Pattern Recognition Letters
journal homepage: www.elsevier.com

Improving Smart Interactive Experiences in Cultural Heritage through Pattern Recognition Techniques

Fabrizio Balducci, Paolo Buono, Giuseppe Desolda, Donato Impedovo, Antonio Piccinno

Dipartimento di Informatica, Università degli Studi di Bari, Via Orabona, 4 – 70125 – Bari, Italy

ABSTRACT

New Information and Communication Technologies have a large potential to improve general public awareness of the importance of Cultural Heritage (CH) and to provide tools that can make visits to historical sites more interesting and enjoyable. The Internet of Things (IoT) technology can further contribute to these goals, by allowing visitors to museum and CH sites to manipulate smart objects by receiving information that stimulates emotions, understanding and appropriation of the contents. In our research, interaction paradigms and innovative methods are developed to allow curators and guides of cultural sites (i.e., domain experts) to manage interactive IoT-based environments, in order to create *Smart Interactive Experiences*, which are usage situations created by synchronizing many available smart objects to specific situations that might better satisfy the needs of the visitors. This article illustrates a system that, by means of a tangible user interface, integrated by pattern recognition and computer vision techniques, supports CH experts in creating Smart Interactive Experiences by properly tailoring the behavior of the involved smart objects. An experimental evaluation of the used techniques has been performed and it is presented and discussed.

1. Introduction

People are becoming increasingly aware that Cultural Heritage (CH) must be preserved and promoted, so that it can be better appreciated [1, 2]. New information and communication technologies have a large potential for helping to reach this goal, offering new communication channels to improve the general public awareness of the importance of CH. Many countries are rich of museums and historical sites, such as archaeological parks, whose visitors are primarily middle-school students. There is evidence that traditional visits do not fully engage such young students, especially in the case of sites whose current appearance as ruins no longer reflects their initial appearance and purpose [3].

Internet of Things (IoT) is emerging as an effective means to connect the physical dimension of museums and exhibitions with digital cross-media information, thus resulting in a very promising technology able to enhance access to CH collections [4-9]. Thanks to the development of 'smart' museums, visitors can manipulate and interact with smart objects reproducing, for example, archaeological artefacts (a cup, a vase, etc.), in order to increase their overall experience and appropriation of contents [10-14].

The availability of interconnected smart objects in a museum makes possible to create the so-called *Smart Interactive Experiences* (SIEs – pronounced "*see-ehs"*), namely usage situations created by synchronizing many available smart objects to specific situations that might better satisfy the needs of the visitors [15]. For example, a SIE could be a game that a professional guide creates by properly orchestrating the behavior of several smart objects available in a museum to let people experience it when visiting the museum.

In most IoT environments, SIEs are pre-packaged and the involved smart objects cannot be easily adapted when either exhibits or type of visitors change. Currently, in the CH domain a few approaches try to facilitate the configuration of smart objects in IoT environments [11]. One big challenge to make IoT have a significant practical and social impact is to develop tools and techniques to support non-technical people to properly manage a great variety of smart devices. In this way, they can create various SIEs able to improve the overall experience of final users interacting with IoT environments.

To address this challenge and promote the creation of SIEs that enhance people fruition of CH assets, we are working on systems that, by implementing different interaction paradigms suitable for non-technical people, enable domain experts, like museum curators or professional guides, to tailor the behavior of smart objects involved in a SIE. To this aim, two iterative phases are devised. First, domain experts enrich SIE resources (e.g., smart devices) with semantic properties relevant to their domain knowledge. For example, referring to the paintings exhibited in a museum, semantic properties might be the artist's name, type of painting, etc. Second, domain experts express the smart object's behavior, i.e., the SIE dynamics, by creating Event-Condition-Action (ECA) rules based on the semantic properties [14]. For example, they can enable specific smart objects to react when they are close to artworks containing some properties (e.g., the artwork of a specific painter). ECA rule creation is supported by a visual interaction paradigm that does not require domain experts to write any code in a formal programming language.

Three prototypes of systems for SIE design have been presented in [15]. The prototypes were instrumental for performing a user study that compared the different design paradigms used in each case, with the aim of identifying their strengths and weaknesses.

Within this context, this article proposes and tests the adoption of innovative module that 1) recognizes and classifies specific elements (smart objects, QR codes, Tangible Attributes, handwritten post-it) within a picture, understanding relationships among them and grouping them hierarchically through their distance and position and 2) recognizes handwritten strings containing characters, location and numbers. This module is essential for the three system prototypes to be used in real contexts, since it facilitates the definition of the smart object semantic properties. One of the three prototype systems is considered as testbed, and it is shown how the new module supports domain experts in both identifying the relevant semantic properties and associating them to the specific IoT resources.

The paper is organized as follows. Section 2 discusses some related work. Section 3 illustrates a SIE and how domain experts can create it, while Section 4 presents the Tangible system. Section 5 describes how pattern recognition algorithms are integrated into a system supporting non-technical users to define the semantic properties of the resources needed to create a SIE. Section 6 presents the results of an experiment carried out to assess the performance of the proposed algorithms. Section 7 discusses these results and concludes the paper by focusing on future work.

2. Related work

The IoT phenomenon has been largely investigated on the technical side [16, 17]. Although some approaches try to facilitate the configuration of single smart objects [11], it is still hard for non-technical stakeholders to synchronize the behavior of multiple physical and virtual resources, installed in the environment or embedded in tangible objects, manipulated by the final users.

There is an increasing interest in addressing this limitation [18]. Task Automation Systems (TASs) offer visual paradigms to assist non-technical users in defining ECA rules [19]. They are used to specify the behavior of a smart object by indicating one or more events and one or more conditions that must occur to activate specific actions, i.e. operations on data or functions available on a smart resource. Some TASs enable people to compose only simple rules, such as *IFTTT* [20], *elastic.io* [21], *Zapier* [22], *itDuzzit* [23], *WigWag* [24]. These tools are far from supporting the creation of SIEs, which involve many physical and digital resources. Advanced TASs support complex rule definitions, addressing more real situations. However, they require specific knowledge and programming skills. Examples are *Node-RED* [25], *Microsoft Flow* [26], *Crosser* [27]. Recent research proposes different paradigms for supporting non-technical people in defining more complex ECA rules. The interested reader may refer to the special issue on "End-User Development for the Internet of Things" [28] (see in particular [29-31]).

