Smartphone-Based Game Development to Introduce K12 Students in Applied Artificial Intelligence

Sara Guerreiro-Santalla, Alma Mallo, Tamara Baamonde, Francisco Bellas
GII, CITIC Research Center, Universidade da Coruña, Spain
sara.guerreiro@udc.es, alma.mallo@udc.es, tamara.bardao@udc.es, francisco.bellas@udc.es

Abstract

This paper presents a structured activity based on a game design to introduce k-12 students in the topic of super-vised machine learning from a practical perspective. The activity has been developed in the scope of an Erasmus+ project called AI+, which aims to develop an AI curriculum for high school students. As established in the AI+ principles, all the teaching activities are based on the use of the student's smartphone as the core element to intro-duce an applied approach to AI in classes. In this case, a smartphone-based game app is developed by students that includes a neural network model obtained with the "Personal Image Classifier" tool of the MIT App Inventor software. From a didactic perspective, the students dealt with supervised learning to solve a problem of image classification. The main learning outcome is the understanding of how relevant is to develop a reliable machine learning model when dealing with real world applications. This activity was tested during 2021 with more than 50 students belonging to six schools across Europe, all of them enrolled in the AI+ project.

Introduction

The AI+ project is an educational innovation initiative which has been funded by the European Union through the Erasmus+ program (AIplus project 2019). It encompasses six secondary schools from five different European countries (Spain, Italy, Slovenia, Lithuania, and Finland), which aim to develop an AI curriculum for students ranging from 15 to 17 years old. The project is leaded by the University of Coruña (Spain), which proposes the specific teaching units (TUs) that make up the curriculum. They contain specific theoretic concepts about the different AI topics, as well as structured activities to train them (Guerreiro-Santalla 2020). These TUs are revised by the teachers of the six schools and tested with their students, which provides feedback to the leading team to improve them. Finally, the TUs are published in the project website to make them available to all the teaching community (Aiplus results 2021). It must be highlighted that the work carried out in this project has been the basis of a new official subject in high school (Introduction to AI), offered to all secondary schools at Galician region in Spain in 2021/22 (Xunta 2021).

The AI approach behind all the TUs is that of *intelligent agent*. Hence, AI systems must be presented to students as agents that receive percepts from the environment, real or simulated, and perform actions to fulfill their objectives in an autonomous way. To achieve a successful response of the AI system, many internal processes occur (representation, reasoning, learning) which make up the core topics of the TUs. This perspective of AI fits with the recommendations of the European Commission (2018) for the future digital education plans, focused on "specific AI", instead a more theoretical and general perspective. This view is also followed by one of the most popular textbooks in the field (Rusell&Norvig 2021), in its last edition.

The AI teaching perspective followed in AI+ is mainly practical, based on programming intelligent agents running on real-world devices and solving specific problems, what is known in the field as *embedded intelligence* (Allipi 2014). This requires relying on some specific hardware elements to implement the teaching material. To this end, a key premise in the project is that of using the student's mobile phone (*Smartphone*) as the central technological element for all educational material to be developed. Current smartphones have the technological level required for AI teaching in terms of sensors, actuators, computing power and communications; and they will have it in the future.

We are aware that reliance on smartphones can be problematic, either because some schools do not allow their use, or because some students may have limited access to these devices due to their cost. However, this is becoming less common, and it is possible for schools to obtain used devices through donation campaigns by other pupils, or from local public institutions. It should be noted that the activities proposed here do not require an active telephone or data connection, as the school's WI-FI is used.

Starting from the remarkable work carried out in the AI4K12 initiative (2021) and including the experience of the leading group of the AI+ project, 8 AI topics that must be considered at this educational level have been already established (Guerreiro-Santalla 2020): perception, actuation, representation, reasoning, learning, collective intelligence, motivation and SEL (sustainability, ethics, and legal aspects of AI). To introduce these topics using the Smartphone, three AI application fields have been selected: intelligent smartphone apps (using App Inventor (2021)), autonomous robotics (using the Robobo robot (2021)) and Internet of Things (using the Home Assistant platform (2021)). Currently, six TUs have been already published, all of them focused on developing intelligent Smartphone apps with App Inventor (Aiplus results 2021). The second block of TUs, centered on intelligent robotics, are under development.

