Energy-Oriented Designs of an Augmented-Reality Application on a VUZIX Blade Smart Glass

Corridon McKelvey[1], Richard Dreyer[1], Donnel Zhu[2], Wei Wang[1] and John Quarles[1]
[1]The University of Texas at San Antonio, Texas, 78249
[2]Health Career High School, San Antonio, Texas 78229
{ wei.wang, john.quarles@utsa.edu}

***Abstract*—The advent of wearable devices introduces many opportunities with unconventional computing paradigms. In this work, we investigate energy-efficient designs of an augmented-reality application, *FaceReminder*, on a VUZIX Blade® smart glass by exploiting its unique see-through display and on-board camera. Powered with well-known facial recognition techniques, FaceReminder aims at helping people with prosopagnosia (i.e., face blindness) or short-memory of face-name connection by showing the names of the person on the see-through display. To cope with the limited resources (especially battery) of the smart glass, we explore designs that offload portion of the computation in facial detection and recognition to another mobile device (e.g., smart phone), which pairs with the glass via Bluetooth. Several optimization techniques, such as resolution adjustment and cropping, have been investigated to improve the latency and energy efficiency with reduced image sizes. We implemented FaceReminder and empirically evaluated its accuracy, latency and energy consumption of the major steps (including photo taking, resizing/cropping, Bluetooth transmission, facial detection and recognition). Compared to the baseline *Glass-Only* design, the most efficient *Paired Glass-Device* design with photos of reduced resolution and MTCNN facial detection technique can reduce the average latency by 73% and energy consumption by 78.9% (i.e., about 5X battery life improvement) while maintaining more than satisfactory 80% recognition accuracy.**

I. Introduction

The advancement of technology and its miniaturization have enabled the incorporation of computing in many unconventional devices. In particular, many wearable devices have been developed in the recent past, such as smartwatches, fitness trackers, smart clothing, head-mounted displays and smart glasses, which introduce unprecedented opportunities for applications in sports, well-being and healthcare [1], [2]. In this work, focusing on a recently released VUZIX Blade® smart glass [3], we investigate the energy efficient designs of an augmented-reality (AR) application, *FaceReminder*.

Although many face recognition apps have been developed for smart phones [4], none of them works for the smart glass. Very recently, by integrating the VUZIX Blade® smart glass with a powerful mobile server, an AI-powered face recognition system has been developed [5], which can help law enforcement officers and security guards to screen the crowd and match faces against a database of violators, missing people or suspects. Following the same principle, FaceReminder aims at helping people with prosopagnosia (i.e., face blindness) or short-memory of face-name connection through analyzing the photo taken with the embedded camera of the glass in real-time and showing the resulting names of family members or old friends on its see-through display.

The key technology that empowers FaceReminder is facial detection and recognition, which has been studied for decades with many well-known algorithms being developed [6], [7], [8]. In general, there are two major steps in identifying a person in a photo: *facial detection* and *facial recognition*. Here, facial detection tries to find the region of interest (ROI) in an image that contains a face (such as the Haar-Cascade Classifier based technique [9]). Then, the facial recognition step normally involves the comparison of the face in the ROI against a set of known faces to identify the person (e.g., FisherFace [10], Local Binary Pattern [11] and Eigenfaces technique [12]). Recently, with the development of machine learning techniques with various neural networks, many efficient deep facial detection/recognition techniques have been developed [13], including the facial detection with Multi-Tasking Cascading Convolutional Neural Network (MTCNN) [14].

The smart glass has its on-board computing components [3] and can perform the facial detection and recognition task in-place, which enables the *Glass-Only* design of FaceReminder. However, the computational intensive process of facial detection and recognition [9], [12] can quickly deplete its small battery capacity and reduce the operation time of the glass. Therefore, the major challenge in designing FaceReminder is to improve its energy efficiency and extend the battery lifetime.

Note that, workload offloading techniques have been widely exploited to address the issue of constrained resources (such as battery capacity) in mobile devices [15], [16], where the computation heavy tasks are normally offloaded to clouds and/or servers in data-centers [17]. In this work, to tackle the strict Size, Weight and Power (SWaP) constraints of the smart glass that has significantly limited its computing power and battery capacity [3], we also explore workload offloading techniques to improve its operation time. Specifically, via Bluetooth connection, the glass will be paired with another mobile device (e.g., smart phone) that has a much larger battery capacity with the Paired Glass-Device design.

With communication of the glass and device via Bluetooth being considered, there are clear trade-offs between computation and communication on their latency and energy efficiency that have to be carefully evaluated, as in the classical workload

This work is supported in part by an Intel gift to The University of Texas at San Antonio.

978-1-7281-5416-9/19 $31.00 © 2019 IEEE; (IGSC 2019, Alexandria, VA)

offloading scenario [17], [18]. In particular, to reduce the amount of data (i.e., the size of photos) to be transmitted from the glass to the device via Bluetooth, several optimization techniques, such as resolution adjustment, cropping and resizing to reduce photo sizes, have been considered. The associated costs and benefits in latency and energy efficiency improvement from the reduced image sizes have to be evaluated. More importantly, to obtain a satisfactory facial recognition accuracy, the impacts of the image size reduction techniques on the resulting accuracy need to be carefully evaluated as well before deploying them in the designs of FaceReminder.

