Deep Artwork Detection and Retrieval for Automatic Context Aware Audio Guides

LORENZO SEIDENARI, University of Florence - MICC
CLAUDIO BAECCHI, University of Florence - MICC
TIBERIO URICCHIO, University of Florence - MICC
ANDREA FERRACANI, University of Florence - MICC
MARCO BERTINI, University of Florence - MICC
ALBERTO DEL BIMBO, University of Florence - MICC

In this paper we address the problem of creating a smart audio guide that adapts to the actions and interests of museum visitors. As an autonomous agent, our guide perceives the context and is able to interact with users in an appropriate fashion. To do so, it understands what the visitor is looking at, if the visitor is moving inside the museum hall or if he is talking with a friend. The guide performs automatic recognition of artworks, and it provides configurable interface features to improve the user experience and the fruition of multimedia materials through semi-automatic interaction.

Our smart audio guide is backed by a computer vision system capable to work in real-time on a mobile device, coupled with audio and motion sensors. We propose the use of a compact Convolutional Neural Network (CNN) that performs object classification and localization. Using the same CNN features computed for these tasks, we perform also robust artwork recognition. To improve the recognition accuracy we perform additional video processing using shape based filtering, artwork tracking and temporal filtering. The system has been deployed on a NVIDIA Jetson TK1 and a NVIDIA Shield Tablet K1, and tested in a real world environment (Bargello Museum of Florence).

CCS Concepts: •**Information systems** → **Mobile information processing systems; Multimedia databases; Retrieval on mobile devices; Image search;** •**Human-centered computing** → *Interactive systems and tools;*

Additional Key Words and Phrases: Deep Learning, Computer Vision, Object Detection, Image Retrieval, Mobile Computing, Cultural Heritage, Audio Guide

ACM Reference Format:
Lorenzo Seidenari, Claudio Baecchi, Tiberio Uricchio, Marco Bertini, Alberto Del Bimbo *ACM Trans. Multimedia Comput. Commun. Appl.* VOL., NUM., Article ART. (March 2017), 21 pages.
DOI: 0000001.0000001

1. INTRODUCTION

Digital and mobile technologies are becoming a key factor to enhance visitors' experiences during a museum visit, e.g. creating interactive and personalized visits. Personalization is viewed as a factor in enabling museums to change from "talking to the visitor" to "talking with the visitors", turning a monologue to a dialogue. This applies especially to audio guides since, similarly to a real museum guide, they must adapt their content to the needs and interests of the visitors [Bowen and Filippini-Fantoni 2004]. Whether personalization addresses on-line exhibitions [Bowen and Filippini-Fantoni 2004], on-site display of artworks [Karaman et al. 2016], or both on-line and

This work is partially supported by the "Social Museum and Smart Tourism" project (CTN01_00034_231545).
Author's addresses: Università degli Studi di Firenze - MICC, Viale Morgagni 65, Firenze, Italy.

DOI: 0000001.0000001

on-site [Wang et al. 2009], there is a need to obtain information about the behavior of the visitor, e.g. what he is looking at, for how long, and what other events happen during the visit. In this paper we address the problem of creating a smart audio guide that adapts to the actions and interests of the visitor of a museum, understanding both the context of the visit and what the visitor is looking at.

The goal of this work is to implement a real-time computer vision system that can run on wearable devices to perform object classification and artwork recognition, to improve the experience of a museum visit through the automatic detection of the behavior of users. Object classification, sensors and voice activity detection help to understand the context of the visit, e.g. differentiating when a visitor is talking with people or his sight is occluded by other visitors, e.g. understanding if he has friends that accompany him during the visit to the museum, or he is just wandering through the museum, or if he is looking at an exhibit that interests him.[1] Artwork recognition allows to provide multimedia insights of the observed item automatically or to create a user profile based on what artworks a user is looking at and for how long.

2. RELATED WORK

Personalized museum experience.

The personalization of a museum visit may address the on-line experience in a virtual museum, the on-site experience in the museum itself, or both cases.

In [Bowen and Filippini-Fantoni 2004] web personalization in museums is motivated by the advantages that it provides in improving usability of museum web sites and the facilitation of the learning process implied in a visit. Personalization is considered a new communication strategy that improves relationships between visitors and the institution. In [Wang et al. 2009] it is presented the Cultural Heritage Information Personalization (CHIP) system, that bridges on-line and on-site tour guides creating a personalized visit tour through a web site and then downloading the guide on a mobile device with RFID sensors that track the visitor in the museum. Tour information and the rating of artworks, if provided by users, are then sent back to the web site to update the user profile. Interactive digital guides have been used in [Zancanaro et al. 2007] and [Kuflik et al. 2012] to analyze and predict the behavioral patterns of museum visitors, according to four main patterns that were initially identified through ethnographic observations by [Eliseo and Martine 1991]. The works show that the four patterns can be identified using features such as average time spent on each artwork, percentage of observed artworks, etc. In [Keil et al. 2013] augmented reality (AR) on a mobile device is coupled with a personalized interactive storytelling experience, e.g. adapting the guide based on the age of the visitor, providing a gamified experience to children. In [Karaman et al. 2016] a non-intrusive computer-vision system has been presented, based on person re-identification of museum visitors observed through surveillance cameras. The system identifies the artworks that are observed by museum visitors and measures how much time is spent looking at each artwork, to create a personalized user profile. At the end of the tour the user profile is used to create a personalized exploration of multimedia content on an interactive table, providing more information on the items that most attracted the visitor, and suggesting additional visits and tours.

Object detection and recognition

After the breakthrough of convolutional neural networks in image classification brought by Krizhevsky *et al.* [Krizhevsky et al. 2012], several works have used similar

[1] https://vimeo.com/187957085

or derived strategies to solve other image and video related tasks [Erhan et al. 2014; Girshick 2015; Girshick et al. 2014; Ren et al. 2015]. A simple, yet dramatically effective strategy pioneered by Girshick *et al.* is to extract CNN features from regions of an image. Further improvements in localization and accuracy are obtained using a bounding box regressor and fine-tuning the CNN features on the detection task. The task of computing a full forward pass for every sub-window is extremely time expensive even for moderately shallow networks. More up-to-date works [Girshick 2015; Ren et al. 2015] avoid this burden by computing a single full resolution convolution on the whole frame and then performing classification and bounding box regression over a region of interest computed over the last convolutional layer. Fast R-CNN avoided the computation of multiple full forward passes, nonetheless it required expensive resources to compute object proposals, often generated with low-level features such as edges [Zitnick and Dollár 2014]. Ren *et al.* [Ren et al. 2015] removed this further computation bottleneck by learning a lightweight object proposal sharing the same features of the network used for object detection.

A more recent class of approaches tries to generate a set of class-labeled bounding boxes with a single pass of a convolutional network [Redmon et al. 2016; Liu et al. 2016]. Redmon *et al.* argue that You should Only Look Once (YOLO) at frames, using an architecture inspired by Inception [Szegedy et al. 2015] focused on reducing the network size and the computation. The main idea is to produce, as an output, a tensor of size $N \times N \times |C| \times 5$, representing the coordinates and probabilities, for each of the C categories, for N^2 evaluated locations. Liu *et al.* proposed an approach named Single-Shot Detection (SSD), which is very similar to YOLO, but differs in the fact that it removes all fully connected layers allowing to predict bounding box using small convolutional filters on the last convolutional activation map. One advantage of SSD is that it allows to evaluate more windows, at multiple scales, by computing convolutions on previous output layers.

Content-based retrieval for Cultural Heritage

Over the years several methods and applications of content-based image retrieval (CBIR) techniques have been applied to the domain of cultural heritage.

A comparison of different techniques, based on engineered and learned features, for image classification and retrieval in cultural heritage archives has been presented in [Picard et al. 2015]. The authors of this work highlight two issues when applying current state-of-the-art CBIR techniques in the cultural heritage domain: *i)* often there is need to account for both micro properties, such as brush strokes, and macro properties, such as scene layout, in the design of similarity metrics; *ii)* datasets are, paradoxically, relatively small, with few images for each item, thus hindering methods that require large scale training datasets. A model to support recognition of complex 3D monuments such as statues was proposed in [Del Bimbo et al. 2009]; in the proposed approach salient SIFT points are selected using a measure of mutual information to reject points that are part of background. A method for painting classification, in terms of artist and style, has been proposed in [Anwer et al. 2016]. Paintings are represented using the concatenation of two Fisher Vectors that represent the whole image and salient parts of the image. In [Liu et al. 2015] a late fusion of global and local CNN features is used to classify images taken during cultural events.

