Watch & Do: A Smart IoT Interaction System With Object Detection and Gaze Estimation

Jung-Hwa Kim, Seung-June Choi, and Jin-Woo Jeong

***Abstract*—The Internet of Things (IoT) attempts to help people access Internet-connected devices, applications, and services anytime and anywhere. However, how providing an efficient and intuitive method of interaction between people and IoT devices is still an open challenge. In this paper, we propose a novel interaction system called *Watch & Do*, where users can control an IoT device by gazing at it and doing simple gestures. The proposed system mainly consists of: 1) object detection module; 2) gaze estimation module; 3) hand gesture recognition module; and 4) IoT controller module. The target device is identified by various deep learning-based gaze estimation and object detection techniques. Afterwards, hand gesture recognition is applied to generate an IoT device control command which is transmitted to the IoT platform. The experimental results and case studies demonstrate the feasibility of the proposed system and imply the future research directions.**

***Index Terms*—Deep learning, gaze estimation, Internet of Things, object detection, smart interaction.**

I. Introduction

THE RECENT developments in network infrastructure and smart devices have resulted in the rapid spread of the Internet of Things (IoT) applications and services. The IoT is defined as a network of Internet-connected things (e.g., computers, vehicles, and sensors) that exchange data and information among themselves with and other services [1]. The number of Internet-connected devices is now dramatically growing. According to a recent study on the prediction of IoT market share, the number of IoT devices will approach 100 billion and the total amount of data generated by the users and devices will reach 35 ZB by 2020 [2]. It is therefore expected that the success of IoT will allow the users and things to be connected anytime, anywhere using any path, network, or service [3]. These characteristics of the IoT ecosystem will improve the quality of services in various application domains such as health-care (e.g., remote patient monitoring and treatment), transportation (e.g., smart transportation systems), and home automation (e.g., smart appliances).

A typical architecture of the IoT comprises a perception layer, a network layer, and an application layer [4]. Numerous studies have been conducted to address the challenging issues in each layer of the architecture. The perception layer first interacts with physical devices like RFID [5], sensors, and actuators and then connects the devices to the network of IoT. The network layer of the IoT is responsible for transmitting data between different things, applications, and services using heterogeneous networks and communication protocols. Finally, the application layer exploits the data from the underlying layers to build and provide the required services.

Interoperability of IoT devices, applications, services, and users is an important factor for the successful implementation of IoT. Therefore, the network layer and its corresponding technologies (e.g., Bluetooth [6], Wi-Fi [7], 6LoWPAN [8], ZigBee [9], Z-Wave [10], MQTT [11], and CoAP [12], etc.) have been usually considered the most important components since they are closely related to the connectivity, interoperability, and scalability of IoT architectures. Similarly, the IoT standards [13], [14], frameworks [15]–[17], and platforms in the application layer aim to maximize the interoperability by abstracting the layers of the IoT architecture and providing an efficient user interface (UI).

However, the current technologies mainly focus on improving the machine-to-machine communication/interaction, rather than the interaction between users and machines. For example, some IoT platforms designed for smart home automation provide a Web-based UI and a mobile application to register, manage, and control the smart home appliances connected to them. Users must first open the website or the mobile application, explore a page to select a menu, find a room or location, and finally select the device to be manipulated from a list. After selecting the device, the users can check the status or control it by touching or clicking the buttons on the webpage or mobile page. However, this UI and procedure will become tedious and time-consuming to the users with the current rapid increase in the number of IoT ready devices. Additionally, users who are not familiar with smart devices, such as children and seniors, or those with limitations in accessing them, such as severely ill patients or the disabled, will encounter difficulties in using the IoT applications and services. This inconvenience will be a major obstacle to the dissemination of the IoT.

In this paper, we focus on the interaction between people and IoT devices, which can affect the usability of the IoT platforms. There have been various studies to enhance the interaction between the users and devices by applying the recent machine learning-based technologies, such as voice recognition, gesture recognition, gaze estimation, etc. These have focused on improving the accuracy of recognizing complex sentences or gestures; however, people with disabilities

Manuscript received September 15, 2018; revised December 15, 2018; accepted January 31, 2019. Date of publication February 5, 2019; date of current version April 23, 2019. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education under Grant 2016R1D1A1B03931672. *(Corresponding author: Jin-Woo Jeong.)*

The authors are with the Department of Computer Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea (e-mail: junghwa96@naver.com; easkr@kumoh.ac.kr; jinw.jeong@kumoh.ac.kr).

Digital Object Identifier 10.1109/TCE.2019.2897758

or severe diseases, the elderly, and children cannot easily perform such tasks. Furthermore, the previous techniques have not been integrated with IoT platform; hence their effects on the IoT ecosystems have not yet been demonstrated.

This paper presents a novel and smart IoT interaction system called *Watch & Do*, where the users can control the devices by gaze and simple hand gestures. The proposed system consists of 1) object detection module, 2) gaze estimation module, 3) hand gesture recognition module, and 4) IoT controller module. The object detection module captures a picture of the room to detect and recognize the installed IoT devices using deep learning approaches. The gaze estimation module records and detects the face of the user from the video stream. Afterwards, based on a deep gaze estimation model and various machine learning classifiers, it predicts the device being gazed at. The hand gesture recognition module is a simple hardware module equipped with proximity-based sensors to detect hand gestures. The user's command to control an IoT device is recognized by this module and then sent to the IoT controller module. Finally, the IoT controller module generates a complete IoT command based on the information received from the object detection module, the gaze estimation module, and the hand gesture recognition module. The target device is identified using the gaze position obtained from the gaze estimation module and a category of the device in that location, from the object detection module. The IoT commands are generated based on the category of the device and a recognized user gesture. For example, "Up" gesture with the device category "air conditioner" is translated into the command "Increase the temperature".

The rest of this paper is organized as follows: Section II briefly reviews various approaches to help users interact with smart devices. Section III provides the details of the proposed smart IoT interaction system with gaze estimation and object detection modules. Section IV discusses the experimental results and user studies; and Section V provides conclusions and future research directions.

II. Related Work

There have been various studies to enhance the interaction between the users and devices. In this section, we briefly review the previous approaches based on head pose estimation, gaze estimation and eye tracking, and gesture recognition. The recent efforts to support the IoT system interaction with user behavior detection techniques are also introduced.

Head pose estimation can be used to monitor the user's attention while interacting with devices. For example, a driver's head pose estimation can be used to detect drowsiness or to monitor the driver's attention. Alioua *et al.* [18] proposed a system to identify a driver's attention level with a set of descriptors representing head pose variations and SVM classifiers for training and testing. The study from [19] also presented a real-time driver's head pose estimation system to prevent traffic accidents. In this system, a deep neural network architecture has been adopted to estimate a driver's head pose even under poor conditions in a vehicle environment. On the other hand, Ruiz *et al.* [20] proposed a general purpose head pose estimation method based on a multi-loss convolutional neural network which computes the intrinsic Euler angles, such as yaw, pitch, and roll.

