, which can be used to identify objects with specific color characteristics.
- Color Moments: are a type of feature extraction method commonly used in color-based object recognition and image retrieval. Color moments are statistical descriptors that capture the statistical properties of color distributions in an image. There are several types of color moments, including:
  - The first order moments (mean)
  - Second order moments (variance)
  - higher order moments (skewness, kurtosis, etc.) These moments are computed separately for each color channel (e.g., red, green, blue) and can be combined into a feature vector to represent the color distribution of an image. Color moments are popular because they are simple to compute, invariant to translation and scaling, and can capture higher-order statistical properties of the color distribution. However, they can be sensitive to noise and may not capture spatial information about the object [23].
- Color Coherence Vectors: are a type of feature extraction method used in color-based object recognition and image retrieval. Color coherence vectors capture the spatial coherence of color distributions in an image by measuring the degree to which neighboring pixels have similar color values.

To compute color coherence vectors, an image is first segmented into regions or objects, and the color distribution of each region is represented as a color histogram. Next, the spatial coherence of the color distribution is measured by computing the similarity between adjacent pixels or regions. This similarity measure can be based on various metrics, such as:

- Euclidean distance
- Mahalanobis distance
- Correlation coefficient

The result of the computation is a set of vectors that represent the spatial coherence of the color distribution in each region. These vectors are called color coherence vectors and can be used as features for object recognition or image retrieval.

Color coherence vectors are effective in capturing the spatial coherence of color distributions and can handle variations in lighting, shadows, and object pose. However, they can be sensitive to image noise and may not capture global color information [24].

11) Texture-based methods: refer to techniques that utilize the texture or patterns in an image to detect objects. These methods are based on the principle that objects in images often have distinctive textures or patterns that can be used to identify them. Texture-based methods are typically used in scenarios where the objects of interest have similar colors or shapes to the background, making it difficult to detect them using traditional color or shape- based methods.

One popular texture-based approach is:

- The Local Binary Pattern (LBP) operator: which involves comparing the intensity values of pixels in an image with their neighboring pixels and encoding the results as binary patterns. The resulting patterns can be used to detect texture variations in the image and identify objects with distinctive textures. that works by computing a binary pattern for each pixel in an image based on the values of its surrounding pixels. The Local Binary Pattern (LBP) operator binary patterns are then used to describe the texture of the image. LBP can be used for object detection by comparing the texture of the object to the texture of the background. [25].

12) Motion-based methods: refer to the techniques that utilize the motion of objects to identify and track them in video sequences. These methods are typically used in scenarios where objects are moving in a dynamic environment and traditional static object detection techniques may not be sufficient.

One popular motion-based approach is:

- Optical flow: which computes the displacement of pixels between consecutive frames in a video sequence. Optical flow can be used to track objects by detecting the regions where the flow vectors are consistent over time. Another approach is background subtraction, which involves subtracting a background image from each frame in the video sequence to highlight moving objects [26].

In conclusion, traditional methods for object detection have their own strengths and weaknesses, and the choice of method depends on the specific application and the nature of the objects to be detected.

While deep learning-based methods have achieved state- of-the-art performance in many areas, traditional methods are still useful in situations where data or hardware resources are limited. Table I: shows the comparison of grouped traditional methods of object detection into different categories and lists their pros and cons.

B. DEEP LEARNING BASED METHODS

There are many Convolutional Neural Networks architectures which are developed by the time to solve object detection problem in a good, accurate and fast way. These architectures could be listed as:

4) Region-based Convolutional Neural Networks (RCNNs): RCNNs are one of the earliest and most popular object detection methods [43]. They use a two-stage approach, where the first stage proposes a set of object regions in the

image, and the second stage classifies each region as containing an object or not [52] [11]. The most famous RCNN models are Faster RCNN, RFCN and Mask RCNN, which have achieved state-of-the-art results on various benchmark datasets [52] [53].

5) Single-Shot Detectors (SSDs): SSDs are a one-stage approach that simultaneously predicts the class and location of objects in an image [44]. Unlike RCNNs, they do not require a separate region proposal step, making them faster and more efficient [47]. Some of the popular SSD models are YOLO (You Only Look Once), RetinaNet, and EfficientDet [48]. EfficientDet is a family of object detection models developed by Google Research that uses efficient architectures and training techniques to achieve high accuracy while using fewer computational resources. It achieves state-of-the-art performance on several benchmark datasets at 2020 [48].

6) Anchor-Free Detectors: Anchor-free detectors are a new class of object detectors that do not rely on predefined anchor boxes for object localization [54]. Instead, they use a set of learnable points to predict object locations and sizes [55]. Some examples of anchor-free detectors are CornerNet, FCOS (Fully Convolutional One-Stage Detector), and RepPoints [56].

7) Transformer-based Detectors: Transformer-based object detectors use self-attention mechanisms to capture long range dependencies between different parts of an image [50]. These models have achieved impressive results on various object detection benchmarks. Some popular transformer-based detectors are DETR (DEtection TRansformer), Deformable DETR, and Sparse R-CNN [51].

8) Hybrid Approaches: Hybrid object detectors combine multiple detection methods to improve detection accuracy and efficiency. For example, CenterNet combines the efficiency of SSDs with the accuracy of RCNNs [57], while Cascade RCNNs use multiple stages of classification to refine object detection results [58].

As figure 2 shows, these different types of object detection methods which are depending on Deep Learning Networks, could be summarized and discussed as follows:

1) Two-stage detectors: These methods include Faster RCNN [43], R-FCN [59], and Mask R-CNN [60].

2) One-stage detectors: Examples of these methods include YOLO [44], SSD [46], and RetinaNet [47].

3) Multi-scale detectors: These methods include FPN [47], RetinaNet [47], and Cascade R-CNN [58].

4) Anchor-based detectors: Examples of these methods include Faster R-CNN [43], RetinaNet [47], and YOLOv3 [61].

5) Anchor-free detectors: Examples of these methods include FCOS [55], CornerNet [54], and CenterNet [62].

6) 3D object detectors: These methods include PointNet [63], MV3D [64], and AVOD [65].

7) Video object detectors: Examples of these methods include Tube-CNN [66], SlowFast [67], and SiamRPN [68].

8) Few-shot object detectors: Examples of these methods include Meta R-CNN [69], FSOD [70], and Few-Shot Object Detection via Feature Reweighting [71].

III. OBJECT DETECTION TECHNIQUES AND DEVELOPMENT TIMELINE

Object detection is an active area of research and development, with many new methods and techniques emerging on a regular basis. This section of the paper covers the development timeline of Object Detection techniques briefly to give the pig picture needed when studying this important and widely used field of computer vision. a brief overview of some of the major milestones in the development of object detection methods over the past few decades could be listed through the time lime as:

- 1990s: The first object detection methods based on hand- crafted features, such as Histograms of Oriented Gradients (HOG) and Haar-like features, were introduced. These methods used traditional machine learning algo- rithms, such as Support Vector Machines (SVM), to detect objects in images [29].
- 2000s: The use of deep neural networks for object detection began to gain popularity. The seminal work on this topic was the Viola Jones algorithm, which used a boosted cascade of simple classifiers to achieve realtime face detection [72]. In the later years of the decade, methods such as Deformable Part Models (DPM) and Fast R-CNN: a Fast Region-based Convolutional Network were introduced, which combined deep neural networks with traditional machine learning techniques to improve accuracy and speed [73] [42].

TABLE I
HISTORY OF SOME FAMOUS TRADITIONAL OBJECT DETECTION METHODS WITH PROS AND CONS

Method	Year	Pros	Cons	References
Feature-based methods	2001	Robust to occlusion. Can detect objects in cluttered environments.	Sensitive to noise. Computationally expensive.	[27]
	1999	Can detect objects of various sizes and orientations. Can handle occlusion and cluttered scenes.	Sensitive to image noise and illumination changes. Computational complexity is high.	[28]
Color-based methods	1999	Can detect objects based on color information.	Limited to objects with distinctive colors. Sensitive to changes in illumination.	[29]
Region-based methods	1998	Can detect objects of various sizes and shapes. Can handle occlusion and cluttered scenes.	Computationally expensive. Sensitive to image noise and illumination changes.	[30]
Scale-invariant methods	1998	Can detect objects at different scales.	Sensitive to image noise and illumination changes. Computationally expensive.	[31]
Contour-based methods	1995	Can detect objects with well-defined contours. Computationally efficient.	Sensitive to noise. Can fail to detect objects with poorly defined contours.	[32]
	1994	Can handle objects with smooth boundaries. Can handle occlusion and cluttered scenes.	Sensitive to noise and poor edge detection in cluttered scenes. Computationally expensive.	[33]
Scale-space method	1987	Can detect objects at different scales.	Computationally expensive.	[34]

Template matching-based methods	2000	Simple and easy to implement. Can detect objects in cluttered environments.	Sensitive to changes in lighting and viewpoint. Limited to detecting objects with a similar appearance to the template image.	[35]
	1983	Simple and efficient. Can detect objects in cluttered scenes.	Limited to specific objects and can't handle variations in scale and orientation.	[36]
Texture-based methods	1983	Can detect objects based on texture information.	Limited to objects with distinctive textures. Sensitive to changes in illumination.	[37]
Motion-based methods	1980	Can detect moving objects in video sequences.	Limited to moving objects. Sensitive to camera motion.	[38]
Correlation-based methods	1980	Can detect objects in cluttered scenes.	Sensitive to image noise and illumination changes. Limited to specific objects.	[39]
Edge-based methods	1986	Can detect objects with well-defined edges. Computationally efficient.	Sensitive to noise. Can fail to detect objects with poorly defined edges.	[40]
	1979	Can detect objects with sharp boundaries. Computationally efficient.	Sensitive to noise and poor edge detection in cluttered scenes.	[40]
Frequency domain-based methods	1987	Can detect objects based on their frequency content and handle occlusion and cluttered environments. Can handle noisy images and detect periodic patterns objects.	Computationally expensive. Sensitive to changes in lighting and viewpoint. Limited to specific objects. Can't handle non-periodic patterns.	[41]

http://dx.doi.org/10.21622/ACE.2023.03.2.064

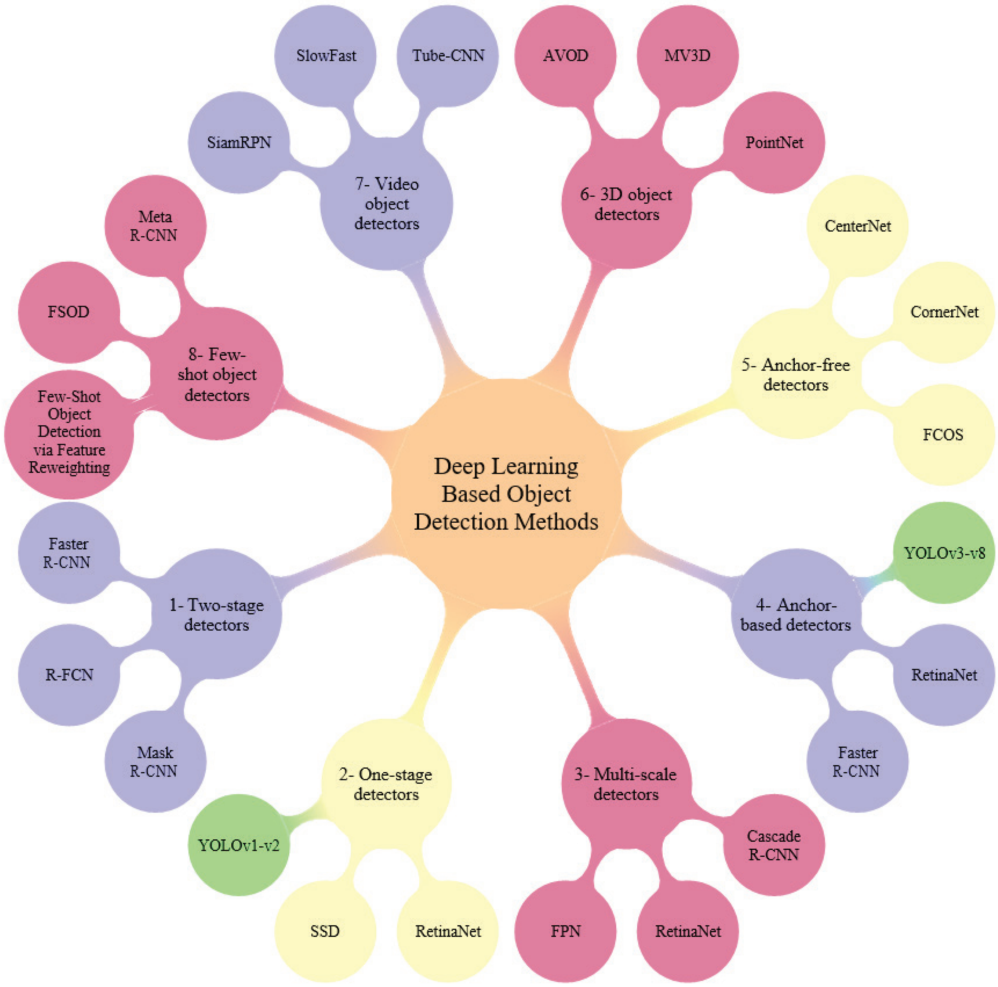

Fig. 2. Deep Learning Based Object Detection Methods.

- 2010s: This decade saw the rise of two-stage object detection methods, such as Faster R-CNN: Region Proposal Networks (RPN) and Mask R-CNN, which separated object proposal generation from object classification. Faster R-CNN: Region Proposal Networks (RPN) are used to generate region proposals, and a Fast R-CNN network is used for object detection. These methods achieved state-of-the-art performance on benchmark datasets such as COCO and Pascal VOC [43]. Mask R-CNN: adds a branch to Faster R-CNN for predicting object masks in addition to bounding boxes and class labels [60]. In the later years of the decade, single-stage methods such as YOLO (You Only Look Once) and SSD (Single Shot Detector) were introduced, which achieved real-time performance by directly predicting object bounding boxes and class labels in a single pass of the network. [44] [46].

- 2020s: In recent years, the focus of object detection research has shifted towards improving efficiency and robustness. Methods such as EfficientDet (starting by EfficientDet-D0 through EfficientDet-D7 models) [74] and DETR (DEtection TRansformer) have achieved state of the art performance on benchmark datasets while reducing the computational cost and improving the generalization ability of the models. Additionally, there has been increasing interest in using self-supervised and unsupervised learning methods for object detection, as these approaches can leverage large amounts of unlabeled data to improve performance. In this decade, many algorithms have been introduced to be a normal result for the tedious and continues work to achieve better and better performance, For example:

http://dx.doi.org/10.21622/ACE.2023.03.2.064

- YOLOv4: an improvement over YOLOv3, achieving high accuracy with faster inference time [75].

- YOLOv5: is a family of object detection models developed by Ultralytics that builds on the success of the previous YOLO models. YOLOv5 uses a novel neural network architecture and training techniques to achieve state-of-the-art performance on several benchmark datasets [45].

- SpineNet: a family of object detectors that use a new architecture to achieve high accuracy with efficient computation and memory usage. It is developed by Facebook AI Research that uses a novel neural network architecture to achieve high accuracy while using fewer computational resources. It achieves state-of-the-art performance at 2020 on several benchmark datasets [49].

- DETR(Detection Transformer): is a novel object detection architecture that uses a transformer-based neural network to perform object detection. It achieves state of the art performance at 2020 on several benchmark datasets while using a simple and unified approach [50].

- Deformable DETR: is an extension of the DETR architecture in 2021 that uses deformable convolutional layers to perform feature extraction. It achieves state- of-the-art performance on several benchmark datasets while improving the detection of small objects [51].

- Detectron2: is a popular open-source object detection framework developed by Facebook AI Research (FAIR). It builds on the success of the original Detectron framework and offers a modular and flexible platform for developing and training state-of-the-art object detection models at 2020. Detectron2 supports a wide range of model architectures and can be easily customized for different tasks and it provides a userfriendly interface and extensive documentation. On the other hand, it requires significant computational resources for training and inference and it may have a steeper learning curve for users unfamiliar with the PyTorch framework [76] [77].

- Sparse R-CNN: is an object detection method that uses a sparse convolutional neural network to perform feature extraction. It achieves state-of-the-art performance on several benchmark datasets at 2021 while using fewer computational resources compared to other methods. It achieves high accuracy with fewer computational resources and can be fine tuned for specific tasks. Furthermore, it is highly modular and it can be easily customized for different tasks. On the other hand, it may require specialized hardware to achieve real-time performance and it can be sensitive to the choice of hyperparameters [78].