Object recognition on mobile devices

The availability of multi-core CPUs and GPUs on mobile devices has recently allowed to implement multimedia and computer vision methods on smartphones, with particular attention to convolutional neural networks.

In [Yanai et al. 2016] an analysis of the best CNN architectures for mobile devices has been performed, evaluating the impact of using NEON SIMD instructions available on ARM CPUs and BLAS routines. The authors propose to use a Network-In-Network (NIN) architecture, where neuron weights are compressed with Product Quantization, to reduce the memory occupation of the CNN network. This solution has been employed to implement a mobile system for food recognition, presented in [Tanno et al. 2016]. The problem of food recognition using mobile devices has been addressed also in [Meyers et al. 2015], where different CNNs are used to segment food, estimate the 3D volume and classify food, so to provide an estimation of the calories; however only the CNN for food classification has been ported to a mobile device. Speed improvement and memory requirements reduction of CNN execution, for mobile devices, has been obtained in [Wu et al. 2016] through weights quantization of fully connected and convolutional layers, and applying an error correction technique to minimize the estimation error of each layer. In [Huynh et al. 2016] a framework to execute deep learning algorithms on mobile devices has been presented. The framework uses OpenCL to exploit the GPUs. The framework addresses the problem of thread divergence in GPUs through data padding. In [Latifi Oskouei et al. 2016] has been presented a framework for GPU-accelerated CNNs on Android devices, that uses SIMD instruction on mobile GPUs, parallelizing some types of layers on GPUs and others, that are less computationally intensive, on CPUs. The framework has been released as open source.

Voice Activity Detection

Voice activity detection (VAD) is the process of detecting when humans are speaking in a given audio stream. It is essential to improve further processing like automatic speech recognition or saving bandwidth in audio coding or conference systems.

The first VAD system was first investigated in the fifties to be used on TASI systems [Bullington and Fraser 1959]. Early approaches to this problem were based on heuristics and simple energy modeling, by thresholding or observing zero-crossing rate rules [Woo et al. 2000]. These methods work well in settings where no background noise is present. More recent methods address this limitation by employing autoregressive models and line spectral frequencies [Mousazadeh and Cohen 2011] to observe signal statistics in current frame and compare it with the estimated noise statistics with some decision rules. However, most of these conventional algorithms assume that noise statistics are stationary over long periods of time, more than those of speech. Given the extreme diversity and rapid changes of noise in different environments, they can't detect occasional presence of speech. The most recent class of approaches for VAD are that of data-driven methods, that avoid to make assumption over the noise distribution. They usually use a classifier trained to predict speech vs non-speech given some acoustic features [Elizalde and Friedland 2013; Misra 2012]. Anyway, their performance degrades when the background noise resembles that of speech. The state-of-the-art methods exploit long-span context features learned through the use of recurrent neural networks [Eyben et al. 2013; Drugman et al. 2016; Vesperini et al. 2016] to adapt the classification on the basis of the previous frames.

The method presented in this paper addresses the problem of creating a personalized on-site museum experience using a non-intrusive computer vision algorithm that can be executed on board of an audio guide. Unlike works such as [Latifi Oskouei et al. 2016] and [Huynh et al. 2016], no special framework has been used, and the problem of computational costs has been addressed using: *i)* a CUDA implementation of a CNN running on NVIDIA portable GPUs, and *ii)* designing the algorithm to exploit the same features used for object detection, classification and retrieval. The problem of the scarcity of training data, highlighted in [Picard et al. 2015], has been solved applying

fine tuning to a pre-trained CNN. Moreover, we exploit on-board sensors and recent recurrent neural networks for voice detection to further understand the context of the wearer, like its movements and its interactions with other people.

The remainder of the paper is organized as follows:in Sect. 3 we describe the overall system architecture and its sub-systems; in Sect. 4 we describe our efficient method for detecting objects and how we obtain a reliable artwork identification using tracking and retrieval. Sect. 5 outlines the context modeling module based on voice and sensor input processing. The full system, comprising also the Android App is described in Sect. 6. Finally in Sect. 7 and Sect. 8 we present quantitative results on our system together with an user experience evaluation, then drawing conclusions in Sect. 9.

3. THE SYSTEM

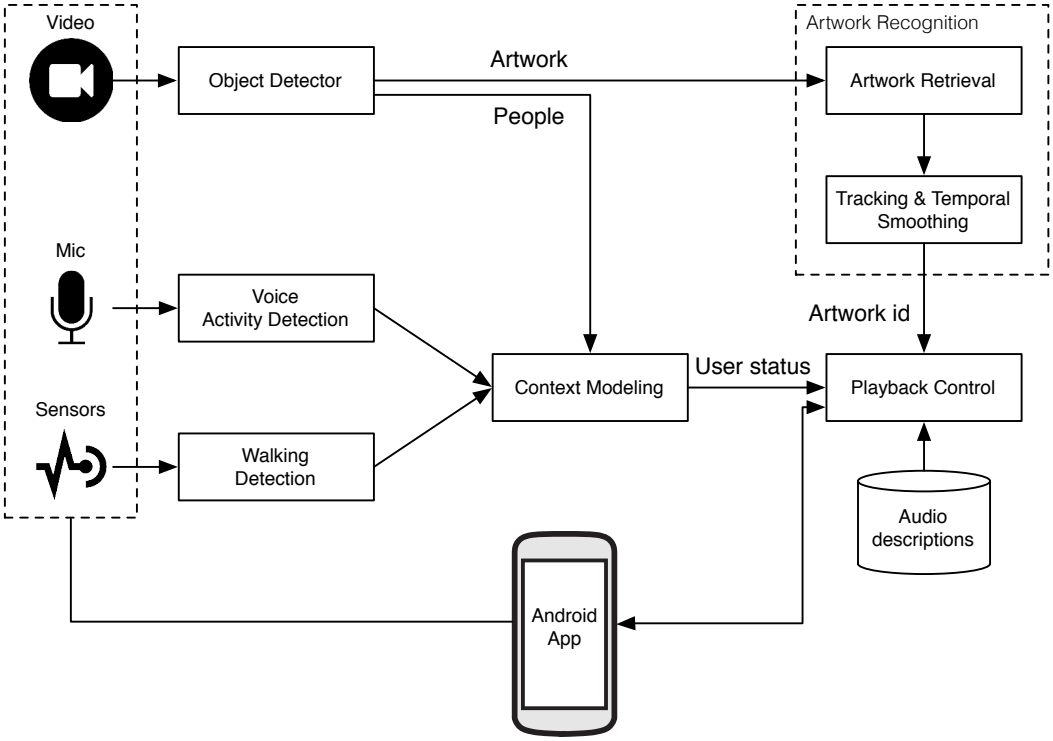

Fig. 1. The overall system architecture.

The system we propose comprises several components that work together to enable a smart experience. Fig. 1 shows an architectural diagram illustrating the main sub-modules of the system. From a higher level view of our system, two main sub-systems are identified, one responsible to recognize artworks, (providing *Artwork id*) and one to model the *User status*. They generate input signals for the *Playback Control* module which is responsible to play descriptions at appropriate time.

Our system senses the environment through three main channels: a camera, a microphone and movement sensors. The three sources are accessed through an Android App which is also responsible as a front-end of the whole system.

The camera is used to understand what the user is looking at. A computer vision system is responsible to detect objects (*Object Detector*) and recognize what artwork the user is looking at (*Artwork Recognition*). Two sub-modules are highlighted in the recognition step: the first retrieves the most similar artwork from a database of known artworks and the second performs tracking to smooth out wrong predictions.

The *Context Modeling* module receives three behavioral signals: *People Detections*, *Voice Activity Detection*, and *Walking Detection*. These signals concur in generating a *User Status* signal. The microphone is used as a source for *Voice Activity Detection*, and movement sensors are necessary for *Walking Detection*.

