ICIC Express Letters
Volume **19**, Number **4**, April **2025**

 ISSN 1881-803X
pp. **419–426**

DPU-BASED HARDWARE IMPLEMENTATION FOR REAL-TIME FACIAL EXPRESSION RECOGNITION SYSTEM

Takuto Ando[1] and Yusuke Inoue[2,*]

[1]Electrical, Electronics Information Engineering Major, Advanced Course
[2]Department of Information Engineering
National Institute of Technology, Oita College
1666 Maki, Oita City, Oita 870-0152, Japan
aes2301@oita.kosen-ac.jp; *Corresponding author: y-inoue@oita-ct.ac.jp

Received May 2024; accepted August 2024

Abstract. *In this paper, we implemented a standalone DPU-based facial expression recognition system on SoC FPGA. The system consists of a face detection step and a facial expression recognition step. In conventional FPGA-based facial expression recognition systems, the Haar Cascade detector is run on the CPU due to FPGA (Field Programmable Gate Array) resource limitations in the face detection step. However, the Haar Cascade detector is less accurate than DNN (Deep Neutral Network)-based face detection for images of profile faces and images with changing lighting conditions. On the other hand, face detection using a DNN such as YOLO requires a long latency when executed on a CPU with low computing performance. Therefore, we offload face detection and facial expression recognition by DNN to DPU, a CNN accelerator on FPGA, to speed up the processing. In this work, we combined face detection with DenseBox and CNN-based facial expression recognition on the same DPU. The same DPU was used to implement the facial expression recognition system, which enabled efficient use of FPGA resources while keeping the size of the circuitry.*
Keywords: DNN, FPGA, DPU, Standalone system, Facial expression recognition

1. **Introduction.** Facial expression recognition in humans is one of the most important and challenging tasks in human-robot communication. Facial expressions serve as an essential non-verbal means for robots to comprehend human emotions and intentions [1]. This capability has been utilized in various interaction systems such as pet robots and medical robots [2, 3]. Machine learning-based methods such as local binary patterns and support vector machines have been proposed for this purpose [4, 5].

Most facial expression recognition systems based on image processing consist of (1) face detection and (2) facial expression recognition. In the first stage, face detection algorithms identify the face region within the given image, which is then used as input for the subsequent stage. The second stage involves classifying the labels of facial expressions based on the detected facial regions. In recent years, methods using DNN (Deep Neural Network) emerged as the leading approach for these processes, achieving high accuracy. On the other hand, they require a huge amount of power for inference. To address this issue, for applications in embedded systems like robotics, a low-power processor is needed for battery operation.

FPGA (Field Programmable Gate Array) is devices that can accelerate inference processing using DNN and are capable of low-power operations. Facial expression recognition systems using FPGA have already been proposed. Vinh and Vinh implemented a standalone facial expression recognition system running on SoC FPGA [6]. In this system, face detection is run on CPU, and facial expression recognition is run on FPGA. The implementation of DNN-based inference on FPGA contributes to achieving high accuracy

DOI: 10.24507/icicel.19.04.419

in facial expression recognition. On the other hand, due to FPGA resource limitations, the OpenCV Haar Cascade detector [7] is used for face detection. This Haar Cascade detector can run on an embedded CPU with low computing performance and is faster than DNN-based face detection. However, it can only detect faces in front of the camera, and the detection accuracy is low when the face is oblique or sideways. Furthermore, the performance of this detector is also affected when the lighting conditions are not consistent.

In contrast, DNN-based face detection is more robust to these challenges and can detect faces with higher accuracy. However, to perform different inferences on FPGA, each inference requires its accelerator. When increasing accelerators on FPGA, the circuit size increases due to the increased consumption of FPGA resources. Therefore, multiple inference processes are executed by the same accelerator. This approach is expected to achieve high processing performance while keeping the circuit area. This work proposes a facial expression recognition system based on DPU (Deep Learning Processor Unit), a general-purpose CNN accelerator capable of performing multiple inference processes.