Eye tracking and gaze estimation are employed to estimate the gaze position by monitoring the user's pupil movement. These technologies are particularly useful for severely ill patients who cannot easily perform daily life voluntary tasks. The eye tracking and gaze estimation techniques can be classified into model-based and appearance-based approaches [21]. Model-based approaches use geometric local features (e.g., corneal reflections [22]–[26], center of pupils, iris edges [27]–[29], etc.) extracted from eye images. Ohno *et al.* [22] presented a method to compute the gaze point position from the pupil and Purkinje image features with an eyeball model. A gaze tracking system that can handle natural head movements was proposed in [23]. On the other hand, Liu *et al.* [24] proposed an infrared sensor based eye tracking system to provide the amyotrophic lateral sclerosis (ALS) patients with a high speed typing system for communication. Similarly, Kavale *et al.* [25] presented an eyeball tracking system which helps the patients make decisions using their eye movements. Choi *et al.* [26] proposed a single-point calibration-based remote eye tracking system for improved human-device interaction. The methods to compute geometric eye shape features (e.g., center of pupils and iris edges) for learning the gaze directions have been proposed in [27]–[30]. However, these shape-based approaches cannot work with low-resolution eye images and hence have limited applicability.

In contrast to the model-based approaches, appearance-based approaches extract visual features from input images to learn a direct mapping function of the image features and gaze points. These approaches do not require high resolution input images or expensive hardware setup and calibration; therefore, can be more suitable for a practical environment. Lu *et al.* proposed learning-based methods to adaptively exploit the training samples to increase the accuracy of gaze estimation [31], [32]. A neural-network based real-time approach with a standard webcam has also been studied in [33]. The work from [34] introduced a way to help people draw lines and type text on the screen with their eye movements using a deep multi-layer perceptron algorithm. As these methods basically rely on feature representations to learn mapping functions, identifying useful features from images is important. The recent advances of deep learning approaches like convolutional neural networks (CNNs) have increased the efficiency of image understanding and classification tasks. The CNN is reportedly successful because it can automatically learn the important features for image understanding [35]. Therefore, some research works have tried to apply CNNs for gaze estimation. Zhang *et al.* [36] used eye images and head pose information for training a multimodal CNN to detect gaze direction. In addition, spatial weights learning with a full face image has been studied to improve the performance of the CNN-based gaze estimation approach [37]. Krafka *et al.* [38] also presented a CNN-based method to estimate gaze directions with a mobile device.

Gesture recognition can improve the efficiency of interaction with devices by reducing the unnecessary steps in the conventional Web-based UI. Rashid and Han [39] proposed a gesture-based system to control smart home devices connected via ZigBee technology. The system recognizes six hand gestures and maps each gesture to a function of a specific device. Similarly, Erden and Çetin [40] presented a multimodal hand gesture detection and recognition system with PIR sensors and a camera. The work from [41] introduced a single-camera dedicated TV which can recognize user's hand gestures to perform TV-related functions such as changing the channel or volume. Morganti *et al.* [42] presented a smart watch based method to recognize the forearm and finger gestures to control a specific device.

Finally, several research works have developed a unified framework with various technologies to support efficient IoT system interaction. The input modalities used in these studies can be categorized into the followings: 1) photo, 2) gesture, and 3) gaze. de Freitas *et al.* [43] presented a smart phone-based device interaction approach. In this, a picture of the target device taken by the users is compared with the reference image available for each device type to compute matching score. The UI of the device with the highest score is then sent to the users. The work from [44] introduced a similar approach where a photo of a target device is sent to a cloud server to retrieve its application ID through a vision-based identification process. Gesture-based approaches generally capture user's hand or arm gestures to infer the target device and its operations. A motion matching technique to select and control an IoT device was introduced in [45]. In this work, each device provides a continuously moving interface in a predefined path and the users need to perform a synchronous body movement to choose the target device. Freeman *et al.* [46] proposed an air-gesture based approach with a smart phone, where the rhythmic gestures are used to select the target IoT device. Alce *et al.* [47] used a smart watch to choose and control an IoT device with user's arm gestures. Budde *et al.* [48] proposed a method to point an IoT device by user's arm gestures captured by a depth camera. The controlling interface is sent to the user's smart phone according to the type of the target device. On the other hand, [49]–[51] used eye gaze to control IoT devices. Khamis *et al.* [49] developed an eye tracker mounted on a rail system to automatically detect and align the tracker with the user's movement. They demonstrated that such a system configuration can improve the user interaction with a large public display. In [50], smart glasses were used for recognizing and selecting the IoT devices. The UI corresponding to the type of the selected device is sent to the user's smartwatch. Velloso *et al.* [51] proposed a gaze-only device interaction approach based on smooth pursuit mechanism [52], where the eye movement was captured and evaluated using smart glasses.

Even though many studies have been conducted to improve the interaction between the users and devices, the successful implementation of IoT environments and services still face some challenges. First, some of the existing works are not suitable for IoT services and applications. The head pose based approaches [18], [19] target small and controlled conditions (i.e., in-vehicle) and cannot be directly applied to IoT environments such as a house or room. The eye tracking and gaze estimation approaches [22]–[29] generally detect the gaze positions on a small screen in front of the users. However, in IoT environment, smart devices and appliances can be installed anywhere in the room or the building; hence these techniques must be improved to work far from. Second, some previous works require the users to wear additional devices, such as a smart watch [42], [45], [47], headband [24], or smart glasses [50], [51]. To achieve a seamless and natural interaction in IoT environment, the number of additional devices to be worn should be minimized. Third, only a specific device or a fixed set of devices can be controlled with the presented techniques [24], [34], [39]–[41]. As the number and category of IoT devices are growing, the scalability and interoperability of IoT interaction systems need to be further addressed. Finally, some of the recent IoT system interaction approaches [43]–[45], [47], [48], [50] provide a UI through smart devices only. However, this requirement is restrictive for those who are not familiar with or not able to use smart devices.

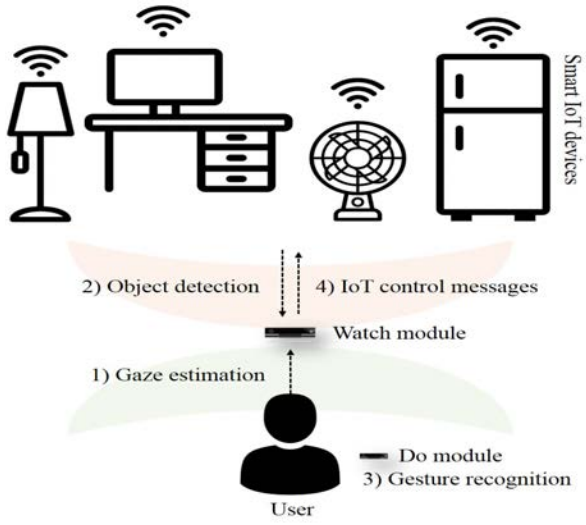


Fig. 1. Overview of the proposed system.

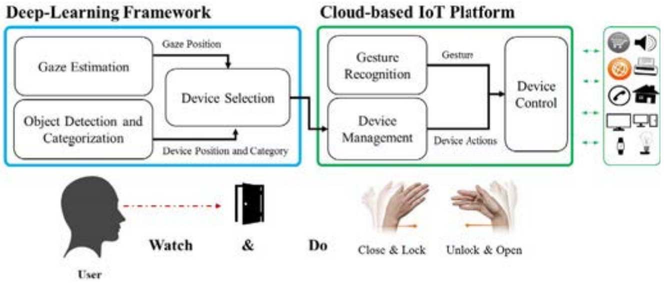


Fig. 2. Architecture of the proposed system.