Ontologies are used to build a semantic layer where high-level concepts provide an abstract and technology-independent representation of the smart objects [32-34]. Thus, users define ECA rules by referring to ontology concepts without worrying about technical details [32]. For this semantic enrichment, experts have to create ontologies and to define the mapping between ontological concepts and smart objects. This requires technical skills and a significant effort, still exposing the system to the risk of not covering the actual needs of SIE designers. In alternative, Ardito et al. propose a visual framework that empowers non-technical SIE designers to build semantic layers for TASs based on the definition of *custom attributes* [14]: they are a means to add domain knowledge that can simplify the definition of complex ECA rules; in other words, they enable SIE designers to express the operational semantics they want to assign to the SIE resources depending on the specific usage situation they are interested in.

The design of SIEs targeted at visitors to CH sites includes tangible objects that visitors can bring with them, touch and manipulate, and also receive personalized information [11, 13, 35]. The interaction with tangible objects activates real-world knowledge improving memory [10, 12], and favors emotions, engagement, understanding, thus increaseing the appropriation of CH content [11]. These motivations are the basis of the prototype systems for SIE design presented in [15]. The user studies performed with these prototypes provided useful hints for our current research that, as reported in this article, is aimed at developing a usable system that can support SIE design in a real context. To this aim, an efficient vision system to recognize objects is required, because both smart objects and custom attributes are represented by physical objects that must be precisely detected. Object detection is, indeed, one of the areas that is maturing rapidly also thanks to deep learning innovation.

Current object detection methods are typically based on Convolutional Neural Network (CNN) models, able to automatically recognize visual features exploiting different architectures [36]. One of the first models featuring convolutions and shared weights was *LeNet* [37]. However, the spread of the deep learning approach for image and object classification was determined by *AlexNet* [38], developed in 2012 as an enhanced version of LeNet. *ZFNet* [39] further improved AlexNet by exploiting deconvolution network, while *GoogLeNet* [40] introduced the Inception module reducing the number of network parameters. In 2016, the residual network *ResNet* became the state-of-the-art for the practical use of such models [41]. Extensions of the CNN model have been introduced as *Recurrent Neural Networks* (RNN) [42] to learn long-term dependencies and their enhancement with *Long Short-Term Memory Networks* (LSTM) [43], while *ConvLSTM* [44] proved to have excellent spatio-temporal sequence prediction capability. As a final remark, RNN must not be confused with *Region-based Convolutional Neural Network* (R-CNN) [45] and their immediate descendant *Fast-R-CNN* [46], where image regions are exploited for feature extraction classifying all regions according to their common features.

3. Smart Interactive Experiences

Smart Interactive Experiences (SIEs) are usage situations created by synchronizing the behavior of multiple smart objects. In CH, SIEs improve visitors' engagement and exhibition appropriation, as they allow people to shape their personal experience while interacting with smart museums, sites or exhibitions [4-8].

To simplify the definition of ECA rules adopted by SIE designers, some authors of this paper recently proposed *custom attributes* as conceptual tools that allow domain experts to externalize their tacit knowledge [14]. These attributes contribute to the definition of a *domain-oriented semantic* which enriches the system for SIE design with operational meanings that allow domain experts to characterize the role of smart objects in a specific usage situation. Similarly to ontology concepts (e.g., see [32]), custom attributes are meant to add knowledge that can simplify the definition of ECA rules.

The solution we propose is that, for each smart object, the domain expert defines properties (which, more technically, are attributes of the object) that can express the meaning and the role of an object according to the SIE dynamics. To better understand the notion of custom attributes and their value for the design of SIEs, we report in the following an example scenario where a SIE is a treasure hunt game designed by a professional guide of a museum. The SIE will be played by visitors of a museum. The professional guide modifies the game by properly configuring the

involved smart objects by means of ECA rules. It is worth remarking that in our research experience we have created various games to improve the fruition of CH content. Games have been proved to be very valuable for transforming a sometimes boring visit to a CH site in engaging user experiences, and are also effective from an educational point of view [47-50].

Scenario: Brando is a guide at the museum of Egnazia (Italy) where visitors can watch artefacts of different periods like the Messapian age, the Roman age and the late Roman age. The display cases are already equipped with RFID tags that visitors can read to obtain additional information. To make the exhibition more engaging, Brando organizes the visit as a treasure hunt. During the tour, Brando asks visitors to identify cases that display artefacts with a specific characteristic, for example, artefacts related to a specific age (e.g., Bronze, Iron, Messapian, Trajan) or to a specific purpose (e.g., fighting, cooking, personal care). To answer Brando's quest, visitors must identify the cases that contain the right artefacts. Visitors will be provided with smart magnifying glasses that are able to recognize display cases and their properties. To answer Brando's quests, visitors have to look at the case through their smart magnifying glass and push a button. If the answer is right, a video file providing additional information about the case artefacts is played on the magnifying glass. Also, the visitor gets some points as a reward. Otherwise, the magnifying glass shows a video indicating that the case does not match the quest. The treasure hunt continues with visitors performing all the proposed quests. The winner is the visitor that, at the end of the treasure hunt, gets the highest score.

To define this visit, i.e., to create this SIE, Brando must configure the involved smart objects, i.e., the smart magnifying glasses and the display cases. Smart magnifying glasses appear like the traditional ones but, instead of the glass, they integrate a rounded display that visualizes the output of a camera installed on the backside, thus creating the effect of looking through a real lens. How can Brando design the treasure hunt by synchronizing the behavior of all such smart objects? The solution we propose is that, for each smart object, he defines properties that can express the meaning and the role of an object according to the game dynamics. For example, each smart magnifying glass is used to identify the group who carries it during the game. Thus, a possible attribute is "Group" (with values: Group1, Group2, etc.). Similarly, it is possible to enrich the display cases with attributes such as "Age" (with values: Iron, Bronze, Trajan, Messapian), "Purpose" (with values: fight, cooking, personal care), "Video file" (with values indicating the name of video file to be played on the smart magnifying glass when the case is the right one). Brando defines these attributes and their values according to the SIE he wants to create, without any constraint (syntactic or semantics) on the type of properties to be specified. This is the reason why they are called *custom attributes*.