This paper presents a structured activity corresponding to TU4. It is focused on introducing supervised machine learning to K12 students by means of a practical activity where they have to develop a smartphone-based game app using the "Personal Image Classifier" of App Inventor (PIC 2021). As explained in (Gresse von Wangenheim 2021), many teaching activities have been developed to introduce machine learning in K12. Most of them are focused on supervised learning and image classification, as the one proposed here. The main reason for this abundance is the existence of several online tools (Marques 2020) that allow to download images from the internet, create the model in an easy way (no programming), train it, and analyze its response quickly. This way, all the stages involved in machine learning can be performed at easily classes and in a short time.

What is new in the activity proposed here is the final "product" that students develop. They use a standard App Inventor extension, like other proposals (Marques 2020), but only as an intermediate step to create a model that is used in a real game app. Consequently, the TU assessment is based, not only on the machine learning model, but on the whole game experience. Hence, this work encourages the relevance of AI topics (learning in this case) associated to their necessity for a practical application. Students must understand that the learned model provides the game with an "intelligent" capacity that is mandatory to make it work properly, having no real interest without it.

As a complementary material, the resource presented here provides the option to introduce students in the field of artificial collective intelligence (multiagent systems), a novel topic in AI education at K12 to our best knowledge (Weiss 2000). To do it, once they have finished a first version of the game app, an improvement is proposed to include non-local information from other players, so a collective version of the intelligent game is developed too. This extra part of the resource corresponds to TU5 in the AI+ curriculum.

Resource Description

The proposed activity is based on the development of a smartphone app, called *Capture it!*, using the MIT App Inventor software, to monitor a Scavenger Hunt game. This is a very popular type of collective game which objective is to find hidden objects in the real world, normally, following clues. The players can be individuals or groups. In this case, the game is monitored by the app, which manages the advance by interacting with the players through the smartphone screen and the speaker. First, the app displays the object that must be found (from a set of predefined ones, as it will be explained later). Once found, the player must take it a picture using the smartphone's camera. The app recognizes the object using a previously learned classification model and notifies the user if it is the correct one or not. If it is not, the user must continue searching for it, while if it is, a new object is proposed. When all the predefined objects are found, the app will inform of the total time used to overcome the challenge. The player/team that has taken the shortest time, is the winner. A video has been created which summarizes the app functioning[1].

As it can be observed, it is not a typical smartphone game, because the game itself takes place in the real world, where the objects must be placed and discovered. But the app must manage the overall functioning, controlling the correct realization and finalization. Two versions of the game are proposed here. In the first one, the game only works with local information. In the second one, optional, information from other players taken from the cloud is used, creating a more challenging collective version of the game to introduce students in multiagent systems. In both cases, the game has been highly interesting and motivating for students.

Target Age Group

This activity has been designed for students in upper secondary school or high school levels (age ranging from 14 to 17) with a basic technical background. To adequately meet the learning objectives of the resource, they should have the following prior knowledge:

- *Programming:* basic level on block-based programming, conditionals, loops, variables, lists and functions.
- *Specific software:* basic knowledge of MIT App Inventor is assumed, for instance, by performing the introductory courses available at their webpage (App Inventor, 2021).
- *Mathematics:* fundamentals of algebra and functions. It is very important that students understand the concept of

[1]https://bit.ly/3ttJpry

mathematical model, and the relation between inputs and outputs. Basic concepts of statistics are welcomed too, to properly understand the fundamentals of data preparation and analysis of results.

- *IT skills*: it is recommended that students have some experience on informatics, to transfer data from the computer to the smartphone, and from the cloud to the computer. Moreover, basic knowledge about photo capturing and storage format is welcomed.