In this work, we propose a novel interaction system for users and devices in IoT environment and services, which addresses the aforementioned issues.

III. The Proposed Method

The ultimate goal of the proposed system is to develop an efficient interaction method between users and IoT devices so that any user including patients, children, and the elderly can control the devices intuitively. The workflow and configuration of the proposed system are depicted in Fig. 1. The hardware module called "*Watch*" is located at the center of a room

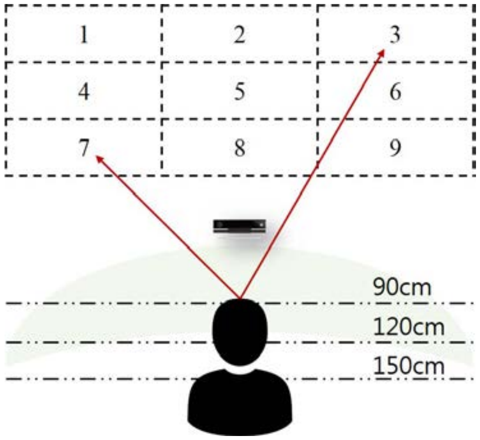


Fig. 3. Overview of indoor gaze estimation by head pose estimation.

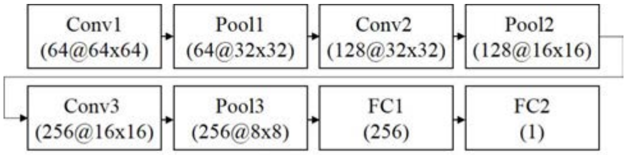


Fig. 4. CNN architecture used in the DeepGaze framework.

Fig. 5. Example images used for training the gaze estimator.

(e.g., a living room or a patient's room). This module estimates user's gaze position and detects the IoT devices installed in the room. Another module called "*Do*" is installed around the user (e.g., near the arms) to detect hand gestures. The proposed system works as follows. First, the *Watch* module records the opposite side of the user to detect and recognize the types of IoT devices installed in the room. Second, the *Watch* module detects the user's head region and then computes a fine-grained head pose information (i.e., pitch, yaw, and roll) to estimate the user's gaze position. With this information, the proposed system can identify the target device. Then, the *Do* module captures the user's hand gestures. A combination of hand gesture information and the type of selected IoT device is then translated into an IoT command and transmitted to IoT platforms. Finally, the device is manipulated according to the command. For example, as depicted in Fig. 1, the *Watch* module recognizes the position and the type of each installed device (i.e., a lamp, monitor, fan, refrigerator from left to right). If a user stares at a device at the left-most side, the *Watch* module detects the user's gaze position and predicts the type of the device to be controlled as a lamp. Finally, the user can turn the lamp on or off with a swipe gesture performed near the *Do* module installed around the user.

Fig. 2 shows a systematic architecture of the proposed method. The proposed system consists of two phases: 1) *Watch* phase and 2) *Do* phase. The goal of the *Watch* phase is to identify the target device by estimating the user's gaze position and classifying the device category. For this, a gaze estimation module and an object detection module based on deep neural networks have been adopted. The *Do* phase is to capture the user's hand gestures to compose an IoT device control command using the device and gesture information. Therefore, the *Do* phase exploits a gesture recognition module with an embedded sensor and an IoT controller module which interacts with the IoT platform. The rest of this section provides details on each module.

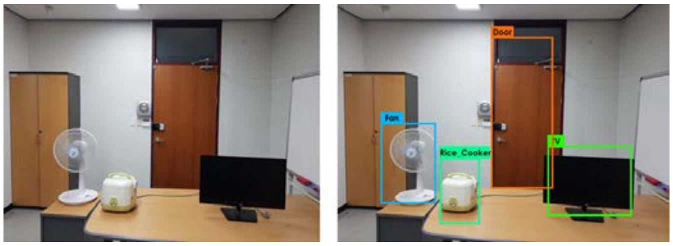


Fig. 6. The result of IoT device detection (left: before, right: after).

A. Gaze Estimation

Most eye tracking and gaze estimation approaches compute the user's gaze position on a display or a screen. In this work, however, the gaze estimation algorithm must be able to compute the gaze position in a room. To achieve this, we divide the physical space of the room into $n \times n$ virtual grids and collect user's head pose information for each grid. Finally, the collected data are utilized to train a gaze estimator. The illustrative example of this setup can be found in Fig. 3.

We use the orientation of the face (i.e., pitch, yaw, and roll) and the distance between the *Watch* module and the user as head pose information for training a gaze estimator. A deep learning-based approach called DeepGaze [53] was employed to compute the value of face orientation. The DeepGaze uses a convolutional neural network (CNN) to output the values of face orientation from the given face image. As can be seen from Fig. 4, the CNN network architecture used in this framework is a slightly modified version of that in [54].

As also described in Fig. 3, the user's head orientation for each grid can vary with the distance between the *Watch* module and the user. Therefore, we consider this distance to training the gaze estimator. This distance can be calculated from the area of the detected face region. The method of face detection is presented in [55]. The final feature vector for gaze estimation thus includes information on the face orientation and distance of the user. A set of feature vectors are then used as input to train the models of final gaze estimators such as decision trees, random forest, support vector machines (SVMs), and k-nearest neighbors (k-NN) algorithms. Fig. 5 shows the examples of facial images used for training the gaze estimator.

B. Object Detection

The *Watch* module also records the opposite side of the user to detect the IoT devices installed in the room and recognizes their types. For object detection and recognition, recent deep learning approaches have shown excellent performance

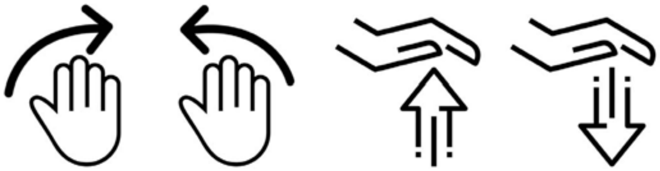


Fig. 7. Supported hand gestures. Each image represents “right”, “left”, “backward”, and “forward” gesture from left to right, respectively.

in terms of accuracy, but the low throughput has been a constraint for real-time use [56], [57]. On the other hand, the regression-based object detector, called YOLO demonstrates a comparable accuracy with improved speed (about 30 $\sim$ 60 fps) [58]. Therefore, we adopted this object detector in the *Watch* module for IoT device detection and classification.

The YOLO framework divides each input image into $S \times S$ grids and each grid predicts N bounding boxes and their confidence scores which indicate whether the bounding box contains an object or not. Further, each grid is assigned a detected label with a class score. The information from the bounding box and class probabilities are then utilized in the final object detection and classification output. In this work, we use the following labels for IoT device categories: a cleaner, air conditioner, rice cooker, TV (monitor), washing machine, fan, and door. We use the convolutional weights pre-trained on ImageNet and train our own network using the images regarding the IoT device labels. Fig. 6 shows the result of IoT device detection and classification using the YOLO framework.

