Received April 29, 2016, accepted May 23, 2016, date of publication July 27, 2016, date of current version August 26, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2594799

Cyber-Physical Signage Interacting With Gesture-Based Human–Machine Interfaces Through Mobile Cloud Computing

LIEN-WU CHEN[1] (Senior Member, IEEE), YU-FAN HO[1], MING-FONG TSAI[1], HSI-MIN CHEN[1], AND CHI-FU HUANG[2]

[1]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[2]Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 62102, Taiwan

Corresponding author: L.-W. Chen (lwuchen@fcu.edu.tw)

This work was supported by MOST under Grant 102-2221-E-035-031-MY3, Grant 104-2221-E-035-021, Grant 104-2218-E-035-016, Grant 105-2622-E-035-008-CC3, and Grant 105-2634-E-035-001.

ABSTRACT In this paper, we propose a cyber-physical signage interacting framework for the interaction between digital signage and mobile users using smart handheld devices, such as smartphones and tablets. The proposed framework can provide diverse multimedia/feedback services to mobile users interacting with digital signage through face detection, classification, and recognition techniques based on mobile cloud computing. Mobile users only need to click and drag the interested service over the face of advertising celebrities/endorsers displayed on digital signage and can obtain supplementary multimedia information or feedback related comments. Using the intuitive way of gesture-based operations, mobile users can directly interact with digital signage through their handheld devices. Our framework reveals an innovative human–machine interface for signage interacting between digital signage and mobile users. In addition, we integrate our framework with a face cache mechanism that can make the interaction delay as small as possible for popular signage. Furthermore, an Android-based signage interacting system is implemented to verify the feasibility and superiority of our framework. Experimental results show that our approach outperforms the existing methods and can significantly reduce the average consumption time of obtaining interested multimedia contents from digital signage.

INDEX TERMS Cloud computing, cyber-physical system, human-machine interaction, mobile device, signage interacting.

I. INTRODUCTION

With the rapid growth of various media information and wireless communication technologies, there is an obvious trend that the way of advertisement dissemination has shifted from traditional static billboards to digital signage, which can be displayed dynamically and updated instantly. Compared to traditional static billboards, digital signage can broadcast diverse and rich information to nearby people with vivid multimedia contents instead of static pictures and texts. In addition, because digital signage has the feature of sensory stimulation that can effectively grab users' attentions, it possesses unique advantages over existing advertising equipment [1].

With respect to the pervasiveness of digital signage, it has been deployed in many public areas, such as airports, railway stations, hospitals, etc., to provide users with instant and useful information (e.g., the schedule of flights, trains, buses, or registrations). However, although digital signage is a platform capable of providing information in various multimedia formats, most of existing applications do not support real-time interaction with audiences. It is simply a one-way broadcasting where audiences can only obtain displayed information from digital signage. There is no instant way to facilitate audiences who can retrieve other detailed information if the contents displayed on digital signage are attractive to them. Hence, the improvement for the interaction between digital signage and audiences to help users rapidly get interested information or instantly feedback comments has become an important issue [2]. In addition, when interacting with digital signage via mobile devices, most users prefer to use intuitive and efficient human-machine interfaces with one hand [3].

References [4] and [5] developed a smart signage system to synchronize the displayed content between digital signage and mobile devices. By using an orientation sensor embedded in a mobile device, a user can use his/her device to point to one of predesigned digital signage and then the content currently shown on that signage can be immediately displayed on the device's screen. Thus, in a public place where a number of digital signage are deployed, a user can leverage his/her mobile device to get interested information from the selected one of digital signage.

Reference [6] designed a digital sensing signage based on Wi-Fi technologies, enabling content delivery from digital signage to mobile devices. When users come close to one of digital signage and execute a dedicated App, mobile devices can search for Wi-Fi signals around them. The signal issued from a specific digital signage is identified by checking the Service Set Identifier (SSID), Medium Access Control (MAC) Address, and Received Signal Strength Indication (RSSI). The content displayed on the selected digital signage can be retrieved by users' mobile devices according to the identified signal.

References [7] and [8] proposed a digital signage system that allows mobile devices to scan a two-dimensional barcode displayed on signage and then directly download and install the corresponding App on the devices. Through this way, a user can see multimedia contents that are currently displaying on digital signage on his/her device screen. In addition, the user can share the retrieved information with others who have the same App installed.

Reference [9] presented a RFID-based signage system that gives navigation information to autonomous robots. A navigation algorithm was designed to specify the searching process of RFID signals and related interpretation. The presented signage system was implemented in an autonomous and social robot for a real indoor environment to support the navigation tasks of the robot.

However, most of existing signage systems play advertising videos repeatedly to their audiences. Although this kind of signage can get audiences' attentions at first glance, it cannot ensure that audiences keep paying attentions on the contents displayed on signage. If audiences intend to know more about the displayed contents, they have to search for related information on their own. Although Quick Response (QR) codes [10] can be shown on digital signage for enabling users to receive additional information about signage contents by scanning QR codes, users are not able to choose the type of information they intend to obtain based on their interests. Hence, through the cooperation between mobile devices and digital signage, audiences can further request customized services from digital signage according to their needs instead of only being aware of displayed information on digital signage.

On the other hand, the system load and response time of signage interacting can be balanced using mobile cloud computing [11], [12], where there are three different architectures as follows [13]. In *local mobile cloud computing* [14], a peer-to-peer (P2P) network can be formed by mobile devices to cooperatively provide their computing resources to all P2P members. In *remote mobile cloud computing* [15], the computation ability and storage space of cloud servers can be directly utilized by mobile devices without involving any intermediate server. In *cloudlet mobile cloud computing* [16], [17], a dedicated cloudlet server is adopted by mobile devices to indirectly transfer their computation tasks to cloud servers.