TABLE II
HISTORY OF SOME FAMOUS DEEP LEARNING-BASED OBJECT DETECTION METHODS WITH PROS AND CONS

Method	Year	Pros	Cons	References
R-CNN (Region-based Convolutional Neural Networks)	2014	Good detection accuracy, high recall, good for complex images.	Slow training and testing, requires selective search for region proposals, not real-time.	[11]
Fast R-CNN	2015	Faster than R-CNN, end- to-end training, good detection accuracy.	Still requires region proposals, not realtime.	[42]
Faster R-CNN	2015	Faster than R-CNN and Fast R-CNN, end-to-end training, good detection accuracy.	Still requires region proposals, not realtime.	[43]

YOLO (You Only Look Once)	2016	Real-time, good detection accuracy, and end-to-end training.	Can miss small objects, and less accurate than other methods on complex images.	[44]
YOLOv5	2020	Achieves high accuracy with faster inference time compared to previous YOLO models. Has a lightweight architecture that requires fewer computational resources. Can be fine-tuned for specific tasks.	May not be as accurate as some other methods. May require specialized hardware to achieve realtime performance.	[45]
SSD (Single Shot MultiBox Detector)	2016	Real-time, good detection accuracy,and end-to-end training.	Can miss small objects, less accurate than other methods on complex images.	[46]
RetinaNet	2017	Good detection accuracy for small objects, less prone to false negatives.	Can miss large objects, slower than some other methods.	[47]
EfficientDet	2020	Achieves high accuracy with less computational resources. Has a good balance between accuracy and efficiency. Can be fine tuned for specific tasks.	May require specialized hardware to achieve real- time performance. Can be sensitive to hyperparameters and initialization.	[48]
SpineNet	2020	Achieves high accuracy with less computational resources. Has a good balance between accuracy and efficiency. Is highly modular and can be easily customized for different tasks.	May require specialized hardware to achieve real- time performance. Can be sensitive to hyperparameters and initialization.	[49]
DETR (DEtection TRansformer)	2020	No need for region pro-posals, end-to-end training, good accuracy for small objects, efficient.	Less accurate for large objects, slower than some other methods.	[50]
Deformable DETR	2021	Achieves high accuracy with improved detection of small objects. Can detect objects of different sizes and shapes. Is highly modular and can be easily customized for different tasks.	May require more computational resources compared to other methods. Can be sensitive to the choice of hyperparameters.	[51]
DETR (DEtection TRansformer)	2020	No need for region pro-posals, end-to-end train- ing, good accuracy for small objects, efficient.	Less accurate for large objects, slower than some other methods.	[50]

In summary, there has been significant progress and improvement in object detection over the past few years. The state-of-the-art in object detection has been rapidly advancing in recent years, with new methods continually emerging that achieve better accuracy, efficiency, and robustness. Overall, these neural network-based methods of object detection represent significant advancements in the field and offer a range of pros and cons depending on the specific application and use case, As shown in Table II.

Furthermore, there is an ongoing process of discovery and innovation in the field of object detection, with researchers and developers constantly exploring new approaches and techniques. Through these techniques:

- Accuracy: The ability of an object detection system to correctly identify objects in an image or video is extremely increased.
- Efficiency: The speed and computational resources required to perform object detection
- Robustness: The ability of an object detection system to perform well under various conditions, such as changes in lighting, camera angle, or object orientation.

The ability of an object detection system to correctly identify objects in an image or video is extremely increased, and new methods and techniques are being developed that are able to achieve these goals more effectively than previous approaches. In addition, a number of benchmark datasets have been instrumental in advancing the field of object detection. These include the Caltech [79], KITTI [80], ImageNet [81], PASCAL VOC [82], MS COCO [83], and Open Images V5 [84] datasets. Recently, a new drone-based dataset was introduced as part of the ECCV VisDrone 2018 contest [85]. This dataset is comprised of a significant amount of images and videos captured from a drone platform [86].

IV. YOLO (YOU ONLY LOOK ONCE) OBJECT DETECTION ALGORITHM: HISTORY AND STATE OF THE ART

YOLO (You Only Look Once) is a popular object detection algorithm that has undergone several iterations since its initial release. Furthermore, it is concluded from the previous section with its subsections and from [74], that the primary singlestage detection networks are:

- Single Shot multiBox Detector (SSD) [87].
- EfficientDet [88].
- You Only Look Once (YOLO) series of networks [89]. Although SSD has good detection accuracy, it lacks sufficient low-level feature convolution layers, resulting in inadequate feature extraction, which makes it less sensitive to small target detection. EfficientDet-D0 through EfficientDet-D7 models can achieve higher accuracy, but this comes at performane weakness factors such that the cost of higher memory consumption and slower inference. In [90], YOLOv5 was used for some object detection applications (palm tree detection) using UAV images and was quantitatively compared with main- stream networks such as YOLOv3, YOLOv4, and SSD300, with YOLOv5 demonstrating the best accuracy. The YOLOv5 network adjusts the perceptual field size and enhances the feature extraction ability, indicating its potential for detection in regions with high canopy coverage [74]. Here are the key differences between each version:

1) YOLOv1: This was the first version of YOLO released in 2015. It used a single deep neural network to perform both object classification and localization in a single stage. It achieved competitive accuracy and speed on the PASCAL VOC 2007 detection dataset. However, it suffered from low recall due to the “grid cell” structure, which made it difficult to detect small objects [44].

2) YOLOv2: Released in 2016, YOLOv2 addressed some of the limitations of the original version. It introduced anchor boxes, which improved the detection of small objects, and a multi-scale feature extraction network, which improved performance on objects of different sizes. It also incorporated batch normalization and residual connections for improved training and performance. furthermore, it addressed the shortcomings of the first version. It achieved a very good results on multiple detection datasets and improved the accuracy and speed of the algorithm [91].

3) YOLOv3: The third version of YOLO further improved the algorithm by

http://dx.doi.org/10.21622/ACE.2023.03.2.064

introducing various enhancements such as multi-scale prediction, feature pyramid networks, and improved training techniques. It achieved state-of-the-art results on multiple detection benchmarks and improved the accuracy, speed, and robustness of the algorithm. YOLOv3 is released in 2018, YOLOv3 introduced several improvements over YOLOv2. It increased the number of anchor boxes, improved the feature extraction network, and introduced skip connections to help detect small objects. It also introduced the concept of "darknet-53," a more complex feature extraction network that improved detection accuracy [61].

4) YOLOv4: Released in 2020, It introduced a number of new features, including the "CSPDarknet-53" archtecture, which improved feature extraction and training efficiency. It also introduced the "Mish" activation function, which outperforms traditional activation functions like ReLU. YOLOv4 also introduced a variety of training techniques, including self-adversarial training, Drop- Block regularization, and improved data augmentation. The fourth version of YOLO introduced a number of enhancements such as a CSPDarknet53 backbone, spatial attention, and dynamic anchor assignment, among others. It achieved fabulous results on multiple detection benchmarks and improved the accuracy, speed, and efficiency of the algorithm at 2020, as mentioned in [75].

5) YOLOv5: Released in 2020, YOLOv5 is an independent project by Ultralytics that is based on the YOLOv4 architecture. It introduced a number of improvements, including a redesigned backbone network, a "Swish" activation function, and improved anchor box placement. It also incorporated a new training pipeline that made it easier to train on custom datasets [92].

6) YOLO Nano: Released in 2020, YOLO Nano is a lightweight version of YOLO designed for resource-constrained devices. It achieves real-time performance on mobile devices with limited computing power by using a simplified architecture and fewer layers. It was presented at the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays [93].

7) YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x: Ultralytics has released several versions of YOLOv5 with varying levels of computational complexity. These versions have different backbone architectures and feature extraction networks, allowing users to choose the version that best suits their needs based on the available hardware resources. The fifth version of YOLO introduced a number of significant improvements such as a new architecture, self-attention, and a novel anchor-free object detection method. It achieved state-of-the-art results on multiple detection benchmarks and improved the accuracy, speed, and simplicity of the algorithm [92] [94] [95] [96] [97].

8) YOLOv6: YOLOv6 is verified to bring more improvements to the YOLO architecture. The sixth version of YOLO introduced various improvements such as custom architecture search, scaled-YOLOv4, and ensemble of models. It achieved results on multiple detection benchmarks and improved the accuracy, speed, and efficiency of the algorithm [98].

9) YOLOv7: The seventh version of YOLO introduced a trainable bag-of-freebies (BoF) module that can be added to any object detection architecture to improve its performance. It achieved state-of-the-art results on multiple detection benchmarks and improved the accuracy, speed, and efficiency of the algorithm [99].

10) YOLOv8: The eighth and most recent version of YOLO introduced various improvements such as deformable convolutional networks, Scaled-YOLOv5, and ensemble of models with different input resolutions. It achieved state-of-the-art results on multiple detection benchmarks and improved the accuracy, speed, and efficiency of the algorithm. The YOLOv8 network is utilized for addressing classification, object detection, and image segmentation challenges. These various approaches enable the identification of objects in

images or videos through distinct means. In addition to providing the object type and probability, the neural network for object detection also outputs the coordinates of the object on the image, including its x and y position, width, and height. This information is demonstrated in the second image. Moreover, object detection neural networks have the capability to identify multiple objects in an image and determine their respective bounding boxes [100].

Furthermore, it is noted that some of the versions mentioned above are not officially released by the original authors of YOLO and might have some variations or differences in their implementation.

V. YOLO 5,7, AND 8 DETECTION RUN COMPARISON FOR OBJECT DETECTION PROCESS OF VEHICLES AND INLAND VESSELS SCENCES

The YOLO model is currently the most widely used realtime object detector due to its lightweight network architecture, effective feature fusion methods, and more accurate detection results. Among the YOLO algorithm variants, YOLOv5 and YOLOv7 have gained significant acceptance in current usage.

YOLOv5 utilizes deep learning technology to achieve realtime and efficient object detection tasks. It improves upon YOLOv4 by enhancing the model structure, training strategy, and overall performance. YOLOv5 adopts the CSP (Cross- Stage Partial) network structure, which effectively reduces redundant calculations and enhances computational efficiency. However, YOLOv5 still has room for improvement in detecting small objects and dense object scenarios, as well as in handling complex situations like occlusion and pose changes. YOLOv7 introduces a novel training strategy, at that time (2022), called Trainable Bag of Freebies (TBoF) to enhance the performance of real-time object detectors. TBoF incorporates various trainable tricks, including data augmentation and MixUp, which significantly improve the accuracy and generalization ability of object detection. However, YOLOv7 is constrained by training data, model structure, and hyperparameters, leading to performance degradation in some cases. Additionally, the proposed method requires more computational resources and training time to achieve optimal performance [101]. Figure V, shows the YOLO Publications Timeline before YOLOv8.

In this section, a comprehensive evaluation is conducted on selected versions of the YOLO platform applied to a specific selected dataset sample of images. A subset of representative images is carefully chosen to determine the most effective YOLO versions for efficient object detection. The selection criteria consider the diversity of scenes, encompassing challenging scenarios where objects may be combined or difficult to detect. The dataset is thoughtfully balanced, featuring waterway scenes for inland vessels, street traffic views capturing vehicles, pedestrians, birds, horses, and traffic lights, among others.

The results of the runs reveal distinct identification outcomes for each object in all selected versions of YOLO. A total of thirty-three (33) images, as illustrated in Figure V, are carefully curated, comprising 25 images related to inland vessels scenes and 8 particularly challenging images representing street traffic views with multiple objects. The subsequent part of this section delves into the detailed analysis of the object detection results, highlighting the capabilities of each selected YOLO version in various challenging conditions.

The selected dataset images, as shown in figure V, are selected as:

- 25 images for inland vessels related images.
- 8 (tricky) images are selected for the other (street traffic views for vehicle, pedestrians, one for a bird, one for a horse, traffic lights, and etc.)

The rest part of this section will cover the different output and shows the ability

Volume 3, Issue 2, December 2023- ISSN 2735-5985
http://dx.doi.org/10.21622/ACE.2023.03.2.064

of each version of the selected ones to detect object in a various set of different conditions.

A. COMMON OBJECTS IN CONTEXT DATASET (COCO)

The Common Objects in Context (COCO) dataset has emerged as a significant milestone in the field of computer vision, specifically for object recognition and detection tasks [83]. With its vast collection of images and meticulously annotated objects, the COCO dataset provides researchers and developers with a standardized benchmark to evaluate and advance the performance of object detection algorithms [102]. By offering a diverse range of object categories and complex scenes, COCO has become an invaluable resource for training and testing state-of-the-art models, pushing the boundaries of object recognition and detection capabilities.

Object recognition and detection form the fundamental building blocks of many computer vision applications, including autonomous driving, surveillance systems, robotics, and augmented reality. The ability to accurately identify and localize objects within an image is crucial for understanding visual scenes and enabling intelligent decision-making. However, this task is inherently challenging due to variations in object appearance, scale, occlusion, and cluttered backgrounds.

The COCO dataset addresses these challenges by providing a large-scale and diverse collection of images spanning 80 object categories, as shown in table IV [83].

These categories encompass a wide range of everyday objects such as people, animals, vehicles, household items, and more. The dataset consists of over 200,000 images from complex real-world scenes, covering a diverse set of visual contexts and capturing a wide variety of object instances. Each image is meticulously annotated with multiple objects bounding boxes, segmentations, and corresponding category labels, enabling fine-grained analysis and evaluation of object detection algorithms [83].

The COCO dataset is a comprehensive and extensive resource for object detection, segmentation, and captioning tasks. It encompasses a range of notable features, including [83]:

1) Object segmentation: The dataset provides precise delineation of objects through segmentation, allowing for pixel-level ground truth information.

2) Recognition in context: COCO captures objects within their surrounding scenes, enabling recognition in diverse visual contexts and promoting a better understanding of object interactions.

3) Super pixel stuff segmentation: The dataset includes detailed annotations for stuff segmentation, providing insights into the distribution and boundaries of regions like sky, water, and vegetation.

4) Scale and diversity: COCO comprises a substantial collection of 330,000 images, encompassing a wide variety of scenes and object instances. This large-scale nature allows for robust training and evaluation of models.

5) Abundance of object instances: With approximately 1.5 million annotated object instances, the COCO dataset offers a rich variety of examples for each object category, facilitating comprehensive analysis and algorithm development.

6) Extensive object and stuff categories: The dataset covers 80 object categories, including people, animals, vehicles, and household items, providing a broad range of objects to detect and recognize. Additionally, it includes 91 stuff categories, capturing various contextual elements.

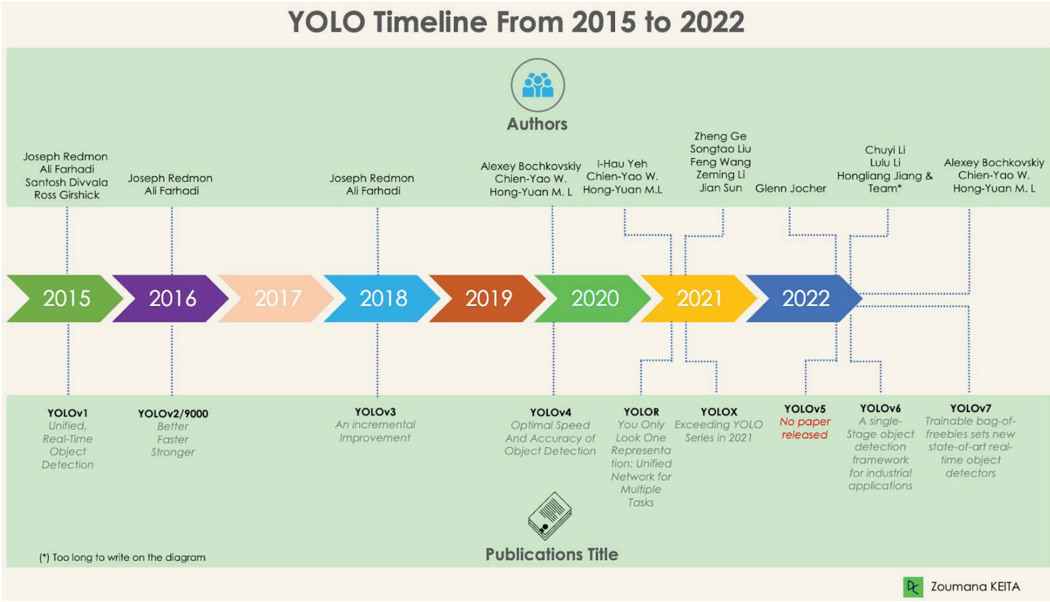

Fig. 3. YOLO Publications and Authors Timeline from 2015 to 2022 (Zoumana KEITA)

7) Caption annotations: Each image in the COCO dataset is associated with five captions, enabling research and development in image captioning and language understanding tasks.