Setup and Resources Needed

The activity, as well as others in AI+, is based on the project-based learning (PBL) methodology. This is in accordance with the project approach to applied AI, so the topics of each TU are trained by facing a challenge to be solved with an embedded device, in this case, the smartphone. Hence, students are organized in groups, and each of them receive the same set of specifications for the solution to achieve. They autonomously develop it by organizing the work in sub-problems, which are monitored by the teacher. Finally, the resulting app is tested by the teacher, which provides an assessment based on the different aspects, as explained later in the evaluation section.

In particular, for this activity groups of two students are suggested, to maximize their implication on the tasks. Each of them will have one role: programmer or manager. Although in this case the main objective is that both students collaborate in the app programming, the programmer will be focused in the development of the program in App Inventor exclusively. On the other hand, the manager must handle all the aspects required to carry out the app properly, like taking notes, asking questions, managing the time, and submitting deliverables to the teacher. It is very important that both members are always in agreement and aware of what each other is doing.

The required resources to follow the activity are: (1) A laptop or computer per group, connected to the WI-FI (2) An Android Smartphone per group, connected to the WI-FI, and with the MIT App Inventor app installed (3) Every group must have an active App Inventor account (4) A small set of objects to be learned with the model and detected during the game, like those than can be present in the school: an apple, a cup, a bottle, or others.

The estimated time required for the development of the activity is 8 hours. Additionally, 4 extra hours could be required for the optional part focused on multiagent systems. All the teaching material related with this resource is available to download at (TU4 2021) and (TU5 2021). These links include a pdf file with the TU explanation for teachers, a second pdf version for students (without solutions), the solution program in App Inventor, and the previously referenced video showing the app expected functioning.

AI Concepts Addressed

The main objective of this activity is to introduce students to the fundamentals of machine learning with an applied perspective. More specifically, students are trained in the concepts of image classification by means of supervised learning with artificial neural networks. As a complementary activity, they are introduced in the scope of artificial collective intelligence, by learning the potential of introducing remote sensing and actuation in an AI system, moving from an intelligent agent to a multiagent system. The AI concepts that must be taught to students from a theoretical perspective are organized in four parts.

First, in the scope of AI, teacher must explain that machine learning relies on algorithms that run in a computer-based system, and that are able to create mathematical models from data. These models generalize their output to other datasets different from those used for learning.

Second, it must be introduced to students that, in the field of Machine Learning, three main types of computational techniques are considered: supervised learning, unsupervised learning and reinforcement learning. The main differences between them should be explained, reinforcing those stages that are common (data preparation, training, validation), and those that are particular to each type.

Third, students must be trained in this activity in the specific case of supervised learning for classification. A previous explanation about the difference between regression and classification problems should be performed. They will train a neural network model for image classification, so the fundamentals of artificial neural networks must be explained. In this scope, only a multilayer perceptron model is introduced, with basic explanation of the feed forward execution. Backpropagation of the error is introduced too, but only in a visual way, without detailing the mathematical background.

Finally, once they understand the type of learning process they will implement, they must go deeper in the 4 main stages of a machine learning process:

(1) *Data preparation:* explaining that the images used for learning must be appropriate in terms of representativity (containing the target object clearly, no interference with others, and adequate size), quality, and diversity (rotations, angles, colors...).

(2) *Artificial Neural Network fundamentals:* explaining the main parameters that can be adjusted in the tool, and their influence in learning (number of layers, number of neurons, learning rate, and epochs).

(3) *Model validation:* introducing the relevance of a formal model validation, including the main performance metrics, like confidence, and highlighting the importance of using cross validation techniques to obtain reliable models. The model validation is a key aspect here because the app functioning relies on it.

(4) *Model usage:* explaining that the result of the learning process is an array of adjusted parameters of the model that can be easily implemented in a program and used in real time in the app.

As an additional activity, students can be introduced on the basics of artificial collective intelligence (multiagent systems), that is, the intelligence that can be achieved when multiple AI systems are interconnected. As an introductory case, this resource includes a cloud database in which the apps can publish and query information. It is a simple type of centralized communication, where all the synchronization and coherence issues are solved, but it allows students to develop a collective version of the game, where each player receive information of the opponents in real time.