In this paper, we propose a cyber-physical signage interacting framework, called AdYou, and implement an Android-based AdYou system to verify the feasibility and superiority of our framework. The architecture of cloudlet mobile cloud computing is adopted in the AdYou framework for signage interacting based on face detection (in the mobile device), classification (in the cloudlet server), and recognition (in the cloudlet server and cloud servers) techniques. AdYou takes advantage of mobile devices to cooperate with digital signage for the interaction between mobile users and digital signage. With the cooperation of digital signage and mobile devices, AdYou can deliver on-demand information to users' mobile devices according to what they are interested in the contents displayed on digital signage. Compared to traditional digital signage, AdYou can provide users with diverse services to obtain corresponding multimedia and realize the interaction between reality and virtuality. Additionally, the contents retrieved from digital signage can be immediately shared between mobile users for human-to-human interaction.

The contributions of our proposed framework are four-fold. First, it reveals an innovative human-machine interface for signage interacting between digital signage and users based on mobile cloud computing. Second, it integrates with local face detection, cloudlet face classification and caching, and cloud face recognition to increase the recognition success ratio of digital signage. Third, it exploits the computing resources of the mobile device, back-end server, and cloud servers to reduce overall interaction time. Finally, an Android-based signage interacting system is implemented to verify the feasibility and superiority of the proposed framework.

The rest of this paper is organized as follows. Section II introduces the system architecture of AdYou. Section III presents the proposed cyber-physical signage interacting framework. An Android-based signage interacting system is implemented in Section IV. Experimental results are discussed in Section V. Finally, Section VI concludes the paper.

II. SYSTEM MODEL

Fig. 1 shows the system architecture of AdYou. On the client side, a user uses a mobile device with Wi-Fi/4G communications to run the dedicated AdYou App. The user can use the camera of his/her mobile device to focus on the faces of advertising celebrities/endorsers displayed on digital signage and then click and drag one of multimedia/feedback service icons (which can be customized for each digital signage in the AdYou App) over one of celebrities'/endorsers' faces. Next, the AdYou App sends the target's face image detected by face detection techniques [18] to the AdYou server for local and remote face recognition [19] through face caching and cloud servers, respectively.

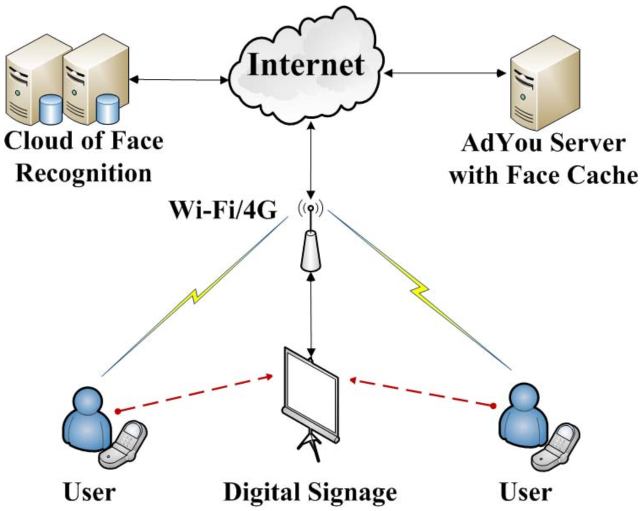

FIGURE 1. System architecture of cyber-physical signage interacting.

On the server side, the received target's face image is forwarded by the AdYou server to the cloud of face recognition and then recognized by cloud servers to obtain the advertisement identifier of AdYou. In addition, the target's face image and its recognized advertisement identifier are cached in the AdYou server in order to reduce the response time of future face recognition. The target's face can be directly recognized by the AdYou server if the face cache contains the target's face and its associated advertisement identifier. Finally, the AdYou server delivers the corresponding services related to the associated advertisement identifier to users. Note that if there is no suitable multimedia content (e.g., the limited number of coupons or the limited time to download virtual items) that users can get at the present time for the dragged service, the AdYou server can promptly recommend alternative services to users for accessing the other types of multimedia contents or feedback information in which they are interested.

Face recognition requires a large amount of database storage and consumes a lot of computing power when performing feature extraction and recognition. Cloud servers are used to handle the tasks of face recognition, while having mobile devices tackle the computing of face detection. Through distributing computation-intensive tasks to cloud servers and mobile devices, the computing efficiency can be optimized to reduce the response time. Thus, mobile users can use an efficient App provided by AdYou to rapidly access various types of multimedia/feedback services according to their demands.

By using AdYou, a user can use a finger to click and drag one of service icons over the face of advertising celebrities/endorsers displayed on digital signage. The user can obtain detailed information about the advertisement, such as movie trailers, discount coupons, mini games, music clips, etc. Compared to existing methods, AdYou allows digital signage to deliver the selected services and information to users. When users use mobile devices to interact with digital signage, AdYou can provide users with customized services and enable information access/feedback on demand. The goal is to minimize the consumption time to download interested contents and feedback related comments for mobile users. The signage interacting process between digital signage and mobile devices in the AdYou framework is optimized by addressing the following issues:

1. Face Detection: How should the face of advertising celebrities/endorsers can be accurately detected when mobile users drag a service icon to a specific face among multiple candidates displayed on digital signage?
2. Face Classification: How should the detected face can be correctly classified by the AdYou server to select a proper face recognition algorithm for improving recognition success ratios?
3. Face Recognition: How should the classified face can be successfully recognized to obtain the associated advertisement identifier through local face caching and remote cloud servers?
4. Face Caching: How should the recognized face and its associated advertisement identifier can be efficiently cached by the AdYou server to reduce the response time of future face recognition for popular signage?

