





# Hardware -Oriented Algorithm for Human Detection using GMM-MRCoHOG features

Ryogo Takemoto<sup>1</sup><sup>a</sup>, Yuya Nagamine<sup>1</sup>, Kazuki Yoshihiro<sup>1</sup>, Masatoshi Shibata<sup>2</sup>, Hideo Yamada<sup>2</sup>, Yuichiro Tanaka<sup>3</sup><sup>b</sup>, Shuichi Enokida<sup>4</sup><sup>c</sup>, and Hakaru Tamukoh<sup>1,3</sup><sup>d</sup>

<sup>1</sup>*Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0196, Japan*

<sup>2</sup>*AISIN CORPORATION, 2-1 Asahi-machi, Kariya, Aichi, 448-8650, Japan*

<sup>3</sup>*Research Center for Neuromorphic AI Hardware, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0196, Japan*

<sup>4</sup>*Department of Artificial Intelligence, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka, 820-8502, Japan*  
*takemoto.ryogo139@mail.kyutech.jp, nagamine.yuya802@mail.kyutech.jp, yoshihiro.kazuki691@mail.kyutech.jp, masatoshi.shibata@aisin.co.jp, hideo.yamada02@aisin.co.jp, tanaka-yuichiro@brain.kyutech.ac.jp, enokida@ics.kyutech.ac.jp, tamukoh@brain.kyutech.ac.jp*

**Keywords:** Image Processing, Human Detection, HOG, MRCoHOG, GMM-MRCoHOG, FPGA

**Abstract:** In this research, we focus on Gaussian mixture model-multiresolution co-occurrence histograms of oriented gradients (GMM-MRCoHOG) features using luminance gradients in images and propose a hardware-oriented algorithm of GMM-MRCoHOG to implement it on a field programmable gate array (FPGA). The proposed method simplifies the calculation of luminance gradients, which is a high-cost operation in the conventional algorithm, by using lookup tables to reduce the circuit size. We also designed a human-detection digital architecture of the proposed algorithm for FPGA implementation using high-level synthesis. The verification results showed that the processing speed of the proposed architecture was approximately 123 times faster than that of the FPGA implementation of VGG-16.


## 1 INTRODUCTION


The demand for home service robots and self-driving cars has been increasing in response to the recent acceleration in the aging population and decline in birthrate. Because these robots and cars with artificial intelligence are expected to operate near humans, high-precision and high-speed human detection functions are required from the viewpoint of safety. However, the more accurate the human detection, the more complex is the computation and the longer the computation time. Parallelization is one of the effective solutions to accelerate the computation.


A typical device for parallel processing is a graphics processing unit (GPU). However, GPUs are not suitable for embedded systems such as home service


robots and self-driving cars in terms of power consumption and heat exhaustion. Instead of software implementation on GPUs, hardware implementation, where a dedicated circuit with parallel architecture for some computation is designed, can achieve a low-power system with high-speed processing because the operation on the dedicated circuit can be more effective than that on GPUs. Therefore, we aim to design a dedicated circuit for human detection and implement it on a field-programmable gate array (FPGA). Because FPGAs have limited physical circuit resources, we need a hardware-oriented algorithm that reduces the number of complex operations in the original algorithm to efficiently utilize the limited resources.

For high-accuracy human detection, histograms of oriented gradients (HOG) features have been proposed (Dalal and Triggs, 2005) and used in multiple applications. This method extracts features of object shapes from luminance gradients in images, and represents the features as histograms of the gradients. For higher-accuracy and smaller-memory re-

<sup>a</sup> <https://orcid.org/0000-0002-6795-0794>

<sup>b</sup> <https://orcid.org/0000-0001-6974-070X>

<sup>c</sup> <https://orcid.org/0000-0001-6309-3185>

<sup>d</sup> <https://orcid.org/0000-0002-3669-1371>

source implementation of human detection compared with HOG features, the Gaussian mixture model-multiresolution co-occurrence histograms of oriented gradients (GMM-MRCoHOG) features that approximate the conventional histogram-based state space with a mixed Gaussian distribution and optimize the feature space have been proposed (Higashi et al., 2018; Nagamine et al., 2019). However, the algorithm still requires a large number of complex operations that are not suitable for FPGA implementation.