In this work, by considering various facial detection and recognition algorithms, offloading and image size reduction optimizations, several designs of FaceReminder have been investigated. First, as the baseline design, the *Glass-Only* approach performs all necessary computation of facial recognition on the VUZIX Blade$^{\circledR}$ smart glass with its on-board computing cores. To improve its performance in latency and energy efficiency, different resolutions of the photos have been considered through resolution adjustment.

Then, with offloading technique being considered, the *Paired Glass-Device* design explores the computational capability of another mobile device to improve the energy efficiency on the smart glass. Here, to reduce the amount of data being transmitted which also takes time and consumes energy, in addition to resolution adjustment, the photos can be cropped on each side with the assumption that the Region-of-Interest (ROI) of the face normally appears in the center as the focus. Moreover, to further reduce the latency of facial detection and improve its efficiency while obtaining reasonably accurate detection result, we exploited a pre-trained MTCNN model that was converted for use in Tensorflow [19].

FaceReminder has been implemented for both Glass-Only and Paired Glass-Device designs with various optimization options. Through the empirical study with extensive experiments, we evaluated the performance of various designs and identified the most efficient option in terms of latency and energy efficiency. Our results show that, the Glass-Only design can take almost 10 seconds to process one photo with the native resolution. Without resolution adjustment, although the Paired Glass-Device design can reduce the average energy consumption per photo by 12.7%, it increases the latency by 48% due to the additional Bluetooth transmission delay. With proper resolution adjustment, both designs can significantly reduce (up to 2/3) the average latency and energy consumption per photo. By incorporating the fast facial detection with MTCNN model, the most efficient Paired Glass-Device design shows 73% and 79% reduction in average latency and energy consumption per photo, respectively.

The contributions of this work are summarized as follows:

- First, FaceReminder (an augmented-reality application) is designed for the VUZIX Blade smart glass considering its unique features. Both Glass-Only and Paired Glass-Device with offloading designs are considered;
- Second, several optimization techniques (such as resolution adjustment, cropping and MTCNN for efficient facial detection) are incorporated into FaceReminder to improve its latency and the energy efficiency;
- Third, through empirical study with extensive experiments on the major steps of FaceReminder, we discovered the key points to obtain its most energy efficient design;
- Finally, we have implemented FaceReminder with several design approaches and evaluated their performance experimentally. The results show that, while maintaining satisfactory facial recognition accuracy, the Paired Glass-Device with MTCNN approach can achieve significant reductions for both latency and energy consumption.

The remainder of this paper is organized as follows. Section II presents the preliminaries and discusses the overview of FaceReminder. The empirical study of facial recognition accuracy, latency and energy consumption of major steps of FaceReminder are discussed in Section III. Section IV presents the two basic designs of FaceReminder and their experimental results. The energy-oriented advanced designs with cropping and MTCNN are discussed in Section V. Section VI concludes this paper and points out our future work.

II. FaceReminder: Overview and Preliminaries

To help people with prosopagnosia (i.e., face blindness) or those with short memory of face-name connection, two features are needed for devices running FaceReminder: a camera to take a photo of the 'forgotten' person (usually in front of the user), and a component (normally video or audio) to deliver the facial recognition result to the user. An ideal device for FaceReminder should have these two features be as less intrusive as possible with a nature and seamless interface to minimize the disturbance of normal inter-person interaction and conversation.

In this section, we first present the recently released VUZIX Blade$^{\circledR}$ smart glass [3] and discuss its unique features that naturally fit the need of FaceReminder. Then, the preliminaries of facial detection/recognition are reviewed. Finally, the overview and major steps of FaceReminder are discussed.

A. VUZIX Blade Smart Glass

TABLE I: Specifications of the glass vs. tablet

	CPUs / GPU	Memory	Battery
VUZIX Blade Smart Glass	4XCortex-A53, 1.2GHz / Vivante GC7000UL	1GB	470mAh
Galaxy TAB E 9.6 (T560NU)	4XCortex-A53, 1.2GHz / Adreno 306	1.5GB	5000mAh

As illustrated in Figure 1, the VUZIX Blade$^{\circledR}$ smart glass has a unique see-through display with 480X480 resolution and 24 bit color [3]. Such transparent display makes it less intrusive compared to other projection based display devices for smart glasses, such as the optical head-mounted display for Google glass [20]. It is ideal for FaceReminder to deliver the facial recognition result. In addition, the front-faced 8 megapixel camera embedded in the glass satisfies another requirement and makes the smart glass a nature fit for FaceReminder. Moreover, Table I shows other specifications of the