After defining custom attributes, Brando specifies the ECA rules controlling the behaviors of the smart objects. He uses a TAS implementing a visual paradigm that simplifies rule creation [30]. An example of a rule, which for brevity we represent here in formal syntax, is:

Rule$_1$: "IF a *smart magnifying glass* is close to a *case*
WHERE *case.Age = quest.Age*
THEN *smart magnifying glass* plays *case.audio_file*"

Assigning custom attributes to objects has two main advantages when creating ECA rules. First, the language used to define the rules is closer to the domain-expert language since the variables used in the rules are the attributes previously defined by the domain expert himself. Without custom attributes, domain experts should deal with a low level of abstractions that force them to be aware of technological aspects (e.g., smart object identifiers, sensor names). Second, the attributes introduce high-level abstractions that favour generalization. Indeed, TASs that are not able to exploit custom attributes are not effective for defining ECA rules for SIEs, where dozens of smart objects are involved. For example, in the above scenario, Brando must create a set of rules like the following, which for brevity we represent here in a formal syntax:

Rule$_a$: "IF the *smart_magnifying_glass_015.RFID_reader*
detects *RFID_tag_012654*
WHERE the current quest is *Age = Iron age*
THEN the *smart_magnifying_glass_015* plays
video029.avi".

This rule has to be adapted and replicated for each case and for each smart magnifying glass. For example, if in Brando's SIE there are *n* lenses and *m* cases, he must replicate Rule$_a$ *n* x *m* times. Thanks to the custom attributes, a single rule addresses this group of rules or, in general, an entire class of devices with the same behaviour.

4. Supporting SIE design

Three prototype systems that implement advanced interaction paradigms for defining custom attributes were recently created. They are implemented as part of a wider architecture that allows designers to flexibly combine those elements of the three design paradigms that best suit the design situations they have to cope with. This architecture allows designers to switch among the three prototypes, or further ones, offering a cross-modality. Details on all the three prototypes and the underlying architecture are reported in [15].

This paper focuses on the recognition algorithms that can be implemented in all of the three prototypes, thus we describe how they are used by only the Tangible system prototype.

4.1. The Tangible system

The Tangible system is based on the idea that domain experts can manipulate real objects to support tangible thinking, which is the ability to think by means of the manipulation of objects augmented with digital information [51], as well as to exploit the capability of tangible interaction to stimulate creative thinking [52, 53]. With the Tangible system, domain experts manipulate two kinds of tangibles. The first ones are the smart objects of the SIE. If some smart objects cannot be manipulated due to their size (e.g. a statue), location (e.g., fix installation), or for other reasons, domain experts can use a small representatives like pawns in a role-playing game. To this aim, we introduced objects with elementary shapes (parallelepiped, cylinder, sphere, pyramid), whose affordance can refer to the original smart object (e.g. a parallelepiped can be used to refer to the smart display case). The second tangible that SIE designers manipulate for defining custom attributes are tangible attributes. Three types of tangible attributes, namely pen, compass and dice, are adopted to define textual, locational and numerical attributes, respectively.

To illustrate how the Tangible system works, in the following we report how Brando defines custom attributes for his game. He puts on a table some of the smart objects involved in the SIE, e.g., a magnifying glass and a display case represented by a red cube (Figure 1). Then, he puts proper tangible attributes close to the smart objects. For example, he puts the pen close to the smart magnifying glass to indicate the definition of a textual attribute; then he attaches a post-it to the pen to specify the name and value of the attribute, e.g., "Group = group 1". He repeats the same actions for each custom attribute he wants to define (see Figure 1).

Then he uses a mobile app to take a picture of all the elements on the table. The pattern recognition algorithms described in Section 5 recognize the elements on the table, which are automatically converted into the definition of custom attributes (<attribute name = value> pairs). Thus, at the end, the smart magnifying glass in the system is enriched with the custom attributes <Group = group 1> (textual). Similarly, the case (represented by the red cube) is characterized by the attributes <Age = Messapian> (textual), <Points = 3> (numerical) and <Purpose = fight> (textual).

After the definition of the custom attributes, Brando proceeds with the creation of the ECA rules (like the one reported in Section 3) that allow him to specify the behaviour of the smart objects in the SIE. ECA rules are created by using one of the visual interfaces presented in [30].

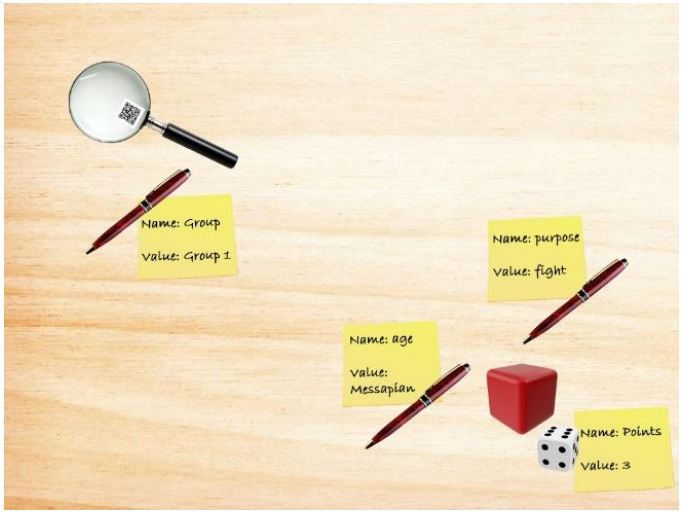

Figure 1. Tangible system: smart object, tangible attributes and post-it notes used to define the custom attributes for a smart magnifying glass and a case represented by a cube. This figure is a reproduction of the real system for clarity purposes.

