논문 22-47-03-09 The Journal of Korean Institute of Communications and Information Sciences '22-03 Vol.47 No.03
https://doi.org/10.7840/kics.2022.47.3.474

Implementation of Emotion Recognition Using Edge-Cloud Computing

Ej Miguel Francisco Caliwag◆, Henar Mike O. Canilang*, Angela C. Caliwag*, Wansu Lim°

ABSTRACT

Emotion recognition systems has been in demand due to its aid in several applications such as health monitoring, workload and drowsiness detection. For these applications, emotion recognition systems require to be deployed on an edge device. For deployment on an edge device, numerous limitations and bottlenecks such as implementation cost, deployment capabilities, and system efficiency affects the performance of the emotion recognition system. That is, emotion recognition on the edge suffers from either low accuracy or high inference time due to the hardware constraints. Hence, several previous studies focus on the deployment of emotion recognition on the edge. Despite that, low accuracy and high inference time still remains an issue. To resolve this, a platform with higher computation capacity must be employed. In this study, we implement an enhanced emotion recognition system by integrating cloud computing platform to the emotion recognition system process, whereby all emotion recognition tasks are performed on the cloud server, can overcome conventional edge device bottlenecks and provide cost-effectiveness, efficient power consumption, and enhanced computing process. Based on the results shown in this study, the proposed system is successful in predicting the emotion of the users in real-time.

Key Words : Application programming interface, cloud computing, emotion recognition, edge computing, machine learning

Ⅰ. Introduction

Edge-cloud computing paradigm deployed in edge AI devices aims to address the increasing demands and existing constraints of the current internet-of-things (IoT) applications[1-3]. To give emphasis on the increasing demands of edge-cloud computing deployed to edge AI devices, market and markets[4] projected an increasing trend for edge AI deployment in terms of industrial applications with 590 million USD to 1,835 million USD from 2020 to 2026 respectively. This represents to a 20.8% compound annual growth rate (CGAR), which is a statement in this AI era. AI applications are widely used nowadays for residential, commercial and industrial applications. [5] emphasized that edge cloud intelligence through edge AI devices transpires to be an engaging computing paradigm for state-of-the-art applications such as the augmentation of human cognition. In [5], deep neural networks (DNNs) are mostly utilized for edge AI applications[6]. DNNs are widely used nowadays for applications such as image processing[7], voice / language processing[8] and recommender systems[9,11].

※ This work was supported by the National Research Foundation of Korea (2020R1A4A101777511, 2021R1I1A3056900)
◆ First Author : Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, 학생회원
° Corresponding Author : Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, wansu.lim@kumoh.ac.kr, 정회원
* Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering
논문번호 : 202110-263-C-RE, Received September 30, 2021; Revised November 30, 2021; Accepted November 30, 2021

A cutting edge application of DNN nowadays allows an AI to identify the emotion of humans through multitude of ways such as using image processing, voice / language processing and body signal processing. As a fact, emotion recognition technology known as "ERT" integrated to AI is a growing billion dollar industry[10]. It is projected that in the post COVID-19 era, emotion detection and recognition will have a market value of USD 19.5 billion from 2020 to USD 37.1 billion for the year 2026. This is equivalent to an annual growth rate of 11.3% for the 2020 to 2026 period. Majority of this market growth pertains to the speech-based emotion detection system, the rapidly developing AI market and the technological advancement for intelligent artificial agents such as edge-cloud computing. There are a lot of possible application for emotion recognition such as for security, automotive industry, gaming industry, smart home and smart city.

Though emotion recognition is in demand nowadays for researchers and developers, there are numerous constraints to consider such as the research cost and the actual implementation cost. These two constraints are present to majority of state-of-the-art application of AI such as at the edge-cloud computing platform. In terms of the emotion recognition and detection application, its complexity is also a relatively major issue pertaining to the processing-to-response-time which is a must for real-time systems. This constraints is carried on when an AI is deployed at the edge or commonly known as edge computing paradigm. Edge AI devices have low resource capability such as their processing power and storage capability[11]. These constraints includes mobile data traffics, real-time data processing critical bottlenecks such as latency and network bandwidth, deployment complexity, optimization problem and data security[12]. The constraints of emotion recognition system and its deployment to edge AI device are supplemental with each other.