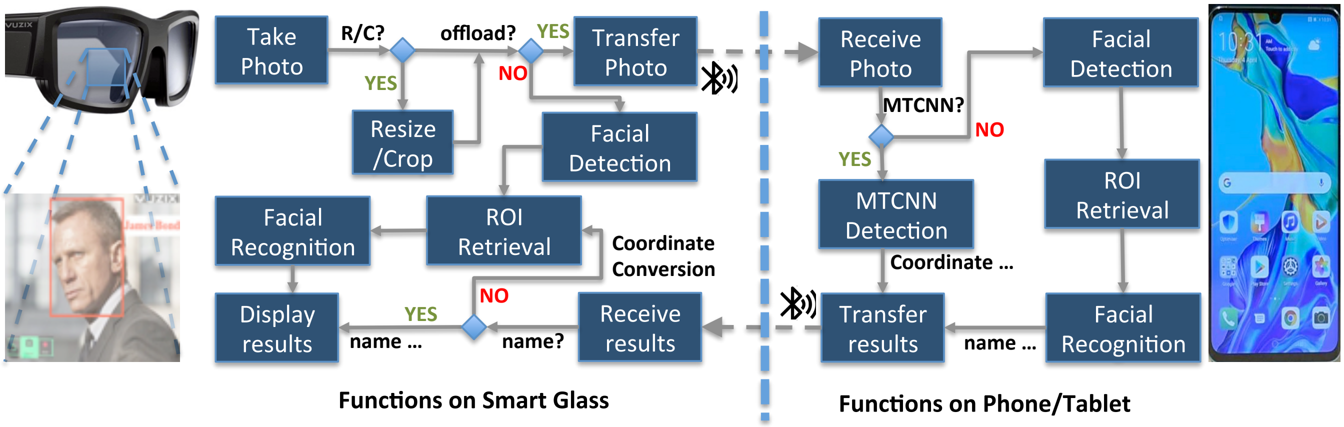

Fig. 1: Overview of FaceReminder and its major steps in the Glass-Only and Paired Glass-Device designs

smart glass, including on-board computing components (such as CPUs/GPU and memory) and battery capacity.

The on-board computer of the smart glass runs Android and can perform the facial detection and recognition for FaceReminder in-place, which enables its Glass-Only design option as discussed further in Section IV-A. However, the battery capacity of the glass is very limited. The 470mAh (with 3.7V) specification gives the total energy of 6260.4 joules, which is only a fraction (from less than 1/10 to about 1/6) of the normal battery capacity of a smart phone or tablet. The computation intensive facial detection and recognition task can quickly deplete its battery with much reduced operation time of the glass.

For its connectivity, the glass is equipped with both Bluetooth (4.1BR/EDR/BLE) and Wi-Fi (2.4Ghz 802.11 b/g/n). With the Bluetooth connection, we can easily pair the glass with a mobile phone or tablet, which generally has a much larger battery capacity (from 3400mAh to 5000 mAh). For comparison, the specifications of the tablet (Samsung Galaxy TAB E 9.6, T560NU, released in 2015) that we used in the experiments are also shown in Table I. As shown in Section IV-B, with the workload offloading technique, the photos can be transmitted to the tablet to perform the facial detection and recognition task to reduce the energy consumption of the glass and extend its battery lifetime.

B. Facial Detection and Recognition

FaceReminder relies on accurate facial detection and recognition, where the related topics have been studied for decades with many well-known techniques being designed and developed [7], [13]. Note that, the main focus of this work is to explore design options of FaceReminder to improve the energy efficiency of FaceReminder and extend the operation time of the smart glass rather than design and develop new facial detection and recognition algorithms. Therefore, we adopted the well-known classical facial detection and recognition algorithms that have been implemented in the OpenCV library for Android platforms [21], [22].

Facial Detection: First, for facial detection, we considered the Haar-Cascade Classifier (HCC) algorithm [9], which has been implemented in OpenCV. It is an object detection algorithm with several pre-trained models, including the one to detect faces in an image. For a given image, the facial detection algorithm will find a set of bounding boxes in the image within which faces have been detected. In this work, focusing on the scenario of one-to-one conversation with the objective to identify a single person in an image, we assume the largest Region of Interest (ROI) contain the target face with its bounding box being returned.

From our experiments, we found that facial detection with the Haar-Cascade Classifier based algorithm is quite computational intensive and can take almost half of the processing time. To improve the facial detection efficiency, we further incorporated a recent studied detection algorithm with the Multi-Tasking Cascading Convolutional Neural Network (MTCNN) [14]. In particular, we utilized an Android implementation of MTCNN [19], where the software package uses Tensorflow [23] and a pre-trained MTCNN model (based on two datasets: WIDER FACE [24] and CelebA [25]) for facial detection. As shown later in the evaluation results, MTCNN can result in much more accurate ROI for photos with even the lowest resolution, and thus can improve the efficiency of facial detection.

Facial Recognition: Once the ROI of an image that contains face is detected, the next step is to identify the person through facial recognition. In the OpenCV library, several classical facial recognition algorithms have been implemented, including EigenFaces [12], FisherFace [10] and Local Binary Pattern [11]. In this work, we utilized the EigenFaces algorithm as it shows relatively high recognition accuracy. In the OpenCV implementation, once the ROI is found, the image is cropped, resized and converted to a 100x100 gray-scale, which is then feed to EigenFaces recognition algorithm to identify the person (if known). To work with EigenFaces algorithm, several photos of each known person are needed to pre-train a recognition model offline. In our experiments, for each known person, we used five photos to train the model (see Section III for details).

