Continuous Emotion Recognition on the Edge

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

Department of Aeronautics, Mechanical and Electronic Convergence Engineering
Kumoh National Institute of Technology, Gumi, South Korea
email: wansu.lim@kumoh.ac.kr

Abstract

Traditional Edge AI implementation methods requires high cost computing machines / device. Aside from these high cost devices, most implementation often requires complex AI models. In this study, a real-time implementation for Edge AI devices is implemented to address the implementation cost and complexity constraints. The Arduino Nano 33 BLE sense is used as an Edge AI Device for this implementation. An open source machine learning platform is used to develop an embedded machine learning implementation which addresses the high implementation and computational complexity of edge AI implementation.

Ⅰ. INTRODUCTION

Billions of mobile devices and IoT devices generates zillion data bytes at the edge of the network. This leads to network constraints such as network bandwidth, distributed computing, latency, security and accessibility, data backup, data accumulation, control management and scale. Edge AI device aims to overcome these constraints by deploying artificial intelligence at the edge to generate, accumulate and process data locally at the edge [1]. To give the trend of Edge AI deployment in terms of small scale and industrial applications, it is expected to grow from USD 590 million (2020) to USD 1,835 million (2026). This pertains to a compound annual growth rate (CAGR) of 20.8% according to market and markets. Deep Neural Networks (DNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs) and generative adversarial networks (GANs) emerged as the top AI methods up to this date [2].

Edge AI applications for computer vision such as face detection and recognition, speech recognition and language processing and data generation is rapidly improving with current hardware architectures and application platforms. Edge AI devices and platforms from different well-known company and manufacturers aims to give a state-of-the-art service to its users. Even though, this continuous growth and development of deploying artificial intelligence in edge AI devices give some limitations to researchers, beginners and some interested individuals. This is because in most cases especially for beginners, Edge AI application is complex and often comes with a high implementation cost.

In this study we implement a continuous emotion recognition on a small-scale edge device. Specifically, we used Arduino Nano BLE 33. Our implementation aims to reduce the latency by splitting the processes between edge and cloud. In addition to that, it also promotes mobility and reduces the overall power consumption. Moreover, we utilize the low-power and low-cost computational capabilities of the microcontroller along with the computational power of the cloud. By connecting the microcontroller to the cloud, it is easier to train a machine learning model in the cloud and deploy it in the edge. Our work will then be applied to a real-time application for user emotion recognition. The following section discusses the details on our implementation.

Ⅱ. PROPOSED METHODOLOGY

This paper aims to exploit the easiest possible way to deploy artificial intelligence at a low-cost and small-scale edge AI device. Specifically, we deploy deep learning-based continuous emotion recognition on a cloud-edge system. To reduce the processing on the cloud, the edge device performs motion detection. Then, the emotion recognition process only starts when a motion is detected. Exploiting deep learning algorithms on the edge gives way to some interested individuals and beginners to be a part of this AI revolution.

A. Experimental Set-up.

In this experiment, we use arduino nano BLE 33 which is a low-cost, low-power, and small-scale microcontroller that is capable of handling machine learning and artificial intelligence processes. By using a serial port, we connect the microcontroller to a 64-bit Microsoft Windows operating system with intel® Core(Tm) i7-10700 CPU @ 2.90GHz processor running with a 32 GB ram and a NVIDIA GeForce RTX 3090 (a Compute Unified Device Architecture-compatible GPU and a CUDA-compatible GPU). This allows us to train our own model faster and more efficient before deploying to the edge device. We used C922 Pro HD stream webcam for gathering sample image data used for training. We deploy the training and verification on the cloud using an open source machine learning platform called Edge Impulse. By running the 'edge-impulse-daemon' command in the windows