8) Key points annotations: COCO provides annotations for key points, specifically for approximately 250,000 people, facilitating the development and evaluation of pose estimation algorithms.

The combination of these features makes the COCO dataset a highly valuable resource for advancing research in computer vision, enabling the development and evaluation of state of the art models across multiple tasks, including object detection, segmentation, captioning, and pose estimation. One of the distinguishing features of the COCO dataset is its emphasis on accurately localizing objects of interest. In addition to category labels, each annotated object is precisely delineated using segmentation masks, providing pixel-level ground truth information [83]. This level of detail allows researchers to explore advanced detection techniques that go beyond simple bounding box estimation, enabling more precise object localization and instance segmentation.

The availability of such a rich and comprehensive dataset has fostered significant advancements in object recognition and detection. Researchers and developers have leveraged the COCO dataset to train and benchmark state-of-the-art models, resulting in remarkable progress in object detection accuracy and efficiency [103] [104]. The dataset's widespread adoption has fueled the development of sophisticated algorithms, including deep learning approaches, which have demonstrated exceptional performance in detecting and recognizing objects across a wide variety of challenging scenarios [105] [106].

Furthermore, the COCO dataset has facilitated the development of robust and generalizable object detection frameworks. By training models on COCO, researchers can leverage the dataset's diversity to enhance the models' ability to handle complex scenes, occlusion, and small object instances [105]. The evaluation metrics provided by COCO, such as aver age precision (AP) and intersection over union (IoU), offer standardized benchmarks for comparing the performance of different object detection algorithms, promoting fair and objective evaluations [102]. The COCO dataset has revolutionized the field of object recognition and detection by providing a comprehensive benchmark for training, testing, and evaluating algorithms. Its large-scale collection of images, diverse object categories, precise annotations, and detailed evaluation metrics have accelerated progress in the development of state-

http://dx.doi.org/10.21622/ACE.2023.03.2.064

of-the- art models. As computer vision applications continue to evolve and demand increasingly accurate and robust object detection capabilities, the COCO dataset will undoubtedly remain a vital resource in driving advancements and pushing the boundaries of object recognition technology [83].

Fig. 4. Dataset samples for vehicles with various behavior patterns, including mixed traffic situations and inland waterway vessels

http://dx.doi.org/10.21622/ACE.2023.03.2.064

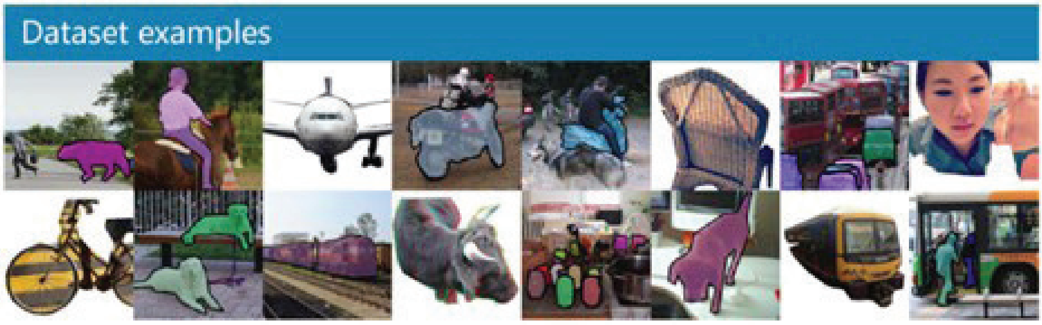

Fig. 5. COCO Objects Examples [83]

B. APPLYING YOLOV5 X, L, S, AND M MODELS FOR THE SELECTED IMAGES SET.

YOLOv5 provides a range of options with four available models: s, m, l, and x. Each model offers distinct levels of detection accuracy and performance, as illustrated in figure V-B Although, there are more models for YOLOv5 but they are dealing with the images width of 1280 pixels. All considered models here to be applied is 460 pixels in size. YOLOv5s,

YOLOv5m, YOLOv5l, and YOLOv5x is used here with the selected 33 images set to test the ability of the system to detect the different objects pretrained in COCO dataset 80 classes.

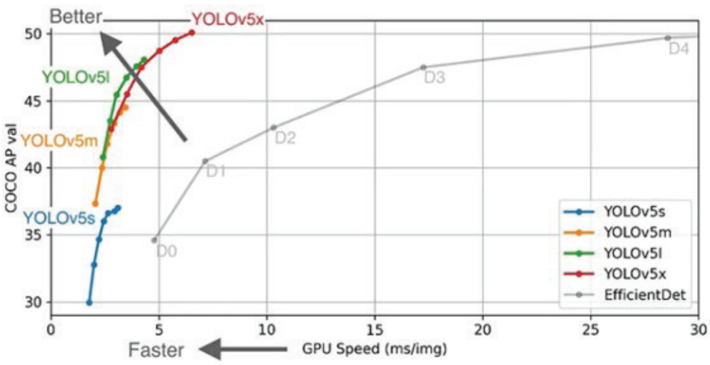

Fig. 6. YOLOv5 Four Models s, m, l, and x Accuracy and Performance (Source: https://github.com/ultralytics/yolov5)

This subsection here aims to analyze and compare the differences between s, m, l, and x models of YOLOv5, employing the COCO 80 classes model. The resulting output provides valuable insights into the variations and characteristics exhibited by these weight models in terms of their object detection capabilities. Detecting of objects in thirty-three (33) images by YOLOv5 models x and l are applied. The utilization of YOLOv5 models x and l yields the capability to accurately detect objects within the sample of the 33 images. Employing YOLOv5 models x and l brings forth a multitude of advantages tailored to precise object detection within the realm of 33 images. Model x excels in achieving faster inference times, allowing for real-time detection in dynamic scenarios. Its optimized architecture efficiently balances accuracy and speed, making it ideal for applications requiring swift decision-making. On the other hand, model l boasts exceptional accuracy in object localization and recognition. Its deeper architecture and refined feature extraction enable it to discern intricate details within the images, ensuring a high level of detection precision, especially in complex scenes with overlapping objects or varying scales. By harnessing the strengths of both models x and l, the overall detection process becomes comprehensive and robust, encompassing scenarios that demand both speed and precision in object identification within the set of the 33 images

Detecting of objects in 33 images by YOLOv5 models sand m are applied. YOLOv5 model s is designed with a focus on speed and efficiency. Its architecture is optimized for rapid inference times, enabling real-time object detection in dynamic scenarios. This advantage is particularly valuable in applications requiring quick decision-making and tracking, such as video surveillance and robotics. Model s efficiently balances speed and accuracy, making it well-suited for scenarios where timely detection is essential. On the other hand, YOLOv5 model m strikes a balance between speed and accuracy. With a slightly deeper architecture and enhanced feature extraction, it excels in accurately localizing and recognizing objects within images. This accuracy advantage is especially useful in scenarios with complex scenes, overlapping objects, or varying object scales. Model m's versatility makes it suitable for a wide range of applications, including autonomous vehicles, industrial automation, and quality control. By leveraging the strengths of both models s and m, the overall object detection process is enriched, catering to scenarios that demand different priorities in terms of speed and accuracy within the specified previous sample set of 33 images

The remaining part of this subsection displays several images generated by four YOLOv5 models: x, l, s, and m. The detected objects in these images are enclosed within bounding boxes and labeled with the class assigned by the respective model. Additionally, an F1 score will be calculated on this sample test to demonstrate the performance and accuracy of these models.

Figures 7 (a), (b), (c), and (d) display the output achieved of applying YOLOv5 model x, l, s, and m respectively, and show true positive detection instant for one bird and false positive for the other one in some cases. Furthermore, the existence of showing multiple labels for the same object by two different classes, one is true which is "car" and the other one is false as well and it is "boat". The existence of water near to the car do this sort of ambiguity and the presence of some wood bars in front of the car, leads to detect the wood as bench in model m, as shown in 7 (d), and it leads falsely the s model to detect the car as a train on the other hand, as Figure 7 (c) illustrates.

A bird in the sky is detected only in one model of YOLOv5 which is x model as appears in 8 (a). but the hand back on the person's shoulder is detected in model x and l as 8 shows in (a) and (b) respectively. True positive person detection is achieved in all models of 8 as well. False Detection of a boat is only achieved in this image when model x applies as 8 (a) illustrates. Model x at this image is most accurate one in this case to detect person, bird, backpack objects very professionally as shown in 8 (a) and (b).

Bridge pillar is detected as boat, as Figures 9 (b) and (c) show. The same vessel is detected as one boat in Figures 9 (c) and (d), but it is counted as two boats in (a) and (b) respectively. Yellow buoy is not detected at all and needs to be added in the training model to be differentiated to the boat. Figures 10 (a), (b), (c), and (d) show that white car on top of the vessel is not detected except the case (a). While the persons in the images are detected accurately.

The trailer part contains the cabin is only detected from the vessel in model x of Figure 11 (a). The person on the vessel surface is not detected in all models except of Model m as Figure 11 (d) illustrates. Building with the boat is delt as one object, as shown in Figure 11 (c). Model x and m are detecting a back shore building as a boat, as shown in Figure 11 (a), (d) respectively. A vessel is sailing in the opposite direction and delt as multiple objects in (b), (d) and as boat partially in (a) as Figure 11 shows.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

Fig. 7. YOLOv5 detection run for selected image 1.

Fig. 8. YOLOv5 detection run for selected image 2.

Fig. 9. YOLOv5 detection run for selected image 3.

C. APPLYING YOLOV7 X MODEL FOR THE SELECTED IMAGES SET.

YOLOv7 demonstrates superior performance in 2022 in terms of both speed and accuracy across a range of frame rates, from 5 FPS to 160 FPS. It achieves the highest accuracy, with an average precision (AP) of 56.8%, among all real-time

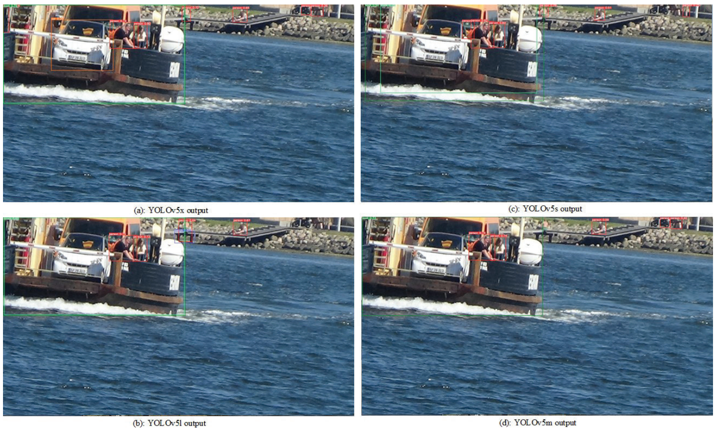

Fig. 10. YOLOv5 detection run for selected image 4.

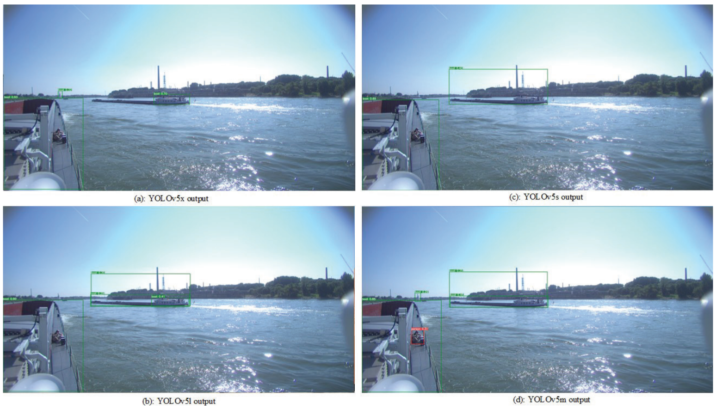

Fig. 11. YOLOv5 detection run for selected image 5.

Deformable DETR, DINO-5scale-R50, ViT-Adapter-B, and several others, in terms of both speed and accuracy as 12 shows Only one model YOLOv7x is used here and as with the previous subsection, limitations of object detection are applied here too with YOLOv7x. The run process to detect objects in 33 selected data set is also applied here by using YOLOv7x. One of the main YOLOv7x model advantagesis could be considered as one of the fastest object detection algorithms available. It can detect objects in real time, making it ideal for applications such as autonomous driving and robotics. YOLOv7x is also very accurate, achieving state-of-the-art results on a variety of object detection benchmarks. This makes it a good choice for applications where accuracy is important, such as medical imaging and security. YOLOv7x has fewer parameters than previous versions of YOLO, making it easier to train and deploy on resource-constrained devices. This makes it a good choice for applications where speed and accuracy are both important, such as mobile phone cameras.

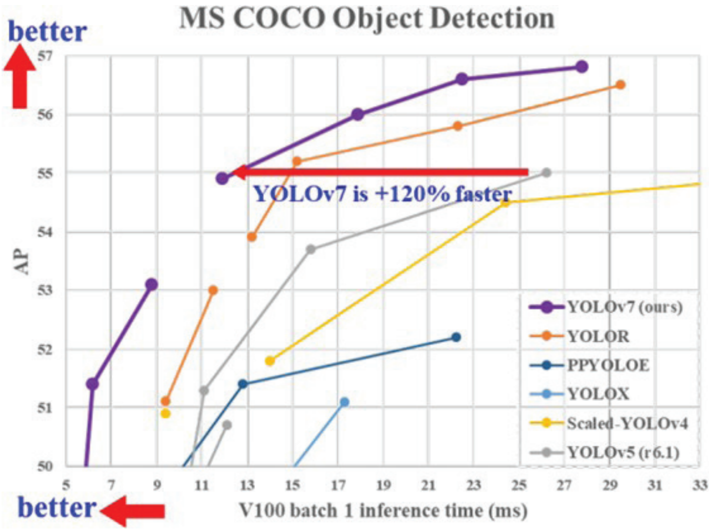

Fig. 12. YOLOv7 in comparison with other real-time object detectors (https://github.com/WongKinYiu/yolov7).

YOLOv7x can detect objects of different sizes at the same time. This makes it a good choice for applications where objects can vary in size, such as traffic or surveillance. YOLOv7x is robust to noise and occlusion. This makes it a good choice for applications where objects may be partially obscured, such as in poor lighting conditions or with moving objects. YOLOv7x achieves its speed by using a simplified model architecture with fewer parameters. This means that the model is less computationally expensive to train and run. YOLOv7x also uses a more efficient implementation of the convolution operations, which further improves its speed

In terms of accuray, YOLOv7x achieves its accuracy by using a larger training dataset with more diverse objects. This helps the model to learn to identify a wider variety of objects. YOLOv7x also uses a more powerful loss function that penalizes inaccurate predictions. This helps the model to learn to make more accurate predictions. YOLOv7x has fewer parameters than previous versions of YOLO because it uses a simplified model architecture. This makes it easier to train and deploy on resource-constrained devices, such as mobile phones or embedded systems. YOLOv7x can detect objects of different sizes at the same time by using a technique called anchor boxes. Anchor boxes are a set of pre defined boxes that are used to represent the possible sizes and aspect ratios of objects. YOLOv7x predicts the probability of each object being present in each anchor box, as well as the coordinates of the object's bounding box. YOLOv7x is robust to noise and occlusion because it uses a technique called spatial pyramid pooling. Spatial pyramid pooling divides the image into a grid of cells, and then averages the predictions from each cell. This helps to reduce the impact of noise and occlusion on the predictions Applying this model on many images is done and Figure 13 shows one image from the selected set and also the same limitations apply here as well.

Detected images can differ based on model accuracy and previous training process: The accuracy of an object detection model can vary depending on various factors, including the training data, model architecture, and optimization techniques used. A model that has been trained on a large, diverse, and well-annotated dataset is likely to perform better in detecting objects compared to a model trained on a smaller or less diverse dataset. Additionally, the training process itself, such as the choice of hyperparameters and training duration, can impact the accuracy of the model.

Fig. 13. YOLOv7 Image Run.

Objects detected multiple times with different classes: It is not uncommon for an object detection model to produce multiple bounding boxes for the same object instance, especially if the object is large or has complex geometry. However, if the model assigns different classes to these multiple detections of the same object, it could indicate issues with the training process or the quality of the training data. In such cases, reevaluating the training pipeline and dataset annotations may help improve the model's performance.