C. Gesture Recognition

Gesture recognition is performed by a commercial gesture sensor module. This sensor supports the following nine hand gestures: “Up”, “Down”, “Left”, “Right”, “Forward”, “Backward”, “Clockwise”, “Counter clockwise”, and “Wave”. User’s hand gestures are associated to different actions according to the type of the selected IoT device. For simplicity, we assigned maximum four gestures (i.e., forward, backward, right, and left) for each device as shown in Fig. 7. A summary of the type of supported devices and their corresponding hand gestures is presented in Table I. The final IoT control message is composed based on this mapping table and transmitted to the IoT platform.

IV. Experiments

To evaluate the performance and feasibility of the proposed system, we conducted quantitative experiments with a prototype. The prototypes of the *Watch* module and *Do* module were configured using commercial webcams and small single-board computers. The deep learning models used for gaze estimation and object detection were trained using a desktop PC with commercial graphic cards supporting parallel computations. The IoT platform used in our experiment is a prototype implementation from the authors’ previous work.

A. Prototype Implementation

First, the *Watch* and the *Do* hardware modules were installed in a room. The experimental condition of the room is similar to Fig. 3 and Fig. 6. The *Watch* module consists of a small single-board computer equipped with a front and back cameras for gaze estimation and object detection, respectively. The *Do* module consists of a unified sensor board and a gesture sensor for hand gesture recognition. The prototype implementations of both modules are presented in Fig. 8 and Fig. 9.

To perform the user study, we also implemented a dashboard application that can display the video frames recorded by the *Watch* module. As depicted in Fig. 10, the dashboard application displays the results of gaze estimation (Label form) and object selection (Object form), gesture recognition (Event form), and the composed IoT command (Details form). In our study, a user was asked to control a device with hand gestures. For example, the instruction given to the user in Fig. 10 is “Turn down the volume of the TV”. It can be found that the

TABLE I
DEVICE TYPE AND ITS CORRESPONDING GESTURES

Device	Gesture	Meaning
Cleaner, rice cooker, washing machine	Forward Backward	Turn on Turn off
Air conditioner	Forward Backward Left Right	Turn on Turn off Temperature up Temperature down
TV/monitor	Forward Backward Left Right	Turn on Turn off Volume up Volume down
Fan	Forward Backward Left Right	Turn on Turn off Speed up Speed down
Door	Forward Backward Left Right	Open Close Lock Unlock

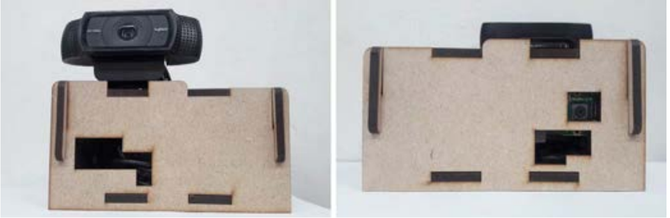


Fig. 8. Prototype implementation of the *Watch* module. The left and right show a front side and a back side of the module, respectively.

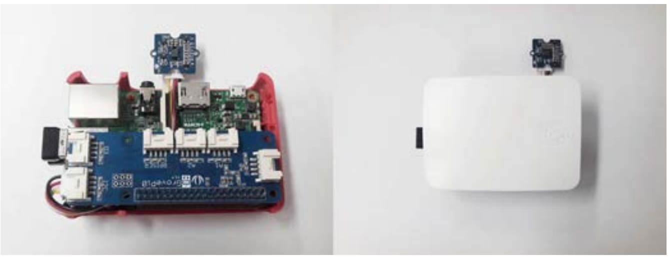
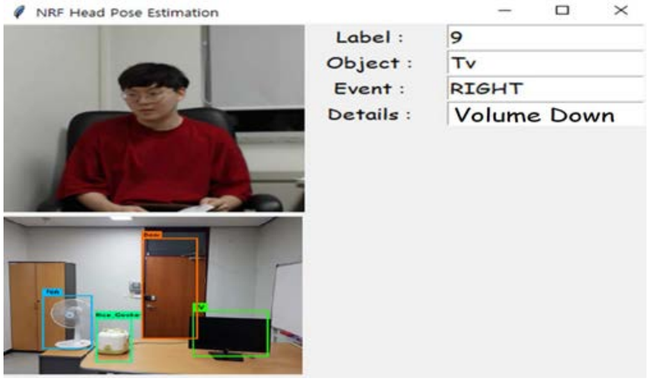


Fig. 9. Prototype implementation of the *Do* module. The left and right show an inside of the module and the final form, respectively.



Label : 9
Object : Tv
Event : RIGHT
Details : Volume Down

Fig. 10. Implementation of the dashboard application.

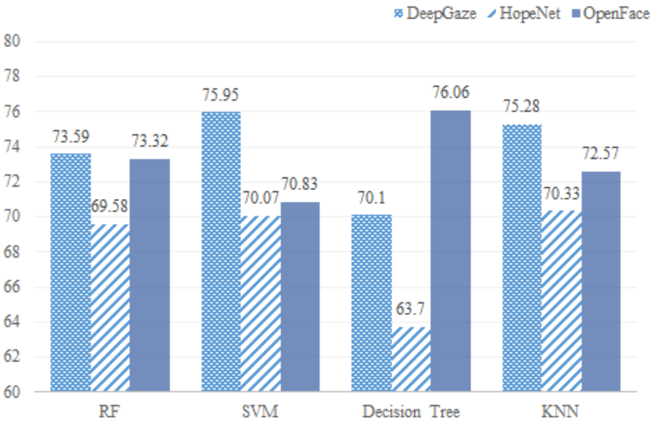


Fig. 11. Classification accuracy with face orientation feature only.

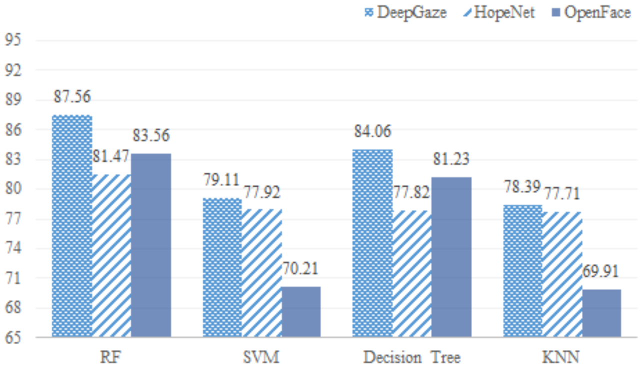


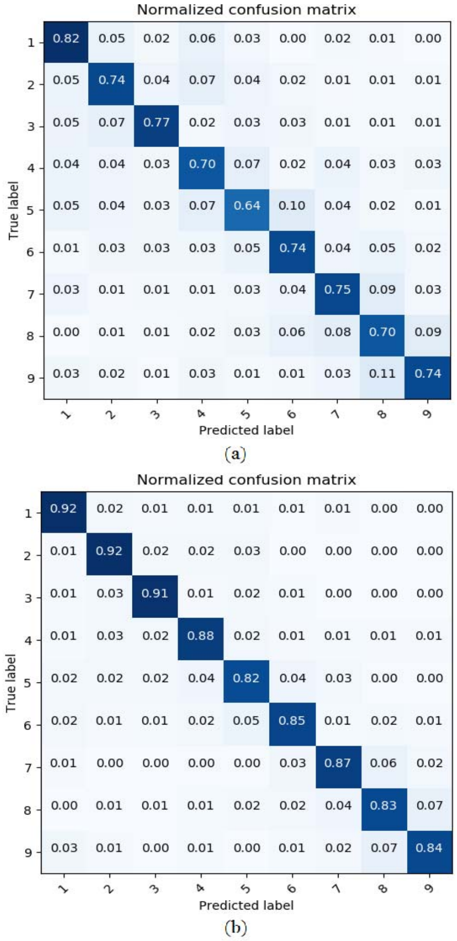
Fig. 12. Classification accuracy with face orientation and distance features.

user gazed at the TV in front of him and then performed the "right" hand gesture to turn down the volume of it.