4. EFFICIENT OBJECT DETECTION AND RECOGNITION

The smart audio guide we developed is based on an efficient computer vision pipeline that simultaneously performs artwork localization and recognition. The guide requires two main computer vision tasks to be solved: *i)* detection of relevant object categories: e.g. persons and artworks; and *ii)* for every detected artwork, reliable recognition of the specific artwork framed. Moreover, since we are dealing with a sequence of frames, in order to improve artwork recognition we take advantage from temporal coherence to make the output more stable.

Our system is based on YOLO [Redmon et al. 2016], that is demonstrated to obtain accurate results even for moderate size networks. The main advantage of YOLO can be read in its acronym, i.e. it requires to look at the image only once. The process to generate scored boxes for each category of interest can be summarized as in the following. The whole image is split in 7×7 blocks. For each of the 49 regions a tensor of $5 \times 2 \times |C|$ is output. This tensor encodes two box predictions for each of the $|C|$ classes. Boxes are represented as a tuple $\langle x, y, h, w, s \rangle$. Non maximal suppression can be used to avoid multiple prediction for the same object. The confidence accounts for the accuracy of the bounding box and the probability of that class being present inside the given.

Differently from SSD [Liu et al. 2015], which is based on VGG-16, our YOLO-based classifier uses a much smaller network that allows the classifier to adhere with the memory requirements of an embedded system like the NVIDIA Tegra TK1 SoC. The architecture is derived from *Tiny Net*, a small CNN pre-trained on ImageNet, which allows the application to run at 10 FPS and fitting on the memory of a Shield Tablet.

The system network was fine-tuned to recognize artworks and people using our dataset. Recognizing people is relevant for two reasons: first we can exploit the presence of people in the field of view to create a better understanding of context, see Sect. 5; secondly, without learning a person model, it is hard to avoid false positives on people, since artwork training data contains statues, which may picture human figures. Learning jointly a person and an artwork model, the network features can be trained to discriminate between this two classes.

4.1. Artwork recognition

The rich features computed by the convolutional layers are exploited and re-used to compute an object descriptor for artwork recognition.

To ensure ease of deployment and update of the system, we base our artwork recognition system on a simple nearest neighbor step. We need to fulfill two important requirements: first our feature should be lightweight, i.e. low dimensional, in order to be stored on the device and reduce the computation time for feature comparison; second we must compute a discriminative representation for a region of the frame that may differ in size and aspect ratio.

To obtain a low dimensional fixed size descriptor of a region, we apply a global max-pooling over convolutional feature activation maps, as shown in Fig. 2. To increase

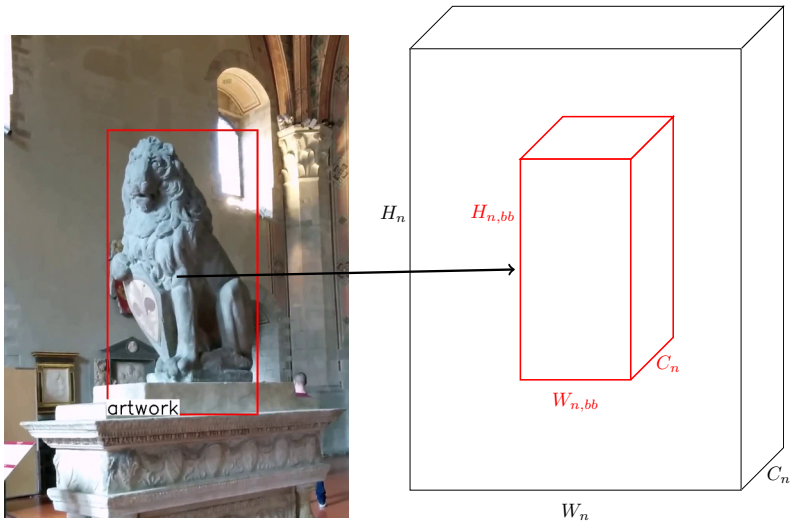

Fig. 2. Feature extraction procedure for an artwork detection on a single convolutional feature map.

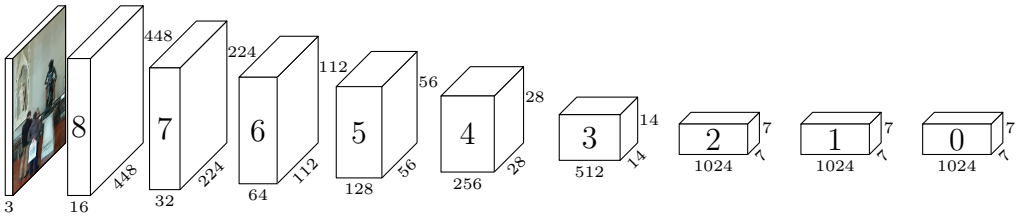

Fig. 3. Our network architecture, with tensor size and layer numbering.

the discriminative power, we concatenate such descriptor computed on two different feature maps. The region is remapped from the frame to the convolutional activation map with a simple similarity transformation.

Considering the activation map of the n^{th} convolutional layer, we have a tensor dimension of $W_n \times H_n \times C_n$. After the reprojection of the bounding box onto the feature map we end up with a smaller tensor with a size of $W_{n,bb} \times H_{n,bb} \times C_n$; $W_{n,bb}$ and $H_{n,bb}$ depend both on the network layer and the bounding box geometry, while C_n depends solely on the network layer and represents its number of channels. The max-pooling operation of the C_n channels over the $W_{n,bb} \times H_{n,bb}$ values generate a feature vector that is independent from the dimension of the bounding box.

Considering the architecture in Fig. 3 one could wonder which features are best to recognize the specific framed artwork, since leftmost layers have higher resolution and mostly represent the low-level structure of the image, while rightmost ones, are low resolution but encode higher level information, closer to the image semantics.

After an experimental evaluation, which is detailed in Sect.7, we selected, as combination, the features from layers 3 and 4, yielding a feature of size 768. The final bounding box descriptor is obtained by concatenation of the two max-pooled regions values and is L^2-normalized.

Considering a pre-acquired dataset of artwork patches $p_i \in \mathcal{D}$ and their artwork labels y, for each detected artwork d we predict a specific artwork label $y_{\hat{p}}$ finding the

nearest neighbor patch

$$\hat{p} = \arg\max_i \langle p_i, d \rangle \tag{1}$$

The recognition system observes each frame independently and predicts artwork labels according to Eq. 1, this approach, in case of motion blur or quick lighting changes may produce incorrect recognition results. In the following we detail how we exploit temporal coherence to produce a more stable recognition output.

4.2. Artwork Tracking and Temporal Smoothing

High recognition accuracy is a requirement for the audio-guide, since mistaking an artwork for another may result in a bad user experience, e.g. this would result, at the interface level, in the audio guide presenting an artwork different from the one that is actually observed.

This is an extremely critical aspect and must be addressed, in order to improve the stability of the recognition system. We devise three strategies, based on the user location with respect to the artwork of interest and the continuous tracking of object bounding boxes.

To reduce the error rate our idea is to avoid performing artwork recognition on objects that may be too far from the user. Farther objects are unlikely to be of interest for the user, moreover the feature computed on a smaller bounding box has little discriminative power and likely leads to erroneous recognition.

Computing the actual metric distance from an artwork requires to perform real-time camera tracking and scene mapping. We believe that this accurate information is not required for our task and therefore we rely on a simple heuristic, comparing the areas of an artwork detection and the whole frame as in the following:

$$\frac{w_{bb} h_{bb}}{WH} > T \tag{2}$$

where WH is the frame area and w_{bb} and h_{bb} are bounding box width and height respectively, and T is a threshold (Fig. 4) empirically fixed. We name this strategy *Distance*. In our experiments we obtained the best results for $T = 0.1$, that as can be seen in Sec. 7.4, allows to reduce false recognitions by 50% w.r.t. not using the heuristic, at the cost of introducing a small number of missed recognitions.

Considering that there is continuity when the user walks around in the area, an artwork recognized frequently across a very short amount of time is probably the most correct. To exploit this, we continuously predict artwork labels as described in Sec. 4.1, but we consider a prediction only after it persists for M frames. We name this strategy *Consistency*. We implement it by tracking all artwork detection boxes with a greedy data association tracking-by-detection algorithm, requiring an IoU of consecutive bounding boxes of 50%. An example of this tracking is shown in Fig. 5.