5. Automatic Scene Understanding and Recognition

The recognition algorithms developed to detect smart objects and tangible attributes have been integrated into a distributed system deployed in a cloud virtual environment. When the SIE designer takes a picture of the table in Figure 1 with his/her mobile device, the picture is sent to the remote system, which recognizes all the objects and translates the original physical composition into custom attributes associated with smart objects. The results are sent back to the mobile devices. To this aim, the remote system runs the following four phases:

1. identification of the position of all the elements in the picture;
2. classification of specific elements (smart object representatives, attribute object, QR code, post-it);
3. understanding of the relationships between the classified elements, grouping them hierarchically;
4. handwriting recognition.

For each of these phases, a specific sub-module has been developed. For the first two phases the *RetinaNet*, a Convolutional Neural Network (CNN) model based on *Resnet-50*, has been employed [54]. *RetinaNet* is a unified network responsible for processing a convolutional feature map of the entire input image. The one-stage detector uses as feature extraction backbone, i.e. the ResNet architecture together with a Feature Pyramids Network (FPN) while at each pyramidal level are attached two specialized FCN subnetworks for classification and bounding box regression which benefits from a focal loss. The model has been pre-trained on COCO (Common Objects in COntext) [55], a large image dataset for object detection, segmentation, key point detection and caption generation. The COCO dataset contains 2.5 million instances belonging to 91 classes, labeled in 328,000 images, allowing the user to customize a CNN model that can be specialized by further training on new classes and object patterns. In this work, the transfer learning technique has been exploited by re-training only the classification layers by adding new classes in the final part of the CNN architecture while the backbone for feature selection is preserved (Figure 2) thus saving computational time and resources. The QR code recognition and translation is performed in the second phase. using brightness and contrast filters and exploiting an open-source standard tool that draws bounding boxes around the QR patterns in the image.

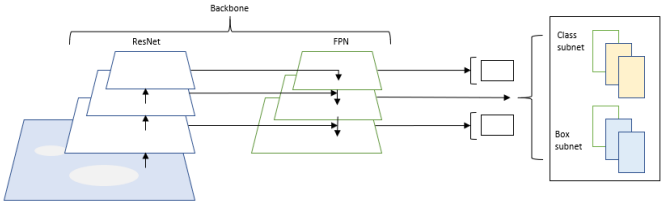

Figure 2. The RetinaNet architecture scheme featuring a CNN and a FPN as backbone followed by two subnetworks for classify anchor boxes and for their regression to ground-truth boxes which are customizable while exploiting the transfer learning technique.

In the third phase, the relationships between the recognized elements is computed using the Euclidean distance. Given a set A of attributes, a set P of post-its and a set O of Smart Objects, each post-it p_i is associated with the nearest attribute object a_i, thus composing the group $\{p_i, a_i\}$. Then, each attribute a_i is associated with the nearest smart object (or QR code) o_i. This process is iterated until all attributes are associated with a smart object (or QR code).

Finally, in the fourth phase, two different models based on LeNet-5 classify the handwritten characters extracted. The two models are pre-trained on MNIST and EMNIST. MNIST dataset consists of 6,000 images for each of 10 classes as a training set and 1,000 images per class as a test set while to classify alphabetic characters, the EMNIST Letters [56] dataset, with 4,800 images for each of the 26 classes for training and 800 images per class was used as a test.

Several image pre-processing functions have been applied to optimize performances. After the post-it area crop and before translating the handwritten text, a series of image processing operations are performed, summarized in Figure 2: the post-it area is resized keeping its proportions, then the image is converted to grayscale color space and the contrast is increased by 40% to make the strokes more evident (Figure 3a); the brightness and contrast are increased to remove the background noise. Finally, the image binarization and colour reversal facilitate the creation of pixel histograms (Figure 3b).

A first general pixel histogram is created by adding the pixel values for each column (image width) as a function of the x-axis with a high threshold to eliminate areas with strong noise; then a histogram is created considering pixel values for each line (image height) according to the y-axis, with the aim of cutting out the horizontal stripes also in case that there are more text lines (Figure 3c). Then, for each stripe, histograms are created for columns (image width) according to the x-axis, allowing the cutting of possible character/digit patterns (Figure 3d). For each cropped piece, the black rows and columns are removed, resizing to 20x20 pixels and maintaining the proportions while, finally, a padding of

black pixels (Figure 3e) that keeps the pattern centred is added due to the compatibility with the MNIST/EMNIST image format.

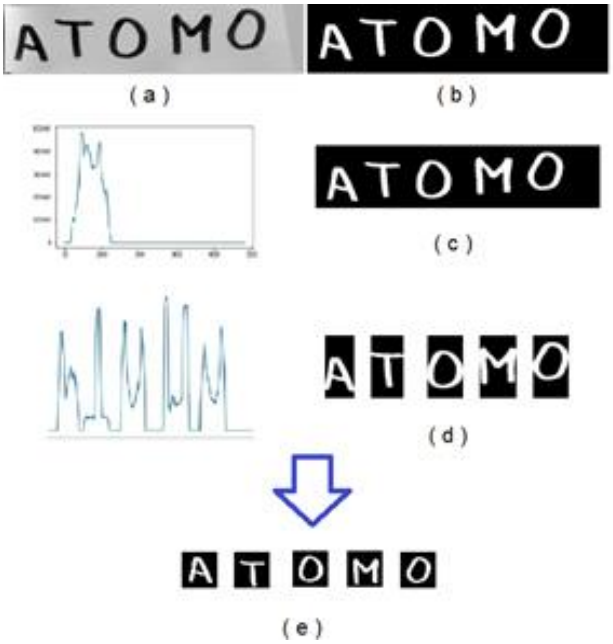

Figure 3. A summary of the image processing pipeline to extract handwritten characters from a post-it image exploiting pixel histograms.

During the analysis and design phases of the recognition system, the following technical constraints have been identified to ensure the practical use of the system:

1. the picture background on which all objects are positioned must preferably have a uniform colour/pattern;
2. the maximum number of smart-objects, QR codes and *tangible attributes* within a single picture should be limited;

The adopted SIE prototype reflects the following limitations in the handwriting tasks:

1. the written stroke follows a horizontal trend having the uniformity and thickness of a felt-tip pen;
2. characters cannot overlap;
3. the characters (alphabetical and numerical) must be written following the MNIST/EMNIST standard.