In this study, we propose a hardware-oriented algorithm of GMM-MRCoHOG that simplifies the complex operation in the original algorithm, such as the calculation of luminance gradients by using a lookup table (LUT); we then design a dedicated circuit for human recognition integrating the hardware-oriented GMM-MRCoHOG with a binarized neural network (BNN) (Hubara et al., 2016), and implement it on an FPGA to achieve a high-accuracy, high-speed, and low-power system.

## 2 RELATED WORKS

MRCoHOG (Iwata and Enokida, 2014), a derivative work of HOG, extracts features by down-sampling an image in two steps; it represents the gradient co-occurrence between images of three resolutions as a two-dimensional co-occurrence histogram. Feature extraction methods using gradient histograms, such as HOG and MRCoHOG, require a manual determination of the optimal class width of the histogram to discretize the luminance gradients. This is difficult because the discretization error of the gradient information and the generalization ability of the features vary depending on the class width. Moreover, these methods require many memory resources to represent gradient histograms.

Conversely, GMM-MRCoHOG constructs an optimal state space by approximating the co-occurrence histogram with a mixed Gaussian distribution, as shown in Fig. 1 and performs feature extraction based on the state space. The approximation results in reducing the required memory resources for gradient histograms in the original algorithm because only a small number of memories is required to represent the mixed Gaussian distribution.

Figures 2 and 3 show the processing flow of GMM-MRCoHOG. First, the co-occurrences of the luminance gradient pairs (36 gradient directions for each axis in Fig. 2) of the positive and negative data of the training images are mapped to the feature space as continuous values, and each feature is approximated by a mixture Gaussian distribution. Then,

using the Jensen–Shannon (JS) information content (Michishita et al., 2018), only features that can effectively separate the positive and negative data are extracted from the respective mixed Gaussian distributions and approximated to a mixed Gaussian distribution using the EM algorithm (Dempster et al., 1977). The resulting mixed Gaussian distribution is then used as the feature space, and the responsibility (described as “resp” in Fig. 3) of each Gaussian distribution is calculated and used as the feature value. In GMM-MRCoHOG, the final number of feature dimensions is determined by the number of Gaussian distributions in 2D space, and not by the number of gradient quantization.

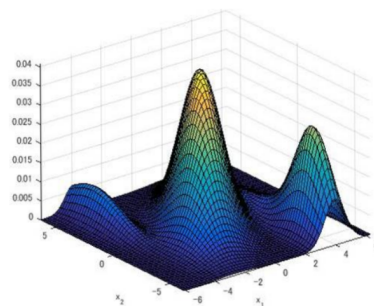


Figure 1: Sample of Gaussian Mixture Model

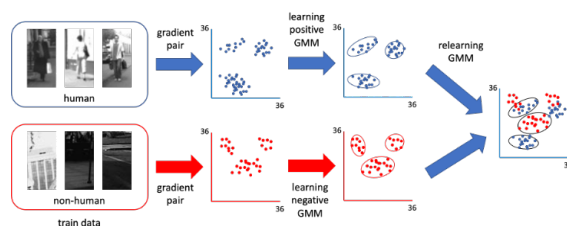


Figure 2: Training Process of State Space in GMM-MRCoHOG

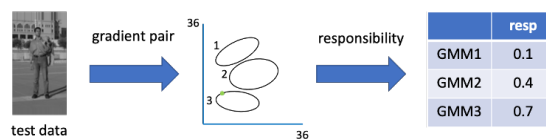


Figure 3: Feature Extraction Process in GMM-MRCoHOG

GMM-MRCoHOG has difficulties in hardware implementation because it includes an arctangent function for the luminance gradient angle decision and the responsibility calculation for the feature value decision, which are complex operations that require considerable circuit resources. Nagamine et al. proposed a hardware-oriented algorithm that approximates these calculations to reduce the circuit resources (Nagamine et al., 2019). The algorithm deter-