In [13], an emotion recognition deployed to the edge realizes the constraints of the emotion recognition model being computationally expensive and the edge AI device which has limited computational resources. This paper highlighted the importance of the emotion recognition to be in real-time thus latency is considered. Emotion recognition implementation for mobile edge computing applications highlighted the constraint of low power consumption characteristic of the subject edge AI device[14]. In line with this, an optimization was applied for their EEG model prior to deployment. Another emotion recognition model using face, [15] highlighted the high computational complexity of this model which is a challenge for edge deployment. Summing it up, the challenge is to run a complex deep learning model like emotion recognition to the edge device which has limited computing capability. The limited computing capability and power consumption of edge AI device requires a lightweight and computationally inexpensive models[16,17].

In line with these constraints of emotion recognition and its deployment to the edge, we propose a low-cost, adaptive and state-of-the-art continuous edge-cloud emotion recognition system that can pave way to multitude of innovations. This study considers the implementation cost, deployment capability and system efficiency which gears towards an enhanced emotion recognition and detection system. In terms of implementation cost, using edge-cloud helps minimize the system management cost and maintenance. Another is that expenses for purchasing equipment to manage a system is reduced by using cloud computing service provider. With this, the overall energy consumption cost is minimized through the application and integration of edge-cloud computing to emotion recognition.

Ⅱ. Methodology

In this section, the overall system process, including the edge, cloud, and the emotion recognition algorithm will be discussed. The overall proposed edge-cloud emotion recognition system is depicted in Fig. 1. In the proposed system, for the edge, an IP camera application feeds video frames to an Ubuntu-based PC. The frames from the IP