In other words, the previous limitations are due to the fact that allowed handwritten text is not based on a Natural Language grammars but admits a limited set of words, so that the described solution can be referred as an "ad-hoc" one. To overcome these limitations and in order to be able to generalize results to a real scenario, different state-of-the-art handwriting recognition models have been considered and compared [57-60]. This solution introduces a different approach to the post-it processing, so that explicit character segmentation can be avoided.

The recognition techniques considered come from the combination of feature extraction backbones (Resnet or VGG) with an optional Thin-plate-spline (TPS) function to normalize input text images, a Bidirectional LSTM (BiLSTM) as sequence modeling and a prediction stage based on the Connectionist Temporal Classification (CTC) or an Attention mechanism (Attn) leading to the following techniques:

1) TPS – Resnet - BiLSTM – Attn [57, 59, 60];

2) TPS – Resnet - BiLSTM – CTC [57, 58, 60];

3) None – VGG - BiLSTM – CTC [57, 59, 60].

In the experimental session, the "ad-hoc" solution is referred as MNIST + EMNIST, while results related to the generalized models are referred as in the previous list.

6. Experimental Phase

In order to test the system functionality illustrated in Section 5, experimental objects with different patterns and sizes have been selected together with three tangible attributes (pen, compass, and dice) having various patterns (see Table 1).

Table 1. The elements (*smart object representatives, tangible attributes, post-it and QR codes*) used in the experiment and their features.

Object	**Size (cm)**	**Colors**	**Function**
Pyramid	1.5 x 1.5 x 2.2	green, brown	smart object represent.
Sphere	1.5	orange, black	smart object represent.
Cylinder	1.5 x 1.5 x 6.1	violet	smart object represent.
Parallelepiped	7.3 x 2.5 x 0.4	blue	smart object represent.
Dice	1.0 x 1.0 x 1.0	red	attribute object
Pen	9.0 x 0.6 x 0.6	black	attribute object
Compass	6.5 x 6.5 x 0.7	white, yellow, blue	attribute object
Post-it	7.5 x 7.5	yellow	handwritten values of a tangible attribute
QR code	3.5 x 3.5	black, white	placeholder objects that are too large

6.1. Setup and Dataset

The recognition system is developed in a Linux environment, exploiting deep learning models with *Keras* and *Python*. CNN training has been performed with an NVIDIA GTX 1060 with 6 GB of VRAM and with an NVIDIA Jetson TX2, which has also been used for testing.

Two experimental datasets have been produced acquiring pictures with a Huawei Mate 20 Lite smartphone at a resolution of 5120x3840 pixels. The Training dataset consists of 70 pictures for each of the eight classes listed in Table 1, used to train the CNN to detect and classify the object's patterns for a total of 560 pictures. The pictures of a single class portray the object from different angles and distances. Considering each 'smart object' class, about 5% of their training images contains the object together with others in order to provide the CNN examples of patterns grouped together. When considering the training pictures for an 'attribute object', on the other hand, the amount increases to about 15% since it is more likely that in a picture the tangible attributes are repeated. The XML ground-truth annotation for each element in the training image consists of 'name' tag and the bounding box coordinates. The optimal parameters found for the CNN training were batch-size=1, steps=653, epochs=9 also exploiting data augmentation techniques (flip, rotation, scaling, translation, shear). Moreover, custom anchors to optimize peculiar pattern recognition have been added because the pen pattern (thin and elongated) was difficult to identify, despite the training.

The Test dataset developed for this study consists in set of 102 pictures that in total contain 290 distinct objects (Smart Objects and QR codes) and 533 Tangible Attributes each of which with an attached handwritten post-it. All the pictures have been taken from two different points-of-view: orthogonal to the surface where objects stand and at 60 degrees between the perpendicular of the surface and the observer. Three experimental classes have been designed (each with 34 images): the Easy class features 1 or 2 objects with a few attributes; the Medium class features 3 associated with multiple (2 or 3) tangible attributes, while the Hard

class presents complex scenarios with 4 elements, even reaching more than 20 objects to recognize at once.

To automate the result checking, each picture has been labeled through a CSV ground-truth file with an ordered list of the present elements, each with a sub-list of the related tangible attributes and text (post-it). The resulting output has the same format as the ground-truth with the addition of the bounding-box coordinates for each recognized element.

6.2. Metrics and Results

To express the effectiveness of the system in the object classification (smart objects and QR codes), the *Accuracy* metric is computed as the ratio between correctly classified elements and their total number in the picture. Attribute and post-it objects are evaluated using a *Group Accuracy* metrics, which indicates that the pattern has been correctly classified and has been assigned to the related smart object. The handwritten text matching employs two metrics: a *total matching* with the ground-truth text and a normalized *Hamming distance* (ratio between recognized characters in the right positions and the longest text).

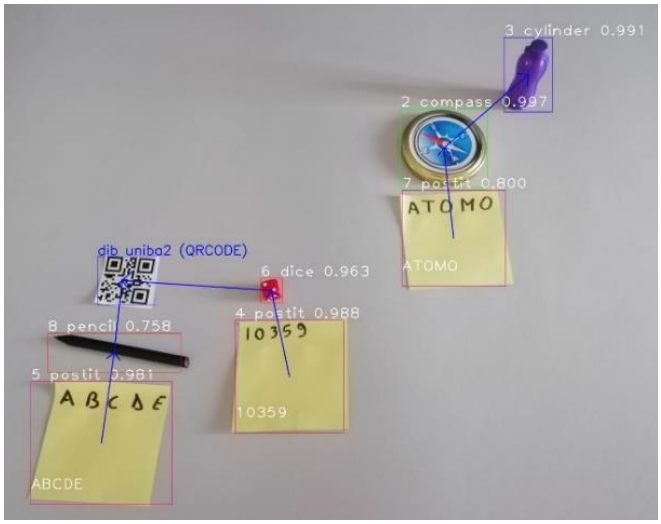

Figure 4**.** Classification and hierarchical grouping done by the System.