With the same principle, it is unlikely that the user moves quickly from an artwork to another in just few frames. So, after the system recognizes an artwork, it continuously output its label proportionally to the elapsed time since the recognition. We call this strategy *Persistence*. We increment a counter p every time the recognition label for a box is unchanged, keeping track of the most frequent label $\overline{y}$. Every time a label $y*$ is different from $\overline{y}$ we decrement p. We predict the artwork identity as $y*$ only if $p > P > M$. This technique greatly reduces the number of false recognitions. In our experiments best results were obtained for $M = 15$ and $P = 20$.

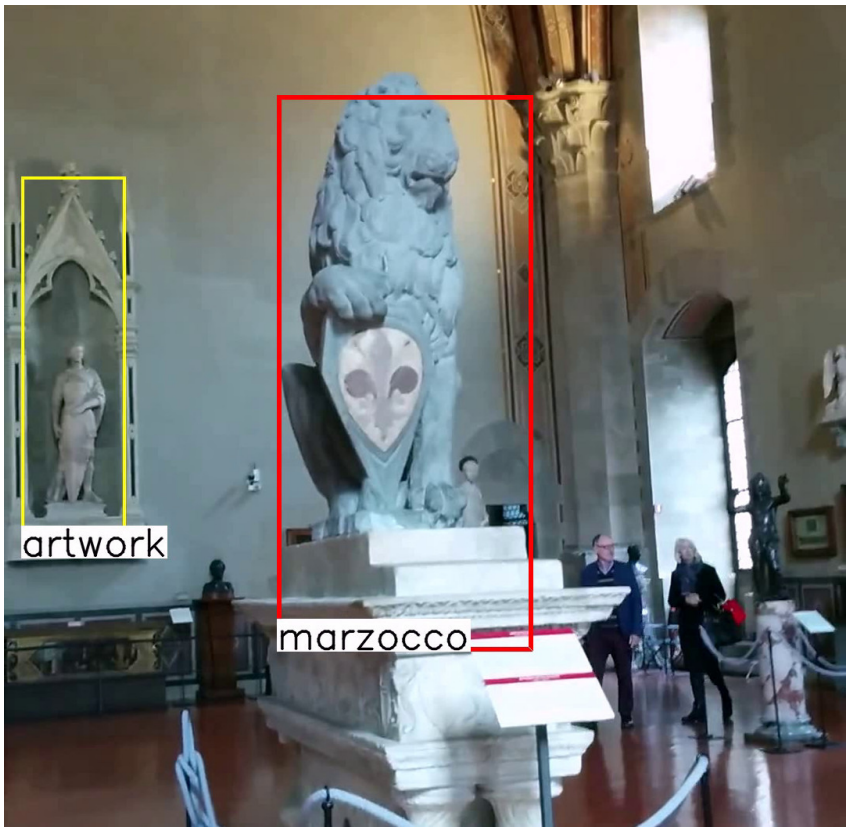

Fig. 4. Shape based filtering: artwork in yellow (left) is not considered for recognition, not satisfying Eq. 2, while the other is recognized as “marzocco” (the heraldic lion symbol of Florence).

5. CONTEXT MODELING

To pursue the idea of an autonomous agent that is able to understand when it is the time to engage the user and when it should be inactive, it is essential to understand the context and the status of the wearer. In addition to the observation of the same scene the user is viewing through the wearable camera, we also try to understand if the user is busy following or participating in a conversation and if he is moving around the room, both independently from the visual data.

5.1. Detecting conversations

Our audio-guide should be able to understand when the user is engaging in a conversation, if his field of view is occluded by other visitors or he is paying attention to another person or an human guide. In that event, it is reasonable to stop the audio-guide or temporarily interrupt the reproduction of any content, in order to let the user carry out his conversation. This should be of high priority and it should be performed even if the user is standing in front of an artwork. To identify this scenario, we use the device microphone to detect the presence of a nearby voice. We chose to employ a Voice Activity Detection (VAD) system for this task.

Typically, museums are mostly quiet environments where people tend to remain silent, to appreciate the artworks, and briefly talk between each other. Nonetheless, in some cases the environment can be noisy with the presence of music in background or

Fig. 5. Example of artwork tracking, with $M = 15$. Only after a stable recognition over M frames the system labels the artwork.

some environmental noise. This requires the adoption of a VAD with automatic noise adaptation. The system will listen continuously to the environment, adapt to the local environment noise and detect when voice is present. Therefore, in order to run in real-time together with the computer vision module, it is essential to provide a lightweight system with low computational complexity. We adopted the system from [Eyben et al. 2013], that is a state-of-the-art method based on a Long Short Term Memory recurrent neural network. This approach is able to model long range dependencies between the inputs (and thus accurately model environmental noise) and is highly scalable. The computational complexity for evaluating the networks is linear with respect to the number of input frames. Only a constant number of operations needs to be performed for every audio frame. We use the open source implementation and model available in the OpenSMILE framework [2].

Considering that a positive voice identification stops the playing of any description, it is imperative that the classifier has a low false positive rate. Unexpectedly stopping the reproduction due to a classifier error may result in a poor user experience. To this end, we evaluate an entire second of audio before emitting the prediction. We choose to use a classifier with a granularity of 0.01, so that, by exploiting all the predictions in this time frame, we can increase the stability of the prediction. The final prediction is the mean over the single classifications. We threshold this value according to the expected false positive rate, measured empirically on our dataset.

5.2. Sensors for walking detection

An important hint for understanding the context of the user is given by its movements. Standing still, walking or sitting can signal if the user is paying attention to some artwork or if he is uninterested in what he is looking at.

We make use of this information for mainly two purposes:

[2]http://audeering.com/research/opensmile/

— If the user is walking fast then he is not probably interested in the visible artworks. This means that even if the visual system detects and recognizes an artwork, the audio description should not be started.
— If the user is standing still in front of an artwork and he is listening to the audio description, this should not be stopped, even if the visual system stops recognizing the artwork. This can happen mostly because of occlusions due to other people walking or standing between the visitor and the artwork.

To perform walking detection we use accelerometer data. We estimate the mean and standard deviation of acceleration magnitude from the training set. We subtract the mean from the acceleration magnitude and then we filter out peaks below the standard deviation. We consider each peak as a step. We then take into account a sliding window of 1 second, and consider the subject walking if at least a step is detected in the given window.

To detect if a person changes the facing direction, we estimate the orientation variation using gyroscope data. We average the orientation vector over the same 1 second sliding window. The facing direction is considered changed if the current orientation vector differs from the average for at least 45°.

6. SYSTEM IMPLEMENTATION

The proposed system has been initially developed using a NVIDIA Jetson TK1 board, to test the performance of the vision system, introduced in Sec. 4, using a device designed for embedded systems. The board has a NVIDIA Kepler GPU with 192 CUDA cores, and an NVIDIA 4-plus-1 Quad-core ARM Cortex A15 CPU. Then the audio-guide application, named SeeForMe, has been deployed on an NVIDIA Shield Tablet K1 that has the same computational capabilities of the TK1 board, but it runs Android 5.0 instead of Linux, and it allows to develop a user friendly application that can support the visitor in his museum experience.