The contributions of this paper can be summarized as follows. First, the system was implemented as a standalone system capable of performing both facial expression recognition and face detection on a single FPGA, achieving higher accuracy and throughput than the previous work. The effectiveness of the proposed hardware configuration compared to the previous work has been demonstrated. Second, measurements of the power consumption and throughput of the entire system showed an improvement of approximately 1.79 times compared to the hardware configuration in the previous work, confirming its effectiveness in terms of power consumption and throughput. The rest of this paper is organized as follows. Section 2 presents the facial expression recognition system using DPU. Section 3 summarizes the experiments and results of our system. Finally, Section 4 concludes our work.

2. Proposed Method.

2.1. System configuration.

The processing of this system is divided into two steps consisting of (1) the face detection and (2) the facial expression recognition, as depicted in Figure 1. In the face detection step, the camera image is used as input, and face detection is run using the DenseBox, a DNN model. Next, in the facial expression recognition step, the CNN model is used to identify the facial expressions on the detected face image. The inference processes performed in these steps are offloaded to DPU running on the FPGA. This DPU is a CNN accelerator provided by AMD and is responsible for accelerating the CNN inference process. DPU supports multiple architectures with different

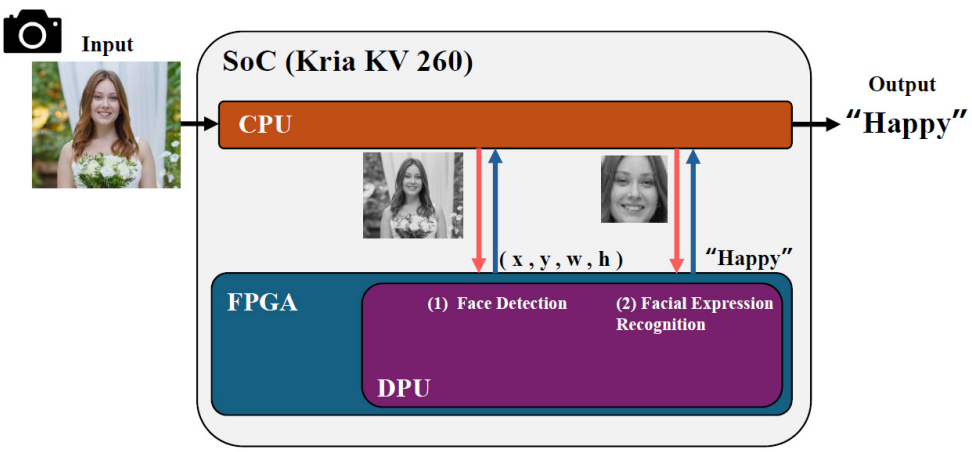

FIGURE 1. System configuration

computing performance and can be selected for different applications. The computational performance of the convolutional architecture is determined by the parallelism in three dimensions: Pixel Parallelism (PP), Input Channel Parallelism (ICP), and Output Channel Parallelism (OCP). PP represents the number of pixels generated in parallel per cycle, while ICP and OCP represent the number of channels processed per cycle at input and output, respectively. The elements used in the computation consist of one pixel from each channel. Also, since multiplication and accumulation are run in one clock cycle, they are counted as two separate operations. Therefore, the operations per clock cycle is calculated using Equation (1) provided.

$$\text{Operations per clock cycle} = PP \times ICP \times OCP \times 2 \tag{1}$$

For instance, the minimum size B512 has $PP = 4$, $ICP = 8$, and $OCP = 8$, resulting in 512 operations per clock cycle. In addition, when high arithmetic performance is incorporated into the circuit, the FPGA resource consumption increases. Therefore, the system incorporates one DPU: B512 on FPGA, and controls two inference processes in a time-division manner.

Next, the DNN for face detection and facial expression recognition is described. The face detection model used is the DenseBox model [8] provided by AMD. The inference with DenseBox is so lightweight that it enables fast detection even when run on DPU with low computational performance. The WIDER FACE dataset [9] was used to train this model. This dataset consists of 32,203 images, annotated with 393,703 faces with large variations in scale, pose, and occlusion.