Figure 4 visually depicts the output of the pipeline. The bounding box, class name and confidence highlight the element classification while the straight blue lines show the hierarchical grouping. The results for the three experimental classes in Table 2 present a Classification Accuracy of 273/290 = **94.1%** (for SO it is 146/154 = 94.8% and for QR codes it is 127/136 = 93.4%), while for tangible attributes the overall Grouping Accuracy is 533/563 = **94.7%**. Notice that the overall Classification Accuracy is beyond **93%** for all the experimental scenarios but, while for the QR codes it decreases when complexity increases, for the Smart Objects the 'Easy' and 'Hard' settings are very close (**96.4%** vs. **96.2%**). The Grouping Accuracy regarding the tangible attributes follows almost the same trend as the Classification Accuracy and always remains over **93%**.

The evaluation of the handwritten text matching is in Table 3 with a Total Matching accuracy of 287/533=**53.8%** and a Normalized Hamming that reaches an average accuracy of **80.1%**. reports a Total Matching metric of 287/533=**53.8%** and a Normalized Hamming that reaches **80.4%**.

Thus, there are some characters that are hard to recognize (for instance 1, 4 and 7; zero is mismatched with O). Table 3 reports the results for the three experimental classes. Notice that while the Total Matching remains about 50%, the Normalized Hamming is always over 79%. Although there are works that achieve better performances using CNN [61, 62], when comparing their time performances, the presented method is faster and computationally inexpensive in the high-complex settings previously exposed, allowing a real-time object classification pipeline with handwritten text recognition facilities. Indeed, the entire pipeline execution requires an average time of about 2 seconds, of which 1.2s for the QR code recognition, which appears to be the bottleneck for real-time usage.

Table 2. Results for the experimental classes with the Classification Accuracy for Smart Object (SO) and QR code (QR), and the Grouping Accuracy for the tangible attributes (Pen, Compass, Dice, post-it elements).

Experimental class	**Objects (Smart Obj. and QR codes)**		**Tangible attributes**
Easy (1/2 elements)	51/52 = **98.1%**	**SO:** 27/28 = **96.4%** **QR**: 24/24 = **100%**	73/75 = **97.3%**
Medium (3 elements)	95/102 = **93.1%**	**SO**: 42/46 = **91.3%** **QR**: 53/56 = **94.6%**	208/222 = **93.7%**
Hard (4 elements)	127/136=**93.4%**	**SO**: 77/80 = **96.2%** **QR**: 50/56 = **89.3%**	252/266 = **94.7%**

Table 3. Results for the handwritten text correspondence exploiting two evaluation metrics: Total Matching and Normalized Hamming.

Metric	**Easy class**	**Medium class**	**Hard class**
Total Matching	44/73 = 60.3%	103/208 = 49.5%	140/252= 55.6%
Normal. Hamming	**79.1%**	**81.9%**	**79.4%**

When considering the explorative study about the new handwritten recognition methods, results in Table 4 show that not all the new techniques used for text recognition achieve better results than the originally MNIST/EMNIST proposed method. The Resnet - BiLSTM – Attn is the combination that outperforms previous results with a +4.4% for the total string matching (between ground truth and the predicted one) and with a +4% considering the Hamming match metric. For techniques 2 and 3 instead, the Normalized Hamming metric is -5.6% and -3.3% respectively and the Total Matching results to be more than 20% lower in technique 3 when compared with the original.

Table 4. Results for the handwritten text correspondence exploiting techniques based on state-of-art CNN with the analysis of the overall post-it written line instead of a character to character approach.

Technique	**Total Matching**	**Normal. Hamming (avg)**
0) MNIST + EMNIST	53.8 %	80.1 %
1) TPS - Resnet - BiLSTM - Attn	**58.6 %**	**84.1 %**
2) None - VGG - BiLSTM – CTC	43.3 %	76.8 %
3) TPS - Resnet - BiLSTM - CTC	29.3 %	74.5 %

It must be underlined that the obtained results confirm the ones reported in the comparative study by Baek et al. [60]. The Resnet - BiLSTM - Attn emerges as the best choice also able to outperform the "ad-hoc" solution (MNIST+EMNIST) initially adopted due to SIE prototype limitations.

7. Discussion and conclusions

This article has presented a system that supports CH experts in creating Smart Interactive Experiences by enabling them to

manage various smart objects in an IoT environment. Pattern recognition and computer vision techniques, like convolutional neural networks, were introduced to automate the definition of semantic properties associated with the smart objects involved in a SIE. Promising and encouraging results emerged during the evaluation of the techniques used. The recognition of smart objects, QR codes and tangible attributes have an accuracy over 90% even in complex configurations with several objects in the same picture. A precision of about 84% resulted for the recognition of handwritten text on post-it notes. At the same time, it must be considered that texts that are wrongly recognized can be quickly edited via the app as well as post-processed by a grammar corrector. Proposed solutions have been considered within a real-time application and can be further extended considering other languages [61, 63, 64].

As future work, we are planning to evaluate the overall system by involving CH experts in the creation of a SIE. Contrary to previous studies conducted in a controlled environment, the system will be evaluated in a more ecological environment during a field study. This will be possible thanks to the advances presented in this paper, which made our initial prototypes evolve into a more powerful system. In addition, we also aim to improve the proposed recognition techniques considering a wider set of real use cases.

Acknowledgments

This work is partially funded by e-Shelf, POR Puglia FESR-FSE 2014-2020 ID: OSW3NO1. We are grateful to the NVIDIA for supporting our research giving us a Jetson TX2 through the Grant Academic Program in 2018.