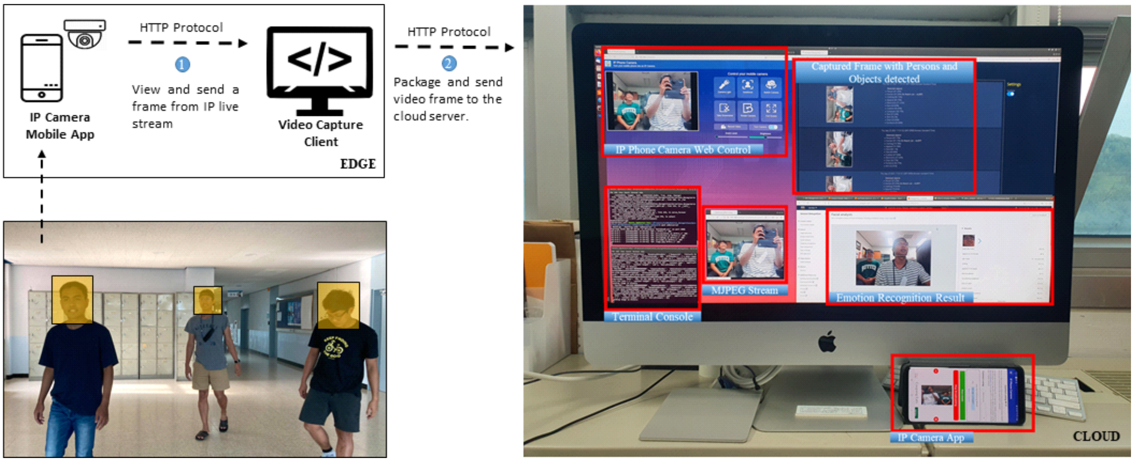


Fig. 1. Overall proposed edge-cloud conFiguration

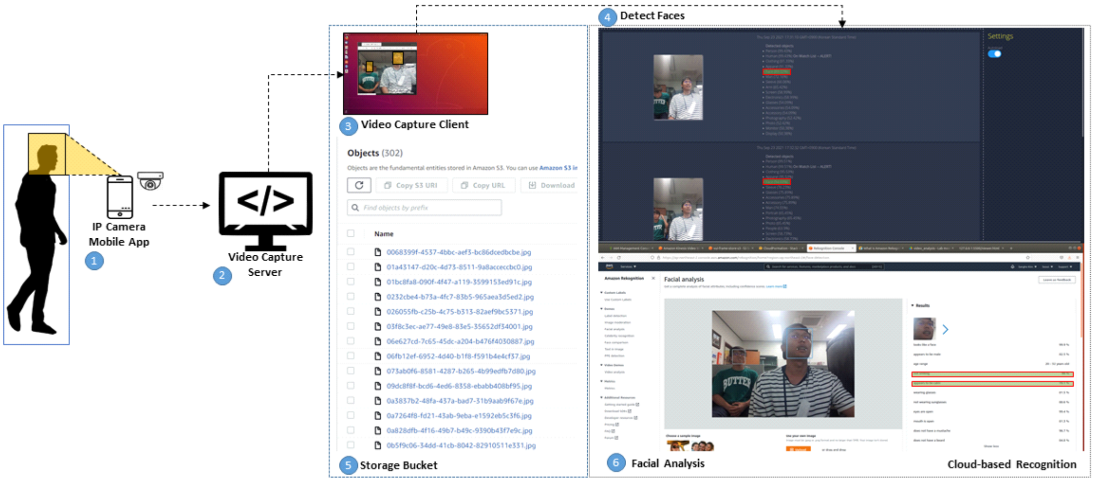


Fig. 2. Step-by-step process of the proposed edge-cloud

camera app are sent to the PC using the HTTP protocol. As soon as the PC receives the frames, it automatically send those frames to the cloud server for further processing. For the cloud, a cloud server is used to deployed a face detection algorithm and emotion recognition model with high computational complexity. A step-by-step process of the proposed edge-cloud is depicted in Fig. 2. Moreover, a system structure shown in Fig. 3 is also discussed in details.

2.1 Edge

In this study, the edge network is used for deploying VideoCapture Server and VideoCapture Client. The VideoCapture Server serves as the frame source. It is where the input frames originated before being transmitted to the VideoCapture Client. The VideoCapture Client serves as the frame destination. It is where the input frames can be displayed and viewed. In this study, a mobile phone and a PC is used as an edge device. For the VideoCapture Server, an IP camera or any android system that runs an IP camera app is eligible to be used as a video stream input. IP camera or internet protocol camera is a digital video camera much like a webcam, which compress files, transmits and receives data over a network or the internet.In addition a local PC is needed to serve as the VideoCapture Client to receive the frames from the mobile IP camera app. As shown in Fig. 1 and Fig.

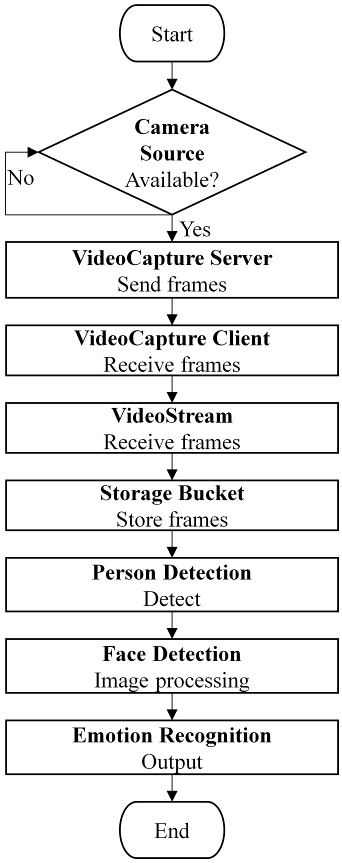


Fig. 3. Flow chart of cloud server process

2, the edge devices supports the VideoCapture Sever and VideoCapture Client. For the purpose of this study, an Android app with an IP camera mobile application is used as the source of the video stream. The video streams catches frame using the mobile phone built-in camera. This frames are then transferred via hypertext transfer protocol (HTTP) from the VideoCapture Server (mobile phone) to the VideoCapture Client (local PC) running Ubuntu 18.04 LTS. All of the frames from the video stream were received by a video capture client and can be visualize through a web graphic user interface (GUI). Concurrently, while vieweng the frames being received by the VideoCapture Client, it also provides the frames IP address. This IP address can then be used to access and transmit those frames to the cloud server. By using IP camera, the size of the frames are then compressed to a network file size which allows the transmission of the frames to be efficient. As shown in Fig. 3, the proposed system fist verifies if there is a camera source connected. If not, the system will be on idle waiting for a camera input to initialize the video stream. If yes, the system will start to receive video frames on the VideoCapture Server. The VideoCapture Client, on the other hand, is already expecting for a frame to receive. Once the VideoCapture Client has received the frames, it can begin transmitting them to the cloud server when the cloud server has established a connection using the IP address.

