FPGA Implementation of a DPU-Based Facial Expression Recognition System

Takuto ANDO **Yusuke INOUE**

National Institute of Technology, Oita College
1666 Maki,Oita, Oita 870-0152 Japan
aes2301@oita.kosen-ac.jp y-inoue@oita-ct.ac.jp

Abstract— In this paper, we implemented a stand-alone DPU-based facial expression recognition system on SoC FPGA. In conventional FPGA-based systems, the Haar Cascade detector is run on the CPU for face detection due to FPGA resource limitations. We offload face detection and facial expression recognition by DNN to DPU, a CNN accelerator on FPGA. The same DPU was used to implement the facial expression recognition system, which enabled efficient use of FPGA resources while minimizing the size of the circuitry.

I. Introduction

Facial expressions are an effective nonverbal means of conveying emotion and intention in human communication[1]. Furthermore, They are effective for communication between humans and computers and have been applied to pet robots and medical robots as an interaction system[2, 3]. For a robot to understand emotions using human facial expressions, it is necessary to perform classification processing of facial expressions, which is called facial expression recognition technology. The technology has been worked extensively due to its practical importance, and methods using local binary patterns and support vector machines have been proposed as machine learning methods[4, 5]. In recent years, Convolutional Neural Network(CNN)-based methods[6, 7] have become the mainstream for facial expression recognition and can achieve more accurate recognition than conventional methods.

Most of the existing facial expression recognition system approaches assume that the face is captured from a frontal angle to the camera. However, in the real world, the head angles can change to various orientations, and the face is often oblique or sideways to the camera. Furthermore, lighting conditions are not constant and may vary in brightness. Therefore, to implement a highly accurate facial expression recognition system in the real world, it is necessary to perform more acceptable face detection considering these issues. Conventional methods such as the Haar Cascade detector perform poorly in real-world detection due to the limitations of the head pose from which faces can be detected.

Compared to conventional methods, Deep Neural Network(DNN)-based face detection is robust to these environmental variables in the real world. Therefore, DNN is effective for both face detection and facial expression recognition, and the DNN-based system is capable of highly accurate identification. The DNN-based system generally requires a processing unit with high computing performance. However, when a facial expression recognition system is implemented in a robot, a low power consumption processing unit is required for long operation. Due to the low computing performance of commonly used embedded CPUs, the inference of DNN requires a huge amount of processing time. Therefore, it is difficult to implement the system as a practical system. An alternative to the embedded CPU implementation is the FPGA implementation with dedicated hardware. FPGA is a device that can sufficiently accelerate the inference using DNN on low power. In this work, the inference of DNN is executed on FPGA and the latency is reduced by using DPU(Deep learning Processor Unit), a CNN accelerator provided by Xilinx. As far as we know, there appears no reported implementation of a stand-alone facial expression recognition system that offloads the inference of two DNNs, one for face detection and the other for facial expression recognition, onto an FPGA.

In this paper, we implemented a DNN-based stand-alone facial expression recognition system on SoC FPGA. The main contributions of this work are as follows.

- Face detection using the DPU quantized YOLOv2 tiny model achieved higher accuracy and faster processing speed than the Haar Cascade detector in the previous work.
- As a stand-alone system, we were able to implement face detection and facial expression recognition using DNN without increasing the size of the circuit.
- The DPU-based facial expression recognition system with the smallest circuit size achieved approximately

12.5 fps for the camera input. The maximum size DPU achieves approximately 50.5 fps, and the system is flexible enough to be implemented according to the intended use.

The structure of this paper is as follows. In Section II, as the previous work, we present an implementation of a facial expression recognition system using the Haar Cascade detector on a SoC FPGA. In Section III, we describe our DPU-based facial expression recognition system on SoC FPGA. Section IV summarizes the experiments and results of our system. Section V discusses our system and Section VI concludes our work.

II. Related Work

Several works have been reported on the implementation of facial expression recognition on FPGA. However, most of the works have implemented only facial expression recognition, since the approach is based on the assumption that face images can be properly acquired by face detection[8, 9]. In contrast, Vinh et al. implemented a stand-alone system that executes facial expression recognition on the detected faces after performing face detection on a SoC FPGA[10]. In the facial expression recognition system proposed by Vinh et al., face detection is executed on an embedded CPU, while facial expression recognition is executed on an FPGA. For face detection, OpenCV's Haar Cascade detector[11] is used. The Haar Cascade detector can be executed on an embedded CPU on low computing power and is faster than DNN-based face detection. On the other hand, CNN-based methods are used for facial expression recognition.