In a transversal way, this activity will also help students reinforce concepts from perception and actuation in AI. Concerning perception, the app uses different sensors: the camera to capture the objects; the microphone to record the user's speech, and the touch screen, where the user can click on buttons. Regarding actuation, the app includes the screen, on which the game information is displayed; and the speaker, communicating the end of the game by voice.

Expected Learning Outcomes

At the end of the activity, students should understand that learning is a key property of any intelligent system to provide it with the capacity of improving from experience. They should understand that situations that were not previously seen or programmed can be addressed, and even anticipated, by including a machine learning model. Moreover, they should obtain specific knowledge about the four stages of machine learning explained above.

This activity encourages the practical relevance of machine learning. Once they finish their app, students understand that the reliability of the image classification model is the key feature of their game, which makes it "intelligent"

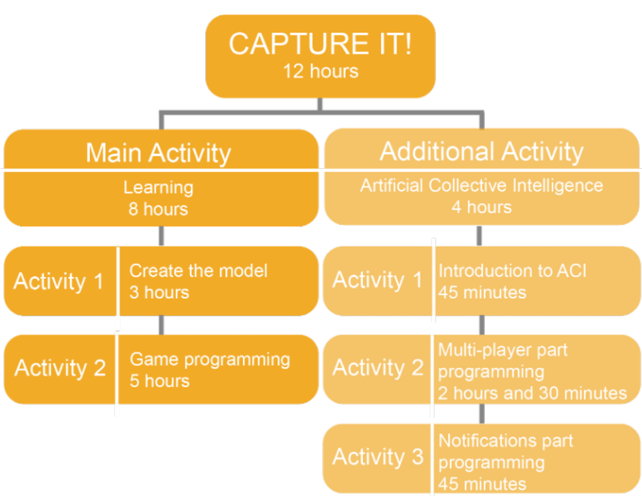

Figure 1. Challenge organization diagram

and interesting for a player. All the remaining aspects of the app programming, like the user interface, and the internal algorithms imply a hard work for them, but the intelligence of the game relies on the learned model.

A third learning outcome is a consequence of developing real-world AI systems that interact with human users: the intelligence of a system depends in a great manner on the user perception. For example, some players with a higher technical background could perceive that the developed game uses a simple matching between objects and pictures, while others could perceive a real intelligence behind it, with a feeling of "astonishment" against AI. In this sense, students must understand their responsibility as AI developers when creating systems dealing with humans, realizing the ethics behind machine learning.

Lastly, regarding multiagent systems, the main aspect that students should learn is that of managing remote information, both for perception and actuation. This concept implies that the intelligent agent does not rely only in local information, and it can communicate with others. From a practical perspective, they will learn how to coordinate the app functioning depending on this global information exchange.

Capture It!

Figure 1 displays a diagram which summarizes the steps that should be followed to develop this structured resource. The main activity, left part of the diagram, has been structured in two tasks, the first devoted with the model and the second one focused on the game programming, where the model is included. This order can be interchanged if preferred.

Activity 1 starts by introducing students with all the machine learning theoretical concepts previously commented. This could be done through a master class, or providing students with material (papers, books, or multimedia), so they can review these concepts autonomously. This last approach is the one proposed in TU4, where the student version of the TU contains links to reference material.