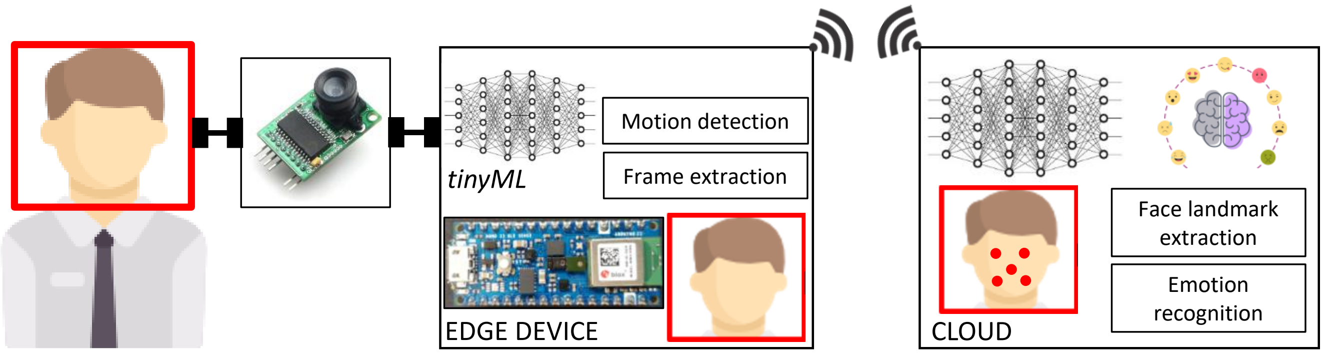

Fig. 1:Proposed Emotion Recognition Model

	HAPPY	SAD	ANGRY	FEAR
HAPPY	100%	0%	0%	0%
SAD	0.9%	96.4%	0%	0%
ANGRY	0%	0%	97.7%	2.3%
FEAR	0%	1.1%	2.0%	96.4%

Fig. 2: Sample Confusion Matrix.

command line, the board will be open for accessibility and configuration.

B. System Model

The system will first perform motion detection on the edge with the help of tinyML. Once a motion was detected, the system will try to catch a frame with a person face using OV7670 CMOS VGA Camera Module connected to Arduino Nano BLE 33. A detection of motion assumes that a person is present within the vicinity. The goal here is to detect only the motion of a human. After that, the detected frame with a face will then undergo to frame extraction focusing mainly on the face. The extracted frame will be sent to the cloud for face landmark extraction. The facial landmarks will be used as features for emotion recognition using Wi-Fi. Finally, live classification will be implemented using the model running on the edge as shown in Fig. 1.

Ⅲ. RESULTS AND DISCUSSIONS

For this experiment, we use AffectNet as our dataset for face recognition. The dataset contains more than 1million facial images collected from the internet by querying three major search engines using 1250 emotion related keywords in six different languages. We only limit the experiment into four classes. We train and run the model with only happy, sad, angry, and fear facial expressions. Fig. 2 shows the confusion matrix of our model. Our current model shows 86.7% accuracy with 0.32 loss. Our proposed method for motion detection returns an inference time of 1 ms and a peak ram usage of 1.5k running on the microcontroller.

Ⅳ. CONCLUSION

In this paper, we proposed a continuous emotion recognition. The tasks are divided into two platforms: 1) Edge and 2) Cloud for faster inference time. Motion detection and frame extraction are deployed on the edge and face landmark extraction and emotion recognition are deployed in the cloud. The proposed model was deployed in a low-cost, low-power consumption and small-scale microcontroller. With the use of an open source machine learning platform we were able to develop an embedded machine learning implementation that addresses high implementation and computational complexity of edge AI applications.

ACKNOWLEDGMENT

This work was supported by the Technology development Program (S2829065, S3010704) funded by the Ministry of SMEs and Start-ups (MSS, Korea), and by the Basic Research Program through the National Research Foundation of Korea (NRF) funded by the MSIT (2020R1A4A101777511).

REFERENCES

[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," in Proceedings of the IEEE, vol. 107, no. 8, pp. 1738-1762, Aug. 2019, doi: 10.1109/JPROC.2019.2918951.

[2] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar and A. Y. Zomaya, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7457-7469, Aug. 2020, doi: 10.1109/JIOT.2020.2984887.

[3] R. Agada and J. Yan, "Edge based mean LBP for valence facial expression detection," 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2015.

[4] AffectNet, available in http://mohammadmahoor.com/affectnet/