mines the luminance gradient angles by using a condition branch of the horizontal and vertical luminance gradients  $f_x$  and  $f_y$ . Figure 4 shows a first quadrant in the luminance gradient space of  $f_x$  and  $f_y$ , which is divided into several areas at intervals of 16 in Manhattan distance. The condition branch determines an angle by subtracting  $f_x$  and  $f_y$  according to the divided area; therefore, the angle decision does not require complex operations. For the feature value decision, the algorithm infers a responsibility from the distance between the input vector and each Gaussian distribution. The algorithm also approximates the Gaussian distribution width as a power of two and changes the Gaussian shape as a rectangle so that the computation can be represented by bit-shift operations and fuzzy inferences. Although the hardware-oriented algorithm reduces most circuit resources from the original algorithm, the condition branch for the angle calculation still requires many LUTs, which worsens the performance of the algorithm because of the imprecise angle approximation.

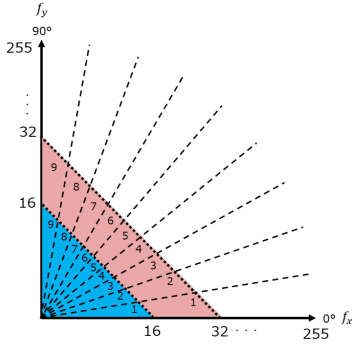


Figure 4: Condition Branch for Luminance Gradient Angle Decision in (Nagamine et al., 2019)

### 3 PROPOSED METHODS

To improve the method proposed by Nagamine et al., we propose a novel coarse angle calculation method using a fixed-point  $\tan\theta$  table. We then construct a hardware-oriented GMM-MRCoHOG-based human recognition circuit using the method for a high-speed and low-power human detection system.

#### 3.1 Coarse Angle Calculation Method Using Fixed-Point Tangent Table

In the GMM-MRCoHOG algorithm, the luminance gradient angle  $\theta$  is calculated as  $\theta = \tan^{-1}(f_y/f_x)$  and discretized in 36 directions. Here, assuming that

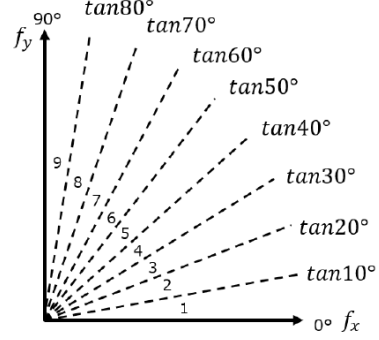


Figure 5: Overview of Discretized  $\tan\theta$

the angle  $\theta$  appears in the first quadrant of the luminance gradient space, we calculate  $\tan\theta$  from  $\tan 0^\circ$  to  $\tan 80^\circ$  in advance, as given by Eq. (1) and discretize it, as shown in Fig. 5.

$$\begin{aligned}
 & \text{if } \tan 0^\circ \leq \frac{f_y}{f_x} < \tan 10^\circ \\
 & \quad \text{direction} = 1(\theta : 0^\circ \sim 10^\circ) \\
 & \text{elif } \tan 10^\circ \leq \frac{f_y}{f_x} < \tan 20^\circ \\
 & \quad \text{direction} = 2(\theta : 10^\circ \sim 20^\circ) \\
 & \quad \vdots \\
 & \text{elif } \tan 80^\circ \leq \frac{f_y}{f_x} \\
 & \quad \text{direction} = 9(\theta : 80^\circ \sim 90^\circ)
 \end{aligned} \tag{1}$$

Then, we create a  $\tan\theta$  table representing the relationship between the discretized  $\tan\theta$ ,  $f_x$ , and  $f_y$ , which enables us to obtain rough angles of luminance gradients. By utilizing the symmetry of the trigonometric functions, the  $\tan\theta$  table can be applied to the second through fourth quadrants.