2.2 Cloud

In this study, the cloud server is used to employ face detection algorithms and emotion recognition models with high computational complexity. As shown in Fig. 3, the cloud server supports VideoStream, Storage Bucket, Person Detection, Face Detection, and Emotion Recognition. As shown in Fig. 2, first, the frames will be stored in a storage bucket once they have been received by the cloud server for backup purposes. Meanwhile, to save the images so that it can be used as a dataset to improve machine learning model training later. Alongside, it will begin streaming them in order to begin processing and detecting people inside the frames. When a person is spotted, a face detection algorithm is activated, which begins completely detecting the individual’s face. One thing to consider while conducting face recognition is the human facial posture, as it has a significant impact on the final outcome. For a better face analysis, the face detection algorithm takes into account the facial posture as well. The detected face’s bounding box will be cropped out, and the image will be enhanced for better facial. The image enhancement has two parameters: sharpness and brightness. Both parameters are returned as values between 0 and 1. This method reduces latency since the model does not have to process the entire frame, which can be large in pixels and take a long time to execute. Following the enhancement of the detected face cropped image, the facial analysis emotion recognition algorithm will be used to determine the person’s emotion. As a contribution for this implementation, first, the algorithm will detect whether the person is smiling or not. Detecting whether the person is smiling or not is beneficial for accurate emotion recognition by allowing the model to first identify the facial status and attribute of a persons' face. If the mouth is relatively wide compared to the specific average face or both upper and lower lips are separated from each other, as well as teeth might appear, then it can serve as a parameter or condition considerations. After that, the emotion of the person will be predicted and detected. These tasks are deployed in the cloud due to the scalable and elastic capability of the cloud. Moreover, cloud is more efficient option for performing such complex algorithm. High comlpexity algorithm requres high computational capacitya and capability which the cloud offers. The emotion recognition algorithm will be covered on the next subsection. After a successful emotion recognition, the metadata will be stored in a database, along with a link to the original image containing the cropped faces. When a user accesses the GUI, a python function is called. This python function is used to capture the frame and its associated metadata from the database for viewing. The GUI displays the cropped image of the persons face as well as the face detection bounding box and the emotion recognition results.

2.3 Emotion recognition

Facial analysis is the process of detecting a face within an image and extracting relevant face attributes from it. Using the DetectFaces API, the bounding box for each face detected in an image along with attributes such as gender and face landmark points will be return as an output. A long the way, enhanced artificial intelligence (A2I) was applied in conjunction with the facial recognition algorithm. This improved artificial intelligence enabling cloud services to communicate with one another, resulting in a more efficient emotion recognition. Low confidence forecasts from emotion recognition are not a concern because the A2I allows for human reviewers to help with low confidence predictions. This is done by the application programming interface (API) that helps a lot in content moderation. This cloud feature enables monitoring of the prediction accuracy as well on a regular basis. At first, the emotion recognition algorithm operates on its own, without the need to examine whether or not the person is smiling. However, in this work, incorporating a person's smile as an additional parameter plays a significant role in recognizing an individual's emotion in order to improve the accuracy of the suggested emotion detection, Therefore, in this study, considering whether a person is smiling or not is performed first before detecting and recognizing the persons' actual emotion. By implementing this model the emotion identification algorithm can easily identify the persons' emotion by validating the accuracy of the output, whether the person is smiling or not.

Ⅲ. Results and Discussions

In this section, the conFiguration setup for both the hardware and software utilized in the experiment, as well as the results obtained will be presented.

3.1 Software ConFiguration

All procedures in conventional emotion recognition are deployed and performed solely on the edge. The main barriers are limited storage, processing capability, and significant power consumption. By enabling edge-cloud computing, these edge computing device barriers can be overcome. In this research, cloud computing's processing capability were utilized to conduct emotion recognition. To begin, the conFiguration of all of the pre-requisite conFigurations and installations, as well as prepare all of the necessary hardware components. On a local PC, a Software Development Kit (SDK) was used to enable coding and cloud provider access. Essential python libraries were installed to access camera, time zone calculations, and cloud setups using the python 'pip' function. Such as:

OpenCV: is an open-source library that includes several hundreds of computer vision algorithms. The document describes the so-called OpenCV 2.x API, which is essentially a C++ API, as opposed to the C-based OpenCV 1.x API. This enables the code to access the camera from a variety of locations and sources.