III. CYBER-PHYSICAL SIGNAGE INTERACTING FRAMEWORK

The proposed cyber-physical signage interacting framework consists of the signage, client, server, and cloud modules. The signage module is used to show advertisement contents including videos, images, texts, etc., which can be instantly updated by the server module. The client module allows users to click and drag one of multimedia/feedback service icons over the signage's displayed contents on the camera preview screen of a mobile device. After a user drags a service icon to a face displayed on the signage module, the client module performs face detection and sends the detected face image to the server module for face recognition.

The server module forwards the received face image to the cloud module for obtaining the associated advertisement identifer through face recognition. In addition, the received face image and its associated advertisement identifier are cached in the server module to reduce the response time of future face recognition. In this way, the new received face image can be directly recognized by the server module if its face cache contains the same face image and its associated advertisement identifier. After obtaining the associated advertisement identifer, the server module delivers the corresponding content related to the associated advertisement identifier to the user.

In the proposed framework, the user operation to accomplish the interaction with digital signage is intuitively clicking and dragging a service icon on the touchscreen of a mobile device. The server module can receive the face image sent by the client module and dispatch it to both the cloud module and face cache (used to keep the recognized results of popular human faces in the server module) that perform face recognition simultaneously. The cloud module maintains a database of all the registered human faces and provides the cloud service of face recognition.

On the other hand, if a face cache hit occurs in the server module, the corresponding content of the associated advertisement identifer can be immediately replied to users. Otherwise, the server module returns the recognition results from the cloud module and store them into its face cache. Through the design of caching face images and identifier information, interacting performance can be further improved to reduce the response time of popular signage with which a lot of people interact. In particular, as both the server and cloud modules perform the task of face recognition in parallel after receiving a recognition request, the response time of cache missing is the same with that spent by the cloud module, which has no additional delay in the server module with face cache.

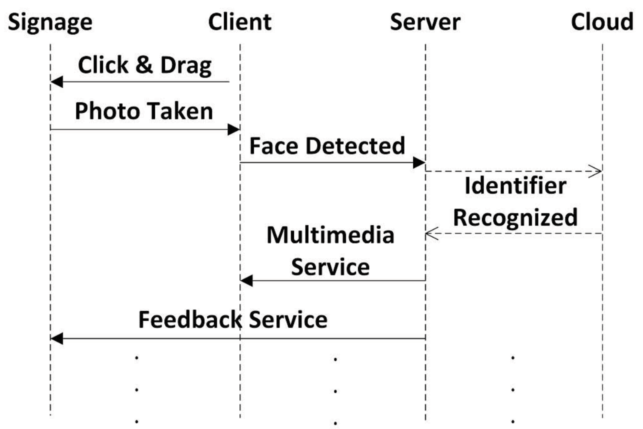

FIGURE 2. Operation flow of cyber-physical signage interacting.

Fig. 2 shows the operation flow of cyber-physical signage interacting. After the client module successfully logins to the server module, a user can use his/her finger to click and drag one of multimedia/feedback service icons on the touchscreen over a target face. Next, the client module activates the camera to take a photo and performs face detection to determine if there exists any human face in the photo. If there is no face detected, the client module prompts an error message regarding the failure of face detection and then the user can repeat the aforementioned actions of clicking and dragging again.

If there are one or more faces detected, the client module locates the face closest to the coordinate where is the last position the user's finger touched after he/she clicked and dragged the selected service icon. Once the closest face is detected, the client module transmits the detected face to the server and cloud modules for face recognition by comparing the received face image with those stored in face cache and databases, respectively. If the received face image is recognized and its associate advertisement identifier is obtained, the server module delivers the corresponding multimedia service to the client module or displays the feedback comment on the signage module.

To achieve accurate face detection and reduce the delay of user-signage interaction, the identifier recognition of the proposed framework is optimized by the following face detection, face classification, and face recognition phases.

A. FACE DETECTION PHASE

In the face detection phase, there are two image processing involved, where the first processing is coarse-grained face detection performed in the client module and the second processing is fine-grained face detection performed in the server module. When a user clicks and drags one of multimedia/feedback service icons over a target face, the client module records the coordinate where the user's finger last touched and captures the entire screen image. To avoid long face transmission and recognition time and to improve recognition success ratios, the client module performs face detection on mobile devices beforehand instead of directly sending the captured screen image to the server module.

For coarse-grained face detection, feature-based techniques [20], [21] are adopted in the client module, which perform training in advance for various features of faces, such as eyes, noses, and mouths. After the client module detects the features of faces, the locations and bounding boxes of detected faces are displayed on the touchscreen. Once the client module determines the target face based on the features of training faces and the coordinate where the user's finger last touched, the target face image is transmitted to the server module for fine-grained face detection.

For fine-grained face detection, the Harr Cascades [22], [23] method is adopted in the server module to further reduce the face part of the received image as much as possible, which can minimize the influence of backgrounds on recognition results. In the Haar Cascades method, the feature values of each image area are identified by using Harr-like features and executes Cascades for classifiers. For face detection, a common Haar feature is a set of two adjacent rectangles that lie above the regions of eyes and cheeks because the eye region is darker than the cheek region. The positions of those rectangles can be used to find a accurate bounding box for the face in fine-grained face detection by

$$f(x) = \sum_{i \in r_w} p(i) - \sum_{i \in r_b} p(i), \tag{1}$$

where $p(i)$ is the pixel value of i, r_w is the white rectangle, and r_b is the black rectangle. The adjacent white and black rectangles horizontally/vertically have the same size and shape. Similarly, different face features can be detected using more than two rectangles. For example, three rectangles can be used to detect a specific face feature (e.g., the eyes are darker than the nose) by calculating the sum in a center (white) rectangle subtracted from the sum within the right and left (black) rectangles.