B. Performance Evaluation

We conducted two experiments to evaluate the performance of the gaze estimation and IoT device detection.

First, we discuss the classification accuracy of gaze estimation for different classifier models and features. To train the final gaze estimator, we adopted decision tree, random forest, SVM, and k-NN algorithms. The dataset consists of 28,350 training images and 9,450 testing images from nine subjects. The classifiers were trained to output the label of the grid (i.e., 1-9) from the input feature vectors (i.e., face orientation and distance) obtained from the DeepGaze framework. For quantitative evaluation of the proposed system, we also developed two more classifiers using gaze features obtained from the head pose estimator (HopeNet) [20] and a model-based gaze tracking method (OpenFace) [30]. Fig. 11 and Fig. 12 show the classification accuracy of each method with and without the face distance feature, respectively. Without the distance feature, the decision tree algorithm with OpenFace features showed the best result (76.06%), which is slightly higher than the SVM algorithm with DeepGaze features (75.95%). On including the distance feature, the performance of every classifier, except the OpenFace-based models, improved significantly, and the maximum gain was 14.12% point for the decision tree algorithm with HopeNet features. The model-based approaches require high-resolution eye images to compute geometric local features; therefore, the performance gain would be limited if the OpenFace-based method cannot detect the eye region from the images. As a result, the random forest model with the face orientation features from DeepGaze and the face distance feature demonstrated the highest classification accuracy (87.56%) among all the models used in the experiment. Fig. 13 depicts the confusion matrix of the classification result from the DeepGaze-based random forest model with and without the



(a)

	1	2	3	4	5	6	7	8	9
1	0.82	0.05	0.02	0.06	0.03	0.00	0.02	0.01	0.00
2	0.05	0.74	0.04	0.07	0.04	0.02	0.01	0.01	0.01
3	0.05	0.07	0.77	0.02	0.03	0.03	0.01	0.01	0.01
4	0.04	0.04	0.03	0.70	0.07	0.02	0.04	0.03	0.03
5	0.05	0.04	0.03	0.07	0.64	0.10	0.04	0.02	0.01
6	0.01	0.03	0.03	0.03	0.05	0.74	0.04	0.05	0.02
7	0.03	0.01	0.01	0.01	0.03	0.04	0.75	0.09	0.03
8	0.00	0.01	0.01	0.02	0.03	0.06	0.08	0.70	0.09
9	0.03	0.02	0.01	0.03	0.01	0.01	0.03	0.11	0.74

(b)

	1	2	3	4	5	6	7	8	9
1	0.92	0.02	0.01	0.01	0.01	0.01	0.01	0.00	0.00
2	0.01	0.92	0.02	0.02	0.03	0.00	0.00	0.00	0.00
3	0.01	0.03	0.91	0.01	0.02	0.01	0.00	0.00	0.00
4	0.01	0.03	0.02	0.88	0.02	0.01	0.01	0.01	0.01
5	0.02	0.02	0.02	0.04	0.82	0.04	0.03	0.00	0.00
6	0.02	0.01	0.01	0.02	0.05	0.85	0.01	0.02	0.01
7	0.01	0.00	0.00	0.00	0.00	0.03	0.87	0.06	0.02
8	0.00	0.01	0.01	0.01	0.02	0.02	0.04	0.83	0.07
9	0.03	0.01	0.00	0.01	0.00	0.01	0.02	0.07	0.84

Fig. 13. Confusion matrix of the classification results from the DeepGaze–based random forest model without (a) and with the face distance feature (b).

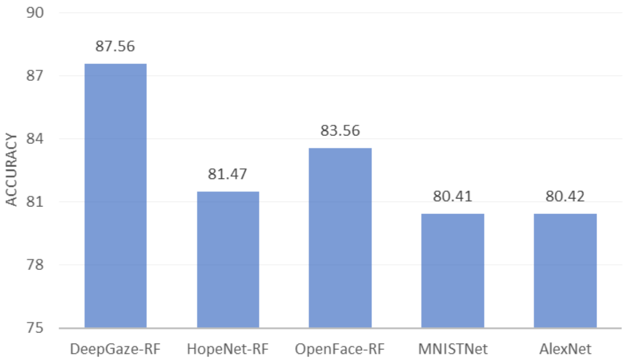


Fig. 14. Classification accuracy of each method with the best model parameters.

face distance feature. It can be found that the face distance feature improves the estimation accuracy for every grid.

We also compared our models with two CNN-based architectures—a four layer (two convolutional and two fully connected layers) CNN architecture called MNISTNet [59] and an AlexNet-like architecture [54]. Fig. 14 presents the classification accuracy of each method with the best model parameters. The experimental results show that the random forest models with DeepGaze, HopeNet, and OpenFace outperform the CNN-based approaches. The maximum accuracy was observed for the random forest model with the DeepGaze features. However, the CNN-based approaches performed competitively without any feature engineering. It is expected that adding optimization techniques will improve the overall performance of the CNN-based approaches.

Next, we discuss the classification accuracy of IoT device detection and classification. The dataset used for our experiment totally consists of 3,065 images of IoT devices (235 images of an air conditioner, 774 images of a rice cooker, 407 images of a washer, 625 images of a cleaner, 315 images of a fan, 530 images of a TV, and 179 images of a door). Due to the lack of a large-scale well-refined publicly available home appliance image dataset, we collected sample images of each device type from various Web search engines. After downloading the images, we manually labeled them to annotate the bounding box and device category information for the YOLO detectors. Fig. 15 presents example training samples used for the IoT device detection and classification.

The classification model was trained for 1,000 iterations. For evaluation of the classification accuracy, we performed a 10-fold cross validation and computed the average precisions. Fig. 16 shows the accuracy of classification results for the IoT devices. On average, the proposed system achieved a high accuracy of 92.25%. Among the devices, the highest classification accuracy was found in the fan (97.17%), while the lowest was for the air conditioner (80.69%). The experimental results on IoT device detection and classification can be interpreted as follows. First, there are no strong correlations between the number of training samples and the classification accuracy in our domain. For example, the device category, door, with the smallest samples (179) shows a high accuracy (93.56%), comparable to that of the rice cooker (93.24%) which has the largest number of samples (774). Second, the visual appearance still affects the classification accuracy. For example, even though both the air conditioner and the door have relatively low number of samples (235 and 179, respectively) there exists a big difference between their classification performances (80.69% and 93.56%, respectively). As depicted in Fig. 15, the door and the fan have unique visual appearances useful for training the detectors. However, it is difficult to find unique characteristics in the appearance of air conditioners, which results in their low classification accuracy.