Misclassifying buoys as boats: Object detection models are trained on large datasets containing various object classes. If a model misclassifies buoys as boats, it might indicate a lack of specific training examples for buoys or similarities between the visual features of buoys and boats. Adding more diverse training examples of buoys and refining the training process can help address this issue.

Need for adding different classes and training the system: If you want the system to detect objects from rivers, waterways, and streets, you would need to include specific classes for those objects during the training process. This would require augmenting the training dataset with relevant images and annotations for objects like riverbanks, bridges, street signs, etc. By training the system on a comprehensive dataset with a wide range of classes, you can enhance its ability to detect and classify objects accurately in different environments.

Based on the results obtained from running the four YOLOv5 models on each image of the sample set in the previous subsection, and after contrast them with the performance of the other x model of YOLOv7, several observations can be drawn:

- The accuracy of detected objects in an image can vary based on the model's accuracy and the training process it underwent.
- In certain cases, an object detection model may detect the same object multiple times but assign different classes to those detections.
- For instance, buoys might be misclassified as boats instead of being recognized as buoys.
- To enhance the system's capability to detect objects from rivers, waterways, and streets, it is necessary to add and train the model in different classes, enabling it to identify various objects in these environments.
- In certain models, static objects like cranes may be included within the bounding box of a boat, while in other cases, they may not be included.
- Sometimes, a single long boat may be treated as a single object, while in other instances, it may be detected as two separate boats.
- Treating a part of an object as a complete object, such as considering a partial boat as a whole boat, is a problem that needs to be addressed.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

- Some objects may be misidentified due to misleading neighboring objects. For example, a car with pieces of wood in front might be incorrectly classified as a train, boat, or another car in different runs of the model.

From all these outcomes and other results, it is clear that object detection models can have varying performance depending on their architecture, training data, and optimization techniques. Addressing these issues may involve refining the training process, augmenting the dataset with diverse examples, and finetuning the model to improve its accuracy and robustness in detecting and classifying objects correctly in both environments, driverless vehicles, and pedestrian streets and for waterway inland vessels, as well.

IV. YOLO8 APPLIED TRAINING AND DETECTING USING COMBINED DATASET OBJECTS FOR DRIVERLESS VEHICLES AND INLAND VESSELS

A. YOLO8 FEATURES AND NETWORK STRUCTURE

The latest iteration in the YOLO model series, YOLOv8, was recently introduced by Ultralytics. Although a reviewed paper is yet to be published, examination of the repository reveals several notable features that distinguish it from other object detection models. In terms of architecture, as shown in Figure 15, YOLOv8 incorporates significant changes, particularly in how it receives and analyzes visual data. Unlike previous YOLO models like YOLOv4, YOLOv8 adopts an anchor- free approach. This is like other variations in the YOLO model series, such as YOLOX, which aims to streamline performance while maintaining high accuracy. Empirically, the best anchor- free approaches have demonstrated comparable or improved performance. However, from a theoretical standpoint, there are certain trade-offs. Anchorfree approaches offer greater flexibility in object detection as they directly identify objects without relying on preset anchors, which can be biased based on previous training and fail to generalize well to new data. However, this flexibility may also lead to biased and misleading predictions that lack the logical foundation inherent in the more traditional process of human object detection and observation.

YOLOv8 implements Anchor-Free instead of Anchor-Based object detection. It employs a dynamic Task Aligned Assigner for the matching strategy, calculating the alignment degree of Anchor-level for each instance. YOLOv8 achieves better accuracy than YOLOv5, making it the most accurate detector to date.

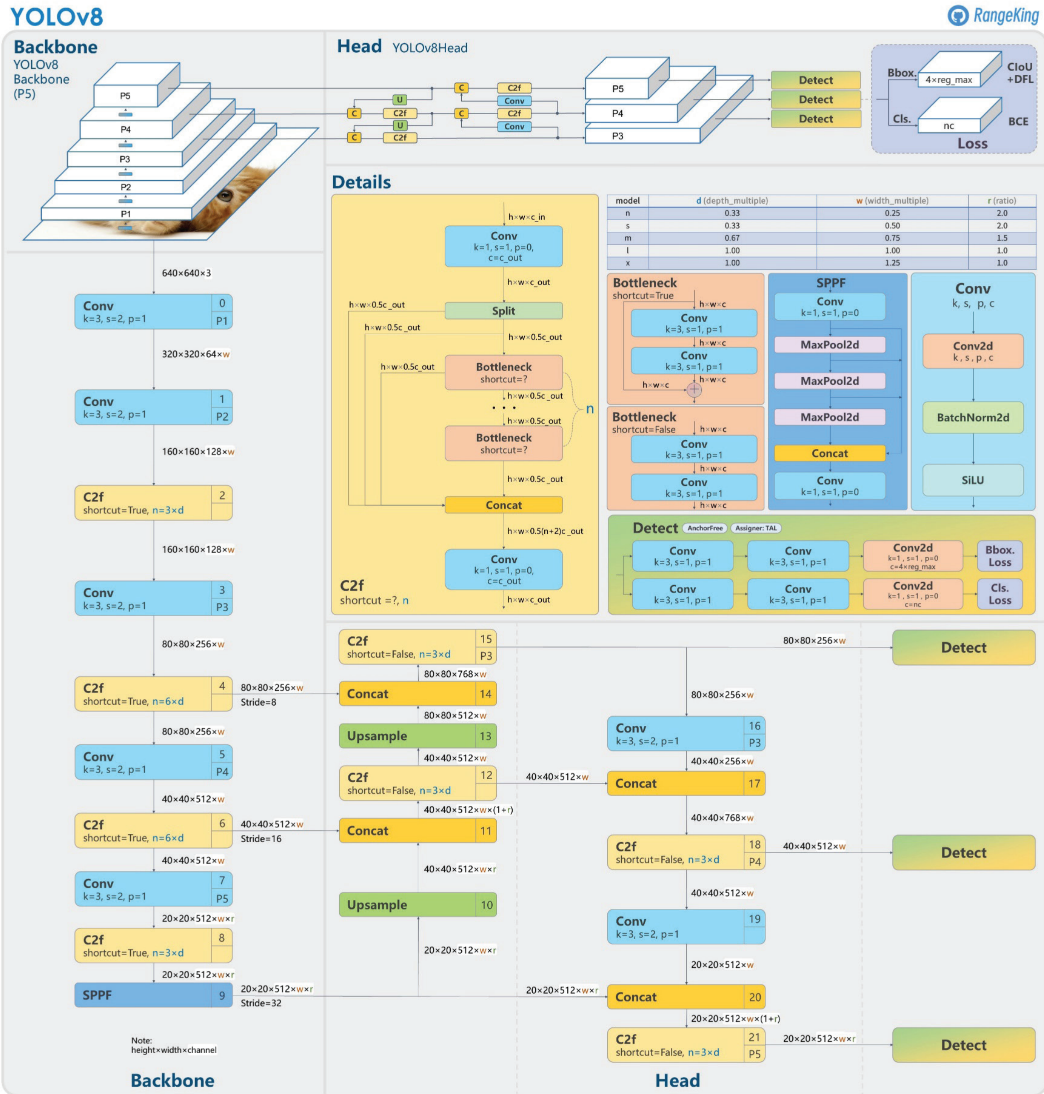

Fig. 14. YOLOv8 Architecture, visualization made (GitHub user Range King)

One key feature of YOLOv8 is its extensibility. It is designed to seamlessly work with all YOLO versions and allows researchers to switch between them, facilitating performance comparisons. Therefore, YOLOv8 was selected as the baseline version used in work implementation.

Furthermore, YOLOv8 benefits from the extensive community support it has garnered, which has contributed to its popularity and widespread usage. This community involvement has facilitated empirical investigations into more effective training schedules and methods. For instance, YOLOv8 does not ad- here to the same training strategy throughout the entire training process. One notable example is the mosaic augmentation, which stitches together images to train the model to detect objects with varying combinations and locations. However, it has been observed that employing this augmentation towards the end of the training process can degrade performance. Consequently, YOLOv8 employs a carefully selected training setup based on empirical experimentation to achieve optimal results.

YOLOv8 offers support for all versions of YOLO but with a focus on speed, accuracy, and user-friendliness, as shown by 15. YOLOv8 presents itself as an exceptional option for addressing diverse requirements in object detection and tracking, instance segmentation, image classification, and pose estimation tasks. and allows seamless switching between different versions, providing researchers with flexibility in their experiments. Furthermore, YOLOv8 is compatible with various hardware platforms (CPU-GPU), enhancing its versa- tility.

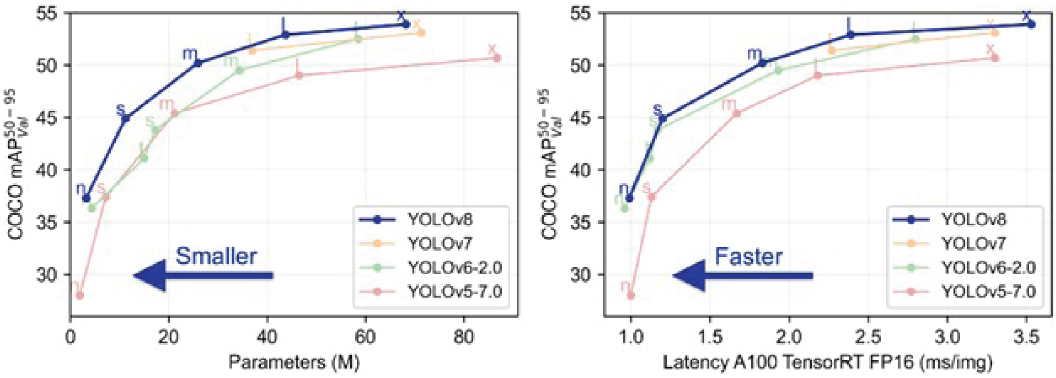

Fig. 15. YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model (https://github.com/ultralytics)

YOLOv8 represents a collection of neural network models developed and trained using PyTorch, which have been exported as files with the .pt extension. These models can be categorized into three types: Classification, Detection, and Segmentation, each serving a distinct purpose. Additionally, there are five models available for each type, differing in size, as shown in Table III.

TABLE III
YOLO8 Three types of models and 5 models of different sizes for each type

Classification	Detection	Segmentation	Kind
yolov8n-cls.pt	yolov8n.pt	yolov8n-seg.pt	Nano
yolov8s-cls.pt	yolov8s.pt	yolov8s-seg.pt	Small
yolov8m-cls.pt	yolov8m.pt	yolov8m-seg.pt	Medium
yolov8l-cls.pt	yolov8l.pt	yolov8l-seg.pt	Large
yolov8x-cls.pt	yolov8x.pt	yolov8x-seg.pt	Huge

B. YOLO8 BASED DETECTION RUN ON A SET OF IMAGES FOR INLAND VESSELS USING COCO 80 CLASSES DATASET READY TRAINED MODEL

In the subsequent sections of this paper, object detection will be performed using YOLOv8. The YOLOv8 models specifically designed for object detection come with pre-training on the COCO dataset, which includes a wide range of images covering 80 different categories. As a result, for the purpose of detecting objects on the street for driverless vehicles, the pre-trained models based on the COCO dataset fulfil the requirements, and there is no need for custom training. The pre-trained models listed in Table III are sufficient for this environment without requiring any additional training. By applying YOLOv8x on objects detecting for the same images, It is noticed that detection is much better than the previously tested versions, as Figure 16 shows, and the existing x model is enough to base on at the train phase to add new ability for detecting the new object.

YOLOv8 employs a novel architecture known as BiFPN (Bidirectional Feature Pyramid Network), which synergistcally amalgamates features from various network strata. This integration equips YOLOv8 with enhanced capabilities in managing objects of varying dimensions and aspect ratios. While YOLOv7 also employs BiFPN, YOLOv8 distinguishes itself by incorporating a higher count of channels within the Feature Pyramid Network (FPN) layers, affording it the ca- pacity to assimilate more intricate features. The adaptability of YOLOv8 is exemplified by its capacity to accommodate higher input resolutions during training compared to its predecessors. This elevated resolution empowers YOLOv8 to discern and precisely identify smaller objects. Though YOLOv7 exhibits similar adaptability, YOLOv8's superior performance in managing higher resolutions is evident

The adoption of the ResNet-50 backbone network in YOLOv8 represents a notable advancement over previous versions. This deeper and more expansive architecture en- ables YOLOv8 to distill complex features that contribute to refined object detection. Although YOLOv7 also integrates the ResNet-50 backbone, YOLOv8's augmented depth furnishes it with an advantage in feature extraction. YOLOv8 leverages anchor-free detection for bounding box prediction, a technique that surpasses predefined anchor boxes in terms of flexibility and precision. While YOLOv7 embraces anchor-free detection, YOLOv8 employs a superior approach termed CenterNet, showcasing enhanced accuracy

The expanded repertoire of detectable object classes in YOLOv8 stems from its utilization of a broader and more diverse training dataset. This augmentation enables YOLOv8 to surpass its predecessors, including YOLOv7, in terms of class diversity. YOLOv8 employs focal loss as a pivotal training mechanism, effectively mitigating the impact of misclassified objects and elevating accuracy. While YOLOv7 also integrates focal loss, YOLOv8 deploys an advanced version, demonstrative of its commitment to precision

The innovative amalgamation of Focal Loss and Smooth L1 Loss functions in YOLOv8 addresses misclassification and inaccurate bounding boxes, resulting in superior performance. The unique weighting of these loss functions in YOLOv8 is carefully tailored for optimal efficacy, distinguishing it from YOLOv7's similar approach. YOLOv8 employs the Swish activation function, known to heighten the precision of object detection algorithms. This function is shared with YOLOv7, albeit contributing to YOLOv8's consistent accuracy enhancements Auto Augment serves as YOLOv8's preprocessing technique, incorporating an automated application of diverse transformations to images. This feature is an evolution over YOLOv7's utilization of Auto Augment, with YOLOv8's iteration displaying a refined effectiveness. Post-processing, YOLOv8 employs Non-Maximum Suppression (NMS) to refine detection outcomes by eliminating redundancy and overlaps. This NMS approach, while comparable to YOLOv7's application, reflects YOLOv8's superior optimization for enhanced efficacy. Overall, YOLOv8 is a significant improvement over previous versions of YOLO. It achieves better accuracy, handles objects of different sizes and aspect ratios better, can be trained on images with a higher input resolution, and can detect more object classes, as Table V shows.

The accuracy of the YOLO object detection algorithms has steadily improved over the years. mAP (mean Average Preci- sion) is a metric used to evaluate the performance and accuracy of object detection algorithms. It is calculated by averaging the average precision (AP) scores for each object category in a dataset. The AP score for a single category is calculated by first calculating the precision and recall scores for that cat- egory. Precision is the fraction of predicted bounding boxes that actually contain the object. Recall is the fraction of ground truth bounding boxes that are predicted by the algorithm. The AP score is then calculated as the area under the precision recall curve. The mAP score is a more robust metric than the AP score because it averages the AP scores for all object categories in a dataset. This helps to reduce the impact of any individual category that may have a low AP score. The mAP score is typically used to compare the performance of different object detection algorithms. A higher mAP score indicates that the algorithm is more accurate. Overall, mAP is a useful metric for evaluating the performance of object detection algorithms. Nevertheless, a crucial consideration necessitates an awareness of its limitations during the interpretation of the outcomes YOLOv8 is the most accurate version of YOLO, achieving an mAP of 65.7% on the COCO dataset, as shown in Table VI.

The COCO dataset is a challenging dataset for object detection, as it contains a wide variety of objects in a variety of settings. The mAP metric measures the accuracy

http://dx.doi.org/10.21622/ACE.2023.03.2.064

of an object detection algorithm by calculating the intersection over union (IoU) between the predicted bounding boxes and the ground truth bounding boxes. An mAP of 100% indicates that the predicted bounding boxes perfectly match the ground truth bounding boxes. It is rarely achieved in real-world scenarios due to various challenges. Object detection algorithms should aim for high mAP values, but it's essential to balance precision and recall. Striking the right balance depends on the specific application and may require fine-tuning and optimization for optimal results.

The improvement in accuracy from YOLOv1 to YOLOv8 can be attributed to a number of factors, including:

- The use of deeper and wider neural networks
- The use of more data augmentation techniques
- The use of more powerful loss functions
- The use of better optimization techniques

YOLOv8 is the most accurate version of YOLO, but it is also the most computationally expensive.