C. Overview of FaceReminder

Figure 1 illustrates the overview of FaceReminder and its major steps in both Glass-Only and Paired Glass-Device designs. As shown in the figure, the first step is to take a photo of the person who is assumed to be in front of the user as in the normal one-to-one conversation scenario. Then, if needed, the photo will be pre-processed to reduce its size by adjusting the resolution and/or cropping off its margins. For the Glass-Only design where all computation is expected to be processed on the smart glass, the (pre-processed) photo will be fed to the facial detection algorithm to find the ROI and its bounding box. In the end, the facial recognition algorithm takes the converted ROI portion of the image and identify the face with a pre-trained model of known persons, where the name of the person (if found) will be shown on the see-through display of the smart glass.

For the case of Paired Glass-Device design, where the glass is paired with another mobile device (smart phone or tablet) via Bluetooth connection, the (pre-processed) photo will be transmitted to the device. After the device receives the photo, it will be processed with similar steps of facial detection and recognition on the mobile device. The resulting names (for recognition if found) or the ROI's coordinates (for MTCNN detection) will be send back to the smart glass (see Section V-B for more details).

III. Empirical Study: Latency, Energy and Accuracy

In order to devise the most energy efficient design for FaceReminder, we conducted an empirical study with extensive experiments on the latency and energy consumption for its major steps. Such an empirical study will provide us insightful ideas on what are the best optimization options to be deployed for energy efficiency and latency reduction. In what follows, to ensure the resulting recognition accuracy of FaceReminder, we first present the accuracy results of the adopted facial detection and recognition algorithms with various optimization options being considered.

A. Accuracy of Facial Detection and Recognition

In our experiments, we collected 760 photos of more than 50 persons (friends and co-workers) using the embedded camera of the smart glass, where at least 10 photos have been taken for each person. These photos will represent similar conditions when FaceReminder is actually deployed, and avoid possible discrepancy for results when other known photo data sets are utilized. For facial recognition with EigenFaces [12], five photos for each person were used to built the known-person database and model offline. The remaining photos are used for evaluating the accuracy of facial detection and recognition.

The native resolution of the photo taken with the smart glass' camera is 3264X2448, which results in the size of a photo in JPEG to be around 5MB. To reduce the size of the photo for efficient facial detection and recognition, we considered three different reduced resolutions: 2048X1536, 1600X1200 and 640X480. Table II shows the accuracy of the adopted Haar-Cascade Classifier facial detection and EigenFaces recognition for photos with different resolutions.

From the table, we can see that the detection accuracy for the photos with native resolution is 89% (that is, on average, 89 out of 100 photos where a valid ROI can be found). The detection accuracy remains almost the same for photos with two reduced resolutions, which is 90% and 88% for 2048X1536 and 1600X1200, respectively. However, for photos with the lowest resolution 640X480, the facial detection with the Haar-Cascade Classifier algorithm has only 61% accuracy, which makes it unsuitable for FaceReminder.

Once a valid ROI is found in a photo, the facial recognition with Eigenfaces algorithm has accuracy of more than 90% for all resolutions. That is, for more than 9 out of 10 photos with a valid ROI, it can correctly identify the person's name. Therefore, the combined detection/recognition accuracy for photos with the tree higher resolutions is no lower than 80%, which is considered to be satisfactory. However, for the photos with the lowest resolution 640X480, the combined detection/recognition accuracy can be as low as 55%, which is not satisfactory. Therefore, in the remaining experiments, we will focus on the three higher resolutions (except for MTCNN as explained below).

Resizing with Cropping: The facial detection with the Haar-Cascade Classifier algorithm is very computation intensive and can take almost half of the total processing time. Given that the detection process heavily relies on the size of the photo, we can crop the margins on all sides with the assumption that the ROI of the face most likely appears in the center of a photo. The more margin being cropped off, the detection can be more efficient, which however may cause reduced detection accuracy. As shown in Table II, by cropping the margins of 10% on each side (where the resulting photo size is only 64% of the original photo with 36% reduction), the detection accuracy is affected by only 2%. However, cropping 15% or more can lead to significantly reduced detection accuracy.

MTCNN for Detection: To further improve the detection efficiency as well as the accuracy for photos with low resolutions, we incorporate facial detection with the MTCNN implementation [14], [19], where the evaluation results are also slown in Table II. It is interesting to see that, the impact of the photo's resolutions on the accuracy of facial detection with MTCNN has much less differentiation. Even for photos with the lowest resolution, the detection accuracy can be as high as 88%. However, the adopted MTCNN is implemented on Tensorflow [23] that utilizes GPU extensively. In this work, we consider MTCNN in the Paired Glass-Device design and run it on the tablet only.