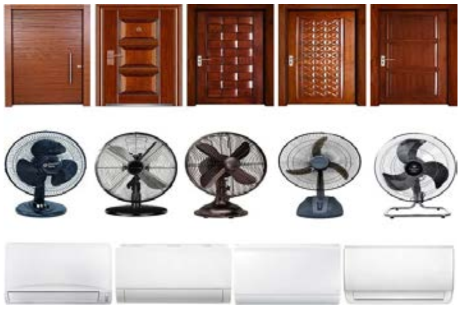


Fig. 15. Example of training samples.

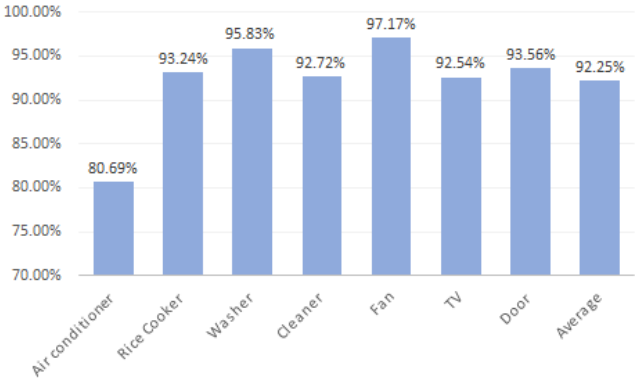


Fig. 16. Classification accuracy of IoT device detection.

Fig. 17 presents the variation in average classification accuracy with the number of iterations. At the 100th iteration, the system produced a poor result of 4.16% on average. However, the performance sharply increases until 300th iteration (73.70%) and then shows a gradual increase until $1,000^{\text{th}}$ iteration (92.25%). The details of the classification accuracy of each device type according to the number of iterations can be found in Table II.

From the experimental results, we can see that the performance of gaze estimation, IoT device detection, and IoT device categorization of the proposed approach is acceptable (around 90%). Therefore, the overall architecture and design of the proposed system should be feasible for use in an IoT environment with further improvements.

C. Comparison With IoT System Interaction Approaches

In this section, we analyze the main concepts of other IoT system interaction approaches and compare them with the proposed system. Table III summarizes the main concepts and components of other IoT system interaction approaches and the proposed system. Generally, an IoT system interaction approach works in the following steps: 1) device discovery, 2) device recognition, and 3) device control.

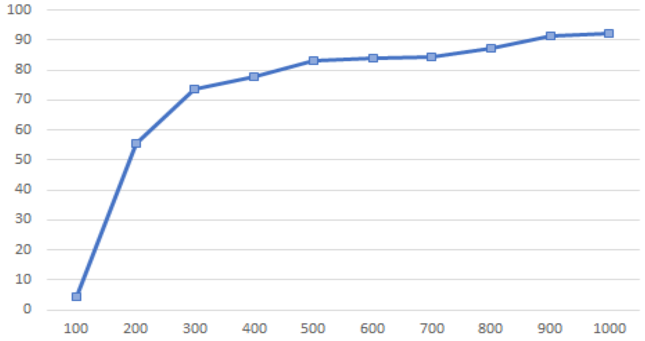


Fig. 17. Average classification accuracy according to the number of iteration.

TABLE II
CLASSIFICATION ACCURACY ACCORDING TO THE NUMBER OF ITERATION

Iter	AC	RC	Washer	Cleaner	Fan	TV	Door
100	3.72	8.84	1.33	2.80	4.71	5.30	2.45
200	18.41	73.18	63.75	54.40	70.31	50.60	57.65
300	25.19	82.95	93.29	83.54	91.42	60.81	78.74
400	32.56	89.31	82.99	78.46	93.47	85.76	81.78
500	41.65	90.82	93.40	88.26	95.01	86.00	87.29
600	38.97	91.34	94.72	90.71	93.48	88.81	90.21
700	37.93	92.15	93.67	91.33	94.80	89.27	91.48
800	48.46	93.27	96.61	92.71	96.34	90.71	93.11
900	78.71	93.38	96.83	92.68	96.22	89.62	93.60
1000	80.69	93.24	95.83	92.72	97.17	92.54	93.56

AC and RC are abbreviations for the air conditioner and the rice cooker

During device discovery, the system identifies and selects the target device. The previous works mainly exploit the device images captured by the users, hand or arm gestures, or user's gaze direction to detect user's intention for device control.

In the device recognition step, the category of the device that a user selected (or intended to select) can be determined in multiple ways. The camera-based approaches [43], [44] used image matching algorithms based on the local features such as SIFT [60] and SURF [61]. However, the local descriptors are generally used for image matching rather than classification, so the reference images and photos should be of a high quality and visual similarity. The gesture based approaches track the user's arms to recognize the predefined gestures for each device [45], [46]. Some works in this category also proposed to compute the direction of arms to find the devices installed at that location [47], [48]. However, in the gesture-based approaches, the usability of the system decreases as the number of devices increases. It becomes difficult to remember the gestures of each device. Also, the systems in [47] and [48] cannot recognize a device if its location is changed from the initial setup. The gaze-based approaches (the proposed system and [50]), on the other hand, detect the target device and classify its category using machine learning methods. Even though these provide the most natural and seamless interaction style, their performance of gaze estimation remains challenging.

After successful recognition of the target device, the users can control the device with a variety of interfaces. References [43], [44], and [48] provide a mobile application to monitor and control the status of a device. References [45], [47], and [50] supports a smart watch interaction where a circular UI and built-in sensors are used for controlling the devices. However, the use of a smart phone and watch as a control interface can be a big hurdle for those who are not familiar with smart devices, such as young children, the elderly, and the disabled. In particular, most of these systems require the user to wear additional devices such as a smart watch or smart glasses, which are uncomfortable and inconvenient. In contrast, the proposed system and [46] allow users to easily control the IoT devices with simple gestures.

The comparison with previous works and the quantitative evaluation demonstrates that the proposed system provides a natural and seamless interaction style with competitive detection and classification performances.

V. DISCUSSION AND CONCLUSION

In this work, we proposed a novel smart IoT interaction system called *Watch & Do*. The proposed method helps users easily control the IoT devices by gazing at a device and performing simple hand gestures. For the *Watch* phase, deep convolutional neural network based approaches were adopted to estimate user's gaze position and to detect and recognize the IoT devices. For the *Do* phase, a simple module with a gesture sensor was implemented. The final IoT command to be transmitted to the IoT platform was composed based on the detected IoT devices and hand gestures. We presented the feasibility of the proposed system through various experiments.

The main contributions of this work are as follows. First, we proposed a novel approach to utilize head pose estimation and object detection technologies for IoT interaction. The previous works on gaze estimation focused on tracking the gaze position on a display. However, these approaches cannot be applied to the IoT environment that requires indoor gaze estimation. In this work, we solve this problem by a combination of object detection and implicit gaze estimation from the fine-grained head pose information. Second, without any additional devices, such as a headband or watch, the users can specify the target device. The previous works used wearable devices to detect user's head pose or gaze direction, which can be uncomfortable for many users. With this work, the users can interact with the IoT devices intuitively by installing simple modules.