After the theoretical introduction, all students start with the model learning. The first step must be to decide the number and type of objects each group wants to detect in their specific game, keeping in mind that there must be at least 3. Some recommendations are provided in the TU4, like using small objects, easily identifiable, and available at schools. For instance, fruits, bottles, or cups are used in the examples of the TU4. The next step is to open the Personal Image Classifier module of App Inventor 2 on a web browser (PIC 2021). It allows to create an artificial neural network model using supervised learning to perform image classification by means of a simple web interface, without requiring programming. To use it, students must follow the next 4 steps:

Add Training Data: students should upload their set of images and assign them their corresponding tag. As it was introduced in the theoretical part, data adequacy is key for a proper learning, both in quality, representativity and number. Students can take pictures of the selected object with the smartphone's camera, or they can download images of equivalent object from the internet. For instance, taking pictures of a single apple will make a model better adapted to this specific type of apple, but with low generalization to other types. If students use a wider range of images of apples, taken or downloaded, the model will be more general, and the game could include different types of apples. In this case, the model learning is more complex, so obtaining a higher confidence is not easy. Teachers should highlight these concepts of generalization vs specificity because they are very important in the real application of the model, and the game experience for the user.

Select Model: students must select the parameters that configure the neural network and the learning algorithm, as explained in the theoretical part. A default set of values is recommended in the TU for 3 objects, but students should try different configurations and reason about the best result.

Add Testing Data: once the model has been trained, the application allows to test it. To do it, students must introduce a set of images different from the training one, that will be recognized by the model.

View Results: finally, the test results are displayed using the confidence measure previously explained. If the result is successful, the model can be downloaded to insert it in the App Inventor program. If it is not, students should analyze what is happening, and return to step 1 to include better images, or to step 2 to change the model. In TU4, assuming that students have used representative images, it is recommended to back to step 2 and try a different parameterization, as recommended by the teacher.

Once activity 1 is finished, it is time for the students to develop the Smartphone app in App Inventor. It is important to point out that the game development requires an intermediate level of programming skills, mainly in the knowledge of App Inventor. To cope with the existence of students with different initial levels and autonomy, two options for developing the app are proposed. Firstly, a guided one, where students can follow a programming template available at (TU4 2021). Secondly, an autonomous one, where only some basic specifications must be respected, but the programming is open. These specifications, valid for both options, are:

- Include the learned model in the app with the Personal Image Classifier component.
- Show the object name to be found on the screen.
- Have a viewfinder to capture pictures of detected objects.
- Show a timer on screen while the user is playing.
- Have a button, or similar, to check whether the captured object is correct or not.
- Show a message indicating if the captured object is correct or not.
- Control that all objects are found and show a message with the time spent.
- Reproduce a voice message when the game is over and ask the player if he/she wants to play again.
- Perform voice recognition to capture player's answer to play again.
- Restart or finish the game according to the user's answer.

The template includes the app structure in terms of screens, sections, program flow, and other functions already implemented. The aim is to avoid the aspects more specific of App Inventor and programming, leaving for implementation all those features that have to do with AI. Figure 2 displays a scheme of the template structure that is available for students, to help them to understand the functions included