B. Major Steps of FaceReminder: Energy and Latency

Next, we evaluate the energy consumption and latency for the major tasks of FaceReminder. First, for energy, we focus on improving the energy efficiency on the smart glass only,

TABLE II: Accuracy for various face detection and recognition techniques

	Haar-Cascade Classifier Detection (%) with cropping					Recognition (%)	Overall (%)	MTCNN based Detection	
Resolutions	0%	5%/side	10%/side	15%/side	20%/side	Eigenfaces	(cropping: 0%)	Detection (%)	Overall (%)
3264X2448	89	88	87	83	75	92	**82**	95	**87**
2048X1536	90	90	88	85	79	94	**85**	87	**82**
1600X1200	88	89	86	83	77	91	**80**	87	**79**
640X480	61	61	61	61	52	90	**55**	88	**79**

TABLE III: Average latency and energy per photo for the two designs of FaceReminder: Glass-Only vs. Paired Glass-Device

	Glass-Only (t in seconds, E in Joules)								Paired Glass-Device (t in seconds, E in Joules)						
Resolutions	t_{photo}	t_{DR}	t_{tot}	$\delta_t(\%)$	E^G_{photo}	E^G_{DR}	E^G_{tot}	$\delta_E(\%)$	t_{bt}	t^{tab}_{DR}	t^{pair}_{tot}	$\delta_t(\%)$	E^G_{bt}	E^G_{tot}	δ_E (%)
3264X2448	0.88	8.71	9.59	-	1.59	16.13	17.72	-	6.18	7.16	14.22	**48.3**	13.88	15.47	**-12.7**
2048X1536	1.50	3.35	4.85	**-49.4**	2.83	8.40	11.23	**-36.6**	2.12	2.64	6.26	**-34.7**	5.03	7.85	**-55.7**
1600X1200	1.70	1.82	3.52	**-63.3**	2.84	4.82	7.66	**-56.8**	1.33	1.60	4.63	**-51.7**	3.05	5.89	**-66.8**

where the *energy consumption* of the tasks on taking photo (E^G_{photo}) and processing photo for detection and recognition on the glass (E^G_{DR}) are experimentally evaluated. To incorporate the workload offloading technique, the energy consumption of the glass (E^G_{bt}) for transmitting photos via Bluetooth to a mobile device has been evaluated as well.

Here, for offloading experiments, we utilized a Samsung Galaxy tablet, where its specifications on CPUs/GPU, memory and battery are also shown in Table I. Clearly, the battery capacity of the smart glass is only about one-tenth (1/10) of that of the tablet. Even for smart phones that normally have their battery capacity in the range of 3400mAh to 4200mAh, the battery capacity of the smart glass is still comparatively much smaller and thus will be our focus to manage.

Similarly, the *latency* of the tasks on taking photo (t_{photo}), processing photo for detection and recognition on the glass (t_{DR}), and transmitting photo via Bluetooth to a mobile device (t_{bt}) has been experimentally measured. In addition, the latency of processing facial detection and recognition task on the mobile device (t^{tab}_{DR}) is also examined.

Given the limited interface to interact with the smart glass, it is difficult to directly measure its power/energy consumption with a power meter. Instead, we exploit the system utility that reports the remaining battery capacity, from which we can derive how much energy has been consumed by certain operations. Specifically, for each major task as mentioned above, the smart glass is programmed to repeatedly execute the task until 50% of the battery capacity is utilized, which corresponds to the total energy consumption being 6260.4/2 = 3130.2 joules. Then, based on the number of photos being processed with the total consumed energy, the average energy consumption for the task to process a single photo can be calculated.

For these experiments, the times (i.e., latency) of the considered task to process each photo are also recorded using a system timer (that has the resolution of milli-seconds). At the end, from the collected execution times of the task to process all the photos, the average latency (i.e., processing time) of the task can be derived accordingly. In the experiments (except for the task of taking photos), the photos are pre-loaded to the smart glass with proper resolution adjustments (if needed). Then, the testing photos are retrieved one by one repetitively to evaluate a given task until the target battery capacity is reached.

The evaluation results are shown in Table III. Here, in addition to the native resolution (3264X2448), the other two reduced resolutions (2048X1536 and 1600X1200) aiming at reducing the size of photos have been considered. Recall that, for the lowest resolution (640X480), the resulting facial detection/recognition accuracy is too low (see Table II) and it is not considered in the experiments.

First, for the task of taking photos with the smart glass' embedded camera, the average latency (see column 2) is 0.88 second to get a JPEG photo with the native resolution (3264X2448). However, to obtain a photo with lower resolutions (such as 2048X1536 and 1600X1200), additional time is needed for the compression where the average latency is increased to 1.5 and 1.7 seconds, respectively. Correspondingly, the average energy consumption for taking a photo is shown in the fifth column, where the numbers are 1.59J, 2.83J and 2.84J for the considered three resolutions, respectively.

Second, for the task of facial detection and recognition on the smart glass, its average latency is shown in the third column. Here, for photos with the native resolution (3264X2448), it takes around 8.71 seconds and detailed analysis shows that majority of the time is used for facial detection to find the ROI of a target face with the Haar-Cascade Classifier (HCC) algorithm [9]. When the resolution of a photo is reduced, the search space for facial detection becomes much smaller. The average latency can be significantly reduced to 3.35 and 1.82 seconds for the resolutions of 2048X1536 and 1600X1200, respectively. With reduced processing time, the energy consumption of facial detection/recognition on the smart glass (see the sixth column) can also be reduced remarkably for photos with lower resolutions. Here, the numbers for the three resolutions are 16.13J, 8.4J and 4.82J, respectively.