C. TRAINING ON A SET OF IMAGES FOR INLAND VESSELS USING FIVE NEW CLASSES DATASET

COCO 80 classes are covering all the objects that are most likely appeared on the street for these autonomous vehicles and for their scene view. Each object that can be detected by the neural network is associated with a numeric ID. In the case of a YOLOv8 pre-trained model, there are 80 different object types, each assigned a unique ID ranging from 0 to 79, as mentioned in Table IV. The object classes in the COCO dataset are widely recognized. Testing this COCO trained model on the detection of the Inland vessels' environment objects is efficient for the objects appeared in both environments like (person, bird, etc.) or already included inside the COCO dataset like (boat, surfboard, etc...), but it will not be sufficient in case of waterway objects like: buoys, shores, cranes and bridge pillars. These added classes will be discussed deeply in the next subsection.

Five more classes from inland vessels' environment by extrapolation, as a contribution of this study, are added to the COCO dataset. These classes are:

- Left Shore (L Shore) with ID 80
- Right Shore (R Shore) with ID 81
- Bridge Pillar with ID 82
- Crane with ID 83
- Buoy with ID 8

http://dx.doi.org/10.21622/ACE.2023.03.2.064

TABLE IV
COCO dataset 80 Classes with preassigned ID's

ID	Object	ID	Object	ID	Object	ID	Object	ID	Object	ID	Object	ID	Object	ID	Object
0	person	1	bicy-cle	2	car	3	motor-cycle	4	airplane	5	bus	6	train	7	truck
8	boat	9	traffic light	10	fire hy-drant	11	stop sign	12	parking meter	13	bench	14	bird	15	cat
16	dog	17	horse	18	sheep	19	cow	20	elephant	21	bear	22	zebra	23	giraffe
24	backpack	25	um-brella	26	handbag	27	tie	28	suitcase	29	fris-bee	30	skis	31	snow-board
32	sports ball	33	kite	34	baseball bat	35	baseball glove	36	skate-board	37	surf-board	38	tennis racket	39	bottle
40	wineglass	41	cup	42	fork	43	knife	44	spoon	45	bowl	46	banana	47	apple
48	sandwich	49	or-ange	50	broccoli	51	carrot	52	hot dog	53	pizza	54	donut	55	cake
56	chair	57	couch	58	potted plant	59	bed	60	dining table	61	toilet	62	tv	63	laptop
64	mouse	65	re-mote	66	keyboard	67	cell-phone	68	micro-wave	69	oven	70	toaster	71	sink
72	refrigerator	73	book	74	clock	75	vase	76	scissors	77	teddy bear	78	hair drier	79	tooth brushes

http://dx.doi.org/10.21622/ACE.2023.03.2.064

Fig. 16. YOLOv8 model x object detection run.

D. COMBINED MODEL AND RESULTS

To combine the trained 80 classes dataset with the new classes data set, a set of processes starting with annotation, then training and predicting are applies in a certain sequence to reduce the annotation time and get the results accurately in both driverless and inland vessels environments. the next subsection shows the block diagram of the applied processes and discussed the methodology of detecting objects base on trained model for combined dataset.

In order to combine the trained model with 80 classes dataset and give it the ability to detect the new classes as well, a set of processes should be used. It is very important to do this as a building block and automate the process as can as possible. To achieve this goal, two main building blocks are needed as figures 17 and 18 show. First of all, as Figure 17 shows, the first block consists of three steps:

1) Annotate the dataset with the new objects with 5 Classes (Buoy, L Shore, R Shore, Crane, and Bridge Pillar).

2) Train the new dataset with new labels generated by the manual annotation.

3) Validate the performance of the new trained model for the newly added five objects.

For this purpose, a dataset of 300 images containing these new objects is chosen and annotated manually. labelImg program is used here for this manual annotation process, as shown in Figure 19, and the program output is with YOLO format for

http://dx.doi.org/10.21622/ACE.2023.03.2.064

labels. This format is class code, and the location of the bounded box of this object with x and y coordinates for the up left corner and down right on of the box.

These labels are saved with the same name of the images as txt file. After finish all of the annotation process, a 300 text files for labels will be saved in the output folder. All objects in these files are coded from 0 to 4, while the main COCO set objects are coded from 0 to 79. It is clear here that it is a must to avoid misleading codes to shift the codes generated in the new set from 0 to 4 to be from 80 to 84. A python program code is written for this purpose. This will make them combined in one list to be ready for the whole training for all objects in the scene for both street and waterway environments.

However, the training process takes a long time (in terms of days) on CPU based machines, and it is very fast in GPU's base ones (12 hours). Google Co-lab Pro package is used in training epochs. Utilizing the Google Colab Pro package for YOLO training offers substantial benefits. First, it's cost-efficient, providing access to powerful GPUs without substantial upfront investment. This accelerates training, crucial for YOLO's computational demands. Second, Colab Pro ensures ample resources, mitigating memory-related issues and enabling efficient handling of larger datasets. Its integration with Jupyter Notebooks fosters collaborative and interactive model development. Lastly, pre-installed libraries expedite setup, and seamless integration with Google Drive aids version control and sharing.

After the training epochs finished, confusion matrix, the losses, and F1 score metric are calculated to measure the model performance, accordingly, as illustrated in Figures 20, 21,22, 23,24, and 25 respectively.

In this stage and after training the 5 classes model with acceptable confidence for the detection of each class and low losses converges until the end of epochs it is time to prepare the dataset for the combined model. This data set includes both 80 classes models labels and the newly generated 5 classes ones. Manual efforts done here but with a very important reward for the system. One of the famous problems here is the double detecting behavior for the same object with the old 80 classes and in the 5 classes system too. Then a label correction phase is required here to distinguish between the old detection (boat, for example) and the new and correct one (buoy, in this case) as figure 26 shows.

The last preparation step here, the training for all of the system depending on the corrected labels and combined labels for each image is required to generate the .pt file which be the best weights file for the neural network to depend on it for further prediction in street environment or in waterway inland vessels one as well. The training and epochs are generated, as the previous training step, as figure 4.36 shows and the losses, confusion matrix and F1 score metric is calculated to measure the model performance, accordingly, as illustrated in Figures 4.41, 4.42, and 4.43 respectively.

The F1 score is a widely used evaluation metric in object detection tasks that combines precision and recall into a single value. It provides a measure of the overall performance and accuracy of an object detection model. In the context of detecting 85 classes on the street and on inland vessels, calculating the F1 score helps assess the model's ability to correctly identify and classify objects within these environments. The F1 score takes into account both the model's ability to correctly detect true positive instances (precision) and its ability to find all relevant instances (recall).

A high F1 score indicates that the model is effectively detesting and classifying objects across 85 classes. It reflects a good balance between precision and recall, where both the number of correctly identified objects and the number of false positives and false negatives are minimized. It is important to note that achieving a high F1 score for such a diverse range of classes in different environments can be challenging. It requires a comprehensive and well-curated training dataset that includes sufficient examples for each class. Additionally, finetuning the model architecture, optimizing hyperparameters, and carefully selecting training techniques can contribute to improving the F1 score.

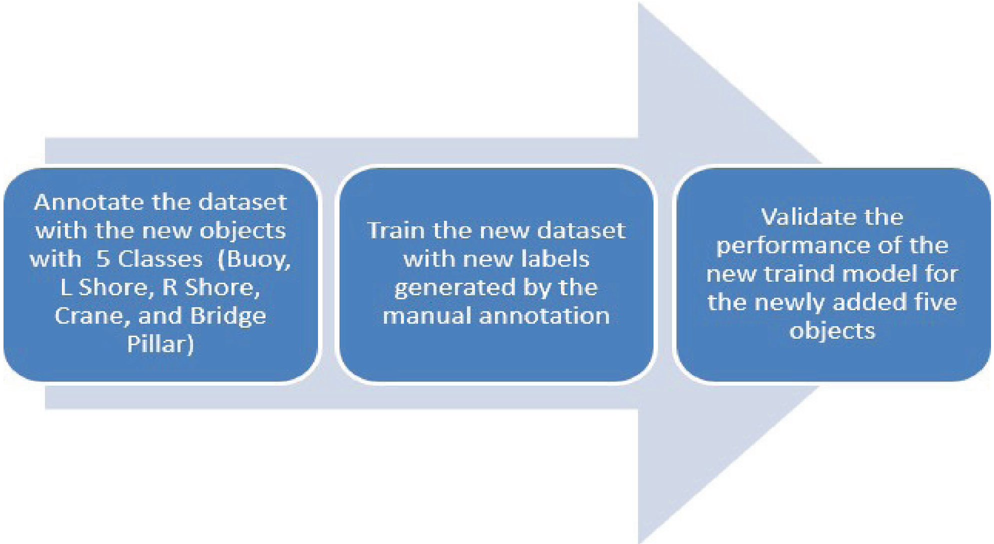

Fig. 17. Training for the new 5 classes model.

Fig. 18. Combined model process flow.

Regular evaluation of the F1 score is crucial during the development and improvement stages of the object detection combined model. It helps identify areas where the model may struggle, such as detecting specific classes or dealing with challenging scenarios. By analyzing and addressing the factors affecting the F1 score, developers can continuously enhance the model's performance and accuracy in detecting the 85 classes on the street and on inland vessels.

V. CONCLUSION

This study has undertaken an in-depth exploration of object detection methodologies, with a particular emphasis on the application of YOLOv7 and YOLOv8 models within street and inland vessel environments. The evaluation of these models' performance and accuracy was conducted using a comprehensive array of evaluation metrics, notably including the F1 score.

The results demonstrated that YOLOv8 surpassed existing object detectors in terms of both speed and accuracy across a range of frame rates. It achieved the highest accuracy among real-time object detectors operating at 30 FPS or higher on the GPU V100. The YOLOv8 models, pre-trained on the COCO dataset, showed promising performance for detecting 85 combined classes in street and inland vessel scenarios, eliminating the need for custom training.

Volume 3, Issue 2, December 2023- ISSN 2735-5985
http://dx.doi.org/10.21622/ACE.2023.03.2.064

The findings unveiled that YOLOv8 surpassed prevailing object detection methods in terms of both speed and accuracy across a diverse range of frame rates. Particularly notable was its achievement of the highest accuracy among real-time object detectors operating at 30 FPS or higher on the GPU V100. The YOLOv8 models, having been pre-trained on the COCO dataset, exhibited promising performance in detecting 85 combined classes across scenarios involving street and inland vessels, obviating the necessity for custom training. Nonetheless, it is imperative to acknowledge that the attainment of elevated accuracy and performance in object detection tasks remains a multifaceted challenge. The process necessitates meticulous model fine-tuning, meticulous hyperparameter optimization, and the curation of a diverse and well-annotated training dataset. These considerations bear significant weight in enhancing both the F1 score and the overarching performance metrics. Future research trajectories might encompass the exploration of advanced architectural paradigms, the integration of supplementary contextual information, and the resolution of context-specific challenges intrinsic to street and inland vessel domains. Addressing these challenges and pursuing innovation stands poised to yield more robust and accurate object detection models, thereby capacitating the adept identification and classification of objects within a spectrum of scenarios.

In summation, this paper profoundly contributes to the field of object detection by rigorously evaluating the performance of the combined YOLOv8 model within the contexts of street and inland vessel applications. This endeavor underscores their potential efficacy in real-time scenarios, thereby charting a course towards safer and more operationally efficient autonomous systems within these domains. The revelations emanating from this study, enriched by insights into the intricacies of YOLOv8, pave a discernible path for the progressive evolution of object detection technology. The implications of these findings are poised to catalyze the advancement of this field, fostering the development of more sophisticated and precise models conducive to safer and more proficient autonomous systems.

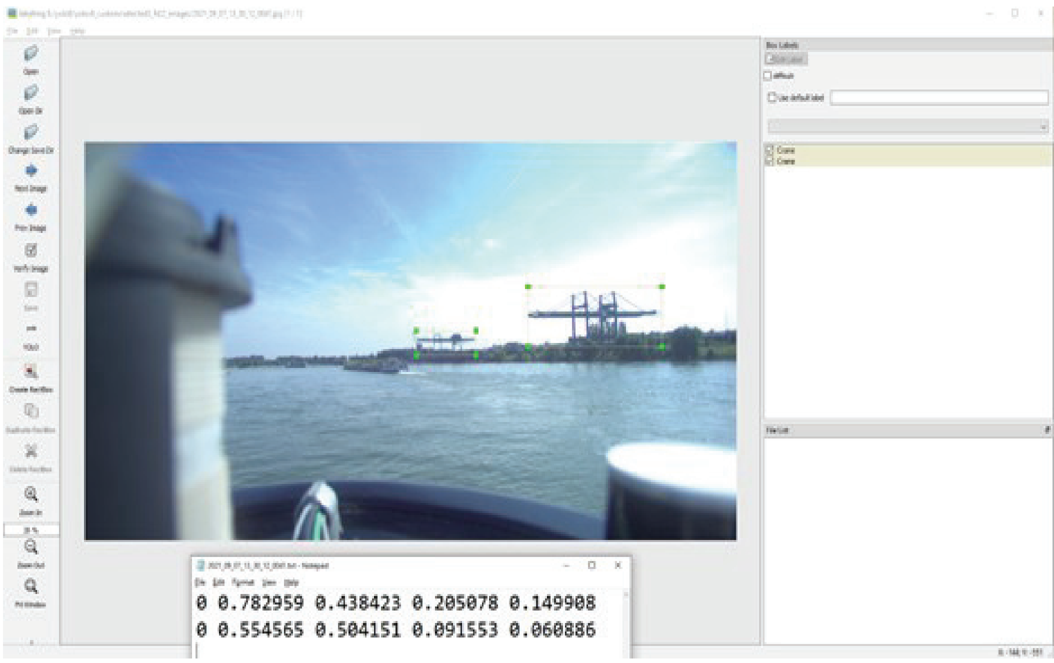

Fig. 19. Annotate new objects in the scene.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

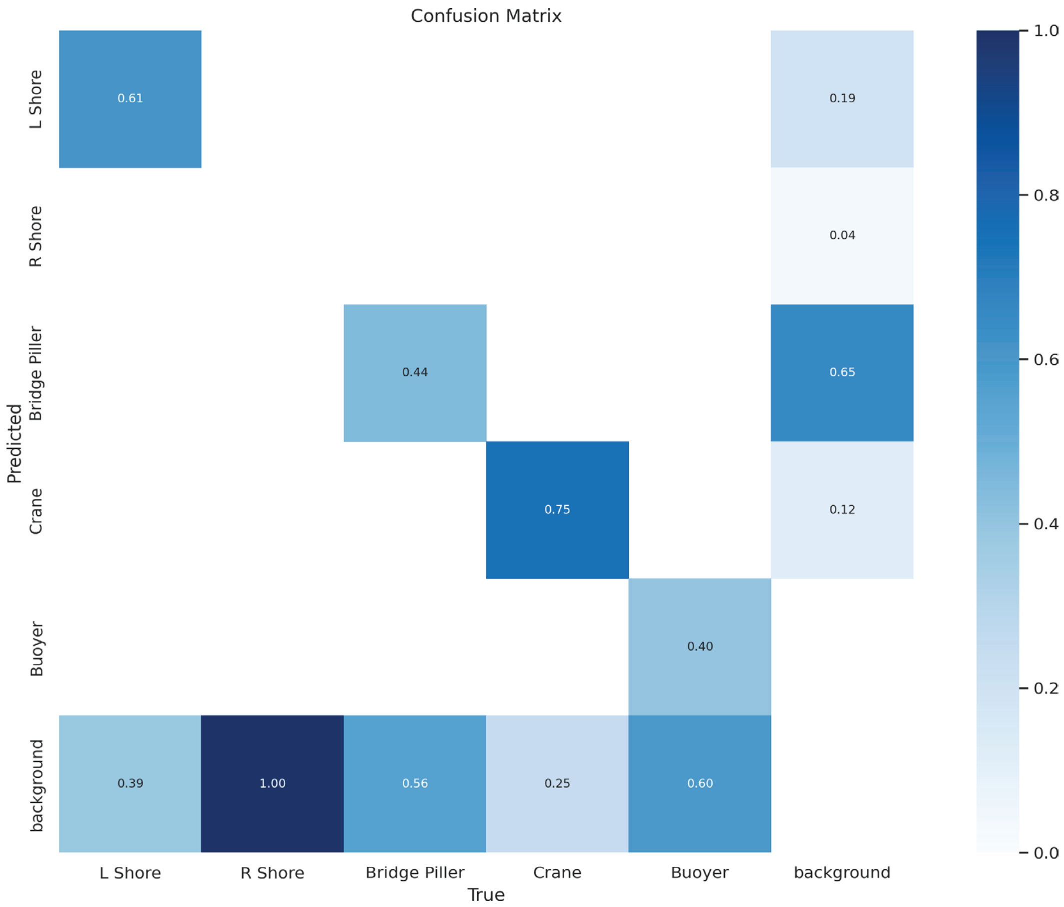

Fig. 20. Confusion Matrix of the 5 Classes Model.