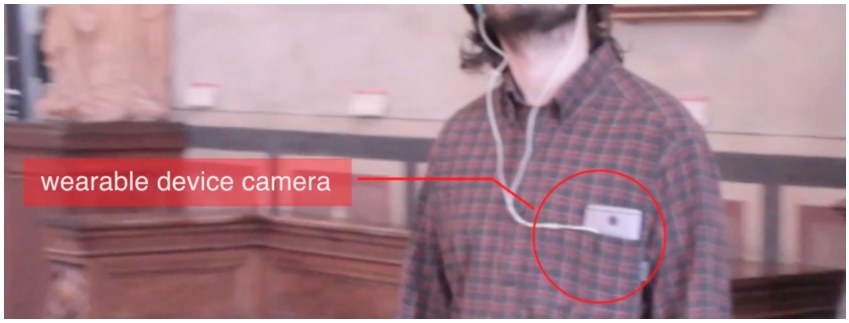

Fig. 6. A visitor with the device camera in the pocket

We designed the application to handle three different user scenarios: *(i)* the user makes use of the application in a fully-automated way (placing the device in a front pocket with the camera facing forward, or hanging it on the chest using a special support), as shown in Fig. 6. In this scenario the system does not need any interaction and continuously observes the surroundings using the camera, choosing when to start and stop the audio by analyzing the user's behavior; in this modality the user can still interact with the application by using voice commands that are elaborated by the operating system and translated in the form of actions such as start/stop the audio; *(ii)*

the user makes use of the application actively in a semi-automated way: after pointing the device towards the artworks the visitor is interested in, the system detects the artwork and provides the contextual audio guide, for which the audio can be started and stopped automatically or manually by the user; *(iii)* the user has completed the tour and wants to revisit his experience: to this end, the application provides a visual history of the tour represented as a carousel of artworks in temporal order. Through the carousel the user can select the artworks he visited, have multimedia insights and replay the audio guide.

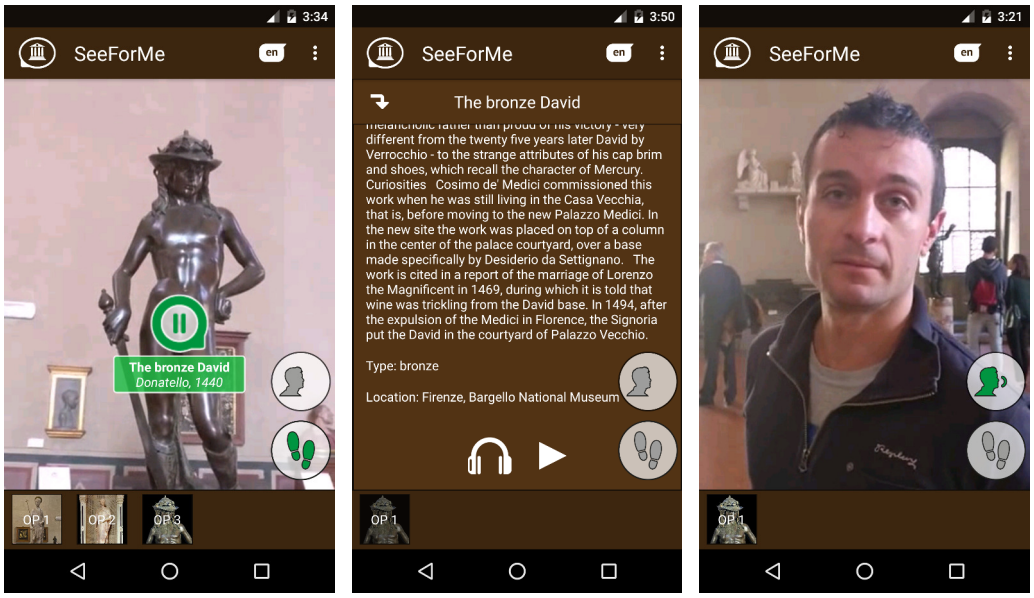

Fig. 7. *(left)* the user is listening to the description of the artwork, *(center)* the user is reviewing an item in the history, *(right)* the user is speaking with someone not focusing on any artwork,

In Fig. 7 we show two use cases of the application GUI. Fig. 7 *left)* the user is listening to the description of an artwork, we can see from the bottom right icon on the screen that the user is also walking, but keeping the focus on the artwork while he moves. For this reason the system continues to provide the audio description of the artwork. The GUI shows the recognized artworks by overlaying a small green indicator that provides both the audio guide status in the form of a play/pause symbol and information about the artwork. Among these the title, author and year of creation are presented to the user. Artwork detected but not still recognized are marked with a red icon stimulating the visitor to approach and to further frame it in order to have it recognized. If the visitor moves away from the artwork the audio played will fade, avoiding an abrupt interruption, so to improve user experience. In case the visitor goes back to the same artwork before listening to any other audio description, the description is resumed automatically from where it stopped.

In the bottom of the screen the application keeps track of the viewing history by showing the previously visited artworks in the form of small thumbnails. By touching one, the user can review the history of the artwork and listen again to the guide as shown in Fig. 7 *center)*. This part of the GUI proposes an image of the artwork and the full text of the audio guide. The user can choose to read the content without the audio or start the playback by pressing the play icon.

In Fig. 7 *right)* the user is speaking to someone close to him. The application recognizes the action and notifies the user by activating the speaking icon on the bottom right of the screen. While the user is interacting with someone else, and while the focus is not on an artwork, any currently active description stops playing leaving the user free to discuss. Once the conversation is finished and the focus is brought back on the artwork the system automatically resumes the artwork description.

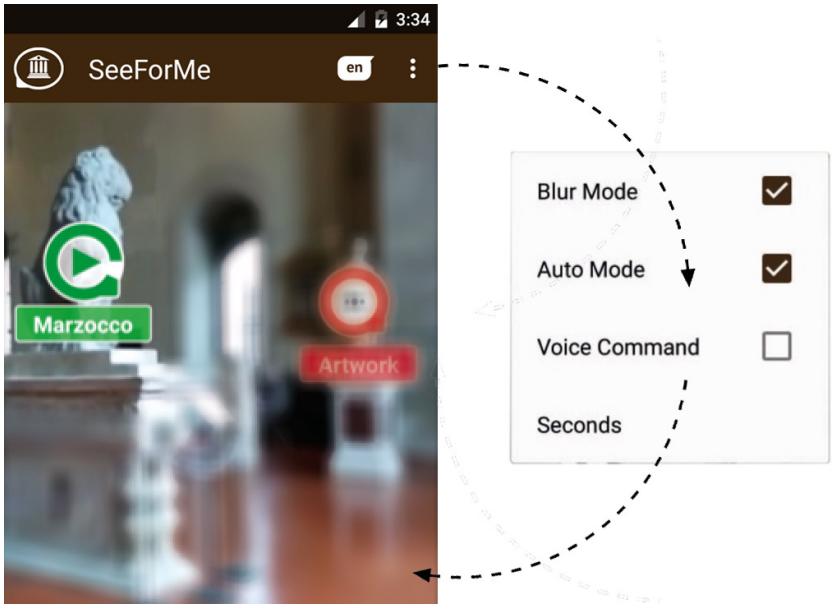

Fig. 8. The contextual menus to configure the app properties. Here it is shown the appearance of the interface in blur mode.

Several application properties and modalities can be configured in the mobile app guide through a contextual menus reachable from the top right corner of the navigation bar. In Fig. 8 it is shown the contextual menus where there are two main modes: *i)* Blur mode, *ii) Auto mode*. The first one enables an app feature which blur the background of the artwork being framed by the visitor in order to focus his attention on the target. The Audio mode instead activates the automatic mode for the control of the guide audio stream. Voice Commands and interaction can also be enabled and disabled. Finally, a range in seconds can be defined to set a custom temporal window between the instant that the system recognizes the artwork and the start of the audio guide reproduction (these delay is marked visually by the green line in the icon which animates until it closes the circle, as shown in Fig. 8).

The mobile app has been developed using the Android SDK. The interface follows the guidelines of material design proposed by Google [3]. SQLite is used to persist the information on the device local storage. Communication between the app and the YOLO module is carried out using Java Native Interface (JNI) which enables the Java code running in the Java Virtual Machine (JVM) to call and be called by native applications. Data-interchange is performed through JSON messages. In particular, the YOLO module communicates with the mobile app passing data related to the current frame of the

[3]https://material.google.com/

camera stream. This data comprises detected and recognized artworks and persons, with the coordinates of their bounding box, and booleans indicating if external speech and user movements have been detected.

7. EXPERIMENTAL RESULTS

7.1. Dataset

We collected a dataset from footage captured in the Bargello Museum in Florence. Bargello Museum hosts a variety of artworks, featuring a large hall (Donatello Hall) with several masterpieces from Donatello. We use this hall as our testing ground. The collected data serves two distinct purposes: train and evaluate the object detector described in Sect. 4, and evaluate the full artwork recognition system.