However, the proposed system has some limitations to be addressed in the future. First, for a gaze-based approach it is still challenging to accurately detect the user's intention to control a device. For example, the user's unintentional gaze on a certain device can be mis-detected by the system. To reduce this ambiguity, several works adopted device activation strategies which allow users to explicitly select a specific device or express the intention to a system. Rhythmic gesture patterns [46] or smooth pursuit techniques [51] are examples of activation strategies. These techniques help users express their intentions as well as select a specific device. However, the users must remember the activation gestures for each device type. In this regard, the *Do* module of the proposed system will be improved by enabling a device activation feature. One possible solution would be to facilitate a hover gesture detection on the *Do* module. Before gazing at a device, the users can

TABLE III
COMPARISON WITH PREVIOUS IoT SYSTEM INTERACTION APPROACHES

Approach	Device discovery	Device recognition	Device control	Extra equipment	Demonstrated devices
[42]	Smart phone (camera)	Image feature matching	Smart phone UI	-	4 (Printer, projector, display, etc.)
[43]					3 (Printer, projector, light bulb)
[44]	Gesture	Motion matching	Motion matching	Smart watch	1 (Lamp)
[45]		Rhythmic gesture	Gesture	Depth camera and finger tracker	1 (Thermostat)
[46]		Direction computation	Smart watch UI	Smart watch	5 (Fan, light bulb, switch, etc.)
[47]		Direction computation	Smart phone UI	Depth camera	1 (Light bulb)
[48]	Gaze	N/A	Gaze	Rail-based eye tracker	1 (Large display)
[49]		SVM-based recognition	Smart watch UI	Smart glass, smart watch	1 (Speaker)
[50]		Smooth pursuit	Smooth pursuit	Smart glass	4 (Music player, TV, lamp, fan)
Ours		CNN-based recognition	Gesture	Watch & Do module	7 (Fan, TV, door, cleaner, etc.)

place their hand over the *Do* module (i.e., hover gesture) for a short time. If the proposed system detects the user's explicit intention and gaze toward a device, then the *Watch* phase will be processed normally.

Second, in contrast to the conventional gaze estimation and tracking systems that work on a small display [21], the proposed system cannot provide immediate display feedbacks. For example, a conventional gaze tracking system prints a red or green dot on the screen to inform the user of current gaze estimation status. The feedback mechanism is closely related with the device activation feature. With a feedback mechanism, the users can be informed that a device has been selected accordingly and is waiting for the user's instructions or that an operation will be carried out soon. The approaches based on smart phones and smart watches can easily provide a device feedback through their application UIs. On the other hand, the works from [45], [46], and [49], which do not have a display, generally provide the feedback through the light effects around the target device. For example, if a user gazes at a fan and the system detects the user's intention to control the fan, then its surrounding LEDs flash for a short period. In the future, we will extend our system with a feedback mechanism for more convenient interface. This may be accomplished by including a tiny LED module attached to the IoT device (similar to that of [51]) or the *Do* module integrated with a TTS framework to inform the category of the selected device.

Third, the performance of indoor gaze estimation should be improved to guarantee the quality of the proposed system. Even though the experimental results show that the proposed system achieves the highest performance, it still produces 10-15% false alarms. Fourth, the current form of object detection and classification approach has limited interaction angle (front 180-degree). For example, a user cannot interact with a device installed at the right or left side of the user. Finally, even though our system interacts with the IoT platform, the derived benefits are currently limited. For example, a new category and its corresponding training images for IoT device object detection can be dynamically updated with the IoT platform support.

To address these limitations, a multimodal appearance model installed with multiple cameras and improved deep neural network architectures will be studied in the future. Furthermore, a new method to interact with IoT platforms to improve the quality of the proposed method will be addressed.

REFERENCES

[1] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018.

[2] J. B. B. Neto, T. H. Silva, R. M. Assunç ao, R. A. F. Mini, and A. A. F. Loureiro, "Sensing in the collaborative Internet of Things," *IEEE Sensors J.*, vol. 15, no. 3, pp. 6607–6632, Mar. 2015.

[3] C. Perera, A. Zaslavsky, M. Compton, P. Christen, and D. Georgakopoulos, "Semantic-driven configuration of Internet of Things middleware," in *Proc. SKG*, Beijing, China, 2013, pp. 66–73.

[4] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of Things (IoT) security: Current status, challenges and prospective measures," in *Proc. ICITST*, London, U.K., 2015, pp. 336–341.

[5] Y. Zou *et al.*, "GRfid: A device-free RFID-based gesture recognition system," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 381–393, Feb. 2017.

[6] K. Wu *et al.*, "HJam: Attachment transmission in WLANs," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 1449–1457.

[7] Y. Zou, W. Liu, K. Wu, and L. M. Ni, "Wi-Fi radar: Recognizing human behavior with commodity Wi-Fi," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 105–111, Oct. 2017.

[8] K. Wu *et al.*, "CSI-based indoor localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1300–1309, Jul. 2013.

[9] W. Kluge *et al.*, "A fully integrated 2.4-GHz IEEE 802.15.4-compliant transceiver for ZigBee^{TM} applications," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2767–2775, Dec. 2006.

[10] H. B. Pandya and T. A. Champaneria, "Internet of Things: Survey and case studies," in *Proc. EESCO*, Visakhapatnam, India, 2015, pp. 1–6.

[11] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[12] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An application protocol for billions of tiny Internet nodes," *IEEE Internet Comput.*, vol. 16, no. 2, pp. 62–67, Mar./Apr. 2012.

[13] OneM2M Partners. (2018). *OneM2M*. [Online]. Available: http://onem2m.org

[14] Open Connectivity Foundation. (2018). *OCF*. [Online]. Available: https://openconnectivity.org/

[15] Open Connectivity Foundation. (2018). *IoTivity*. [Online]. Available: http://iotivity.org

[16] Open Connectivity Foundation. (2018). *AllJoyn*. [Online]. Available: https://github.com/alljoyn

[17] OpenIoT Consortium. (2018). *OpenIoT*. [Online]. Available: http://www.openiot.eu

[18] N. Alioua, A. Amine, A. Rogozan, A. Bensrhair, and M. Rziza, "Driver head pose estimation using efficient descriptor fusion," *EURASIP J. Image Video Process.*, vol. 2016, no. 1, Dec. 2016, Art. no 2.

[19] B. Ahn, D.-G. Choi, J. Park, and I. S. Kweon, "Real-time head pose estimation using multi-task deep neural network," *Robot. Auton. Syst.*, vol. 103, pp. 1–12, May 2018.

[20] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *Proc. CVPRW*, Salt Lake City, UT, USA, 2018, pp. 2074–2083.

[21] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[22] T. Ohno, N. Mukawa, and A. Yoshikawa, “FreeGaze: A gaze tracking system for everyday gaze interaction,” in *Proc. ACM ETRA*, New Orleans, LA, USA, 2002, pp. 125–132.

[23] Z. Zhu and Q. Ji, “Novel eye gaze tracking techniques under natural head movement,” *IEEE Trans. Biomed. Eng.*, vol. 54, no. 12, pp. 2246–2260, Dec. 2007.

[24] S. S. Liu *et al.*, “An eye-gaze tracking and human computer interface system for people with ALS and other locked-in diseases,” *J. Med. Biol. Eng.*, vol. 32, no. 2, pp. 111–116, Apr. 2012.