By using the feature detection of two or more rectangles, a cascade of feature classifiers can be constructed based on a training set of positive and negative images. A classifier $h_i(x)$ consists of a feature f_i and a threshold θ_i as

$$h_i(x) = \begin{cases} 1 & \text{if } f_i(x) \geq \theta_i \\ -1 & \text{if } f_i(x) < \theta_i, \end{cases} \quad (2)$$

where x is sub-window of an image. In particular, a negative result from the first classifier leads to the immediate rejection of sub-windows. Otherwise, a second classifier is triggered by the positive result of the first classifier. A third classifier is triggered by the positive result from the second classifier, and so on. In this way, the computing time of face detection can be reduced by training classifiers using the machine learning algorithm of AdaBoost (i.e., Adaptive Boosting) and by adjusting the feature threshold to reject sub-windows as many (and early) as possible.

B. FACE CLASSIFICATION PHASE

The face photos taken under various environmental conditions may cause incorrect recognition results when applying different face recognition algorithms, where the Eigenface [24], [25], Fisherface [26], and Local Binary Patterns Histograms (LBPH) [27], [28] algorithms are used in the server module. After coarse-grained and fine-grained face detection, the detected face images are further classified to determine which one of the used algorithms is the most appropriate for the next phase of face recognition. As shown in Fig. 3, the server module determines the variation of lightness between the detected face and training faces (e.g., Fig. 3(a) and Fig. 3(b)) and the similarity of colors between the face part and its background (e.g., Fig. 3(c) and Fig. 3(d)). Note that all original face images used in the paper are from the open database of PICS [29].

FIGURE 3. Influences of lightness and backgrounds in face images.

On the basis of our experimental results, the lightness variation has a significant influence on the Eigenface algorithm, which results in the decrease of recognition success ratios. In contrast, the Fisherface algorithm is more tolerant in the different degrees of lightness than the Eigenface algorithm. In particular, the LBPH algorithm can produce accurate recognition results on the face image with high lightness variation and dissimilar backgrounds. To obtain the variation of lightness, we convert the face images from the color space of Red, Green, and Blue (RGB) to that of Hue, Saturation, and Lightness (HSL) and then use the Lightness values to determine the lightness differences between the detected face and training faces.

However, the face and background colors could be very similar to each other, as shown in Fig. 3(b), which causes a problem that fails to distinguish a face from its background and thus decreases the success ratios of face recognition algorithms. To solve this problem, we determine the similarity of a face and its background by considering the differences of pixel colors between them. Through performing fine-grained face detection, we can obtain the bounded face part to be recognized. Thus, the average pixel value inside the bounded face part can be compared with the pixel colors outside the bounded face part. Therefore, we can determine whether the colors of the face and its background are too similar to be distinguished in the feature extraction of face recognition.

According to our realistic trials, if more than 36% of the pixel colors outside the bounded face part are within the range between plus and minus 6% of the average pixel value inside the bounded face part, the face image is determined to be similar to its background. In addition, if the lightness difference between the detected face and training faces is less than 14%, between 14% and 33%, or greater than 33%, the face image is determined to has a low, median, or high lightness variation, respectively.

C. FACE RECOGNITION PHASE

After the face classification phase is done, a feasible method is selected among Eigenface, Fisherface, and LBPH algorithms to perform face recognition for the classified images based on realistic trials. The set of face samples stored in the database is taken in an indoor environment with normal lightness and trained by the application programming interfaces of Eigenface, Fisherface, and LBPH algorithms in the Open Source Computer Vision (OpenCV) library [30]. The Eigenface algorithm extracts features through training faces in a Principle Components Analysis (PCA) manner. Based on the composition of features appeared in each group of face samples, a featured face can be derived as a representative face, as shown in Fig. 4a. However, the recognition success ratio of the Eigenface algorithm decreases when the face images are taken with different face angles and environmental lightness, which cause different extracted features from face samples.

FIGURE 4. Feature composition in (a) Eigenface and (b) Fisherface algorithms.

The Fisherface algorithm is designed on the top of PCA to combine with the Linear Discriminant Analysis (LDA) method, which can solve the aforementioned problem in the Eigenface algorithm. When the Fisherface algorithm performs face training, only the major features for different faces are kept, as shown in Fig. 4b, which can tolerate the influences of various face angles and environmental lightness. Therefore, the Fisherface algorithm can achieve higher recognition success ratios than the Eigenface algorithm for the face images taken under different environmental conditions from training faces. The LBPH algorithm aims at extracting local features whereas the influence of lightness is global in general, as shown in Fig. 5. Thus, the LBPH algorithm can achieve higher recognition success ratios than Eigenface and Fisherface algorithms even if there exists high lightness on face images to be recognized.

FIGURE 5. Comparison of high lightness variation in face images using the LBPH algorithm.

TABLE 1. Adaptive mechanism based on face classification.

Environmental conditions	Similar backgrounds	Dissimilar backgrounds
Low lightness variation	Fisherface	Eigenface
Median lightness variation	Fisherface	Fisherface
High lightness variation	Fisherface	LBPH

Tab. 1 lists each feasible algorithm selected in the proposed *Adaptive* mechanism for the face images classified based on various backgrounds and lightness. When the detected face has dissimilar colors to its background and similar lightness to face samples, the Eigenface algorithm is used to perform face recognition. On the contrary, when the lightness of the detected face is moderately higher/lower than that of face samples, the difference of lightness may cause face recognition failures in the Eigenface algorithm. In this case, the Fisherface algorithm is applied to face recognition instead of the Eigenface algorithm. In particular, when the lightness of the detected face is much higher/lower than that of face samples, the LBPH algorithm is adopted to recognize the detected face because it can tolerate the high level of lightness.