Additionally, we eliminate divisions that require most circuit resources in the conditional branch in the  $\tan\theta$  table. As  $f_x \geq 0$  and  $f_y \geq 0$ , we can replace Eq. (1) with Eq. (2), where no division is required.

$$\begin{aligned}
 & \text{if } f_x \times \tan 0^\circ \leq f_y < f_x \times \tan 10^\circ \\
 & \quad \text{direction} = 1(\theta : 0^\circ \sim 10^\circ) \\
 & \text{elif } f_x \times \tan 10^\circ \leq f_y < f_x \times \tan 20^\circ \\
 & \quad \text{direction} = 2(\theta : 10^\circ \sim 20^\circ) \\
 & \quad \vdots \\
 & \text{elif } f_x \times \tan 80^\circ \leq f_y \\
 & \quad \text{direction} = 9(\theta : 80^\circ \sim 90^\circ)
 \end{aligned} \tag{2}$$

The values in the  $\tan\theta$  table are then approximated with fixed-point numbers that enable faster computation and fewer circuit resource implementations compared with floating-point numbers.

### 3.2 Human Recognition Circuit Integrating Hardware-Oriented GMM-MRCoHOG and BNN

We designed a dedicated human recognition circuit using the proposed coarse angle calculation algorithm and the responsibility inference method proposed by Nagamine et al. (Nagamine et al., 2019), as shown in Fig. 6.

This circuit receives a  $32 \times 64$  pixels image as input and continuously transfers one pixel at a clock cycle from the top-left to the bottom-right pixel of the image to the image buffers. Here, we set the GMM-MRCoHOG extract features from three resolution images: the original size image, a 1/2-resized image, and a 1/4-resized image; therefore, we implemented three image buffers for these resolutions. Each of the buffers is a three-line buffer to calculate the luminance gradient from  $3 \times 3$  pixels in the image. The derivative filter blocks receive three lines of pixels and calculate the horizontal and vertical luminance differences. The angle calculation blocks calculate the angles of the luminance gradients, and the results are stored in the two-line buffers of the second stage. Then, the gradient co-occurrence is calculated, and the GMM-MRCoHOG feature is extracted. The obtained feature is fed into the BNN, which classifies the input image as human or not human. The synaptic weights and activation of the BNN are binarized such that the circuit requires small memory resources.

Here, the number of mixtures of the Gaussian distribution used in the GMM-MRCoHOG is 6. The BNN has three layers: input, hidden, and output layers, and the number of neurons in the hidden layer is 1.

## 4 EXPERIMENT

We verified the proposed coarse angle calculation method, implemented a human recognition circuit integrating the hardware-oriented GMM-MRCoHOG and the BNN using high-level synthesis, and estimated the processing speed and circuit size. The experimental environment is presented in Table 1.

### 4.1 Coarse Angle Calculation Method Using the Fixed-Point Tangent Table

In this experiment, we verified the proposed coarse angle calculation method with respect to circuit size, estimated the angle matching rate to true angles, processing speed of the circuit, and the approximation

effect on accuracy for human recognition tasks.

First, we verified the circuit sizes of the  $\tan\theta$  table when the integer part of the fixed-point numbers in the table was fixed to three bits, and the fraction part was varied from zero to seven bits. The target device was a Xilinx Zynq XC7Z020 FPGA on a Zedboard with a clock frequency of 200 MHz.

Next, we verified the matching rate between the estimated angles calculated using the proposed method and the true angle values. The bit width setting of the fixed-point numbers in the table was the same as the circuit size verification. The true angle values were calculated by feeding  $f_x$  and  $f_y$  into the  $\text{atan2}$  function of the  $\text{cmath}$  library in C language and discretized in 36 directions. In addition, we compared the proposed method with the angle approximation method from a previous study (Nagamine et al., 2019).

Next, we compared processing speeds of angle calculations of the following three methods:

1. Software implementation of angle calculation by  $\text{atan2}$  function
2. Software implementation of angle calculation by the proposed method
3. Hardware implementation of angle calculation by the proposed method

In the software implementation, the average of the calculation times of all 261,121 input luminance gradients executed on an Intel Core i7-8700K central processing unit (CPU) was used as the angle calculation time for software. In the hardware implementation, clock cycles to calculate an angle by the circuit multiplied by the clock cycle time was used as the angle calculation time. Here, the fraction part of the fixed-point numbers in the table was set to six bits, and the target board and its clock frequency were the same as the circuit size verification. Thus, the clock cycle time was set to 5 ns.