On the other hand, the facial expression recognition model was created concerning the network [10] created by Guarniz. The network structure of the facial expression recognition model is shown in Figure 2. Seven different classes (Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral) are identified from the input 48×48 px grayscale images. First, feature extraction is performed by repeating the block consisting of a convolutional layer, a batch normalization layer, an activation layer, a pooling layer, and a dropout layer four times. Then, the feature map is transformed into a one-dimensional vector by the flatten layer and classified into seven facial expression classes by a full concatenation layer. FER-2013 [11] was used to train this model. As shown in Figure 3, the dataset consists of a total of 35,887 grayscale images of 48×48 px, with seven facial expression labels assigned.

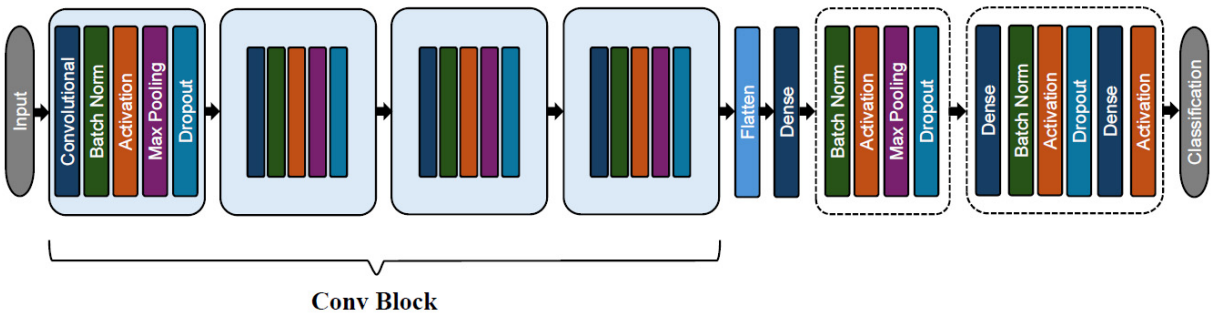

FIGURE 2. Network configuration of CNN models for facial expression recognition

The processing flow is then described in detail. In the overall processing, pre-processing such as image resizing, cropping, and grayscale conversion is run on the CPU. In contrast, DNN-based inference, such as face detection and facial expression recognition, is run on DPU. First, the input image is acquired and resized to match the DenseBox input size (640×360). Next, face detection is performed using the DenseBox. The face is then cropped based on the face coordinates obtained from the face detection step. Finally,

FIGURE 3. Sample images of the FER-2013 dataset

facial expression recognition is run using a CNN-based model and the classification results are output.

2.2. **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. Vitis AI is a DNN framework provided by AMD. It is a development environment that facilitates the conversion of built models into a format that is compatible with DPU. 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 leveraging these tools, DNN models can be run on DPU through the quantization and compilation processes.

3. Experiments and Results.

3.1. **Experimental environment.** The development environment for the DNN model was Python 3.7.12. We used Darknet [12] 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.

3.2. **Evaluation of inference for face detection.** The purpose of this experiment is to evaluate the face detection model run on DPU and to compare it with the previous work. Face detection is evaluated in terms of recognition accuracy and latency. We compare the performance of face detection by DPU with our method to the previous work, using the AFW dataset [13] for evaluation. 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.

A comparison is made between the model run on DPU and the Haar Cascade detector used in the previous work. The input size of the DenseBox is 416×416 px, which requires the input image to be resized. In contrast, the Haar Cascade detector performs inference without resizing. Consequently, the latency cannot be impartially compared due to the differing input sizes. Therefore, a fair latency comparison is made by adding the resizing time to the latency of the DenseBox.

The accuracy of the DenseBox run on DPU was 0.917. Compared to the method with the Haar Cascade detector in the previous work [6], the accuracy was improved by approximately 1.73 times. The latency of the model run on DPU was 42.10 ms, which is about 18.95 times shorter than that of the previous work. Therefore, it has been verified that the face detection of the DNN model with DPU is superior to the previous work in terms of recognition accuracy and latency.