However, when the colors of the detected face are similar to those of its background, the similarity may cause the failures of feature extraction in the LBPH algorithm. Therefore, when the colors of the detected face and its background are similar, the Fisherface algorithm is used to perform face recognition. Through applying feasible algorithms to classified face images, the face photos taken under different environmental conditions can be correctly recognized, which can achieve higher recognition success ratios than traditional approaches that always use the same face recognition algorithm.

IV. SYSTEM IMPLEMENTATION

We have developed an Android-based AdYou system [31] consisting of digital signage, mobile App, back-end server, and cloud servers. Digital signage is responsible for displaying advertisement contents, such as videos, images, texts, coupons, virtual items, etc. In our system, the contents of digital signage are conducted by using HTML5 to generate advertisement webpages, as shown in Fig. 6, where the content types are including videos, images, and texts. In the mobile App, we adopt Android smartphones as a client-side platform and develop multimedia/feedback services enabling users to interact with digital signage.

FIGURE 6. Digital signage of AdYou.

For face detection executed in the developed mobile App, two different methods, camera hardware detection and Android software detection, are implemented. Camera hardware detection can detect faces in real time and has the advantage of speeding up the process of face detection, but not all smartphones can support this method. Alternatively, if a smartphone cannot support camera hardware detection, it can use Android software detection to detect faces. To avoid the incompatibility problem of smartphones, we set Android software detection as the default method (but can be switched to camera hardware detection) to ensure that all smartphones can successfully perform face detection.

Fig. 7 shows the system demonstration of AdYou. The graphical user interface of the mobile App consists of two parts that are a camera preview screen and a set of multimedia/feedback service icons, as shown in Fig. 7a. Mobile users can click and drag one of service icons over the camera preview screen, as shown in Fig. 7b. Next, the mobile App performs face detection to look for the face that is closest to the coordinate where the user's finger last touched, as shown in Fig. 7c. After the target face is detected, the image of the

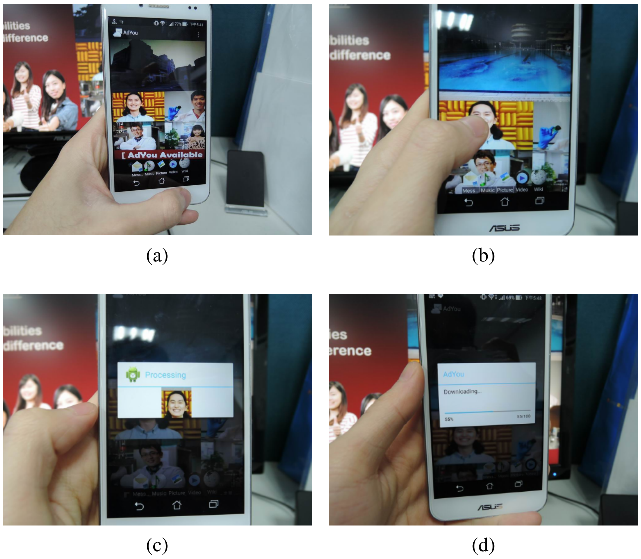

FIGURE 7. **(a) User interface, (b) service dragging, (c) face detection, and (d) multimedia downloading in AdYou.**

detected face is sent to back-end and cloud servers for face recognition.

When the back-end server receives the image of a detected face, it forwards the detected face to cloud servers that perform face recognition to recognize the associated advertisement identifier. After the back-end server receives the returned advertisement identifier from cloud servers. the recognized face image and its associated advertisement identifier are cached in the back-end server to reduce the interaction time of popular signage through the proposed Adaptive mechanism for local face recognition. As shown in Fig. 7d, according to the service icon (i.e., the video icon) dragged by the user and the target face (i.e., the male face) recognized by cloud servers, the back-end server can deliver the corresponding multimedia service to the user or feedback comments to digital signage. A demo video is provided online to show how signage interacting can be done by mobile users and how multimedia contents can be shared between mobile users through AdYou as fast as possible (see https://youtu.be/4kSIH7v81ho).

V. EXPERIMENTATIONS

In this section, we evaluate the consumption time to download multimedia contents by comparing AdYou with existing methods including TubeMate, FireFox, and NeoReader. TubeMate is an Android App facilitating users to retrieve multimedia contents by specifying keywords on a multimedia website. FireFox is a well-known Web browser that can direct users to file hyperlink pages and download related files presented on digital signage through keyword input for searching. NeoReader is an application that can scan the various formats of QR codes and barcodes. After scanning a QR code that contains a file hyperlink, NeoReader delegates the task of file downloading to the FireFox browser.

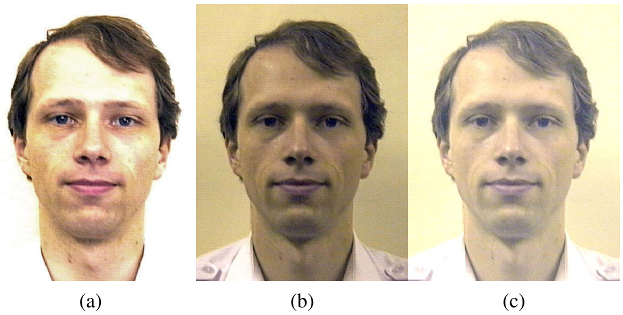

FIGURE 8. **Face images with (a) normal lightness, (b) similar backgrounds, and (c) high lightness.**

Compared with TubeMate and FireFox, the operations required by NeoReader do not include the step of keyword search. In our experiments, the open face database of PICS [29] is adopted. Each experiment is repeated 10 times by realistic trial and we take the average value.