[25] K. Kavale, K. Kokambe, and S. Jadhav, “taskEYE: ‘A novel approach to help people interact with their surrounding through their eyes,”’ in *Proc. ICALT*, Mumbai, India, 2018, pp. 311–313.

[26] K.-A. Choi, C. Ma, and S.-J. Ko, “Improving the usability of remote eye gaze tracking for human-device interaction,” *IEEE Trans. Consum. Electron.*, vol. 60, no. 3, pp. 493–498, Aug. 2014.

[27] J. Chen and Q. Ji, “3D gaze estimation with a single camera without IR illumination,” in *Proc. ICPR*, Tampa, FL, USA, 2008, pp. 1–4.

[28] L. Sesma-Sanchez, A. Villanueva, and R. Cabeza, “Gaze estimation interpolation methods based on binocular data,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 8, pp. 2235–2243, Aug. 2012.

[29] J. J. Cerrolaza, A. Villanueva, M. Villanueva, and R. Cabeza, “Error characterization and compensation in eye tracking systems,” in *Proc. ACM ETRA*, Santa Barbara, CA, USA, 2012, pp. 205–208.

[30] E. Wood *et al.*, “Rendering of eyes for eye-shape registration and gaze estimation,” in *Proc. ICCV*, Santiago, Chile, 2015, pp. 3756–3764.

[31] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, “Gaze estimation from eye appearance: A head pose-free method via eye image synthesis,” *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3680–3693, Nov. 2015.

[32] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, “Learning gaze biases with head motion for head pose-free gaze estimation,” *Image Vis. Comput.*, vol. 32, no. 3, pp. 169–179, Mar. 2014.

[33] W. Sewell and O. Komogortsev, “Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network,” in *Proc. ACM CHI EA*, Atlanta, GA, USA, 2010, pp. 3739–3744.

[34] M. B. Ahmad, Saifullah, M. A. Raja, M. W. Asif, and K. Khurshid, “i-Riter: Machine learning based novel eye tracking and calibration,” in *Proc. IEEE I2MTC*, Houston, TX, USA, 2018, pp. 1–5.

[35] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[36] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, “Appearance-based gaze estimation in the wild,” in *Proc. CVPR*, Boston, MA, USA, 2015, pp. 4511–4520.

[37] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, “It’s written all over your face: Full-face appearance-based gaze estimation,” in *Proc. CVPRW*, Honolulu, HI, USA, 2017, pp. 2299–2308.

[38] K. Krafka *et al.*, “Eye tracking for everyone,” in *Proc. CVPR*, Las Vegas, NV, USA, 2016, pp. 2176–2184.

[39] M. A. Rashid and X. Han, “Gesture control of ZigBee connected smart home Internet of Things,” in *Proc. ICIEV*, Dhaka, Bangladesh, 2016, pp. 667–670.

[40] F. Erden and A. E. Çetin, “Hand gesture based remote control system using infrared sensors and a camera,” *IEEE Trans. Consum. Electron.*, vol. 60, no. 4, pp. 675–680, Nov. 2014.

[41] S. Jeong, J. Jin, T. Song, K. Kwon, and J. W. Jeon, “Single-camera dedicated television control system using gesture drawing,” *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1129–1137, Nov. 2012.

[42] E. Morganti *et al.*, “A smart watch with embedded sensors to recognize objects, grasps and forearm gestures,” *Procedia Eng.*, vol. 41, pp. 1169–1175, Jan. 2012.

[43] A. A. de Freitas *et al.*, “Snap-to-it: A user-inspired platform for opportunistic device interactions,” in *Proc. ACM CHI*, San Jose, CA, USA, 2016, pp. 5909–5920.

[44] K. Chen *et al.*, “SnapLink: Fast and accurate vision-based appliance control in large commercial buildings,” *ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–27, Jan. 2018.

[45] D. Verweij, A. Esteves, V.-J. Khan, and S. Bakker, “WaveTrace: Motion matching input using wrist-worn motion sensors,” in *Proc. ACM CHI*, Denver, CO, USA, 2017, pp. 2180–2186.

[46] E. Freeman, S. Brewster, and V. Lantz, “Do that, there: An interaction technique for addressing in-air gesture systems,” in *Proc. ACM CHI*, San Jose, CA, USA, 2016, pp. 2319–2331.

[47] G. Alce *et al.*, “UbiCompass: An IoT interaction concept,” *Adv. Human Comput. Interact.*, vol. 2018, Apr. 2018, Art. no. 5781363.

[48] M. Budde *et al.*, “Point&control—Interaction in smart environments,” in *Proc. UbiComp*, Zürich, Switzerland, 2013, pp. 303–306.

[49] M. Khamis *et al.*, “Eyescout: Active eye tracking for position and movement independent gaze interaction with large public displays,” in *Proc. UIST*, Quebec City, QC, Canada, 2017, pp. 155–166.

[50] S. Mayer and G. Soros, “User interface beaming—Seamless interaction with smart things using personal wearable computers,” in *Proc. BSN*, Zürich, Switzerland, 2014, pp. 46–49.

[51] E. Velloso, M. Wirth, C. Weichel, A. Esteves, and H. Gellersen, “AmbiGaze: Direct control of ambient devices by gaze,” in *Proc. DIS*, Brisbane, QLD, Australia, 2016, pp. 812–817.

[52] M. Vidal, A. Bulling, and H. Gellersen, “Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets,” in *Proc. UbiComp*, Zürich, Switzerland, 2013, pp. 439–448.

[53] M. Patacchiola and A. Cangelosi, “Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods,” *Pattern Recognit.*, vol. 71, pp. 132–143, Nov. 2017.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. NIPS*, 2012, pp. 1097–1105.

[55] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. CVPR*, Kauai, HI, USA, 2001, pp. 511–518.

[56] R. Girshick, “Fast R-CNN,” in *Proc. ICCV*, Santiago, Chile, 2015, pp. 1440–1448.

[57] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. NIPS*, Montreal, QC, Canada, 2015, pp. 91–99.

[58] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. CVPR*, Las Vegas, NV, USA, 2016, pp. 779–788.

[59] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[60] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

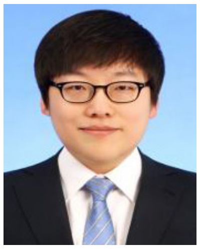
[61] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

Jung-Hwa Kim was born in Gumi, South Korea. She is currently pursuing the B.E. degree in computer engineering with the Kumoh National Institute of Technology, Gumi.

Her current research interests include Internet of Things, deep learning, and artificial intelligence.

Seung-June Choi was born in Gumi, South Korea. He is currently pursuing the B.E. degree in computer engineering with the Kumoh National Institute of Technology, Gumi.

His current research interests include Internet of Things, deep learning, and artificial intelligence.



Jin-Woo Jeong received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Hanyang University, South Korea, in 2006, 2008, and 2013, respectively.

From 2013 to 2016, he was a Senior Research Engineer with the Software R&D Center, Samsung Electronics. Since 2016, he has been an Assistant Professor with the Department of Computer Engineering, Kumoh National Institute of Technology, Gumi, South Korea. His research interests include Internet of Things, data mining, deep learning, information retrieval, and smart interaction.