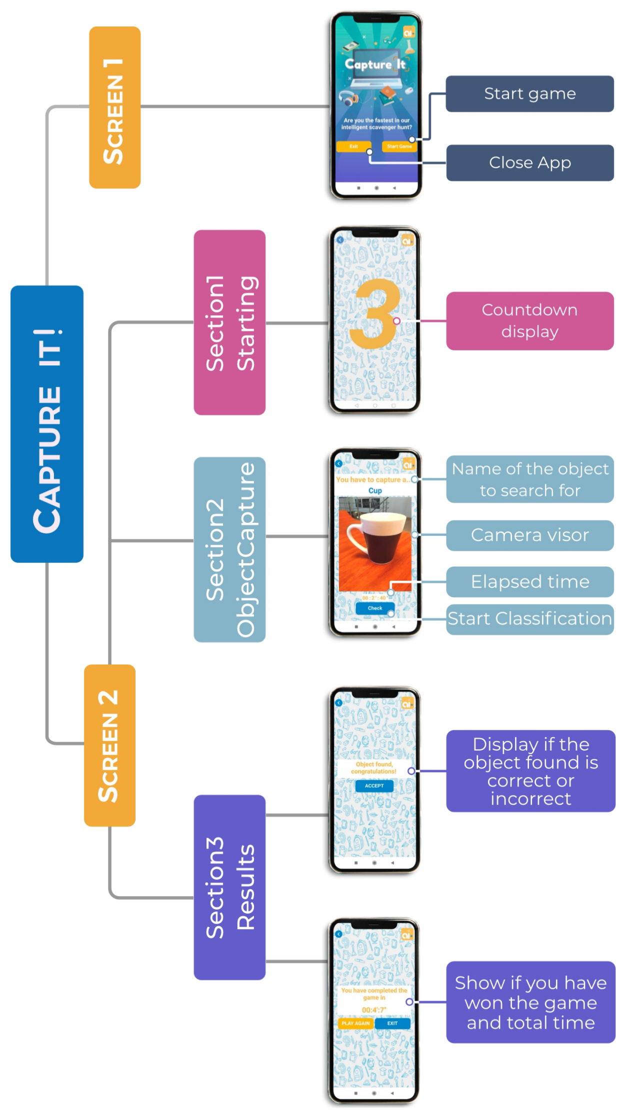

Figure 2. App Inventor template structure

on it. Right labels correspond to game functionalities that have code associated to them. Students must develop, mainly, all those under the Section2 “Object Capture”, as explained with detail in the TU pdf file (TU4 2021).

As this programming can be tricky, a total of 5 hours is recommended for students to perform it. A global solution is provided in the resource, mainly for the teachers (TU4 2021). Screen captures of the final game aspect are shown in Figure 4, which correspond to the template design. Left image corresponds to screen1, the opening screen. When the user clicks the start button, the middle image appears (screen2), where the user can see the name of the object to find, the camera viewer, and a button to check if the focused object is the correct one. This check button fires the model execution, that provides a confidence value. This value must be properly used by students to show if the detection was correct. This is what can be seen in the right image of Figure 4, in this case, displaying that the object was the expected one. The correct game functioning must be tested by playing the game with different students in several executions, so drawbacks can be detected and solved. It must be remarked to students to validate the model operation in real situations.

Additional Activity

Once the main version of the *Capture it!* game has been developed and tested, an optional activity can be proposed to improve the game experience, focused on artificial collective intelligence. The right part of the diagram of Figure 1 displays the steps that should be followed to develop this part, which has been organized in three sub-activities.

Activity 1 encompasses the theoretical introduction to students to artificial collective intelligence (multiagent systems). Again, in the proposal included at TU5, the student version of the TU contains links to reference material that can be autonomously checked by them.

Activities 2 and 3 are devoted with the main programming aspects to include the collective features in the game. They are based on the use of a cloud database in the program. In this case, it has been decided to use the one provided by App