References

1. Copeland T. 2004. Presenting archaeology to the public. In: Merriman T. (Eds), *Public Archaeology*. Routledge, 132-44.
2. Merriman T. 2004. *Public Archaeology*. Routledge, London, UK.
3. Costabile M. F., De Angeli A., Lanzilotti R., Ardito C., Buono P., and Pederson T. 2008. Explore! possibilities and challenges of mobile learning. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 145-54.
4. Uskov A. and Sekar B. 2015. Smart Gamification and Smart Serious Games. In: Sharma D., Favorskaya M., Jain L.C., andHowlett R.J. (Eds), *Fusion of Smart, Multimedia and Computer Gaming Technologies: Research, Systems and Perspectives*. Springer, 7-36.
5. Madeira R. N., Correia N., Dias A. C., Guerra M., Postolache O., and Postolache G. 2011. Designing personalized therapeutic serious games for a pervasive assistive environment. In *Proc. of Conference on Serious Games and Applications for Health (SeGAH '11)*, 1-10.
6. Andreoli R., Corolla A., Faggiano A., Malandrino D., Pirozzi D., Ranaldi M., Santangelo G., and Scarano V. 2017. A Framework to Design, Develop, and Evaluate Immersive and Collaborative Serious Games in Cultural Heritage. *ACM Journal on Computing and Cultural Heritage*, 11, 1 (2017), 1-22.
7. Cuomo S., Michele P. D., Piccialli F., Galletti A., and Jung J. E. 2017. IoT-based collaborative reputation system for associating visitors and artworks in a cultural scenario. *Expert Systems with Applications*, 79 (2017), 101-11.
8. Marshall M. T., Dulake N., Ciolfi L., Duranti D., Kockelkorn H., and Petrelli D. 2016. Using Tangible Smart Replicas as Controls for an Interactive Museum Exhibition. In *Proc. of International Conference on Tangible, Embedded, and Embodied Interaction (TEI '16)*. ACM, New York, NY, USA, 159-67.
9. Colace F., Santo M. D., Greco L., Lemma S., Lombardi M., Moscato V., and Picariello A. 2014. A Context-Aware Framework for Cultural Heritage Applications. In *Proc. of International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, 469-76.
10. Manches A. 2011. Digital manipulatives: tools to transform early learning experiences. *Internation Journal of Technology Enhanced Learning (IJTEL)*, 3, 6 (2011), 608-26.
11. Petrelli D. and Lechner M. 2014. The meSch project – Material EncounterS with digital Cultural Heritage: Reusing existing digital resources in the creation of novel forms of visitor's experiences. In *Proc. of International Committee for Documentation of ICOM (CIDOC '14)*.
12. Yannier N., Hudson S. E., Wiese E. S., and Koedinger K. R. 2016. Adding Physical Objects to an Interactive Game Improves Learning and Enjoyment: Evidence from EarthShake. *ACM Transaction on Computer-Human Interaction (TOCHI)*, 23, 4 (2016), 32 pages.
13. Zancanaro M., Not E., Petrelli D., Marshall M., van Dijk T., Risseeuw M., van Dijk D., Venturini A., Cavada D., and Kubitza T. 2015. Recipes for tangible and embodied visit experiences. In *Proc. of Museums and the Web conference (MW '15)*.
14. Ardito C., Buono P., Desolda G., and Matera M. 2017. From smart objects to smart experiences: An end-user development approach. *International Journal of Human-Computer Studies*, 114 (2017), 51-68.
15. Ardito C., Desolda G., Lanzilottia R., Malizia A., and Matera M. 2019. Analysing Trade-offs in Frameworks for the Design of Smart Environments. *Behaviour & Information Technology* (2019).
16. Mighali V., Fiore G. D., Patrono L., Mainetti L., Alletto S., Serra G., and Cucchiara R. 2015. Innovative IoT-aware Services for a Smart Museum. In *Proc. of International Conference on World Wide Web (WWW '15)*. ACM, New York, NY, USA, 547-50.
17. Piccialli F. and Chianese A. 2017. The Internet of Things Supporting Context-Aware Computing: A Cultural Heritage Case Study. *Mobile Networks and Applications*, 22, 2 (April 01 2017), 332-43.
18. Fröhlich P., Baldauf M., Meneweger T., Erickson I., Tscheligi M., Gable T., Ruyter B. d., and Paternò F. 2019. Everyday Automation Experience: Non-Expert Users Encountering Ubiquitous Automated Systems. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems - Extended Abstracts (CHI '19)*. ACM, New York, NY, USA, 1-8.
19. Coronado M. and Iglesias C. A. 2016. Task Automation Services: Automation for the Masses. *IEEE Internet Computing*, 20, 1 (2016), 52-8.
20. IFTTT. https://ifttt.com/. Last access: June 10, 2019
21. elastic.io. http://www.elastic.io/. Last access: March 25, 2016
22. Zapier. https://zapier.com/. Last access: May 9, 2018
23. itDuzzit. http://cloud.itduzzit.com/. Last access: Sept 10, 2017
24. WigWag Smart Home. http://www.wigwag.com/. Last access: Sept 10, 2017
25. Node-RED. http://nodered.org/. Last access: Sept 10, 2019
26. Microsoft Flow. https://flow.microsoft.com/. Last access: June 28, 2019
27. Crosser. https://crosser.io/platform/crosser-node/. Last access: June 28, 2019
28. Markopoulos P., Nichols J., Paternò F., and Pipek V. 2017. Editorial: End-User Development for the Internet of Things. *ACM Transactions on Computer-Human Interaction*, 24, 2 (2017), 1-3.
29. Ghiani G., Manca M., Paternò F., and Santoro C. 2017. Personalization of Context-Dependent Applications Through Trigger-Action Rules. *ACM Transaction on Computer-Human Interaction*, 24, 2 (2017), 33 pages.
30. Desolda G., Ardito C., and Matera M. 2017. Empowering end users to customize their smart environments: model, composition paradigms and domain-specific tools. *ACM Transactions on Computer-Human Interaction*, 24, 2 (2017), 52 pages.
31. Brich J., Walch M., Rietzler M., Weber M., and Schaub F. 2017. Exploring End User Programming Needs in Home Automation. *ACM Transaction on Computer-Human Interaction*, 24, 2 (2017), 1-35.
32. Corno F., Russis L. D., and Roffarello A. M. 2017. A Semantic Web Approach to Simplifying Trigger-Action Programming in the IoT. *Computer*, 50, 11 (2017), 18-24.
33. Tutenel T., Bidarra R., Smelik R. M., and Kraker K. J. D. 2008. The role of semantics in games and simulations. *Computer Entertainment*, 6, 4 (2008), 1-35.
34. Corno F., De Russis L., and Roffarello A. M. 2017. A High-Level Approach Towards End User Development in the IoT. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems - Extended Abstracts (CHI '17)*. ACM, New York, NY, USA, 1546-52.
35. Risseeuw M., Cavada D., Not E., Zancanaro M., Marshall M., Petrelli D., and Kubitza T. 2016. An authoring environment for smart objects in museums: the meSch approach. In *Proc. of Workshop on Smart Ecosystems cReation by Visual dEsign (SERVE '16)*, CEUR-WS, 25-30.
36. Balducci F., Impedovo D., and Pirlo G. 2018. Detection and Validation of Tow-Away Road Sign Licenses through Deep Learning Methods. *Sensors (Basel, Switzerland)*, 18, 12 (2018), 4147.
37. Lecun Y., Bottou L., Bengio Y., and Haffner P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 11 (1998), 2278-324.
38. Krizhevsky A., Sutskever I., and Hinton G. E. 2012. ImageNet classification with deep convolutional neural networks. In *Proc. of*