Artwork imagery has been collected in a diverse set of illumination and viewpoint conditions. In fact, the Donatello Hall is an extremely challenging environment featuring a high ceiling with large glass windows. Therefore depending on the time of the day and the weather condition, artwork appearance may change significantly, because of light diffusion and camera sensor saturation. We collect $1,237$ images from all the statues in the Donatello Hall, in different lighting and viewpoint conditions.

For the object detection task we extract a subset of annotated images, splitting the data in training and testing. Artworks appearing in the training set do not appear in the testing set to correctly evaluate the performance of the detector. We added person images from PASCAL VOC2007 in order to have a more balanced training set. Fine-tuning our small network does not require a huge amount of data; we simply collected a balanced dataset of ~ 300 person and ~ 300 artwork images. We used vertical image flipping in training as data augmentation.

To evaluate the recognition system we annotated a larger set of images with the artwork id. To easily collect our recognition database we developed a tool based on our detection pipeline. We use our artwork detector to generate bounding boxes and the tracker described in Sect. 4.2, to link all boxes in a sequence, after a sufficient amount of frames of an artwork has been collected the user may simply select an existing id or enter a new record in the database. Considering the non-parametric nature of the recognition system discussed in Sect. 4.1, this process can be run multiple times to enrich the dataset.

Finally, to test our full pipeline, we use sequences accounting for $8,820$ frames. We also pay attention to include shots where multiple artworks are visible. In each frame, we annotated the bounding box and the label of each visible artwork. At the end of the process we collected a total of ~ 250 seconds of video with $7,956$ detections.

7.2. Artwork detection

In the first experiment, we evaluate the performance of the artwork detection system. After performing the fine-tuning of the network on our dataset, we run the trained detector on the test set and measure the average precision. As described in Sec. 4, we aim at detecting the artworks that are in front of the wearer and give less importance to the ones in the distance. As a result, we only consider detections of a minimum area T that are indicative of a small distance from the user. We report in Fig. 9 the average precision obtained by the detection system when varying the minimum area of the considered detections. The area is normalized with respect to the dimension of the video frame. It can be observed that the average precision increases with the minimum area of the box and reaches the maximum value of 0.9 at 40% of the area. This means that the classifier is more effective at recognizing nearer artworks. We note that increasing the minimum box size area is not always a guarantee that the detector will be more precise. While far detections are very prone to errors due to the

small object scale, some detection errors may also be present at a near distance due to blur.

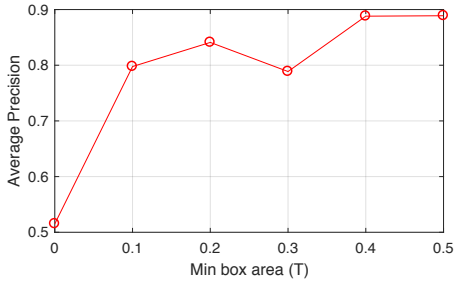

Fig. 9. Average precision of artwork detection when varying the minimum box area.

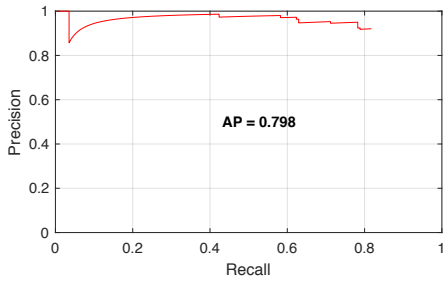

Fig. 10. Precision-recall curve for artwork detection using a threshold $T = 0.1$.

Selecting a good value for the minimum box area is therefore a trade-off between a good precision and the proximity that a wearer has to be to an artwork. We chose the final value of $T = 0.1$, that provides a significant improvement of precision over the bare detector output and a maximum distance of ~ 5 meters. In Fig. 10 we show the precision-recall curve relative to the final T value. Our system has a very good precision at high recall rates. Hence, the curve exhibit only a small amount of loss in terms of precision until 0.8 recall. We note that higher recall can not be reached due to the T threshold selected according the results reported in Fig. 9. For this reason, the curve is truncated at that point.

7.3. Artwork recognition: nearest neighbour evaluation

In this experiment we evaluate the effect of the number of nearest neighbors on artwork recognition, in terms of precision. Descriptors are computed concatenating two layers of the network, according to the approach described in Sect. 4.1. Results are plotted in Figure 11 with accuracy using 1 nearest neighbour, where features extracted from layers 3 or 4 are combined with the other layers. The figure shows again that the combination of the 3^{rd} and 4^{th} layers provides the best results. In Figure 12 we report the accuracy when varying the number of nearest neighbours using the just selected best combination. We observe that 1 nearest neighbor provides the best performance in recognizing an artwork. Accuracy degrades when more nearest neighbours are used in voting the correct artwork id. This is due to the fact that the environment we are testing the system in, has high variability in lighting conditions. Moreover we acquired multiple poses for each artwork. It is clear that for each query only a few samples will be in the similar pose/lighting conditions while increasing the amount of neighbours will just add noisy data to the vote pool.

7.4. Temporal Processing Evaluation

In order to measure the effectiveness of the three strategies for temporal processing described in Sec. 4.2, we perform an experiment where several of their combinations are tested. The annotated video sequences are thus fed to a simulation of the system, where each combination of output bounding box and label is tracked and compared to the ground truth data. The thresholds are fixed at $T = 0.1$, $M = 15$ and $P = 20$. We measure the number of detections where the artworks are correctly and incorrectly labeled, and the number of times the system chose to output the “generic” artwork label.

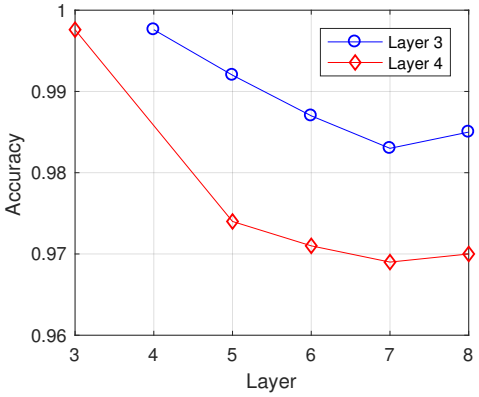

Fig. 11. Recognition accuracy of combinations of layer 3 and 4 with layers $[3, \ldots, 8]$.

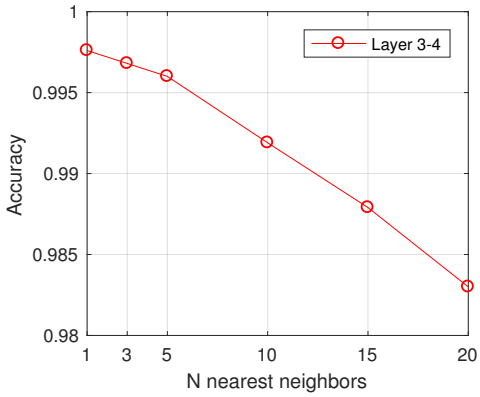

Fig. 12. Recognition accuracy of the best layer combination (layers 3-4), varying the number of nearest neighbors.

We report in Table I the result of the evaluation. In the first test (T1), we measure the performance of the system without any additional criterion as baseline, i.e. the frame by frame output of the recognition system. We observe that the system outputs the correct artwork for the majority of the detections ($\sim$ 70%), while about 30% were labeled as an incorrect artwork. By adding the *Distance* criterion, we see in test T2 that a slightly lower amount of detections were correctly labeled, but about half of the incorrect recognitions were considered generic, instead. This confirms that a large amount of errors are made on farthest artworks, since they are more difficult to recognize. In test T3, we observe the outcome of the *Consistency* strategy. In this case, almost all the incorrect artwork recognition are successfully exposed and classified as generic artwork output. This is due to the uncertainty of the vision system that rapidly shifts its prediction from frame to frame. In test T4, as seen in T2, adding the *Distance* criterion to *Consistency*, reduces the Incorrect recognitions. While the *Consistency* criterion by itself is able to almost nullify the incorrect recognitions, it is not robust to sparse errors. In fact, the system often swings from the correct recognitions to the generic label. This issue is resolved when combining this stringent strategy with the *Persistence* one, in test T5. This is visible quantitatively in the gain of the number of correct recognitions and the relative decrease of generic outputs, at the expense of increasing the incorrect ones. Combining all the criteria, as in T6, leads to a very low number of incorrect detections and a reasonable number of neutral artwork outputs, confirming our intuition about the efficacy of the three strategies. With only 22 wrong detections, the system predicts a wrong label approximately less than one cumulative second every ~ 5 minutes of video.