The facial expression recognition system proposed by Vinh et al. uses the Haar Cascade detector for face detection, which has low detection accuracy. This is because the Haar Cascade detector can only detect frontal faces and cannot detect oblique or sideways faces. Therefore, in the real world, proper face detection becomes impossible as the head posture changes. One approach to this problem is to use a DNN that is more robust to real-world than the Haar Cascade detector. As in the previous work, in the case of SoC FPGA, DNN-based face detection is executed either on the CPU or on the FPGA. When executed on a CPU, a very lightweight DNN is required due to performance limitations. However, on an embedded CPU, that kind of DNN that meets this requirement may have low recognition accuracy. On the other hand, When executed an FPGA, these issues can be solved by using a CNN accelerator to execute the inference of the DNN.

Therefore, this work implements a system that executes face detection and facial expression recognition in a time-division manner using the same DPU(CNN accelerator). DPU is a general-purpose CNN accelerator from Xilinx that can execute several different CNN inferences on the same DPU. Using a DPU with this feature, a facial expression recognition system was implemented without increasing the size of the circuitry.

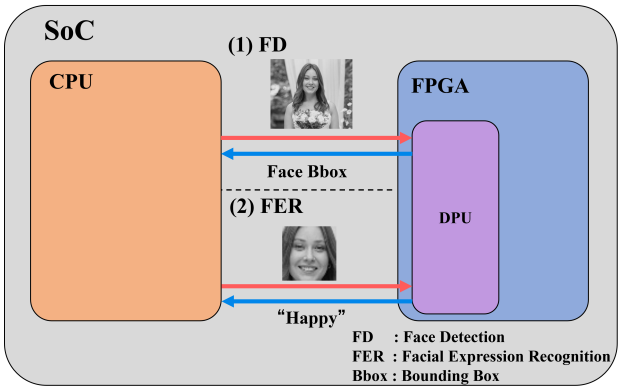

Fig. 1. Hardware configuration of the facial expression recognition system

III. Facial Expression Recognition System on SoC FPGA

A. System Configuration

The processing of this system is divided into two steps, consisting of (1)a face detection step and (2)a facial expression recognition step. In the face detection step, YOLOv2 tiny is used to detect faces. In the facial expression recognition step, a CNN-based inference model is used to identify facial expressions. The configuration of this system is shown in the Fig. 1. The inference in the two steps is offloaded respectively to the same DPU executed in the FPGA. DPU is a CNN accelerator provided by Xilinx and is responsible for speeding up the CNN inference. DPU support multiple architecture sizes with different computing performance, allowing users to select an architecture for each application. This system has the smallest circuit size, DPU: B512 (512 operations are run per clock cycle. Here in after as B512). The system incorporates a single B512 in the FPGA section, and controls two inference in a time-division manner.

We use DNN models for face detection and facial expression recognition. The face detection model in our work used is YOLOv2 tiny[12]. YOLOv2 tiny is a lightweight model, allowing for fast detection even on a DPU with a small circuit size. The FDDB dataset[13] was used to train this model. This dataset consists of 2,845 images of various resolutions, annotated with 5,171 faces. The facial expression recognition model was created based on the network of models proposed by Guarniz et al[14]. FER-2013[15] was used to train the model. The dataset consists of a total of 35,887 grayscale images of 48×48 pixels, with seven facial expression labels(angry, disgust, fear, happy, sad, surprised, and neutral).