Next, we verified the approximation effect of the proposed method on the accuracy of human recognition tasks. To avoid the effect of the binarization of the discriminator using the BNN, we used a support vector machine (SVM) (Cristianini and Shawe-Taylor, 2000), which is a floating-point number model, as a discriminator. Here, we compared the accuracy of three algorithms for GMM-MRCoHOG: the original algorithm, the hardware-oriented algorithm of the previous study (Nagamine et al., 2019), and the proposed algorithm. We set the number of mixtures of Gaussian distribution as 16 and 32. The datasets used in this experiment were the Daimler Pedestrian Classification Benchmark Dataset (Gavrila and Enzweiler, 2008) and INRIA Person Dataset (Dalal and Triggs,

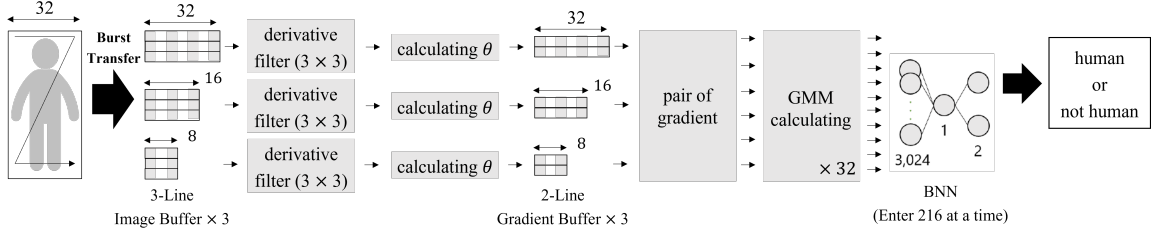


Figure 6: Human Recognition Circuit Integrating Hardware-Oriented GMM-MRCoHOG and BNN

Table 1: Experimental Environment

CPU	Intel Core i7-8700K 3.70[GHz]
Memory	16GB
OS	Windows 10
Circuit Synthesis Environment	Vivado HLS 2018.2 GUINNESS
FPGA Board	ZedBoard, XC7Z020CLG484-1 (200[MHz])
FPGA Board	ZCU102, XCZU9EG-2FFVB1156 (100[MHz])



Figure 7: Examples of Daimler Pedestrian Classification Benchmark Dataset



Figure 8: Examples of INRIA Person Dataset

2005), which consist of human and non-human images of size  $32 \times 64$  pixels. The details of these datasets are summarized in Table 2, and example images of these datasets are shown in Figs. 7 and 8.

We also verified the accuracy of a human recognition system using the BNN as a discriminator and compared it with that of a binarized version of the VGG-16 network (Simonyan and Zisserman, 2015).

Table 2: Dataset

Train		
Dataset	Images	Resolution
Daimler	human: 10,000	$32 \times 64$ pixels
INRIA	not human: 10,000	
Test		
Dataset	Images	Resolution
Daimler	human: 1,126	$32 \times 64$ pixels
INRIA	not human: 4,840	

## 4.2 Human Recognition Circuit Integrating Hardware-Oriented GMM-MRCoHOG and BNN

The designed human recognition circuit was synthesized using Vivado HLS 2018.2, to estimate the processing speed and circuit size. The target device was a Xilinx Zynq UltraScale+ MPSoC XCZU9EG FPGA on a ZCU102 board with a clock frequency of 100 MHz. For comparison, we also implemented the binarized VGG-16 in the XCZU9EG FPGA using GUINNESS (Nakahara et al., 2019).

For the speed comparison between software and hardware implementations of the human recognition systems, the average of the software execution time to process 5,955 images of size  $32 \times 64$  pixels on an Intel Core i7 8700 K CPU was used as the image processing time for the software implementation. For the hardware implementation, clock cycles to process an image of  $32 \times 64$  pixels, estimated by C Synthesis of Vivado HLS 2018.2, multiplied by the clock cycle time 10 ns, was used as the image processing time. For the binarized VGG-16, clock cycles to process an image of  $48 \times 48$  pixels, estimated by GUINNESS,

multiplied by the clock cycle time 10 ns, was used as the image processing time.