First, we conduct experiments to evaluate the recognition success ratios of face recognition algorithms including Eigenface, Fisherface, LBPH, and the proposed Adaptive mechanism that can use face classification to select a feasible algorithm for recognizing face images. In the face database, the number of individuals is varied from 5 to 30 and each individual has eight face samples. As shown in Fig. 8, the face images with different lightness and backgrounds are used to be recognized by using Eigenface, Fisherface, LBPH, and our Adaptive mechanism. Fig. 9 shows that with the increase of the number of individuals, the Adaptive mechanism has the highest recognition success ratio and can be applied to face photos taken in various environmental conditions. In contrast, the results recognized by Eigenface, Fisherface, and LBPH algorithms are incorrect in some cases when the detected face has different lightness variation or background similarity from face samples.

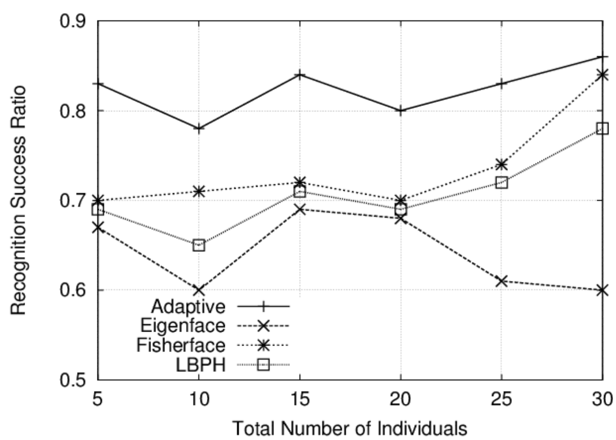

FIGURE 9. **Comparisons of the recognition success ratios with different numbers of individuals in face databases.**

On the other hand, Fig. 10 shows the comparisons of recognition success ratios with different numbers of face

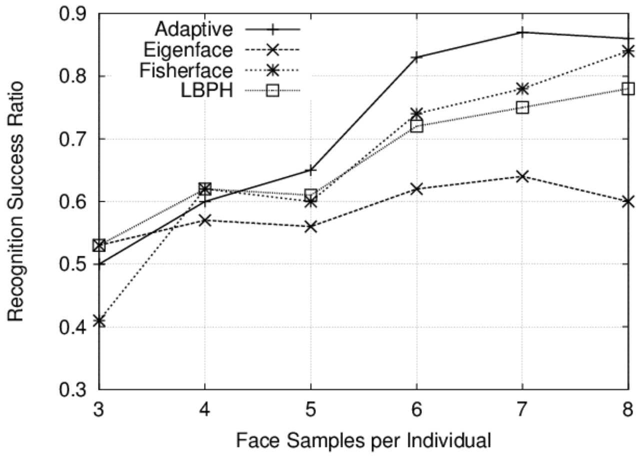

FIGURE 10. Comparisons of the recognition success ratios with different numbers of face samples for each individual.

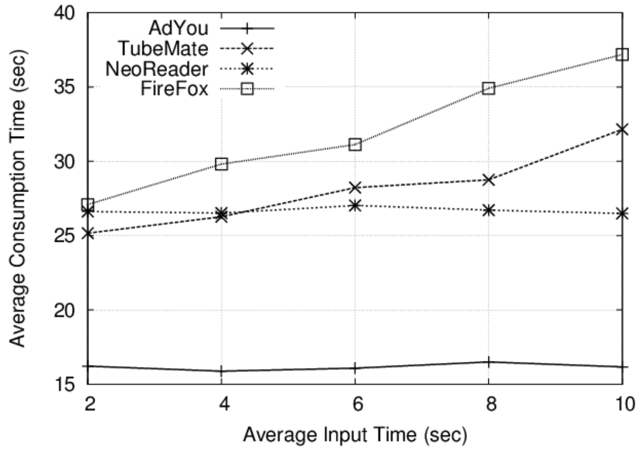

FIGURE 12. Comparisons of the average consumption time under different input times.

samples for each individual. The total number of individuals is 30 and the number of face samples for each individual is varied from 3 to 8. Similar to Fig. 9, the Adaptive mechanism achieves higher recognition success ratios than Eigenface, Fisherface, and LBPH algorithms. This is because the Adaptive mechanism can select the more suitable algorithm of face recognition according the similarity of environmental conditions between the detect face and face samples. In particular, when there are only few face samples in the database, the Adaptive mechanism can be used to keep recognition success ratios as high as possible. We use the Adaptive mechanism to recognize the detected faces in the following experiments of AdYou.

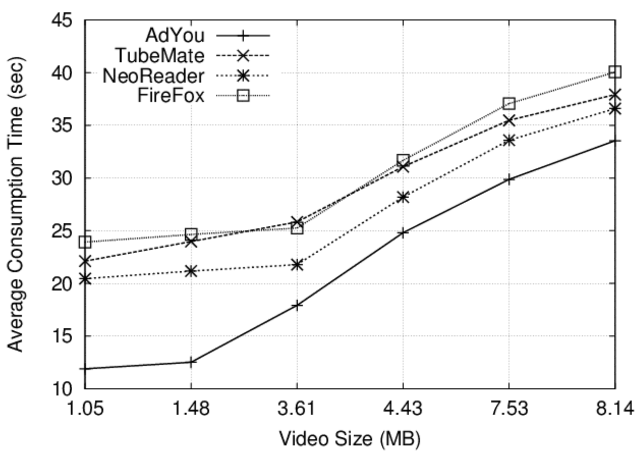

FIGURE 11. Comparisons of the average consumption time for video files with different sizes.

Fig. 11 shows the comparisons of the average consumption time using TubeMate, FireFox, NeoReader, and AdYou to download different sizes of advertisement videos from digital signage. Because both TubeMate and FireFox need to perform keyword search regarding the interested contents on digital signage, the average consumption time spent by TubeMate and FireFox are longer than NeoReader and AdYou. Although NeoReader can reduce the consumption time by scanning QR codes instead of keyword input, it still relies on a Web browser (e.g., FireFox) to accomplish file downloading. On the contrary, by using AdYou, users only need to perform a click-and-drag action and then immediately obtain the video contents of digital signage. Therefore, it can significantly reduce the average consumption time to download interested advertisement videos.