TABLE V
YOLO8 and Evolution of Previous YOLO Algorithms

Key features	YOLO (V1)	YOLO (V2)	YOLO (V3)	YOLO (V4)	YOLO (V5)	YOLO (V6)	YOLO (V7)	YOLO (V8)
Architecture	Single neural network for both object localization and classification	Single neural network for both object localization and classification	Darknet-53 backbone and three detection heads with feature maps of different resolutions	Single-shot detection neural network for object localization and classification	Single neural network for both object detection and classification	Single-stage object detection framework	ELAN-efficient layer aggregation network	The modified version of the SPP-YOLO
Input resolution	448 X 448 pixels	416 X 416 pixels	Configurable input resolution up to 608 X 608 pixels	640 X 640 pixels	The configurable resolution, typically 640 X 640	Can be trained on multiple resolutions (up to 640 X 640)	Can be trained on multiple resolutions (1280 X 1280)	Can be trained on multiple resolutions (608 X 608)
Backbone network	"Darknet-19" as its backbone network	Darknet-19 architecture as its backbone network	Darknet-53 architecture as its backbone network	CSPDarknet53 or CSPRes- NeXt50 (depending on configuration)	CSPDarknet53 (improved Darknet network)	EfficientRep	CBS, E-ELAN, MP, and SPPCSPC modules	CSPDarknet53 architecture
Bounding boxes	Predicts bounding boxes with class probabilities	Predicts bounding boxes with class probabilities and anchor boxes	Predicts bounding boxes with class probabilities and objectness score Object classes	Predicts bounding boxes with class probabilities	Predicts bounding boxes with class probabilities and confidence score	Predicts bounding boxes with class probabilities and confidence score	Predicts bounding boxes with class probabilities and confidence score	Predicts bounding boxes and class probabilities for an object
Object classes	20	80	COCO dataset with 80 object classes	80 (COCO dataset)	Customizable, depending on the dataset	80	80	80
Training	Trained on VOC 2012 and ImageNet datasets using darknet framework	Trained on COCO dataset using darknet framework with data augmentation	Trained on COCO dataset using darknet framework	Trained on COCO dataset using the PyTorch framework	Trained using PyTorch framework with various datasets, such as COCO, VOC, and Open Images	Trained on COCO dataset using PyTorch framework	v1	Trained on COCO dataset using PyTorch framework
Loss function	Includes three main components: Localization Loss (Lloc), Confidence Loss (Lconf), and Classification Loss (Lcls)	includes several main components: Localization Loss (Lloc), Confidence Loss (Lconf), Classification Loss (Lcls), and Class Probability Loss (Lprob).	The multi-scale loss combines binary cross-entropy, confidence loss, and regression loss	IoU loss, GIoU loss, and objectness loss	Combined loss function consisting of focal loss, binary cross-entropy, and smooth L1 loss	yolov8n-seg.pt	YOLOv6 used Varifocal loss (VFL) for classification and Distribution Focal loss (DFL) for detection	VFL Loss as classification loss and DFL loss + CIOU loss as classification loss
Activation function	Leaky ReLU	Leaky ReLU	Leaky ReLU	Mish activation	SiLU (Swish)	SiLU	Leaky Relu	Leaky Relu
Pre-processing	Resizing and normalization	Resizing and normalization with data augmentation	Random resizing and data augmentation	Resizing and normalization plus Random crop, resize with letter boxing, and color distortion	Random scaling, translation, rotation techniques are used	Padding Gray borders	Compound scaling method	Mosaic data augmentation and class-specific anchor boxes
Post-processing	Non-max suppression	Non-max suppression with region proposal network (RPN) and anchor boxes	Non-max suppression with dynamic threshold based on objectness score and IoU threshold	NMS with an threshold of 0.5	Non- maximum suppression and confidence thresholding	Non- maximum suppression	Non- maximum suppression	Non- maximum suppression

TABLE VI
Comparison of YOLO8 with Previous Versions Accuracies

YOLO version	mAP on COCO (test-dev)
YOLOv1	21.2%
YOLOv2	33.1%
YOLOv3	35.5%
YOLOv4	43.5%
YOLOv5	48.6%
YOLOv6	50.4%
YOLOv7	57.9%
YOLOv8	65.7%

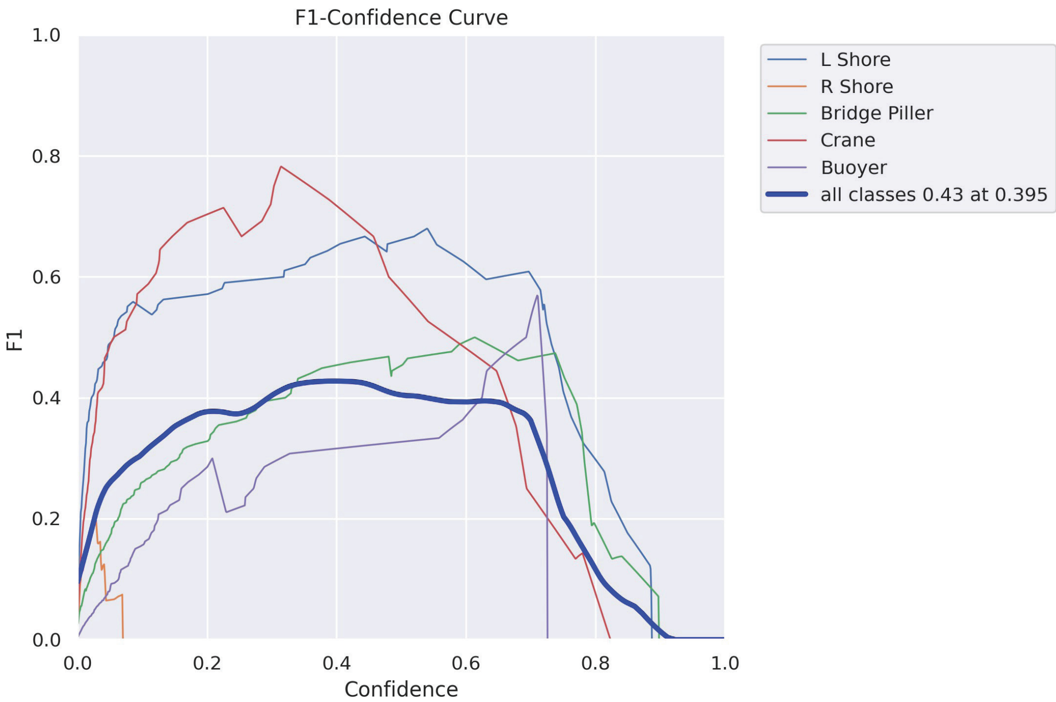

Fig. 21. F1-Confidence curve for the 5 classes model.

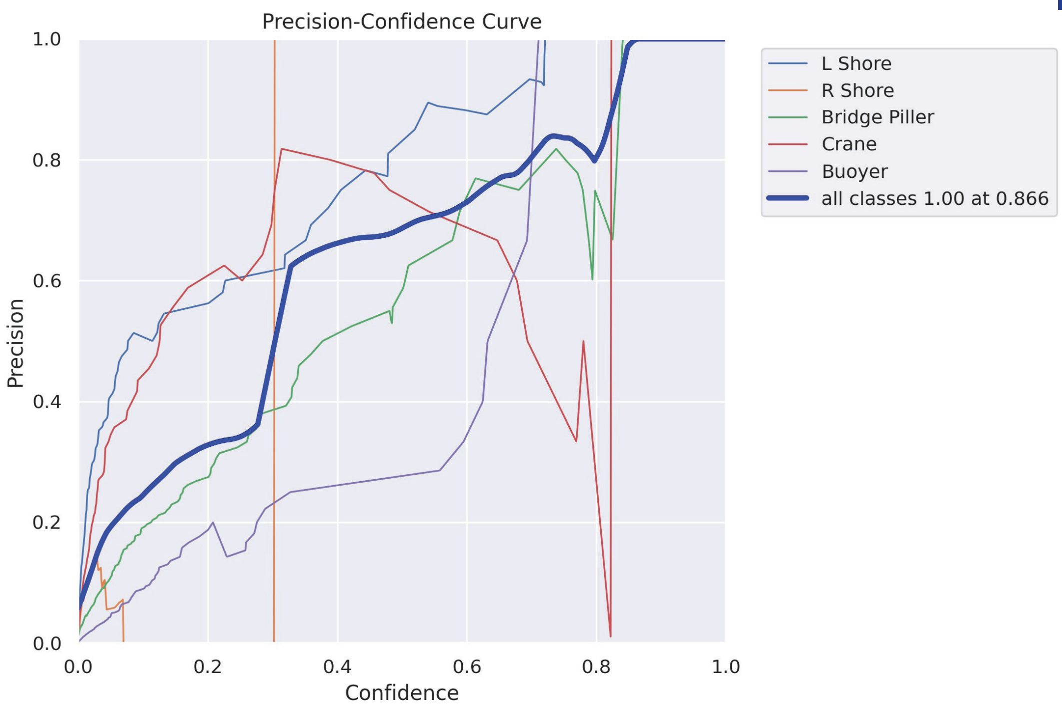

Fig. 22. Precision-Confidence curve of the 5 Classes Model

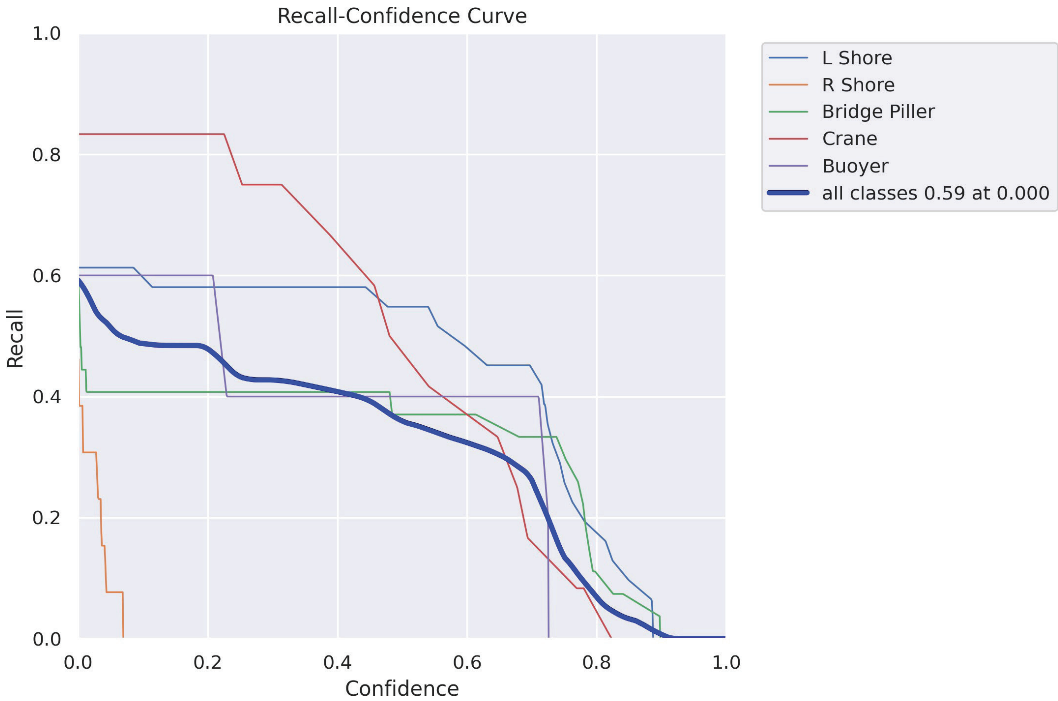

Fig. 23. Recall-Confidence curve of the 5 Classes Model.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

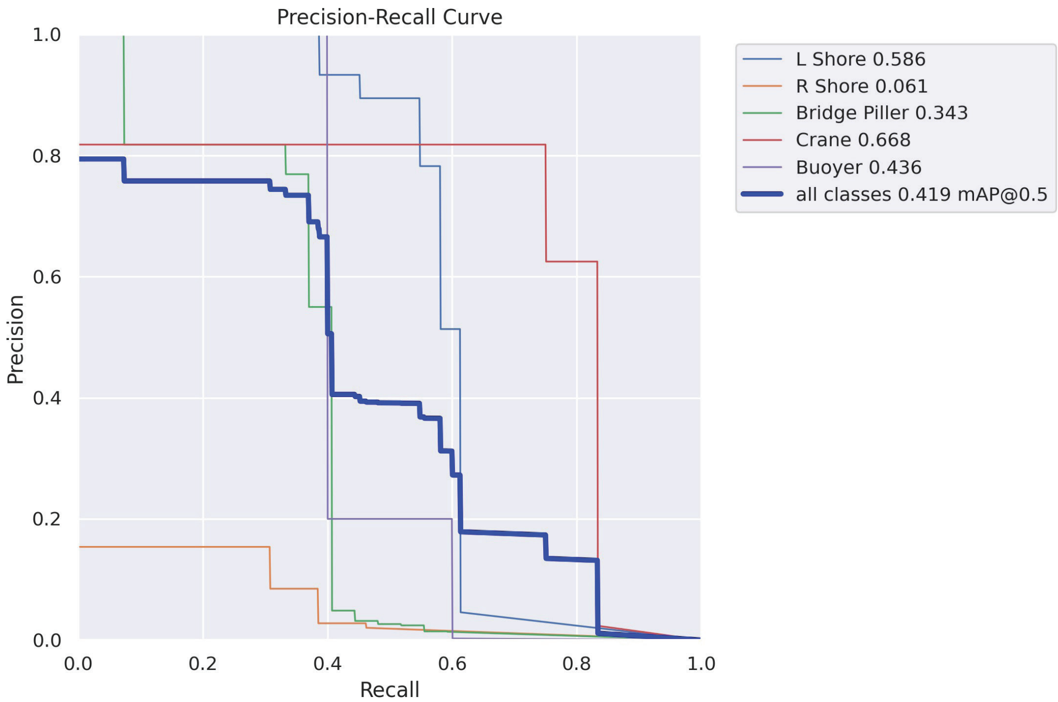

Fig. 24. Precision-Recall curves of the 5 Classes Model.