Next, to support workload offloading, the average latency of transmitting a photo via Bluetooth to the tablet is shown in the 10^{th} column. Not surprisingly, the transmission time of a photo directly depends on its resolution, where lower resolutions mean smaller photo sizes, which lead to reduced latency. For the native resolution (3264X2448), it takes 6.18 second on average to transmit one photo, where the latency can be significantly reduced to 2.12 and 1.33 seconds for the resolutions of 2048X1536 and 1600X1200, respectively. Correspondingly, the average energy consumed to transmit a photo for the three resolutions are 13.88J, 5.03J and 3.05J, respectively, (as shown in the column 14).

Finally, when the facial detection/recognition is offloaded to the tablet, its latency is shown in the column 11. For each resolution setting, the processing time is slightly improved compared to having the smart glass to perform the same task.

IV. Glass-Only vs. Paired Glass-Device

From the empirical study in the last section, we can see that the resolution of photos will have a great impact on the performance of FaceReminder in terms of its latency and energy efficiency on the smart glass. In addition, although offloading the task of facial detection and recognition to a mobile device can increase the latency due to the additional photo transmission time via Bluetooth, it can improve the energy efficiency on the smart glass. In this section, we compare two basic designs of FaceReminder: *Glass-Only* vs. *Paired Glass-Device*, where the second design incorporates the workload offloading technique.

A. Glass-Only Design: Resolution Adjustment

Glass-Only with Native Resolution (Baseline): First, we consider the Glass-Only design that performs all necessary processing on the glass. In particular, for the case of utilizing photos with the camera's native resolution (3264X2448), it is considered as the *baseline* design of FaceReminder.

The baseline's total latency and energy consumption to processing one photo on the smart glass are also shown in Table III (see columns 4 and 8, respectively). Here, the baseline design takes on average 9.59 seconds in total to process a single photo, which is unbearably slow. Moreover, it takes on average 17.72J to process one photo. Assuming that FaceReminder runs consecutively to process photos for scenarios where the user meets different persons in a meeting or a party gathering. With the baseline design, the glass can only process 353 photos in total with its full battery capacity (6260.4J). This translates to an operation time of only 56 minutes, which is significantly less than the specified battery life of 8 hours [3].

Optimization with Resolution Adjustment: From the results of the empirical study in the last section, the first optimization technique for improving both energy efficiency and latency of FaceReminder is to reduce the resolution of the photos to be processed. Here, by reducing the photo resolution to 2048X1536, the total processing time (i.e., latency) of FaceReminder can decrease to 4.85 seconds with 49.4% improvement over the baseline (columns 4 and 5 of Table III).

For energy (see columns 8 and 9), it is reduced to 11.23J for processing one photo on average with an improvement of 36.6%. With the even lower resolution of 1600X1200, the average latency and energy consumption to process a single photo can be further reduced to 3.52 seconds and 7.66J, which are improved by 63.3% and 56.8%, respectively, over those of the baseline design. With reduced energy consumption, more photos can be processed with longer operation time. For instance, for the resolution of 1600X1200, the smart glass can process 817 photos in total with its full battery capacity. Assuming that a photo is taken every 10 seconds (similar to the baseline's latency), the operation time can increase to 136 minutes, a 2.4X improvement over the baseline.

B. Paired Glass-Device Design: Workload Offloading

It has been well-known that the task of facial detection and recognition is rather computationally intensive and energy hungry [7], [13], which has also been confirmed in our empirical study (see Section III). Given the fact that the battery capacity of the smart glass is very limited [3], to further improve the energy efficiency of FaceReminder and increase the operating time of the smart glass, the second design *Paired Glass-Device* exploits the workload offloading technique, which follows a similar idea in traditional offloading studies [17], [18]. As illustrated in Figure 1, the basic idea of Paired Glass-Device is to offload the facial detection and recognition task to another mobile device (a smart phone or tablet) by exploiting its higher computational power and much larger battery capacity, which is similar to iFalcon [5].

Clearly, there is a trade-off regarding to energy consumption of processing a photo locally on the smart glass compared to that of transmitting the photo via Bluetooth to the mobile device. Fortunately, from our empirical study as discussed in the last section, it has been shown that transmitting a photo is relatively energy efficient compared to processing it locally on the smart glass.

As shown in the last two columns in Table III, for the Paired Glass-Device design utilizing photos of the native resolution (3264X2448), the total energy consumption on the smart glass to process one photo is 15.47J on average, which leads to 12.7% improvement compared to that of the baseline design. However, as the tablet has similar computation power and takes roughly the same time on the facial detection (with the Haar-Cascade Classifier algorithm [9]) and recognition (with the Eigenfaces algorithm [12]) task as that of the smart glass, the average latency for the Paired Glass-Device design increases to 14.22 seconds due to the additional photo transmission time via Bluetooth, which leads to a slowdown of 48.3% (columns 12 and 13).

When lower resolutions of the photos are considered for optimizations, the latency of FaceReminder can be much reduced due to reductions on both Bluetooth transmission time and the processing time of facial detection and recognition on the tablet. The average latency can be found as 6.26 and 4.63 seconds for the two reduced resolutions 2048X1536 and 1600X1200, respectively, which are still higher than those of the Glass-Only design.