Fig. 12 shows the comparisons of the average consumption time under different keyword input times (from 2 to 10 seconds) for getting an advertisement video of size 3.61 MB. Because NeoReader has to further use FireFox to download files, it spends a longer time to get the advertisement video than TubeMate with low keyword input time in average. However, with the increase of keyword input time, it is obvious that NeoReader can achieve better performance than TubeMate and Firefox. This is because NeoReader can use QR code scanning to get the advertisement video instead of keyword input and search. Among the evaluated methods, AdYou can directly get the advertisement video without keyword input so that its average consumption time is the lowest. From Fig. 11 and Fig. 12, it can be seen that AdYou can save average consumption time as much as possible, no matter what size of multimedia contents is downloaded and what speed of keyword input is used.

Fig. 13 shows the comparisons of the average consumption time to obtain the advertisement video of 3.61 MB with and without local face cache under different cache hit rates. It can be seen that the average consumption time can be reduced when certain celebrities'/endorsers' faces of digital signage are cached in the AdYou server. Since the celebrities'/endorsers' faces displayed on popular signage are supposed to be recognized more often than others, which can make cache hit rates high, the recognition result of popular signage can be returned quickly without waiting for the

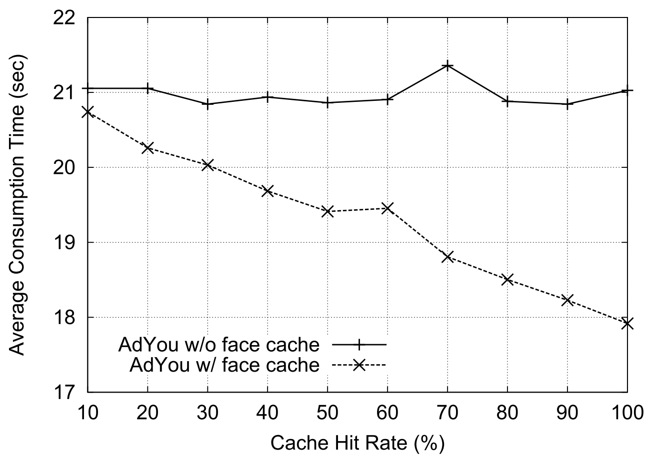

FIGURE 13. Comparisons of the average consumption time under different cache hit rates.

response of cloud servers, which can reduce the average consumption time of AdYou.

VI. CONCLUSIONS

In this paper, we design, implement, and evaluate a cyber-physical signage interacting framework based on mobile cloud computing. The computing resources of the mobile device, back-end server, and cloud servers are exploited to reduce overall interaction time consisting of local face detection, cloudlet face classification and caching, and cloud face recognition. Through the proposed framework, mobile users can interact with digital signage as fast as possible by using gesture-based operations. From realistic experiments, it can be concluded that the proposed framework can significantly reduce the total consumption time to obtain interested multimedia contents from digital signage. In addition, our framework provides mobile users with an intuitive and efficient human-machine interface for user-signage interaction.

REFERENCES

[1] R. Want and B. N. Schilit, "Interactive digital signage," *Computer*, vol. 45, no. 5, pp. 21–24, May 2012.

[2] T. Tung, R. Gomez, T. Kawahara, and T. Matsuyama, "Multiparty interaction understanding using smart multimodal digital signage," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 5, pp. 625–637, Oct. 2014.

[3] J. Lai and D. Zhang, "ExtendedThumb: A target acquisition approach for one-handed interaction with touch-screen mobile phones," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 3, pp. 362–370, Jun. 2015.

[4] J. She, J. Crowcroft, H. Fu, and P.-H. Ho, "Smart signage: An interactive signage system with multiple displays," in *Proc. IEEE Int. Conf. Green Comput. Commun., IEEE Internet Things, IEEE Cyber, Phys. Social Comput.*, Aug. 2013, pp. 737–742.

[5] J. She, J. Crowcroft, H. Fu, and P.-H. Ho, "Smart signage: A draggable cyber-physical broadcast/multicast media system," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 2, pp. 232–243, Dec. 2013.

[6] G. Kim, S. Lee, J. Park, S. Jung, S. Kim, and Y.-C. Shin, "Thumbnail selection: Delivering digital signage contents to mobile phone," in *Proc. Int. Conf. Adv. Commun. Technol. (ICACT)*, Jan. 2013, pp. 396–400.

[7] J. Lee, J. Lee, H. Jung, S. Moon, and K. Yoon, "Smart digital signage using smartphone," in *Proc. 15th Int. Conf. Adv. Commun. Technol. (ICACT)*, Jan. 2013, pp. 978–981.

[8] J. Lee and K. Yoon, "The application of digital signage system using smart device," in *Proc. 16th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2014, pp. 675–677.

[9] A. C. Paredes, M. Malfaz, and M. A. Salichs, "Signage system for the navigation of autonomous robots in indoor environments," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 680–688, Feb. 2014.

[10] S.-S. Lin, M.-C. Hu, C.-H. Lee, and T.-Y. Lee, "Efficient QR code beautification with high quality visual content," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1515–1524, Sep. 2015.

[11] A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393–413, 1st Quart. 2014.

[12] E. Ahmed, A. Gani, M. Sookhak, S. H. A. Hamid, and F. Xia, "Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges," *J. Netw. Comput. Appl.*, vol. 52, pp. 52–68, Jun. 2015.