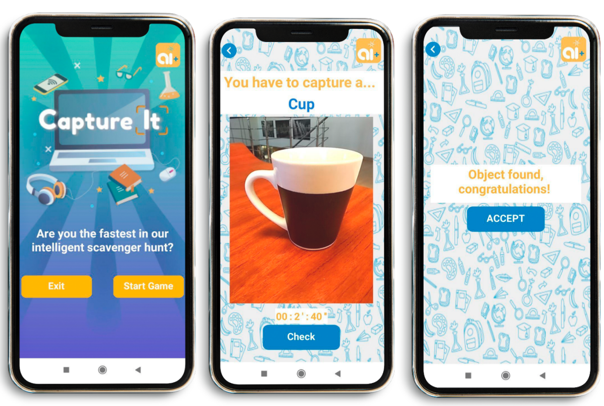

Figure 4. Final game aspect when using the template

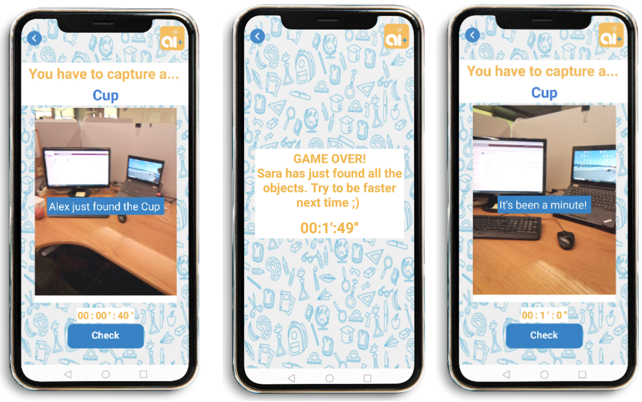

Figure 3. Game aspect in the additional activity

Inventor, *CloudDB*, which includes a component to store data in the cloud and allows multiple users of the app to communicate over multiple devices. As commented above, the learning outcome of this optional part is to realize the potential of using remote information and actuation in an AI system. It is important to remark that the intelligent features of this version do not come from using a database, but from the learned model in the individual apps.

Again, this optional part can be developed by students in a guided manner using a provided template, or in autonomous fashion. In both cases, they must start from their app version created in the previous activity, and modify it to include the three following improvements:

(1) Before starting the game, each player will have to log in with a username. This way, as shown in the left capture of Figure 3, every time an object is found, the specific object and the username can be remotely notified to the rest of players (Activity 2).

(2) The game finalization should be notified to all players at the same time (middle image). In the previous version, as players can be in different locations, the only way to establish the winner was to meet and compare the elapsed times (Activity 2).

(3) As time goes by, the application should send warnings to the players to let them know that they should hurry up. These messages should be different and more alarming as the minutes pass to capture the attention of the players. For instance, the right image of Figure 3 shows an example of message. In this improvement, it could be also notified that other team has already found a given object (Activity 3).

The final version of the collective game should be properly tested, in this case, by playing the game several times with different objects and different number of teams to validate the information exchange in real time.

Evaluation

To take the maximum advantage of this resource, a specific evaluation scheme is proposed, which focuses the assessment accordingly to the activity approach and methodology:

- *Final test of the app*: each group must show their app in normal operation, that is, playing the game. The teacher must perform a check of the final solution, reviewing if all the specifications have been considered, and if it can be used with reliability, mainly, if the learned model responds as expected. In addition, the students must submit their code, so the teacher can test it if required. As this is not a programming activity, the assessment emphasis should not be on the code quality, but in its functioning.
- *Final test of theoretical concepts*: at the end of the activity, students should fill a short survey focused in the main AI concepts treated, mainly, supervised machine learning and, optionally, artificial collective intelligence. An example was included in the evaluation section of the TUs.
- *Ongoing work*: the implication of students during the activity development should be considered too, being a very important aspect in the evaluation of any PBL methodology. This assessment should consider the work developed in each of the assigned roles. To this end, the TU contains individual rubrics that the teacher should fill every week.

Implementation Results

All the TUs already implemented in the scope of the AI+ project have been tested by the six partner schools before they were made available. In each of these schools, there is a group of teachers and students involved in their implementation and testing. Specifically, eight teachers and fifty students were enrolled in the AI+ during 2021, with different digital skills and ages (ranging always from 14 to 17). All of them carried out the two activities presented here during 2021. They were developed mainly in online fashion due to the COVID-19 restrictions, but most of them were carried out in groups of two, as recommended.

The version of the activity that has been described here was not the original one, but it has been improved after receiving feedback from the teachers. For instance, accordingly to the PBL methodology, the original version was mainly autonomous in terms of the app programming. Only design specifications were provided, and students could create the app on their own. But following teachers' recommendations, a more guided option was included, as explained above. One of the schools took this decision to focus learning in AI topics, avoiding programming ones. The remaining schools considered that their students did not have enough programming background in App Inventor to create a reliable app in an autonomous way, being this final functioning a key aspect in the final evaluation.

At the end of the activity realization, students answered a questionnaire to find out what they learned terms of machine learning. Analyzing the results, the main learning outcomes were successfully obtained, mainly in terms of understanding of the machine learning involved processes, and in the relevance of creating a reliable model that can be used in a real application, in this case, a game. The feedback forms provided by teachers, the results of this questionnaire for one of the groups (15 students), and some of the final implementations are available for review[2].