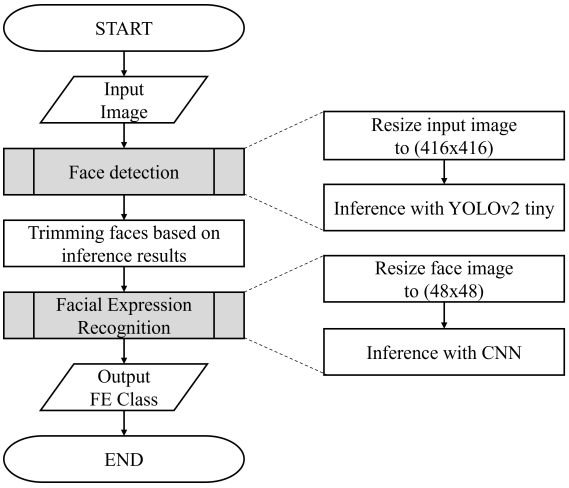

Fig. 2. Processing flow

B. Processing Workflow

The flowchart of the facial expression recognition system is shown in Fig. 2. In the processing flow, preprocessing such as image resizing and cropping is run on the CPU, while face image detection and facial expression recognition are run on the DPU. First, the input image is acquired and resized to fit the input size of YOLOv2 tiny. Successively, face detection is run on a YOLOv2 tiny. The face is cropped based on the facial coordinates obtained in the face detection step. Finally, the facial expression recognition step runs inference using the DNN model and outputs the classification results.

C. Offloading of Inference to DPU

In order to offload inference by DNN to DPU, it is necessary to transform the model using Vitis AI's tools. The DNN models were converted using AI Quantizer and AI Compiler contained in the tools. AI Quantizer is a tool that quantizes a 32 bit floating-point model into an 8 bit fixed-integer model, maintaining as high accuracy as possible. Quantization of the model can optimize memory usage and reduce the number of hardware operations. AI Compiler is a tool that converts a model quantized to an 8 bit integer by AI Quantizer into a model that can be run on DPU. By quantizing and compiling the model, the inference of the model can be run on a DPU, allowing for fast inference on low power consumption.

IV. Experiments and Results

This section evaluates the performance of this system by describing the experimental environment and its results.

A. Experimental Environment

The development environment for the DNN model was Python 3.7.12. We used Darknet[16] to train the face detection model and Keras 2.8.0 and Tensorflow 2.8.0 to train the facial expression recognition model. Xilinx Vivado 2021.1 was used to create the hardware design incorporating the DPU. A facial expression recognition system will be implemented on a Xilinx Zynq Ultrascale+ MPSoC with an architecture that integrates an ARM-based processor and Xilinx's UltraScale+ FPGA. We implemented the system using Xilinx Kria KV260, an evaluation board equipped with this SoC. The CPU portion of the SoC is an ARM Cortex-A53 with an operating frequency of 1.3 GHz.

B. Inference for face detection

The purpose of this experiment is to confirm the effectiveness of offloading to DPU in this system. Face detection and facial expression recognition are evaluated in terms of latency and recognition accuracy. In this subsection, we compare the performance of face detection by DPU with other face detection methods. The AFW dataset[17] was used to evaluate the experiments for performance comparison. This dataset consists of 205 images of various resolutions, annotated with 473 faces. The images include faces, especially those angled from the side, as well as those with scale, illumination, and occlusion challenges.

The results of face detection using the AFW dataset are shown in Table I. The quantization model on DPU is compared to the non-quantization model on CPU and to the Haar Cascade detector used in the previous work[10] and to the commonly used MTCNN[18]. The input size of YOLOv2 tiny is 416×416 pixels, so the input image must be resized. On the other hand, other methods perform inference without resizing, hindering the comparison of latency for different input sizes. Therefore, a comparison is made by adding the resize time to the latency of YOLOv2 tiny.

The accuracy was 0.783 for the non-quantized YOLOv2 tiny model run on the CPU, the highest accuracy. In contrast, the accuracy of the quantized model run on the DPU was 0.777, which is lower but still acceptable than that of the run on the CPU. The latency of the quantization model run on DPU was 132 ms, the shortest of all these methods. Furthermore, the accuracy was improved by a factor of 1.46 and the latency was reduced by a factor of 6.08 compared to the Haar Cascade detector method used in the previous work. Therefore, we can say that the DPU-based quantization model face detection is the best among these methods in terms of detection accuracy and latency.

C. Inference for Facial Expression Recognition

The FER-2013 dataset[15] was used for the evaluation. We compare the recognition accuracy and latency between the non-quantized model run on the CPU, the quantized model run on the DPU, and the previous work.