Table 1. Accuracy and latency results for face detection

Method		Average precision	Latency [ms]
CPU	Haar Cascade (Previous work)	0.531	798
DPU	**DenseBox (Our work)**	**0.917**	**42.10**

3.3. **Evaluation of inference for facial expression recognition.** The purpose of this experiment is to evaluate the facial expression recognition model run on DPU and to compare it with the previous work. Facial expression recognition is evaluated in terms of recognition accuracy and latency. The FER-2013 dataset [11] was used for the evaluation. The CNN model run on DPU is compared with the CNN models of the previous work in terms of recognition accuracy and latency. The accuracy and latency of expression recognition by each DNN are shown in Table 2.

Table 2. Accuracy and latency results for facial expression recognition

Method	Accuracy [%]	Latency [ms]
CNN (Previous work)	66	6.36
CNN (Our work)	**67.4**	**7.34**

The facial expression recognition model run on DPU had an accuracy of 67.4% and a latency of 7.34 ms. The confusion matrix in the inference results for this model is shown in Figure 4. The true label refers to the correct answer class, while the prediction

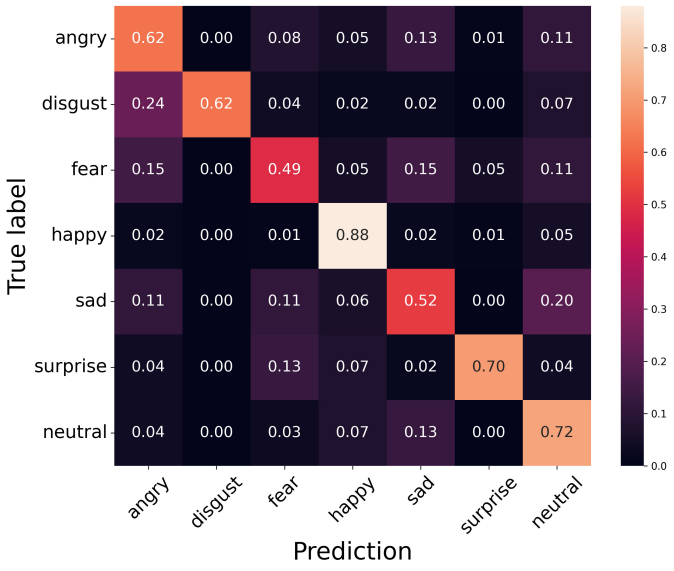

Figure 4. Normalized confusion matrix

indicates the label of the inference result. The correctness rate of each class shows that the recognition accuracy of the fear and sad classes is low, while the accuracy of the other classes is high. The model in the previous work performed with an accuracy of 66% and a latency of 6.36 ms on the FER-2013 test dataset. The results show that our system outperforms the previous work in terms of accuracy, but the latency is longer than that of the previous work.

3.4. **Evaluation of the overall system.** The purpose of this experiment is to validate the effectiveness of the hardware configuration of this system using B512. DPU of different sizes from previous work is evaluated on FPGA resource consumption, system power consumption, and throughput as indicators. Table 3 shows the results of these comparisons. However, as it is not possible to perfectly simulate the implementation environment of the previous work, only the hardware configuration was simulated. Specifically, we implemented a system with a configuration in which face detection by Haar Cascade detector on CPU and facial expression recognition by CNN on DPU and compared them. The power consumption is measured by connecting the KETOTEK KTEM02 digital energy meter to the power socket on the KV260 board. The difference between the power consumption of the entire board in an idle state and with the system running was measured. The system power consumption was calculated from these measurements. The throughput of the entire system was measured, not only for face detection and facial expression recognition but also for processes other than these.

Table 3. Power consumption and throughput in different models of system face detection

Method		FPGA resources		Power [W]	Throughput [FPS]
		ALMs or LUTs *	DSPs		
DPU	B4096	51,561	710	2.8	26.25
	B2304	41,861	438	2.4	24.7
	B1024	34,593	230	2.2	21.6
	B512	**27,023**	**118**	**1.7**	**19.68**
CPU	Haar Cascade (Previous work)	22,465	112	1.8	11.67

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

First, FPGA resource consumption and throughput are evaluated. It can be seen that as the size of DPU increases, the processing performance improves, but the FPGA resource consumption increases. Also, the previous work consumed 22,465 ALMs and 112 DSPs. In contrast, our system consumed 27,023 LUTs and 118 DSPs. Comparing the circuit sizes, it was found that the previous work consumed fewer FPGA resources. Although the circuit scale of our system is larger than that of the previous work, it is possible to run facial expression recognition and face detection inference on the same DPU. Therefore, the DNN-based face detection is run on DPU, which is superior to the previous work that runs the Haar Cascade detector-based face detection on CPU in terms of recognition accuracy and latency.