However, the energy efficiency of FaceReminder with the Paired Glass-Device design can be improved with the average

TABLE IV: Advanced designs of FaceReminder: Paired-Cropping vs. Paired-MTCNN (both start with 3264X2448)

	t_{photo}	t_{resize}	t_{bt}	t_D^{tab}	t_{tot}	t_δ(%)	E^G_{photo}	E^G_{other}	E^G_{tot}	E_δ (%)	Overall accuracy (%)
Paired w. Cropping (10%)	0.88	0.39	2.43	4.68	8.38	**-12.6**	1.59	8.06	9.65	**-45.5**	80
Paired w. MTCNN	0.88	0.42	0.80	0.69	2.59	**-73.0**	1.59	2.14	3.73	**-78.9**	83

energy consumption per photo being reduced to 7.85J and 5.89J for the two reduced resolutions, respectively. These are 19.1% and 10% more energy reductions compared to those of the Glass-Only design. For the case of utilizing photos with the resolution of 1600X1200, the smart glass can process 1062 photos in total with its full battery capacity. Again, assuming that a photo is taken every 10 seconds, it leads to an operation time of 177 minutes, which is a 3.16X improvement over that of the baseline design.

V. Optimizations: Cropping and MTCNN

Note that, the task of taking photos has to be performed on the smart glass with its embedded camera, and its latency and energy consumption are rather fixed to get JPEG photos of different resolutions. Therefore, for the Paired Glass-Device design, to further improve its performance in both latency reduction and energy efficiency, we can either reduce the transmission time with even smaller photos or adopt more efficient facial detection and recognition algorithms on the mobile device as in iFalcon [5]. In this section, we first consider the photo cropping optimization that can lead to smaller photos to be transmitted. Then, the more efficient facial detection with MTCNN [14] is incorporated that exploits the GPU on the tablet.

A. Optimization with Cropping

Recall that, the empirical study in Section III shows that the facial detection is still relatively accurate with 10% cropping on each side of a photo. Here, we consider only the case of photos with native resolution (3264X2448). With 10% cropping on each side, the resulting photo will have the size of 64% (i.e., 0.8X0.8=0.64) as that of the original photo, which leads of an effective reduction of 36% in photo size.

As shown in Table IV, with the resulting smaller photos from the cropping, the average transmission time is reduced to 2.43 seconds (compared to 6.18 seconds for transmitting the original photo). Moreover, the time for facial detection and recognition is reduced to 4.68 seconds (compared to 7.16 seconds for processing the original photo).

With the cropping time of 0.39 second on average, the total latency of the Paired Glass-Device design becomes to be 8.38 seconds on average, a reduction of 12.6%. Moreover, the average energy consumption per photo is only 9.65J, an improvement of 45.5% compared to that of the baseline. Finally, the overall accuracy of facial detection and recognition with cropping optimization is 80% and satisfactory.

Although cropping can reduce the size of photos, such reduction seems to be not as effective as resolution adjustment. From the results in Table III, the average transmission times for photos with the two reduced resolutions are 2.12 seconds and 1.33 seconds, respectively, which indicate smaller photos can be obtained with the reduced resolutions. The cropping optimization could also be applied to photos with reduced resolutions and better latency/energy results could be expected for the Paired Glass-Device design. However, such combined optimization has to be carefully evaluated to ensure the resulting facial recognition accuracy be properly maintained.

B. Optimization with MTCNN on Device

From the last section, we can see that, even with the reduced resolution of 1600X1200, the facial detection and recognition on the tablet can still take around 1.6 seconds. Here, detailed analysis shows that majority of the time is used by the facial detection with the Haar-Cascade Classifier algorithm [9] and the Eigenfaces recognition algorithm [12] takes only a couple of milli-seconds. To get more efficient facial detection, we have adopted a more recently developed Multi-Tasking Cascading Convolutional Neural Network (MTCNN) [14] with its Android implementation [19] on top of Tensorflow [23], which can effectively exploits the GPU on the tablet.

Moreover, from Table II, it can be seen that the MTCNN based facial detection has excellent detection accuracy, even for photos with the lowest resolution of 640X480. This provides a great opportunity to exploit photos with much reduced sizes for faster transmission via Bluetooth. However, for photos with 640X480 resolution, the overall accuracy with performing both the MTCNN based detection and Eigenfaces recognition on the tablet will be lower than 80%.

Therefore, to tackle this problem and achieve a satisfactory overall accuracy of 80%, we separate the Eigenfaces facial recognition with photos of the native resolution (3264X2448) on the smart glass from the MTCNN based facial detection with 640X480 resolution photos on the tablet. Specifically, the mixed optimizations for the Paired Glass-Device design have the following steps (see Figure 1).

First, once a photo with the native 3264X2448 resolution is taken, a copy with the lowest resolution 640X480 is generated while keeping the original photo unchanged. Then, the lowest resolution photo copy with much reduced size is transmitted via Bluetooth to the tablet. On the tablet, the MTCNN based facial detection will take the photo with 640X480 resolution and find the ROI that contains a face. Instead of utilizing the content of the ROI for facial recognition on the tablet, the coordinates of the ROI are transmitted back to the smart glass via Bluetooth.

After that, the ROI coordinates will be converted to find the corresponding ROI in the original photo of the native resolution 3264X2448. Then, the ROI will be cropped, resized to 100x100, and converted to gray-scale for being use in the Eigenfaces facial recognition step [12], [21].