[13] M. Ayad, M. Taher, and A. Salem, "Real-time mobile cloud computing: A case study in face recognition," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2014, pp. 73–78.

[14] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. ACM Mobile Cloud Comput. Services (MCS)*, Jun. 2010, Art. no. 6.

[15] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[16] L.-W. Chen, Y.-F. Ho, W.-T. Kuo, and M.-F. Tsai, "Intelligent file transfer for smart handheld devices based on mobile cloud computing," *Int. J. Commun. Syst.*, Feb. 2015.

[17] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *J. Netw. Comput. Appl.*, vol. 59, pp. 46–54, Jan. 2016.

[18] J.-S. Jang and J.-H. Kim, "Fast and robust face detection using evolutionary pruning," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 562–571, Oct. 2008.

[19] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 10, pp. 1042–1052, Oct. 1993.

[20] C. Schneider, N. Esau, L. Kleinjohann, and B. Kleinjohann, "Feature based face localization and recognition on mobile devices," in *Proc. Int. Conf. Autom., Robot., Control Vis. (ICARCV)*, Dec. 2006, pp. 1–6.

[21] S. Chaudhari, A. Kale, K. S. Kinage, and S. G. Bhirud, "Face feature detection and normalization based on eyeball center and recognition," in *Proc. Int. Conf. Future Comput. Commun. (ICFCC)*, May 2010, pp. V3-503–V3-507.

[22] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. I-511–I-518.

[23] A. Sharifara, M. S. M. Rahim, and Y. Anisi, "A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection," in *Proc. Int. Symp. Biometrics Secur. Technol. (ISBAST)*, Aug. 2014, pp. 73–78.

[24] L. H. Xuan and S. Nitsuwat, "Face recognition in video, a combination of eigenface and adaptive skin-color model," in *Proc. Int. Conf. Intell. Adv. Syst. (ICIAS)*, Nov. 2007, pp. 742–747.

[25] W.-D. Yang, "A simple approach to a small-scaled face recognition system," M.S. thesis, Dept. Comput. Sci. Inf. Eng., Nat. Central Univ., Taoyuan City, Taiwan, Jun. 2007.

[26] H.-J. Lee, W.-S. Lee, and J.-H. Chung, "Face recognition using Fisherface algorithm and elastic graph matching," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2001, pp. 998–1001.

[27] L. Chen, Y.-H. Wang, Y.-D. Wang, and D. Huang, "Face recognition with statistical local binary patterns," in *Proc. IEEE Int. Conf. Mach. Learn. (ICMLC)*, Jul. 2009, pp. 2433–2439.

[28] A. R. Ardakany, M. Nicolescu, and M. Nicolescu, "An extended local binary pattern for gender classification," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2013, pp. 315–320.

[29] *Psychological Image Collection at Stirling (PICS)*, accessed on Apr. 29, 2016. [Online]. Available: http://pics.psych.stir.ac.uk/

[30] *Face Recognition With OpenCV*, accessed on Apr. 29, 2016. [Online]. Available: http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html

[31] L.-W. Chen and Y.-F. Ho, "A face-based signage interacting system for mobile users using smartphones," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016.

LIEN-WU CHEN (SM'16) received the B.S. degree from Fu-Jen Catholic University in 1998, the M.S. degree from National Central University in 2000, and the Ph.D. degree from National Chiao Tung University in 2008, all in computer science and information engineering. He was a Research Assistant with the Institute of Information Science, Academia Sinica from 2001 to 2009. He joined the Department of Computer Science, National Chiao Tung University as a Post-Doctoral Research Associate in 2009, and qualified as an Assistant Research Fellow in 2010. From 2012 to 2015, he was an Assistant Professor with the Department of Information Engineering and Computer Science, Feng Chia University. He has been an Associate Professor since 2015. His research interests include wireless communication and mobile computing, especially in the Internet of Vehicles, Internet of Things, and Internet of People.

Dr. Chen has authored over 60 journal and conference papers and has 20 granted invention patents. Among those patents, the technologies for file transfer, application execution, and goods data searching methods based on augmented reality and cloud computing have been transferred to Acer Incorporated in 2011. He was a Guest Researcher with the Industrial Technology Research Institute in 2009, and the Chief Executive Officer with the RFID Resource Center, National Chiao Tung University, from 2009 to 2011. He is an Honorary Member of the Phi Tau Phi Scholastic Honor Society, a Lifetime Member of the CIEE and IICM, and a member of the ACM.

YU-FAN HO received the B.S. degree from the Department of Information Engineering and Computer Science, Feng Chia University, in 2014, where he is currently pursuing the M.S degree.

His research interests include wireless networks and vehicular ad hoc networks. He was a recipient of Best Demo Award at the 13rd IEEE International Conference on Pervasive Computing and Communications (2015) and the Merit Paper Award at the 20th Taiwan Academic Network Conference (2014).

MING-FONG TSAI received the Ph.D. degree from the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan. He is currently an Assistant Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His current research interests include error-control coding, multimedia communications, and vehicular communications.

HSI-MIN CHEN received the B.S. and Ph.D. degrees in computer science and information engineering from National Central University, Taoyuan, in 2000 and 2010, respectively. He is currently an Assistant Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His research interests include software engineering, object-oriented technology, service computing, and distributed computing.

CHI-FU HUANG received the B.S. degree from Feng Chia University in 1999, the M.S. degree from National Central University in 2001, and the Ph.D. degree in computer science and information engineering from National Chiao-Tung University in 2004. He was a Research Assistant Professor with the Department of Computer Science, National Chiao-Tung University, from 2005 to 2009. He has been an Assistant Professor with the Department of Computer Science and Information Engineering, National Chung Cheng University, since 2009. His research interests include wireless communication and mobile computing, especially in IoT and IoV.

• • •