Inclusivity

In the AI+ project, and with the activity presented here, it is not only intended that students learn about AI and improve their digital skills, but also to reduce the gender gap and promote equal opportunities between regions. In the first issue, the global training in AI that is proposed for high school students, will provide girls with a clear view of this field before they enter to the tertiary education, opening them new opportunities to jobs requiring high quality specializations.

In the second case, providing all regions with open-access teaching material of high quality and technological impact, will help to provide equal opportunities between different states regardless of their economic capacity. In addition, a large majority of secondary school students have their own Smartphone, so they can use it for learning, as proposed in the project. This significantly reduces the cost of introducing this discipline and equalizes regions again.

Conclusions

A structured activity to introduce K12 students in supervised machine learning from an applied perspective has been described. This resource was implemented by 50 students in 6 different European schools in the realm of an Erasmus+ innovation project called AI+. The learning outcomes were very successful, mainly in the students' understanding about the potential of including machine learning models in real applications to provide them with valuable intelligent features. Moreover, and additional activity has been described, to introduce the basics of artificial collective intelligence by including remote sensing and actuation to the original app.

Acknowledgments

This work was partially funded by the Ministerio de Ciencia, Innovación y Universidades of Spain/FEDER (grant RTI2018-101114-B-I00), and Co-funded by the Erasmus+ Programme of the European Union through grant number 2019-1-ES01-KA201-065742. Moreover, the authors wish to acknowledge the support received from the CITIC research center, funded by Xunta de Galicia and European Regional Development Fund by grant ED431G 2019/01.

[2]https://bit.ly/3tpAKXo

References

AIplus project. 2019. Developing an Artificial Intelligence curriculum adapted to European high schools. http://aiplus.udc.es. Accessed: 2021-09-01.

AIplus results. 2021. https://aiplus.udc.es/results/. Accessed: 2021-09-01.

AI4K12. 2021. https://ai4k12.org/. Accessed: 2021-09-01.

Alippi. C. 2014. Intelligence for Embedded Systems: A Methodological Approach. *Springer Publishing Company*.

App Inventor. 2021. https://appinventor.mit.edu. Accessed: 2021-09-01.

European Commission. 2018. Coordinated Plan on Artificial Intelligence (COM (2018) 795 Final). https://bit.ly/3hdLbsg. Accessed: 2021-09-01.

Guerreiro-Santalla, S., Bellas, F., Duro, R.J. 2020. Artificial Intelligence in Pre-University Education: What and how to teach. *Proceedings 3rd XoveTIC*, vol. 54, pp. 48. MDPI.

Gresse von Wangenheim, C., Hauck, J.C.R., Pacheco, F.S. et al. 2021. Visual tools for teaching machine learning in K-12: A ten-year systematic mapping. *Educ Inf Technol* 26, 5733–5778. https://doi.org/10.1007/s10639-021-10570-8

Home Assistant. 2021. https://www.home-assistant.io. Accessed: 2021-09-01.

Marques, L. Gresse von Wangenheim, C., Hauck, J. 2020. Teaching Machine Learning in School: A Systematic Mapping of the State of the Art, *Informatics in Education 19*, no. 2, 283-321

PIC. 2021. Personal Image Classifier module. https://classifier.appinventor.mit.edu. Accessed: 2021-09-01.

Russell, S. & Norvig, P. 2021. Artificial Intelligence. A Modern Approach. *Pearson*

Robobo. 2021. The Robobo Project website. https://theroboboproject.com/en/. Accessed: 2021-09-01.

TU4. 2021. Resource material for teaching unit 4. https://bit.ly/3lcBXh7. Accessed: 2021-09-01.

TU5. 2021. Resource material for teaching unit 5. https://bit.ly/38URhJe. Accessed: 2021-09-01.

Weiss, G. 2000. Multiagent systems: a modern approach to distributed artificial intelligence. *MIT Press*, Cambridge, MA, USA.

Xunta. 2021. Official website with the new subject formal publication. https://bit.ly/38RCmiW. Accessed: 2021-09-01.