International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12). Curran Associates Inc., Lake Tahoe, Nevada, 1097-105.
39. Zeiler M. D. and Fergus R. 2014. Visualizing and Understanding Convolutional Networks. In *Proc. of European Conference on Computer Vision (ECCV '14)*. Springer International Publishing, 818-33.
40. Szegedy C., Wei L., Yangqing J., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., and Rabinovich A. 2015. Going deeper with convolutions. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, 1-9.
41. He K., Zhang X., Ren S., and Sun J. 2016. Deep Residual Learning for Image Recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, 770-8.
42. Pascanu R., Mikolov T., and Bengio Y. 2013. On the difficulty of training recurrent neural networks. In *Proc. of International Conference on International Conference on Machine Learning - Volume 28 (ICML'13)*. JMLR.org, Atlanta, GA, USA, III-1310-III-8.
43. Hochreiter S. and Schmidhuber J. 1997. Long short-term memory. *Neural computation*, 9, 8 (1997), 1735-80.
44. Xingjian S., Chen Z., Wang H., Yeung D.-Y., Wong W.-K., and Woo W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proc. of Advances in Neural Information Processing Systems (NIPS '15)*, 802-10.
45. Cao Y., Niu X., and Dou Y. 2016. Region-based convolutional neural networks for object detection in very high resolution remote sensing images. In *Proc. of International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD '16)*, 548-54.
46. Ren S., He K., Girshick R., and Sun J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39, 6 (2017), 1137-49.
47. Ardito C., Costabile M. F., De Angeli A., and Lanzilotti R. 2012. Enriching exploration of archaeological parks with mobile technology. *ACM Transaction on Computer-Human Interaction (TOCHI)*, 19, 4 (2012), 31 pages.
48. Oviatt S. 2013. *The design of future educational interfaces*. Routledge.
49. Anderson E. F., McLoughlin L., Liarokapis F., Peters C., Petridis P., and de Freitas S. 2010. Developing serious games for cultural heritage: a state-of-the-art review. *Virtual Reality*, 14, 4 (December 01 2010), 255-75.
50. Mortara M., Catalano C. E., Bellotti F., Fiucci G., Houry-Panchetti M., and Petridis P. 2014. Learning cultural heritage by serious games. *Journal of Cultural Heritage*, 15, 3 (2014/05/01/ 2014), 318-25.
51. Ishii H. and Ullmer B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. ACM, New York, NY, USA, 234-41.
52. Kim M. J. and Maher M. L. 2008. The impact of tangible user interfaces on spatial cognition during collaborative design. *Design Studies*, 29, 3 (2008), 222-53.
53. Doering T., Beckhaus S., and Schmidt A. 2009. Towards a sensible integration of paper-based tangible user interfaces into creative work processes. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems - Extended Abstracts (CHI '09)*. ACM, New York, NY, USA, 4627-32.
54. Lin T., Goyal P., Girshick R., He K., and Dollar P. 2018. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2018), 2980-8.
55. Lin T.-Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., and Zitnick C. L. 2014. Microsoft COCO: Common Objects in Context. In *Proc. of European Conference on Computer Vision (ECCV '14)*. Springer International Publishing, 740-55.
56. Cohen G., Afshar S., Tapson J., and van Schaik A. 2017. EMNIST: Extending MNIST to handwritten letters. In *Proc. of International Joint Conference on Neural Networks (IJCNN '17)*. IEEE, 2921-6.
57. Shi B., Bai X., and Yao C. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 11 (2016), 2298-304.
58. Tian Z., Huang W., He T., He P., and Qiao Y. 2016. Detecting text in natural image with connectionist text proposal network. In *Proc. of European Conference on Computer Vision (ECCV '16)*. Springer, 56-72.
59. Shi B., Wang X., Lyu P., Yao C., and Bai X. 2016. Robust scene text recognition with automatic rectification. In *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, 4168-76.
60. Baek J., Kim G., Lee J., Park S., Han D., Yun S., Oh S. J., and Lee H. 2019. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proc. of International Conference on Computer Vision (ICCV '16)*, to appear.
61. Jaderberg M., Simonyan K., Vedaldi A., and Zisserman A. 2016. Reading Text in the Wild with Convolutional Neural Networks. *International Journal of Computer Vision (IJCV)*, 116, 1 (2016), 1-20.
62. Puigcerver J. 2017. Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition? In *Proc. of International Conference on Document Analysis and Recognition (ICDAR '17)*, 67-72.
63. Ubul K., Tursun G., Aysa A., Impedovo D., Pirlo G., and Yibulayin T. 2017. Script Identification of Multi-Script Documents: A Survey. *IEEE Access*, 5 (2017), 6546-59.
64. Colace F., De Santo M., Greco L., and Napoletano P. 2015. Improving relevance feedback-based query expansion by the use of a weighted word pairs approach. *Journal of the Association for Information Science and Technology (JASIST)*, 66, 11 (2015), 2223-34.