TABLE I
Accuracy and Latency Results for face detection

Method		Accuracy (AP)	Latency [ms]
CPU	Haar Cascade (Previous work)	0.531	802
	MTCNN	0.741	8,004
	YOLOv2 tiny	0.783	989
DPU	**YOLOv2 tiny (Our work)**	**0.777**	**132**

TABLE II
Accuracy and Latency Results for Expression Recognition

Method		Accuracy[%]	Latency[ms]
CPU	CNN:Non-quantized	70.6	349
FPGA	CNN(Previous work)	66	6.36
	CNN:quantized (Our work)	**67.4**	**7.34**

Table II shows the accuracy and latency of facial expression recognition by each DNN. The accuracy of the non-quantized facial expression recognition model was 70.6 %, and that of the quantized facial expression recognition model was 67.4 %. As for the latency required for each image, the non-quantized model took 349 ms, while the quantized model took 7.34 ms. The model was about 47 times faster by quantizing the model and running with the DPU. The DNN model in the previous work, which was run on an FPGA, had the accuracy of 66 % and the latency of 6.36 ms. Although our system outperformed the previous work in facial expression recognition the accuracy, the latency was longer than that of the previous work.

V. Discussions

A. *Comparison of hardware configuration with the previous work*

To verify the effectiveness of the hardware configuration with the same DPU in our system, we first compare the performance and FPGA resource usage of our system and that of the previous work[10]. The consumption of each FPGA resource is shown in Table III. In the previous work, Intel's SoC Cyclone V was used, and the logic unit is ALMs. On the other hand, the logic unit of Xilinx's Kria KV260 used in this system is LUTs, so the comparison by the number of resources consumed is for reference only.

Comparing the hardware resource usage, the previous work consumed 22,465 ALMs and 112 DSPs. In contrast, our system consumed 27,023 LUTs and 118 DSPs.

TABLE III
Comparison of FPGA resources consumed with the previous work

Method	ALMs or LUTs *	DSPs
Previous work	22,465	112
Our work	**27,023**	**118**

(*The previous work consists of ALMs because the board is Intel boards.)

The previous work consumed less FPGA resources. Although our system requires more circuitry than the previous work, it is possible to run facial expression recognition and face detection inference on the same DPU. Therefore, face detection with DNN is run on a DPU, which is superior to the previous work that runs face detection with the Haar Cascade detector on a CPU in terms of accuracy and latency.

Also, we compare the hardware configuration of our system with that of the previous work. In the hardware configuration of the previous work, we consider the case where face detection is replaced by a DNN, as in our system. In this case, there are two options run the inference on the CPU or implement a new CNN accelerator on an FPGA. Running on a CPU improves the accuracy of face detection, but incurs a long latency. Therefore, it is difficult to apply it for practical purposes. On the other hand, if a new dedicated CNN accelerator is implemented and run on an FPGA, detection accuracy and latency are not a problem, but the circuit size becomes larger. In contrast, our system uses the same DPU(CNN accelerator) for face detection and facial expression recognition, so these problems do not occur. In facial expression recognition, the latency is not as short as the latency in the previous work, but it is kept within an acceptable margin of error. Therefore, the hardware configuration with the same DPU in this system is more effective than the hardware configuration in the previous work.

B. *DPU size Analysis*

This system was evaluated with an FPGA incorporating B512. We clarify which size is the most effective in terms of latency and FPGA resource usage when a DPU of other sizes is used. A comparison is made between the FPGA incorporating B512 in this system and a DPU of other sizes. Table IV shows the FPGA resources used in the case DPUs are incorporated and the latency by DPU size for face detection and facial expression recognition. The DNN models for face detection and facial expression recognition are quantized models used in the system, respectively, and compiled by AI Compiler for each DPU size. The AFW dataset contains a wide range of resolution sizes, resulting in variations in the processing time for face detection, including resizing each image.

Comparing the FPGA resources that incorporate the

TABLE IV
FPGA RESOURCES USED AND PROCESSING PERFORMANCE BY DPU SIZE

Size	FPGA resource			Latency [ms]	
	LUTs	DSPs	BRAMs	Face Detection	FER
				Webcam(640×480)	FER-2013
B512	**27,023**	**118**	**12**	**73.6**	**7.34**
B1024	34,593	230	44	45.9	4.42
B2304	41,861	438	60.5	26.2	2.41
B4096	51,561	710	82.5	17.9	1.60

DPU, the B512 used in this system has the lowest computational performance since the circuit size is quite small. In case the size of the DPU is increased, the DSPs and BRAMs usage increases, in particular. Comparing B4096 with B512, the LUTs have increased by a factor of 1.9, the DSPs by a factor of 6.0, and the BRAMs by a factor of 6.9, indicating that B4096 consumes a very large amount of FPGA resources. On the other hand, latency was reduced by only a factor of 4.1 for face detection and a factor of 4.0 for facial expression recognition. Similar patterns were observed for DPU of other sizes, indicating that the amount of latency reduction was small in relation to the increase in circuit size. Therefore, B512 is the most effective in terms of latency and FPGA resource usage.