7.5. Voice Detection Evaluation

In this experiment we test the performance of the voice activity detection system on our dataset. We consider two simple strategies to emit a classification per second, namely Sample and Mean. The Sample strategy is just evaluating the classifier on a single audio frame per second, sampled at the beginning of a new second. This has the advantage to require only a single evaluation of the net. The Mean strategy, instead, consider all the predictions of net in a second and finally emits the mean of the values. This is more robust to the fluctuations of the classifier, at the expense of running the net continuously. With both strategy, in order to minimize the number of false positives, we measure the performance of the classifier varying the positive threshold.

Table I. **Performance by applying the three strategies for temporal smoothing:** C stands for Consistency, D for Distance and P for Persistence. We report the number of detections where, respectively, the artwork was correctly recognized, the artwork was misplaced for another one and where the system chose to output a generic "artwork" label.

Test	Strategy			Correct	Incorrect	Skipped
	C	D	P			
T1	✗	✗	✗	5,598 (∼70%)	2,358 (∼30%)	0 (0%)
T2	✗	✓	✗	5,334 (∼67%)	1,267 (∼16%)	1,355 (∼17%)
T3	✓	✗	✗	4,475 (∼56%)	36 (∼0%)	3,445 (∼43%)
T4	✓	✓	✗	4,363 (∼55%)	11 (∼0%)	3,582 (∼45%)
T5	✓	✗	✓	5,141 (∼65%)	61 (∼1%)	2,754 (∼35%)
T6	✓	✓	✓	4,966 (∼62%)	22 (∼0%)	2,968 (∼37%)

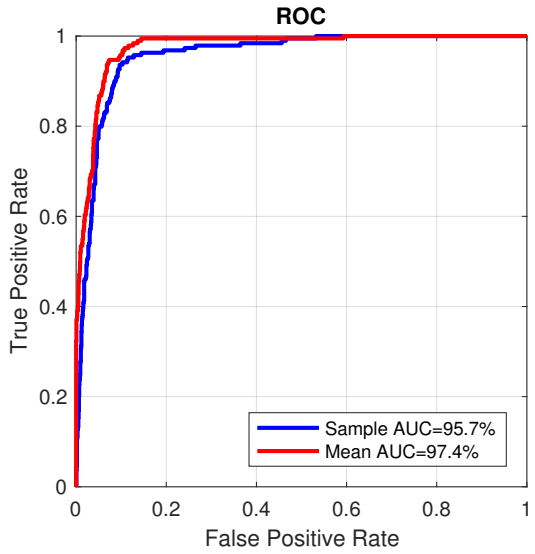

Fig. 13. Receiver operating characteristic curve of the tested voice activity classifiers.

We report the receiver operating characteristic (ROC) curve of the two strategies in Fig. 13. We observe that both strategies have a high area under the curve (AUC), meaning that they correctly predict the presence of the voice most of the time. The Mean strategy has a higher AUC and has always an higher true positive rate at the same false positive rate than Sample. This confirms that the Mean strategy is more robust than Sample.

8. USER EXPERIENCE EVALUATION

Modern tourist guides have their origins dating up to 17^{th} and 18^{th} centuries *Grand Tour*, and their role has become a key component in modern tourism experiences and applications. Guide functions can be highly specialized and require a lot of expertise and interpersonal skills to satisfy tourist needs. [Cohen 1985] describes guide roles as characterized by instrumental (guide), social (animator), interactional (leader) and communicative (intermediator) functions. Instrumental functions represent services capable to convey essential tourism information such as path finder to artworks location and related infos. Interactional features offer the ability to create a relation between the user and the contextual environment (e.g. informations about artworks). Improving this ability also means improving interaction. Sociality involves all the ac-

Table II. **Functions comparison of our guide with respect to human and traditional audio guides**

Type	Instrumental	Social	Interactional	Communicative
Human Guide	* * *	* * *	* * *	* * *
Audio Guide	*	–	–	* *
SeeForMe	* *	* *	* *	* *

tivities aiming at engaging the users with collaborative and not isolated experiences. Communicative functions facilitate access to artwork insights and targeted content, e.g. pointing out objects of interest. All these functionalities are fulfilled at their best by humans and modern audio guides have only partially replaced the complex role of the human guide. On the other hand the use of technology has improved aspects such as efficiency, sociality and autonomy in providing information communication under the so-called smart tourism paradigm. In Table II we compare guide-role functions as provided by human and traditional audio-guides with those available in our system.

The main differences between traditional audio guides and our system can be found in the interactional and social aspects of the provided experience. In traditional audio-guides the fruition of content is for the most part passive and the user has a low control on the reproduction of content. As regard to this, SeeForMe offers a more user friendly experience giving the possibility to interrupt the audio playback manually and automatically, and to restart the reproduction from the last point. Playback control can be achieved also using voice commands. Furthermore, activation of contents in audio-guides is cumbersome: locations or room numbers have to be searched and inserted manually reducing usability whilst SeeForMe allows automatic artwork recognition; this fact results in further differentiation of the Instrumental function, even in case of automatically triggered guides (e.g. those using RFID): an audioguide, being completely passive can not direct the visitor, while SeeForMe, highlighting the presence of other artworks as shown in Fig. 8, can direct the visitor within the museum. As for sociality, if it is true that social networking mechanisms are commonly provided in tourism apps for mobile phones, these functionalities are intended for virtual or remote users and not real companions. Indeed, audio-guides hinder communication between visitors (especially group visitors) and make people feel isolated, causing them to stop using devices and applications in order to join others. SeeForMe in this sense is more social because it automatically understands the context detecting if the user loses attention or simply is speaking with someone else, adapting the interaction with the system consequently.

In order to assess the whole experience offered by the system in a real environment, we conducted an evaluation of its usability. According to ISO, usability is defined as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use”. However, there are several usability models and types of assessments, like ISO standards on quality models (ISO 9126), user-centered design (ISO 9241) or user-centered approaches. A review of techniques for mobile application usability evaluation is provided in [Nayebi et al. 2012].

The usability study was performed with the popular Standard Usability Scale (SUS) [Brooke 1996], that follows a user-centered approach. Testing a user interface with SUS means, given a scenario of use and one or more tasks to solve, administer a 10 point questionnaire to a group of users. SUS is a Likert scale [Trochim et al. 2006], therefore questions address extreme cases, with opposite meaning and alternating positive with negative sentences. Answers to questions are numbers from 1 to 5, expressing all ranges from “Strongly Disagree” to “Strongly Agree”. This testing

strategy has been proved effective in removing acquiescence bias. The alternation of positive and negative items makes sure that users read it carefully. Nielsen states that it is sufficient to collect 5 polls to find the 85% of design errors of an interface or experience [Nielsen and Molich 1990]; in these tests we recruited twelve persons, divided in two groups of six people each.

We tested two different scenarios, supervised and unsupervised, in which users were asked to perform two simple tasks: *i*) "Activate the audio-guide for one or more artworks of your interest", *ii*) "After the visit, use the app to find again the information about one artwork you have seen". In the former a group of people receives from the authors of the paper a spoken detailed description of our system, thoroughly explaining the Android app functionalities and also detailing insights on the recognition engine. In the latter scenario instead, users are given the same two simple tasks but without any explanation on the application functions.

After normalization SUS scores are expressed in the range $[0 - 100]$. They do not represent percentages but can be interpreted with an adjective rating [Bangor et al. 2009]. A score over 68 means that the user interface or experience design is above average [Sauro and Lewis 2012] and that tasks can be completed without too much fatigue. Scores above 80 usually means that the interface is correctly designed and that the user experience is enjoyable.

We obtained an average SUS of 74.0 for the unsupervised scenario and 79.5 for the supervised scenario. The small gap in scores measured in the two scenarios, and their closeness to 80, means that the user interface is easy to use and that the training provided by expert users is not strictly required to perform tasks correctly. Nonetheless, considering that the user experience increased when users received a brief tutorial on the features and technical details, means that there is some room for further improving the design of interface and user experience of our app.