The evaluation results for the mixed optimizations with MTCNN for the Paired Glass-Device design are also shown in Table IV. Here, the overall average latency can be reduced to 2.59 seconds, which is an improvement of 73% over that of the baseline design. The breakdown of the latency includes the transmission time of 0.8 second for a photo with 640X480 resolution and 0.69 second of the MTCNN facial detection on the reduced resolution photo. With such a mixed processing of MTCNN based facial detection on tablet and Eigenfaces recognition on the smart glass, the resulting overall recognition accuracy from the experiments is shown to be 83%.

In terms of energy, the average energy consumption of the smart glass for processing one photo is only 3.73J due to the reduced transmission energy of photos with much reduced size. Compared to the baseline, the energy reduction reaches 78.9%. Similarly, the smart glass can process 1678 photos with its full battery capacity. This translates to an operation time of 280 minutes (again, assuming a photo is taken every 10 seconds), which is an improvement of 5X compared to that of the baseline design.

VI. Conclusions

By exploiting its unique see-through display and on-board camera of a recently released VUZIX Blade® smart glass, we designed and developed an augmented-reality application, *FaceReminder*. It aims at helping people with prosopagnosia (i.e., face blindness) or short-memory of face-name connection with the supporting of well-known facial detection and recognition techniques. To tackle the rather limited battery of the smart glass and find the most energy efficient design, we exploited several optimization strategies, including resolution adjustment and cropping for reduced photo sizes, workload offloading to another mobile device via Bluetooth, and efficient facial detection with MTCNN with GPU processing. For the two base designs: Glass-Only and Paired Glass-Device, we have implemented FaceReminder on the smart glass and a tablet with much larger battery capacity. From our empirical study on the latency and energy consumption of the major steps in FaceReminder, we identified the energy efficient optimization techniques, and have evaluated them through extensive experiments. The results show that, both resolution adjustment and cropping can effectively reduce the photo sizes and improve efficiency of FaceReminder. The most efficient *Paired Glass-Device* design with mixed optimization of reduced resolution and MTCNN facial detection technique can reduce the average latency by 73% and energy consumption by 78.9% (i.e., 5X battery lifetime improvement) while maintaining a satisfactory 80% recognition accuracy.

In our future work, we will explore more efficient facial detection and recognition algorithms, especially those based on deep neural networks (DNN). Moreover, given that taking photo can take about 50% overall time for the most efficient design, we will investigate schemes to further improve photo processing. For example, using raw bitmaps instead of JPEG to avoid de/compression overheads.

References

[1] V. Genaro Motti, S. Kohn, and K. Caine, "Wearable computing: A human-centered view of key concepts, application domains, and quality factors," in *Proc. of the 16th Int'l Conf. on Human-computer Interaction with Mobile Devices and Services*, 2014, pp. 563–564.

[2] D. Ravi, C. Wong, B. Lo, and G. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE J. of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 56–64, 2017.

[3] "Vuzix blade smart glasses," 2019. [Online]. Available: https://www.vuzix.com/products/blade-smart-glasses

[4] "Best mobile face recognition apps for 2019," 2019. [Online]. Available: https://www.facefirst.com/blog/19-incredible-face-recognition-apps/

[5] "ifalcon face control mobile," 2019. [Online]. Available: https://nntc.digital/products/ifalcon-face-recognition-solutions/

[6] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, Dec. 2003.

[7] R. Jafri and H. Arabnia, "A survey of face recognition techniques," *Journal of Information Processing Systems*, vol. 5, pp. 41–68, 06 2009.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CoRR*, 2015.

[9] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proc. Int'l Conf. on Image Processing*, vol. 1, 2002.

[10] H. Sahoolizadeh and Y. Aliyari Ghassabeh, "Face recognition using eigen-faces, fisher-faces and neural networks," in *Proc. of the 7th IEEE Int'l Conf. on Cybernetic Intelligent Systems*, 2008.

[11] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Prof. of the European Conference on Computer Vision (ECCV)*, 2004, pp. 469–481.

[12] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[13] M. Wang and W. Deng, "Deep face recognition: A survey," *CoRR*, 2018.

[14] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," *CoRR*, vol. abs/1604.02878, 2016. [Online]. Available: http://arxiv.org/abs/1604.02878

[15] C. Xian, Y.-H. Lu, and Z. Li, "Adaptive computation offloading for energy conservation on battery-powered systems," in *Proc. of the 13th Int'l Conf. on Parallel and Distributed Systems*, 2007.

[16] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 319–333, Feb 2019.

[17] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, Aug 2017.

[18] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, Dec 2016.

[19] Jianfeng Chen, "Mtcnn4android," 2019. [Online]. Available: https://github.com/vcvycy/MTCNN4Android

[20] "Google glass," 2019. [Online]. Available: https://www.google.com/glass/start/

[21] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

[22] Quickbird Studios, "Opencv-android," 2019. [Online]. Available: https://github.com/quickbirdstudios/opencv-android

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/

[24] "Wider face: A face detection benchmark," 2019. [Online]. Available: http://shuoyang1213.me/WIDERFACE/

[25] "Large-scale celebfaces attributes (celeba) dataset," 2019. [Online]. Available: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html