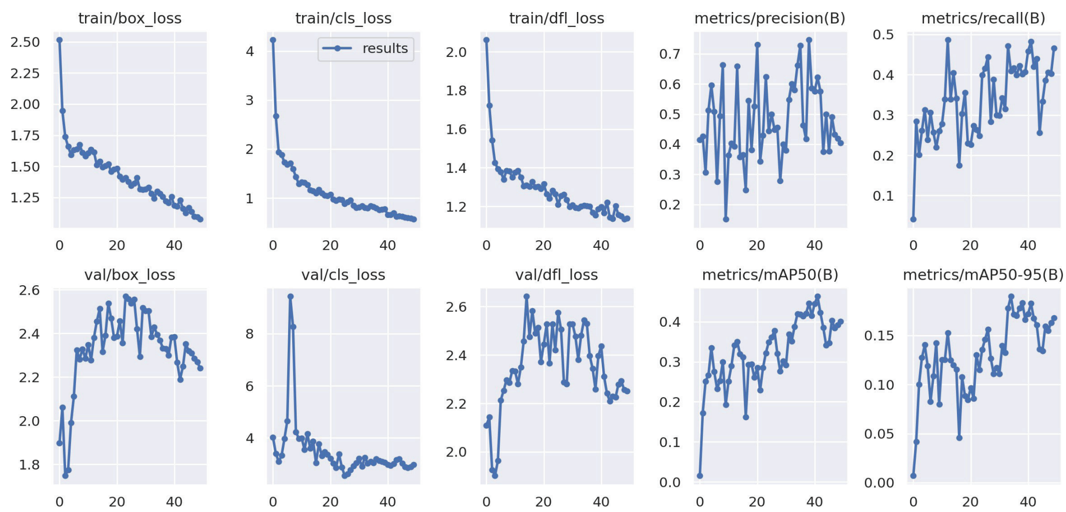

Fig. 25. Losses different curves of the 5 Classes Model

Volume 3, Issue 2, December 2023- ISSN 2735-5985
http://dx.doi.org/10.21622/ACE.2023.03.2.064

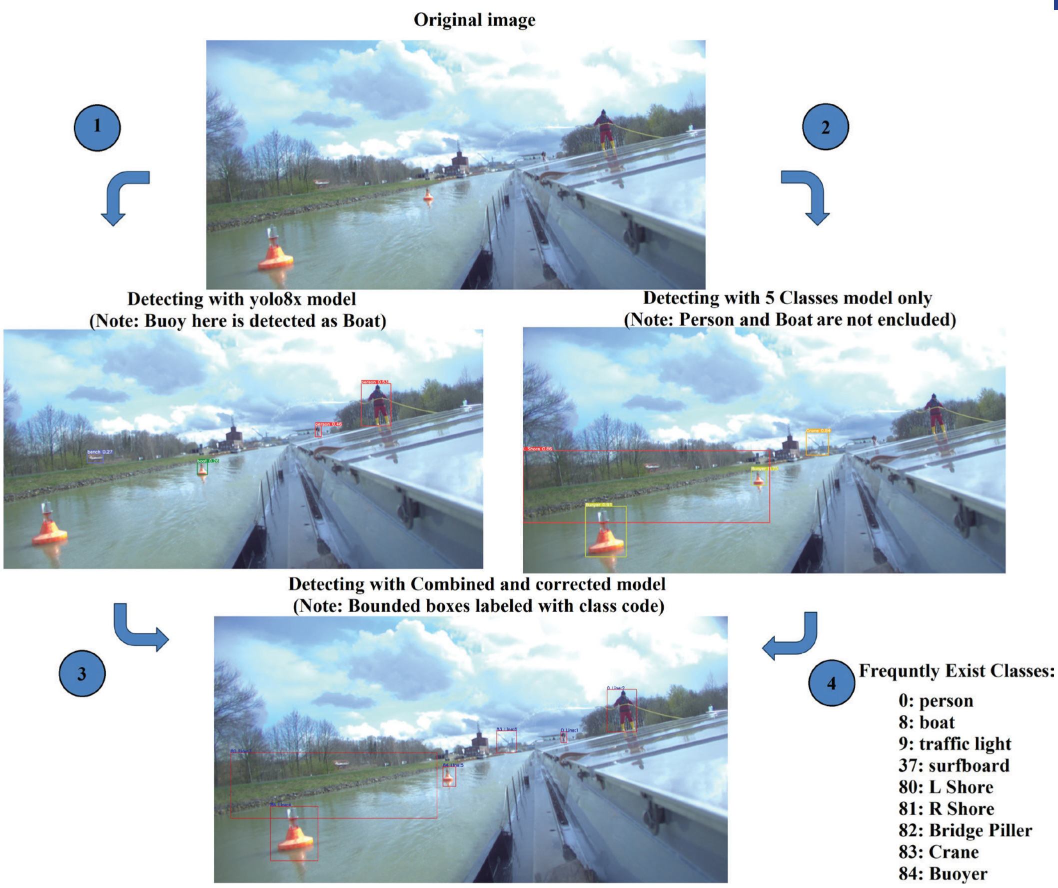

Fig. 26. Correcting multiple labels error and construct the combined model.

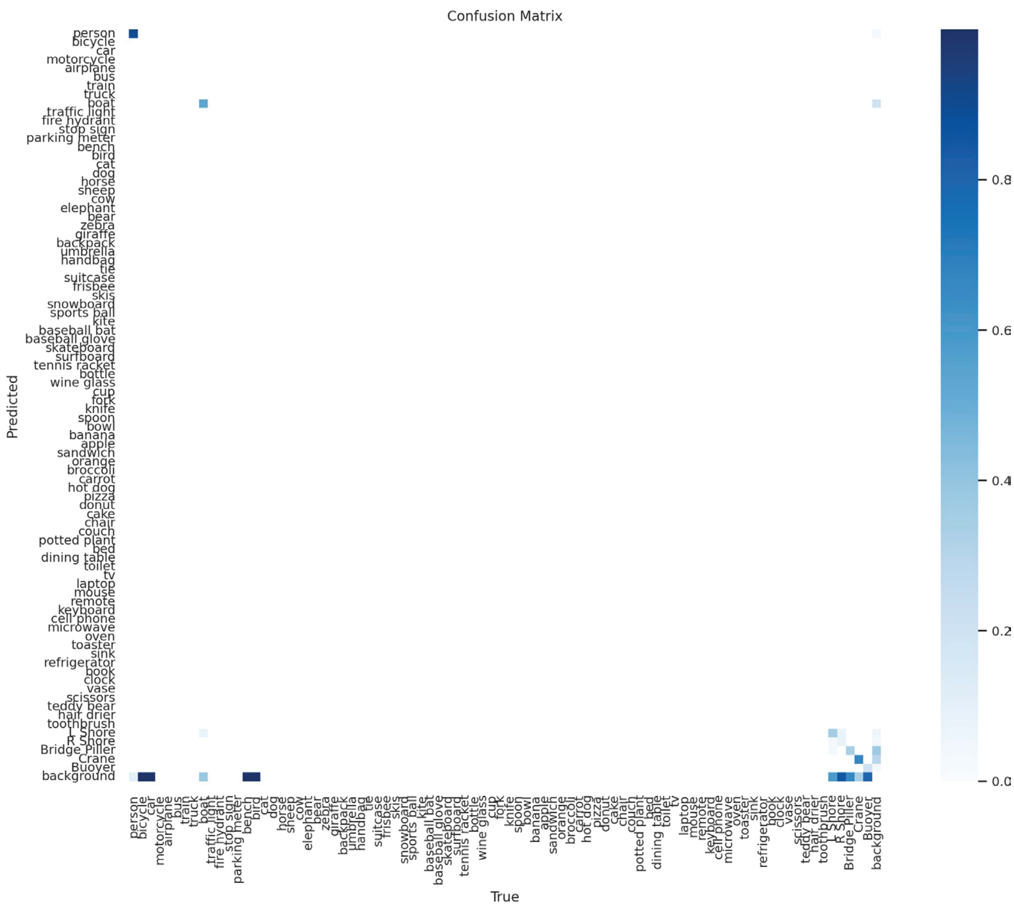

Fig. 27. Confusion Matrix of the 85 Classes Model.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

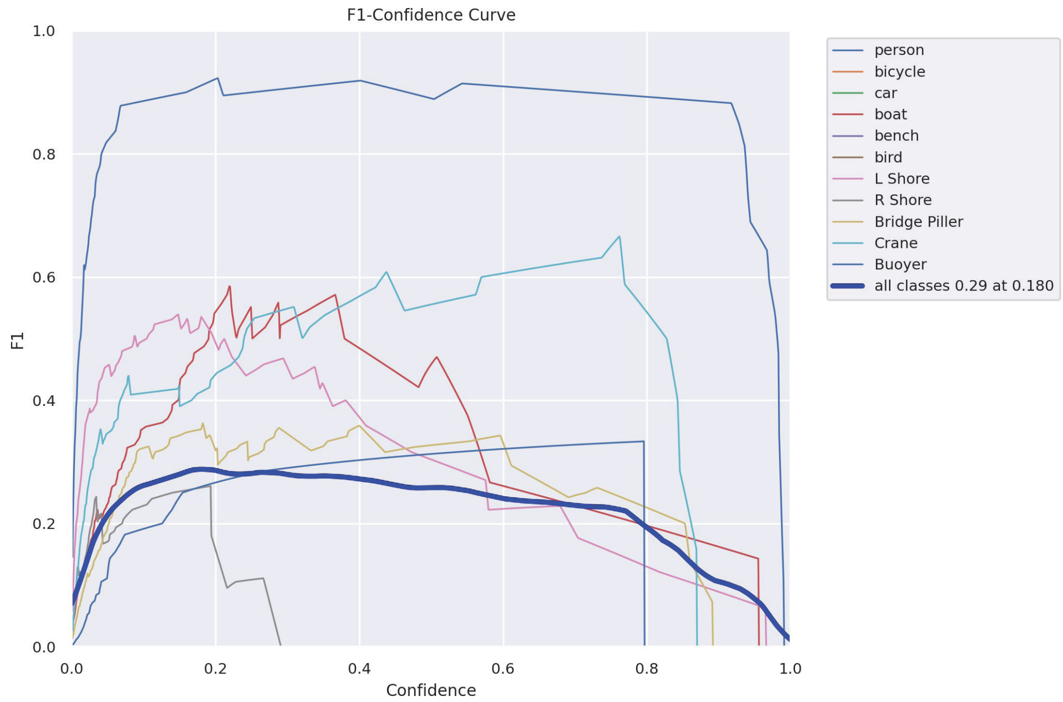

Fig. 28. F1-Confidence curve for the 85 classes model.

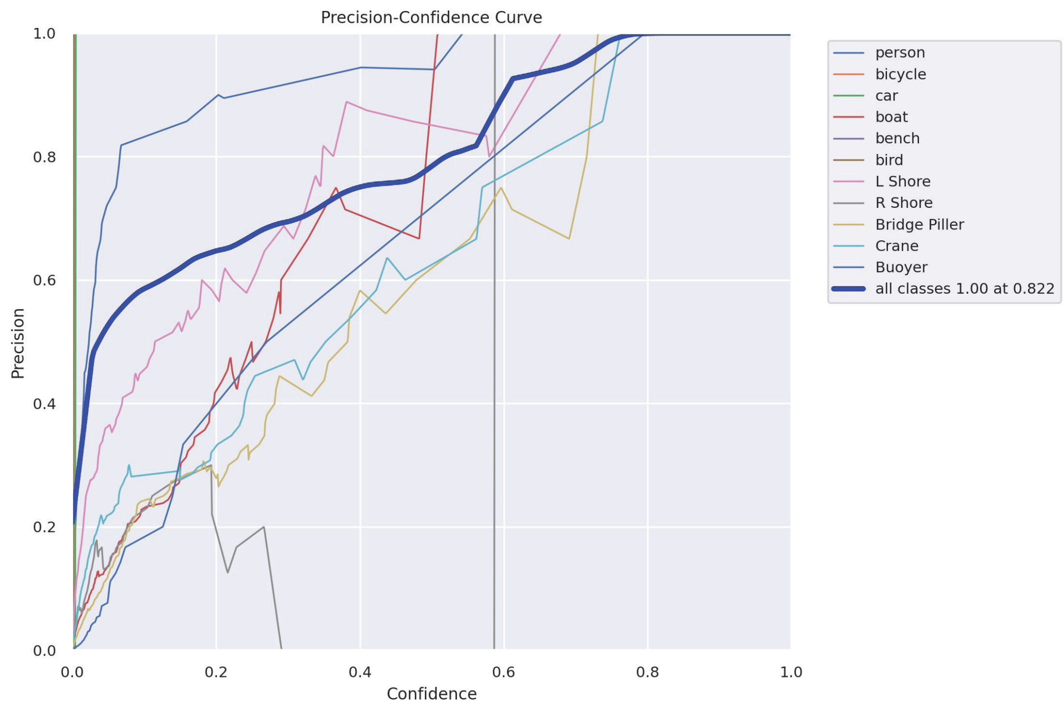

Fig. 29. Precision-Confidence curve of the 85 Classes Model

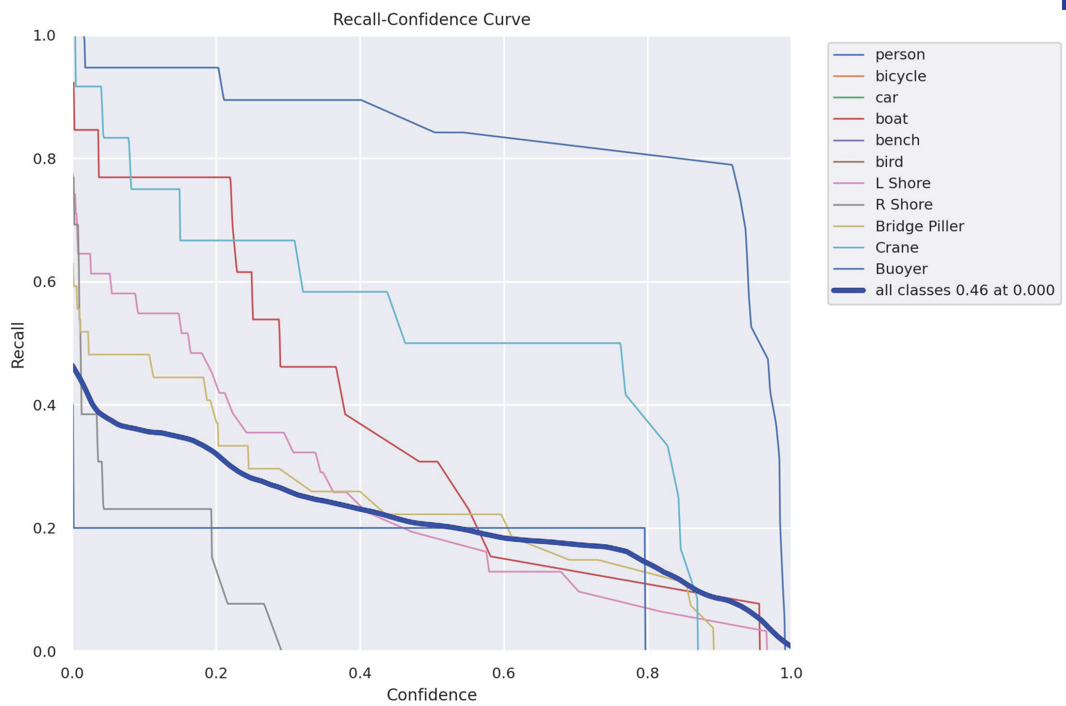

Fig. 30. Recall-Confidence curve of the 85 Classes Model.

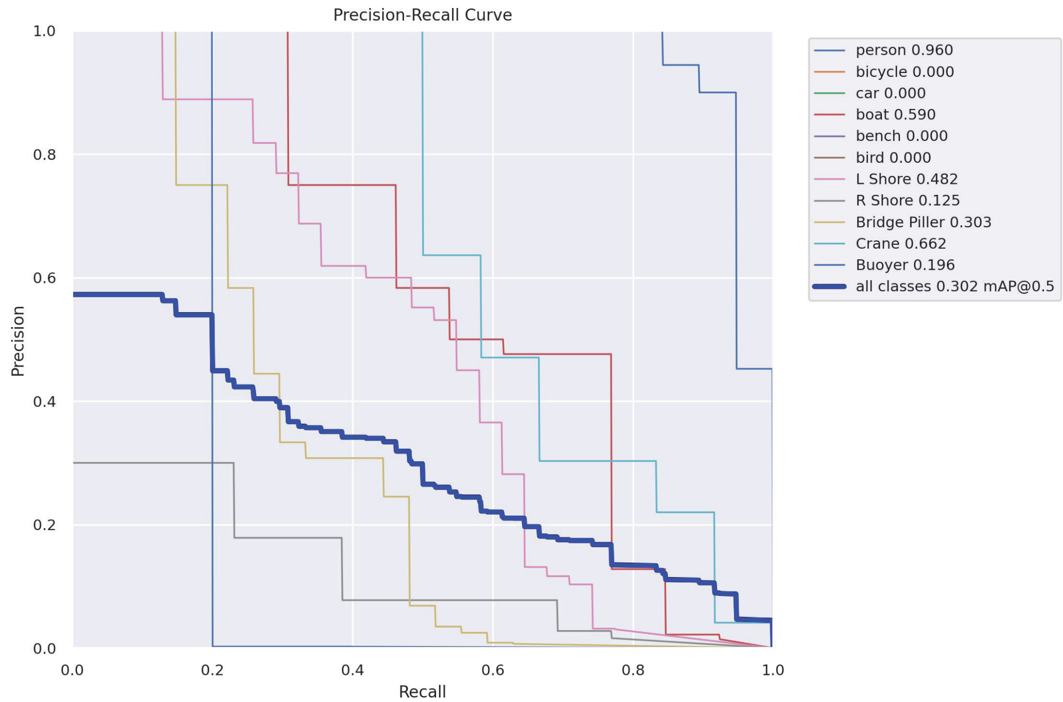

Fig. 31. Precision-Recall curves of the 85 Classes Mode

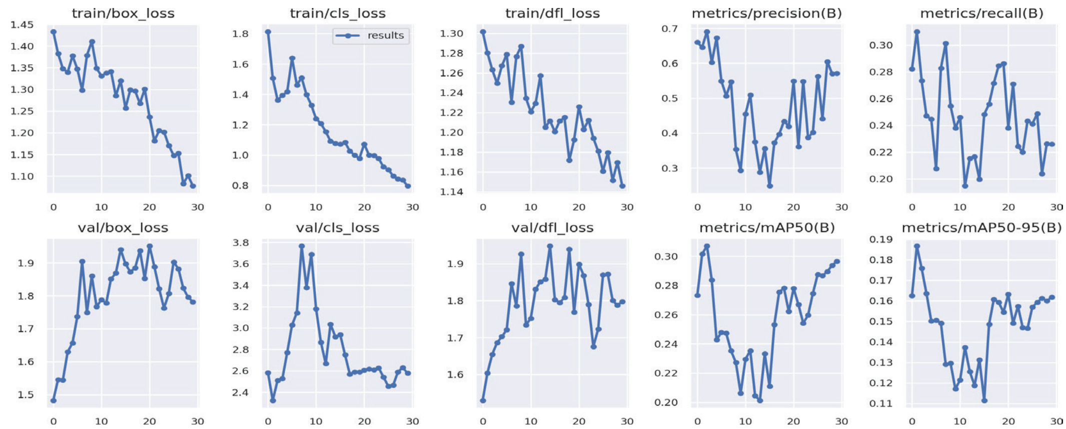

Fig. 32. Losses different curves of the 85 Classes Model

VI. REFERENCES

[1] X. Zhao, G. Wang, Z. He, and H. Jiang, "A survey of moving object detection methods: A practical perspective," Neurocomputing, vol. 503, pp. 28–48, 2022. [Online]. Available: https://doi.org/10.1016/j. neucom.2022.06.104

[2] W. Han, J. Chen, L. Wang, R. Feng, F. Li, L. Wu, T. Tian, and J. Yan, "Methods for Small, Weak Object Detection in Optical High- Resolution Remote Sensing Images: A survey of advances and chal- lenges," IEEE Geoscience and Remote Sensing Magazine, vol. 9, no. 4, pp. 8–34, 2021.