Users, when interviewed, mostly agreed that the automatic start/stop of the guide is the feature that makes the experience smooth. Regarding negative aspects of our system, most of the points made by users were about the need to access menus to change the language or other options.

9. CONCLUSION

We have presented a system running on the NVIDIA Jetson TK1 and on NVIDIA Shield Tablet K1. Our approach jointly solves two problems: contextual analysis and object recognition. We apply our efficient video processing pipeline and multi-sensor analysis to improve museum experience. Our method allows to profile in real-time visitor interests and to provide instantaneous feedback on the artworks of interest. We exploit audio and sensor data to improve the user experience reducing the intrusiveness of the smart audioguide.

Our Android app, allows users to switch between a fully automated experience to a more interactive mode. Moreover, after a visit is completed it is possible to for the user to look back and listen, or read, again about the artwork that gathered his interest.

Usability testing revealed few pitfalls of our experience design, but users where satisfied on average and provided some suggestions to improve the user interface further.

ACKNOWLEDGMENTS

The authors acknowledge and thank Giovanni Taverriti and Stefano Lombini for their work on the early version of the vision system, and Alessandro Sestini for his help in the development of the initial version of the Android front-end.

REFERENCES

Rao Muhammad Anwer, Fahad Shahbaz Khan, Joost van de Weijer, and Jorma Laaksonen. 2016. Combining Holistic and Part-based Deep Representations for Computational Painting Categorization. In *Proc. of ACM on International Conference on Multimedia Retrieval (ICMR)*.

Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies* 4, 3 (2009), 114–123.

Jonathan P Bowen and Silvia Filippini-Fantoni. 2004. Personalization and the web from a museum perspective. In *Proc. of Museums and the Web (MW)*.

John Brooke. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* (1996), 189–194.

Kenneth Bullington and JM Fraser. 1959. Engineering aspects of TASI. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics* 78, 3 (1959), 256–260.

Erik Cohen. 1985. The tourist guide: The origins, structure and dynamics of a role. *Annals of Tourism Research* 12, 1 (1985), 5–29.

Alberto Del Bimbo, Walter Nunziati, and Pietro Pala. 2009. David: Discriminant analysis for verification of monuments in image data. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*.

Thomas Drugman, Yannis Stylianou, Yusuke Kida, and Masami Akamine. 2016. Voice Activity Detection: Merging Source and Filter-based Information. *IEEE Signal Processing Letters* 23, 2 (2016), 252–256.

Veron Eliseo and Levasseur Martine. 1991. Ethnographie de l'exposition. *Études et recherche, Centre Georges Pompidou, Bibliothèque publique d'information* (1991).

Benjamin Elizalde and Gerald Friedland. 2013. Lost in segmentation: Three approaches for speech/non-speech detection in consumer-produced videos. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*.

Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. 2014. Scalable object detection using deep neural networks. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2155–2162.

Florian Eyben, Felix Weninger, Stefano Squartini, and Björn Schuller. 2013. Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Ross Girshick. 2015. Fast R-CNN. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*.

Loc Nguyen Huynh, Rajesh Krishna Balan, and Youngki Lee. 2016. DeepSense: A GPU-based Deep Convolutional Neural Network Framework on Commodity Mobile Devices. In *Proc. of Workshop on Wearable Systems and Applications (WearSys)*.

Svebor Karaman, Andrew D. Bagdanov, Lea Landucci, Gianpaolo D'Amico, Andrea Ferracani, Daniele Pezzatini, and Alberto Del Bimbo. 2016. Personalized multimedia content delivery on an interactive table by passive observation of museum visitors. *Multimedia Tools and Applications* 75, 7 (2016), 3787–3811.

Jens Keil, Laia Pujol, Maria Roussou, Timo Engelke, Michael Schmitt, Ulrich Bockholt, and Stamatia Eleftheratou. 2013. A digital look at physical museum exhibits: Designing personalized stories with handheld Augmented Reality in museums. In *Proc. of Digital Heritage International Congress (DigitalHeritage)*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*. 1097–1105.

Tsvi Kuflik, Zvi Boger, and Massimo Zancanaro. 2012. *Analysis and Prediction of Museum Visitors' Behavioral Pattern Types*. Springer Berlin Heidelberg, 161–176.

Seyyed Salar Latifi Oskouei, Hossein Golestani, Matin Hashemi, and Soheil Ghiasi. 2016. CNNdroid: GPU-Accelerated Execution of Trained Deep Convolutional Neural Networks on Android. In *Proc. of ACM Multimedia (MM)*.

Mengyi Liu, Xin Liu, Yan Li, Xilin Chen, Alexander G. Hauptmann, and Shiguang Shan. 2015. Exploiting Feature Hierarchies With Convolutional Neural Networks for Cultural Event Recognition. In *Proc. of IEEE International Conference on Computer Vision (ICCV) Workshops*.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. In *Proc. of European Conference on Computer Vision (ECCV)*. http://arxiv.org/abs/1512.02325

Austin Meyers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, and Kevin P. Murphy. 2015. Im2Calories: Towards

an Automated Mobile Vision Food Diary. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*.

Ananya Misra. 2012. Speech/Nonspeech Segmentation in Web Videos. In *Proc. of Annual Conference of the International Speech Communication Association (Interspeech)*.

Saman Mousazadeh and Israel Cohen. 2011. AR-GARCH in presence of noise: parameter estimation and its application to voice activity detection. *IEEE Transactions on Audio, Speech, and Language Processing* 19, 4 (2011), 916–926.

Fatih Nayebi, Jean-Marc Desharnais, and Alain Abran. 2012. The state of the art of mobile application usability evaluation. In *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*.

Jakob Nielsen and Rolf Molich. 1990. Heuristic Evaluation of User Interfaces. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems (CHI)*. DOI:http://dx.doi.org/10.1145/97243.97281

David Picard, Philippe-Henri Gosselin, and Marie-Claude Gaspard. 2015. Challenges in Content-Based Image Indexing of Cultural Heritage Collections. *IEEE Signal Processing Magazine* 32, 4 (2015), 95–102.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*.

Jeff Sauro and James R. Lewis. 2012. *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*. 1–9.

Ryosuke Tanno, Koichi Okamoto, and Keiji Yanai. 2016. DeepFoodCam: A DCNN-based Real-time Mobile Food Recognition System. In *Proc. of International Workshop on Multimedia Assisted Dietary Management (MADiMa)*.

William M. Trochim and others. 2006. Likert scaling. *Research methods knowledge base, 2nd edition* (2006).

Fabio Vesperini, Paolo Vecchiotti, Emanuele Principi, Stefano Squartini, and Francesco Piazza. 2016. Deep neural networks for Multi-Room Voice Activity Detection: Advancements and comparative evaluation. In *Proc. of International Joint Conference on Neural Networks (IJCNN)*.

Yiwen Wang, Natalia Stash, Rody Sambeek, Yuri Schuurmans, Lora Aroyo, Guus Schreiber, and Peter Gorgels. 2009. Cultivating Personalized Museum Tours Online and On-Site. *Interdisciplinary Science Reviews* 34, 2-3 (2009), 139–153.

Kyoung-Ho Woo, Tae-Young Yang, Kun-Jung Park, and Chungyong Lee. 2000. Robust voice activity detection algorithm for estimating noise spectrum. *Electronics Letters* 36, 2 (2000), 180–181.

Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. 2016. Quantized Convolutional Neural Networks for Mobile Devices. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*.

Keiji Yanai, Ryosuke Tanno, and Koichi Okamoto. 2016. Efficient Mobile Implementation of A CNN-based Object Recognition System. In *Proc. of ACM Multimedia (MM)*.

Massimo Zancanaro, Tsvi Kuflik, Zvi Boger, Dina Goren-Bar, and Dan Goldwasser. 2007. Analyzing Museum Visitors' Behavior Patterns. In *Proc. of International Conference User Modeling (UM)*.

C. Lawrence Zitnick and Piotr Dollár. 2014. Edge boxes: Locating object proposals from edges. In *Proc. of European Conference on Computer Vision (ECCV)*. 391–405.

Received November 2016; revised ? 2017; accepted ? 2017