If a facial expression recognition system with real-time performance is required, it can be expected to be handled by increasing the size of the DPU, although the circuit size will increase. The latency of facial expression recognition is shorter than in the previous work when using a DPU with a size larger than B1024. The facial expression recognition system using B512 achieves approximately 12.5 fps for a camera input with a resolution size of 640×480. Furthermore, when B4096 is used, the DPU size is approximately 50.5 fps, indicating that a larger DPU size is required to achieve a system with real-time performance. Therefore, this system can be evaluated as having the flexibility to be implemented according to the intended use.

VI. CONCLUSIONS

In this paper, we implemented a stand-alone DPU-based facial expression recognition system on SoC FPGA. We offloaded the inference of two DNNs for face detection and facial expression recognition onto an FPGA. As a result, face detection accuracy was improved by a factor of 1.46, and the latency was reduced by a factor of 6.08 compared to the conventional Haar Cascade detector. On the other hand, facial expression recognition achieved the accuracy of 67.4 % and the latency per image was 7.34 ms. The accuracy of our system exceeded that of the previous work, and the latency was kept within an acceptable margin compared to the previous work. Although the circuit size was slightly larger than in the previous work, the same DPU can be used to perform facial expression recognition and face detection inference. Therefore, we concluded that the hardware configuration using the same DPU achieves better results than the previous work while minimizing the size of the circuitry.

Future work includes improving the accuracy of face detection by revising the architecture of the DNN for face detection. In addition, multiple DPUs with low computing performance will be parallelized to improve accuracy while reducing the circuit size.

REFERENCES

[1] Y.-I. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 2, pp. 97–115, 2001.

[2] D. O. Melinte and L. Vladareanu. Facial expressions recognition for human–robot interaction using deep convolutional neural networks with rectified adam optimizer. *Sensors*, Vol. 20, No. 8, p. 2393, 2020.

[3] O. Arriaga, M. Valdenegro-Toro, and P. Plöger. Real-time convolutional neural networks for emotion and gender classification. *arXiv preprint arXiv:1710.07557*, 2017.

[4] C. Shan, S. Gong, and P. W. McOwan. Robust facial expression recognition using local binary patterns. In *IEEE International Conference on Image Processing 2005*, Vol. 2, pp. II–370. IEEE, 2005.

[5] D. Ghimire, S. Jeong, J. Lee, and S. H. Park. Facial expression recognition based on local region specific features and support vector machines. *Multimedia Tools and Applications*, Vol. 76, pp. 7803–7821, 2017.

[6] J. Li, K. Jin, D. Zhou, N. Kubota, and Z. Ju. Attention mechanism-based CNN for facial expression recognition. *Neurocomputing*, Vol. 411, pp. 340–350, 2020.

[7] J. Shao and Y. Qian. Three convolutional neural network models for facial expression recognition in the wild. *Neurocomputing*, Vol. 355, pp. 82–92, 2019.

[8] J. Kim, J.-K. Kang, and Y. Kim. A resource efficient integer-arithmetic-only FPGA-based CNN accelerator for real-time facial emotion recognition. *IEEE Access*, Vol. 9, pp. 104367–104381, 2021.

[9] H. Phan-Xuan, T. Le-Tien, and S. Nguyen-Tan. FPGA platform applied for facial expression recognition system using convolutional neural networks. *Procedia computer science*, Vol. 151, pp. 651–658, 2019.

[10] P. T. Vinh and T. Q. Vinh. Facial expression recognition system on SoC FPGA. In *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, pp. 1–4. IEEE, 2019.

[11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conf Comput Vis Pattern Recognit*, Vol. 1, pp. I–511, 02 2001.

[12] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[13] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[14] Facial-expression-recognition-2018. `https://github.com/kckeiks/Facial-Expression-Recognition-2018/tree/master`. (Accessed on 09/07/2023).

[15] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. arXiv 1307.0414 stat.ML, 2013.

[16] J. Redmon. Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`, 2013–2016. (Accessed on 11/04/2023).

[17] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2879–2886, 2012.

[18] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, Vol. 23, No. 10, pp. 1499–1503, oct 2016.