[3] M. N. Chan and T. Tint, "A Review on Advanced Detection Meth- ods in Vehicle Traffic Scenes," Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021, pp. 642–649, 2021.

[4] M. S. Karis, N. R. A. Razif, N. M. Ali, M. A. Rosli, M. S. M. Aras, and M. M. Ghazaly, "Local Binary Pattern (LBP) with application to variant object detection: A survey and method," Proceeding - 2016 IEEE 12th International Colloquium on Signal Processing and its Applications, CSPA 2016, no. March, pp. 221–226, 2016.

[5] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," Digital Signal Processing, vol. 132, p. 103812, 2022. [Online]. Available: https://doi.org/10.1016/j.dsp.2022.103812

[6] K. Li and L. Cao, "A review of object detection techniques," in 2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT), 2020, pp. 385–390.

[7] D. G. Lowe, "Distinctive image features from scale-invariant key- points," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in CVPR, 2005.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346–359, 2008.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," Proceedings of the IEEE International Conference on Computer Vision, p. 2564, 2011.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for

accurate object detection and semantic segmentation," IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, 2014.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.

[13] B. Zitova and J. Flusser, "Image registration methods: a survey," Image and Vision Computing, vol. 21, no. 11, pp. 977–1000, 2003.

[14] P. Kaur and S. Singh, "A survey of edge detection techniques," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 11, pp. 274–281, 2012.

[15] J. Canny, "A computational approach to edge detection," IEEE Trans- actions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[16] R. C. Gonzalez and R. E. Woods, Digital image processing, 3rd ed. Pearson Prentice Hall, 2008.

[17] D. Marr and E. Hildreth, "Theory of edge detection," Proceedings of the Royal Society of London B, vol. 207, no. 1167, pp. 187–217, 1980.

[18] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 898–916, 2011.

[19] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect objects with a scale-invariant template," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 11, pp. 2109–2119, 2009.

[20] T. Lindeberg, "Feature detection with automatic scale selection," in International journal of computer vision, vol. 30, no. 2. Springer, 1998, pp. 77–116.

[21] V. Balasubramanian and S. Z. Li, "Object detection using correlation filters," IEEE Signal Processing Magazine, vol. 29, no. 4, pp. 75–86, 2012.

[22] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Frequency-based object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2006, pp. 311–318.

[23] T. Arici, S. Dikbas, and Y. Altunbasak, "Color moments for content- based image retrieval: A review," Signal Processing, vol. 88, no. 8, pp. 2008–2021, 2008.

[24] J. Liu, Y. Xu, and H. Zhang, "Color coherence vector based image retrieval using multivariate gaussian model," IEEE Transactions on Multimedia, vol. 10, no. 4, pp. 629–640, 2008.

[25] T. Ojala, M. Pietik"ainen, and T. M"aenp"a"a, "Multiresolution gray- scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelli- gence, vol. 24, no. 7, pp. 971–987, 2002.

[26] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," inACM Computing Surveys (CSUR), vol. 38, no. 4. ACM, 2006, p. 13.

[27] D. G. Lowe, "Object recognition from local scale-invariant features," in Proceedings of the Seventh IEEE International Conference on Computer Vision.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

IEEE, 2001, pp. 1150–1157.

[28] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 1999, pp. I–511–I–518.

[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proceedings of the 2005 IEEE Computer Society Con- ference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886–893.

[30] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2004, pp. II–IV.

[31] D. G. Lowe, "Object recognition from local scale-invariant features," in Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2. IEEE, 1999, pp. 1150–1157.

[32] S. Shaheen, "Object recognition using contour-based representation,"

[33] Image and vision computing, vol. 13, no. 9, pp. 751–757, 1995.

[34] D. Mumford and J. Shah, "Boundary detection by minimizing func- tionals," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 22–26.

[35] T. Lindeberg, "Scale-space theory: A basic tool for analyzing struc- tures at different scales," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, pp. 610–625, 1994.

[36] G. Bradski and A. Kaehler, "The opencv library," Dr. Dobb's Journal of Software Tools, 2000.

[37] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," Image and vision computing, vol. 22, no. 10, pp. 761–767, 2004.

[38] T. Ojala, M. Pietik"ainen, and T. M"aenp"a"a, "Multiresolution gray- scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelli- gence, vol. 24, no. 7, pp. 971–987, 2002.

[39] Z. Zivkovic, "Improved adaptive gausian mixture model for back- ground subtraction," in Proceedings of the 17th International Confer- ence on Pattern Recognition, vol. 2, 2004, pp. 28–31.

[40] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," Proceedings of the IEEE, vol. 72, no. 10, pp. 1273–1284, 1983.

[41] J. Canny, "A computational approach to edge detection," IEEE Transac- tions on Pattern Analysis and Machine Intelligence, no. 6, pp. 679–698, 1986.

[42] D. Gabor, "Theory of communication. part 1: The analysis of informa- tion," Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering, vol. 93, no. 26, pp. 429–441, 1946.

[43] R. Girshick, "Fast r-cnn," in ICCV, 2015.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real- time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[45] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779– 788.

[46] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov5: Improved real-time object detection," arXiv preprint arXiv:2011.08036, 2020.

[47] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in European conference on computer vision. Springer, 2016, pp. 21–37.

[48] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[49] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard,

[50] M. Zhu, and Q. V. Le, "Efficientdet: Scalable and efficient object detec- tion," IEEE Conference on Computer Vision and Pattern Recognition, pp. 10 781–10 790, 2020.

[51] X. Du, Z. Zhang, H. Li, L. Zhang, W. Wen, R. Wang, and M. Tan, "Spinenet: Learning scale-permuted backbone for recognition and localization," in CVPR, 2020.

[52] N. Carion, F. Massa, A. Kirillov, and R. Girshick, "End-to-end ob- ject detection with transformers," European Conference on Computer Vision, pp. 213–229, 2020.

[53] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," IEEE Conference on Computer Vision and Pattern Recognition, pp. 13 152– 13 161, 2021.

[54] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region- based fully convolutional networks," Advances in Neural Information Processing Systems, pp. 379–387, 2016.

[55] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125, 2017.

[56] H. Law and J. Deng, "Cornernet: Detecting objects as paired key- points," in Proceedings of the European Conference on Computer Vision. Springer, 2018, pp. 734–750.

[57] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one- stage object detection," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9627–9636.

[58] K. Yang, W. Zhang, Z. Li, P. Sun, H. Liu, and C.-C. J. Kuo, "Reppoints: Point set representation for object detection," IEEE International Conference on Computer Vision.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

[59] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," IEEE Conference on Computer Vision and Pattern Recognition, pp. 9626- 9635, 2019.

[60] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.

[61] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region- based fully convolutional networks," in Advances in neural information processing systems, 2016, pp. 379–387.

[62] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[63] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[64] S. Zhou, Q. Huang, J. Yao, and Y. Guo, "Multi-modal multi-scale deep learning for large-scale image annotation," IEEE Transactions on Image Processing, vol. 28, no. 3, pp. 1276–1289, 2019.

[65] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660.

[66] Z. Chen, Y. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1382–1391.

[67] J. Ku, M. Mozafari, J. Lee, A. Harati, and V. Bhat, "Inception-v4, inception-resnet and the impact of residual connections on learning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018, pp. 4278–4285.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[69] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6202- 6211.

[70] Z. Li, C. Ma, and Y. Sheikh, "Deepgcns: Can gcns go as deep as cnns?" in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9267–9276.

[71] Z. Yan and T. Kanade, "Learning to segment under various forms of weak supervision," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 926–935.

[72] B. Deng, K. Li, X. Liu, and D. Tao, "Arcface: Additive angular margin loss for deep face recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 4690- 4699.

[73] H. Li, C. Li, G. Li, and L. Chen, "A real-time table grape detection method based on improved YOLOv4-tiny network in complex background," Biosystems Engineering, vol. 212, pp. 347–359, 2021. [Online]. Available: https://doi.org/10.1016/j.biosystemseng.2021.11. 011

[74] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," CVPR, 2001.

[75] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," in PAMI, 2010.

[76] X. Wang, Q. Zhao, P. Jiang, Y. Zheng, L. Yuan, and P. Yuan, "LDS-YOLO: A lightweight small object detection method for dead trees from shelter forest," Computers and Electronics in Agriculture, vol. 198, no. January, p. 107035, 2022. [Online]. Available: https://doi.org/10.1016/j.compag.2022.107035

[77] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Liao, "Yolov4: Optimal speed and accuracy of object detection," IEEE Conference on Computer Vision and Pattern Recognition, pp. 10 934–10 944, 2020.

[78] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," in NeurIPS, 2019.

[79] Y. Zhou, Y. Cui, M. Wang, and J. Xu, "Detectron2: A pytorch-based modular object detection library," arXiv preprint arXiv:2006.03137, 2020.

[80] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Sparse r-cnn: End-to-end object detection with learnable proposals," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14 783–14 792.

[81] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 4, pp. 743–761, 2012.

[82] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012, pp. 3354– 3361.

[83] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma,Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.

[84] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisser- man, "The pascal visual object classes (voc) challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, 2010.

[85] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan,

P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in Computer Vision–ECCV 2014. Springer, 2014, pp. 740– 755.

[86] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont- Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," arXiv preprint arXiv:1811.00982, 2018.

[87] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge," arXiv preprint arXiv:1804.07437, 2018.

[88] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," IEEE Access, vol. 7, pp. 128 837–128 868, 2019.

http://dx.doi.org/10.21622/ACE.2023.03.2.064

[89] X. Lou, Y. Huang, L. Fang, S. Huang, H. Gao, L. Yang, Y. Weng, and I.-K. Hung, "Measuring loblolly pine crowns with drone imagery through deep learning," J. For. Res., vol. 33, no. 1, pp. 227–238, 2021.

[90] A. Ammar, A. Koubaa, and B. Benjdira, "Deep-learning-based auto- mated palm tree counting and geolocation in large farms from aerial geotagged images," Agronomy, vol. 11, no. 8, p. 1458, 2021.

[91] W. Chen, S. Lu, B. Liu, G. Li, T. Qian, and J. Huang, "Detecting citrus in orchard environment by using improved yolov4," Sci. Program., vol. 2020, pp. 1–13, 2020.

[92] T. Jintasuttisak, E. Edirisinghe, and A. Elbattay, "Deep neural network based date palm tree detection in drone imagery," Comput. Electron. Agric., vol. 192, p. 106560, 2022.

[93] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.

[94] G. Jocher, A. Wong, I. Vasilev, and M. Green, "Yolov5: A universal object detection framework," in Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 429–437.

[95] B. Wong, S. Wu, Q. Lin, J. Liang, W. Cheng, Z. Li, and J. Ho, "Yolo nano: a highly compact you only look once convolutional neural network for object detection," in Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. ACM, 2020, pp. 161–170.

[96] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov5s: Improved version of yolov4 tiny," arXiv preprint arXiv:2103.16857, 2021.

[97] Wang, Chien-Yao and Bochkovskiy, Alexey and Liao, Hong-Yuan Mark, "Yolov5m: Yolov5 with more parameters," arXiv preprint arXiv:2104.08501, 2021.

[98] Wang, Chien-Yao and Bochkovskiy, Alexey and Liao, Hong-Yuan Mark, "Yolov5l: Stronger yolov5 with longer anchor boxes," arXiv preprint arXiv:2104.10472, 2021.

[99] Wang, Chien-Yao and Bochkovskiy, Alexey and Liao, Hong-Yuan Mark, "Yolov5x: Improved yolov5 backbone with dropblock," arXiv preprint arXiv:2105.04217, 2021.

[100] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov6: Re- designing yolo for improved accuracy and speed," arXiv preprint arXiv:2104.06580, 2021.

[101] Wang, Chien-Yao and Bochkovskiy, Alexey and Liao, Hong-Yuan Mark, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv preprint arXiv:2105.04206, 2021.

[102] D.-Y. Chen, W.-C. Chiu, Y.-D. Chen, Y.-H. Liu, S.-H. Lai, S.-H. Lai, and Y.-T. Chen, "Yolov8: A scalable object detection system for complex environments," arXiv preprint arXiv:2211.00623, 2022.

[103] H. Lou, X. Duan, J. Guo, H. Liu, J. Gu, L. Bi, and H. Chen, "Dc- yolov8: Small-size object detection algorithm based on camera sensor," Journal of Computer Science and Technology, 2023.

[104] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in con- text," IEEE Transactions on

Pattern Analysis and Machine Intelligence (TPAMI), vol. 39, no. 12, pp. 2414–2425, 2017.

[105] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in European Conference on Computer Vision (ECCV), 2014.

[106] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, and D. Lin, "Psanet: Point- wise spatial attention network for scene parsing," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[107] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[108] Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in Neural Information Processing Systems (NIPS), 2016.

[109] R. M. Alamgir, A. A. Shuvro, M. A. Mushabbir, M. A. Raiyan, N. J. Rani, M. M. Rahman, M. H. Kabir, and S. Ahmed, "Performance Analysis of YOLO-based Architectures for Vehicle Detection from Traffic Images in Bangladesh," 2022. [Online]. Available: http://arxiv.org/abs/2212.09144.

[110] M. C. Keles, B. Salmanoglu, M. S. Guzel, B. Gursoy, and G. E. Bostanci, "Evaluation of YOLO Models with Sliced Inference for Small Object Detection," arXiv e-prints, p. arXiv:2203.04799, 2022. [Online]. Available: http://arxiv.org/abs/2203.04799

[111] C. Wang, A. Bochkovskiy, and H. Liao, "Yolov7: Trainable bag-of- freebies sets new state-of-the-art for real-time object detectors," in Proceedings, 2022.

[112] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical analysis of design aspects of various yolo-based deep learning models for object detection," International Journal of Computational Intelligence Systems, vol. 16, no. 1, p. 126, 2023.

[113] A. Bochkovskiy, H.-Y. Liao, and C.-Y. Wang, "Yolov8: Towards real- time object detection," arXiv preprint arXiv:2301.05417, 2023.

[114] A. Bochkovskiy, C.-Y. Wang, H.-Y. Liao, and R. Girshick, "Yolov5: A unified, fast, and accurate object detection system," arXiv preprint arXiv:2004.10934, 2020.

[115] A. Bochkovskiy, H.-Y. Liao, and C.-Y. Wang, "Yolov6: An improved implementation of yolov5," arXiv preprint arXiv:2101.08697, 2021.

[116] J. Redmon and A. Farhadi, "Yolov4: Optimal speed and accuracy trade- offs for object detection," arXiv preprint arXiv:1804.04381, 2018.

[117] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "Yolo: Real-time object detection," arXiv preprint arXiv:1506.02640, 2015.

[118] R. Girshick, J. Malik, and T. Darrell, "Mean average precision (map) for object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2014. [Online]. Available: https://arxiv.org/abs/1412.0076

ACE

Advances in Computing and Engineering

Volume 3, Issue 2, December 2023 - ISSN 2735 - 5977

مركز الأكاديمية للنشر
ACADEMY Publishing Center