Next, power consumption and throughput are evaluated. The smaller the DPU size, the lower the throughput, and the lower the power consumption. Compared to the hardware configuration in the previous work, it can be seen that the throughput is improved by approximately 1.69 times, while the power consumption is almost the same. The throughput per power consumption is shown in Figure 5. The power consumption per throughput of our system with B512 was 11.58 fps/W. This is the best result among the four different DPU sizes and the previous work. Compared to the previous work, this is an improvement

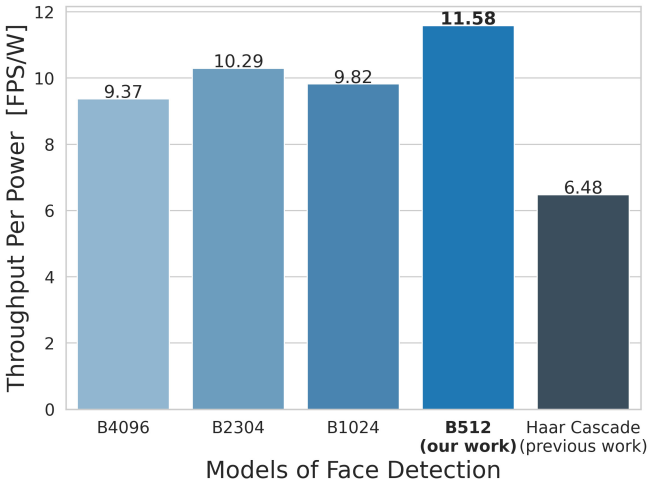

FIGURE 5. Throughput per power for different models

of approximately 1.79 times, confirming the effectiveness of the system in terms of power consumption and throughput.

4. **Conclusions.** In this paper, we implemented a standalone 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 using DenseBox has successfully improved the accuracy by approximately 1.73 times and reduced the latency by approximately 18.95 times compared to the conventional method using the Haar Cascade detector. On the other hand, the system has achieved an accuracy of 67.4% in facial expression recognition and a latency of 7.34 ms per image. Furthermore, the throughput per power of the entire system is approximately 1.79 times better than in the previous work, confirming the effectiveness of the system in terms of power consumption and throughput. 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 keeping the size of the circuitry. Future work includes improving the throughput of the entire system by improving the speed of face detection. Specifically, the processing time can be reduced by multi-threading the inference task and its task scheduling.

REFERENCES

[1] Y.-I. Tian, T. Kanade and J. F. 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, 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, *IEEE International Conference on Image Processing*, Genova, Italy, 2005.

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

[6] P. T. Vinh and T. Q. Vinh, Facial expression recognition system on SoC FPGA, *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, Ho Chi Minh City, Vietnam, pp.1-4, 2019.

[7] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Kauai, HI, USA, 2001.

[8] *Vitis AI Library User Guide G1354 (V3.5)*, https://docs.amd.com/viewer/book-attachment/KR753m2y6vGxH3r37gIq3A/cBdlK8Cf7joC9NuWU_MHjg, 2023, Accessed on 04/07/2024.

[9] S. Yang, P. Luo, C. C. Loy and X. Tang, WIDER FACE: A face detection benchmark, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp.5525-5533, 2016.

[10] *Facial-Expression-Recognition-2018*, https://github.com/kckeiks/Facial-Expression-Recognition-2018/tree/master, Accessed on 09/07/2023.

[11] I. J. Goodfellow, D. Erhan, P. L. Carrier et al., Challenges in representation learning: A report on three machine learning contests, *arXiv Preprint*, arXiv 1307.0414, 2013.

[12] *Darknet: Open Source Neural Networks in C*, http://pjreddie.com/darknet/, Accessed on 11/04/2023.

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