Pynt: is a scripting language for automating build processes. This library was used to generate and call JAVA (a general-purpose programming language that allows developers to write once and run anywhere (WORA)) methods.

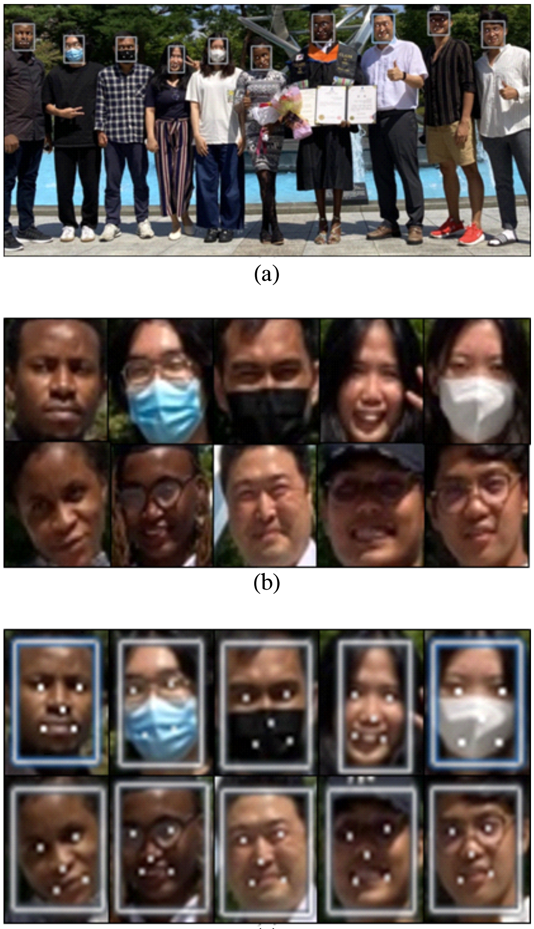
Pytz: is a python library that allows accurate and cross platform time zone calculations using Python 2.4 or higher. It also solves the issue of ambiguous times at the end of daylight-saving time.

User entities to represent the person or application were being utilized to interact on the cloud provider using an access key and a security access. A user with full access to the cloud services required as well as the policies were created as a prerequisite. Policies are the certificates required to access cloud services such as Amazon, Microsoft, and Google, among others. Then, through a JavaScript Object Notation (JSON) format file, conFiguration of the cloud services required using a command-line interface (CLI), an application programming interface (API) and SDK were performed. A JSON file is used to implement conFigurations. The model was built using the python package 'pynt,' which provides build instructions that are implemented as python scripts.

3.2 Hardware ConFiguration

Only two hardware components were employed in the hardware conFiguration: 1) a mobile phone with an IP camera app and 2) a local PC. For the video capture server, an Android phone running Android version 10, a rear camera with dual pixel 12MP AF with a pixel size of 1.4m and Video Digital Image Stabilization (VDIS), and an IP camera app installed were used. For the video capture client, a desktop with Ubuntu 18.04.5 LTS equipped with an Intel i5-4690 CPU @ 3.50GHz and a RAM size of 32GB were employed.

In Fig. 4, a sample implementation of the proposed edge-cloud emotion recognition system is



(a) (b) (c)

Fig. 4. Cropped faces via the face detection algorithm

deployed and tested on ten different individuals. The initial frame received on the cloud server has been processed and analyzed in order to detect faces, as seen in Fig. 4a. A frame with face detected is stored inside a storage bucket deployed on the cloud. The cropped images of the detected faces from the frame are shown in Fig. 4b. Finally, on Fig. 4c, the bounding box for the face detection algorithm was illustrated. Table I shows thee results on the accuracy of the proposed edge-cloud emotion recognition. The detection accuracy and emotion recognition accuracy percentages are presented. The table shows that detecting whether or not an individual is smiling in the first place has a significant impact on the final output of emotion recognition. Looking at the ten individuals, the

Table 1. Emotion recognition results for each cropped faces

Person	Smiling	Accuracy (%)	Emotion	Percentage (%)
1	No	97.5	Calm	53
2	No	64.7	Sad	53.9
3	No	89.8	Calm	89.8
4	Yes	62.2	Happy	87.3
5	No	97.8	Calm	95.8
6	Yes	59.1	Happy	80.1
7	No	59.2	Happy	53.2
8	Yes	65.9	Happy	85
9	Yes	62.1	Happy	91.5
10	No	52.6	Happy	62.2

model detects whether a person is smiling or not with corresponding accuracy shown in the accuracy (%) column. These results now play an important role on how the emotion of an individual will be predicted. The percentage (%) column shows the confidence of the model indicating the emotion of an individual.

Ⅳ. Conclusion

In this paper, a real-time edge-cloud emotion recognition application was developed and deployed. The performance of the application was evaluated on the accuracy of the recognition on the cloud. We provide the results by implementing and testing the proposed edge-cloud emotion recognition on ten individuals standing in one frame. The results demonstrate that the edge-cloud deployment is efficient enough in recognizing emotions of an individual. In addition, detecting whether a person is smiling or not contributes in the emotion recognition algorithm by providing a persons' facial attributes as an additional parameter for emotion recognition. Moreover, the proposed system includes the following features as well: 1) cost-effectiveness, 2) efficient power consumption, and 3) enhanced computing process.

References

[1] S. Jung, "Edge computing and vehicles: Opportunities and challenges for the future," *J. KICS*, vol. 46, no. 5, pp. 834-847, May 2021.

[2] J. Hwang, et al., "Internet of things identification and edge gateway architecture for smart city infrastructure management," *J. KICS*, vol. 44, no. 10, pp. 1981-1991, Oct. 2019.

[3] Y. Liu, et al., "Toward edge intelligence: Multiaccess edge computing for 5g and internet of things," in *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6722-6747, Aug. 2020.

[4] "*Edge computing market*," Market Research Firm, [Online], Available: https://www.marketsandmarkets.com/Market/market-133384090.html.

[5] T. Rausch, et al., "Edge intelligence: The convergence of humans, things, and AI," *2019 IEEE Int. Conf. Cloud Eng.*, Prague, Czech Republic, Jun. 2019.

[6] D. Liu, et al., "HierTrain: Fast hierarchical edge AI learning with hybrid parallelism in mobile-edge-cloud computing," in *IEEE Open J. Commun. Soc.*, vol. 1, pp. 634-645, 2020.

[7] M. D. Zeiler, et al., "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, pp. 818-833, 2014.

[8] J. Devlin, et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," *Comput. and Lang.*, May

2019.
[9] P. Covington, et al., "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, pp. 191-198, 2016.
[10] "*Emotion detection and recognition market*," Market Research Firm, [Online], Available: https://www.marketsandmarkets.com/Market-Reports/detection-recognition-market-23376176.html.
[11] A. Ahmed, et al., "A survey on mobile edge computing," in *Proc. Int. Conf. Intell. Syst. Contr.*, pp. 1-8, 2016.
[12] M. De Donno, et al., "Foundations and evolution of modern computing paradigms: Cloud, IoT, Edge, and Fog," in *IEEE Access*, vol. 7, pp. 150936-150948, 2019.
[13] G. Muhammad, et al., "Emotion recognition for cognitive edge computing using deep learning," in *IEEE Internet Things J.*, vol. 8, no. 23, Dec. 2021.
[14] Z. Li, et al., "Intelligent emotion detection method in mobile edge computing networks," *2020 IEEE/CIC Int. Conf. Commun. in Chin.*, pp. 1214-1219, 2020.
[15] H. Zhang, et al., "A face emotion recognition method using convolutional neural network and image edge computing," in *IEEE Access*, vol. 7, pp. 159081-159089, 2019.
[16] P. Hu, et al., "Software-defined edge computing: Principles, open system architecture and challenges," *2019 IEEE SmartWorld, Ubiquitous Intell. Commun.*, pp. 8-16, 2019.
[17] Y. Liu, et al., "Toward edge intelligence: multiaccess edge computing for 5g and internet of things," in *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6722-6747, Aug. 2020.

이 제 이 (Ej Miguel Francisco Caliwag)
2020년 2월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어
[ORCID:0000-0003-4563-8705]

헤 나 르 (Henar Mike O. Canilang)
2020년 2월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어
[ORCID:0000-0002-9774-5118]

안 젤 라 (Angela C. Caliwag)
2019년 10월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어
[ORCID:0000-0002-0279-935X]

임 완 수 (Wansu Lim)
2014년 9월~현재 : 금오공대 전자공학부 부교수
<관심분야> 임베디드 시스템, 지능형 제어
[ORCID:0000-0003-2533-3496]