We also estimated the circuit size of the human recognition system using the Export RTL of Vivado HLS 2018.2, and the circuit size of the binarized VGG-16 using GUINNESS. Moreover, we estimated the power consumption of the circuit using Vivado 2018.2.

## 5 RESULTS

### 5.1 Coarse Angle Calculation Method by Using Fixed-Point Tangent Table

Figures 9 and 10 show the circuit resource utilization of the  $\tan\theta$  table. As shown in Fig. 9, both the LUT and flip-flop (FF) utilization increased almost linearly whereas the bit width of the fraction part of the fixed-point numbers was zero to six bits. However, in the case of the seven-bit model, the number of resources was lower than that in the six-bit model. As shown in Fig. 10, a digital signal processor (DSP) was required only in the case of the seven-bit model whereas no DSP was required in the range of zero to six bits.

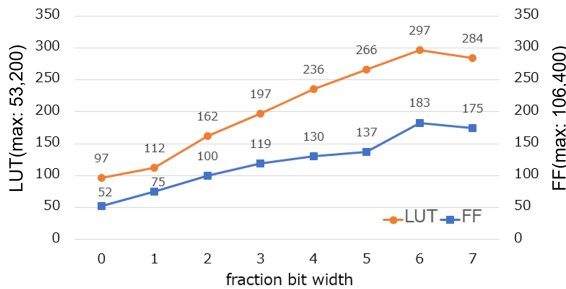


Figure 9: Circuit Resource Utilization of LUTs and FFs

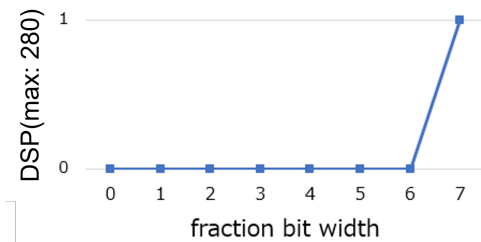


Figure 10: Circuit Resources Utilization of DSPs

Figure 11 shows the angle matching rate between approximated angles by the proposed method and true angles obtained by atan2 function. According to a previous study (Nagamine et al., 2019), the matching rate was 91%. Therefore, the matching rate of the pro-

posed method was higher than that of the previous study when the bit width of the fraction part of the fixed-point numbers was four or more, and it was approximately 99 % when the bit width was six or more. The maximum error of the angle in the figure represents the maximum absolute difference between the angles approximated by the proposed method and the true angles. For example, if some angle is classified by atan2 function in the third direction while the angle is classified by the proposed method as the fourth direction, the error is 1. From the figure, the maximum error of the angle was 1 in cases of more than two bits for the fraction part of the fixed-point numbers.

Table 3 shows the processing time of the angle calculation. As shown in the table, the proposed hardware-oriented algorithm on the CPU required approximately 14 times longer processing time than that of the atan2 function. The proposed hardware-oriented algorithm on the FPGA was approximately twice as fast as the atan2 function, and approximately 28 times faster than the proposed algorithm on the CPU.

Table 3: Processing Time of Angle Calculation

Methods	Time [ns]
atan2 (software)	59.6
Proposed (software)	837.7
Proposed (hardware)	30

Figures 12 and 13 show the accuracy of the human recognition system with 16 and 32 Gaussian mixtures with the SVM implemented by MATLAB. As shown in these figures, the proposed method improved the accuracy of the human recognition task from the previous study in both mixture cases.

Table 4 presents the human recognition accuracy of the proposed method with the BNN where the number of mixtures was set as six, and Table 5 shows the human recognition accuracy of the binarized VGG-16. The proposed human recognition system with a BNN having one neuron in the hidden layer was able to classify humans with high accuracy and outperform the binarized VGG-16.

Table 4: Human Recognition Accuracy by Hardware-Oriented GMM-MRCoHOG with BNN

	Accuracy rate
train	99.4 [%]
test	97.1 [%]



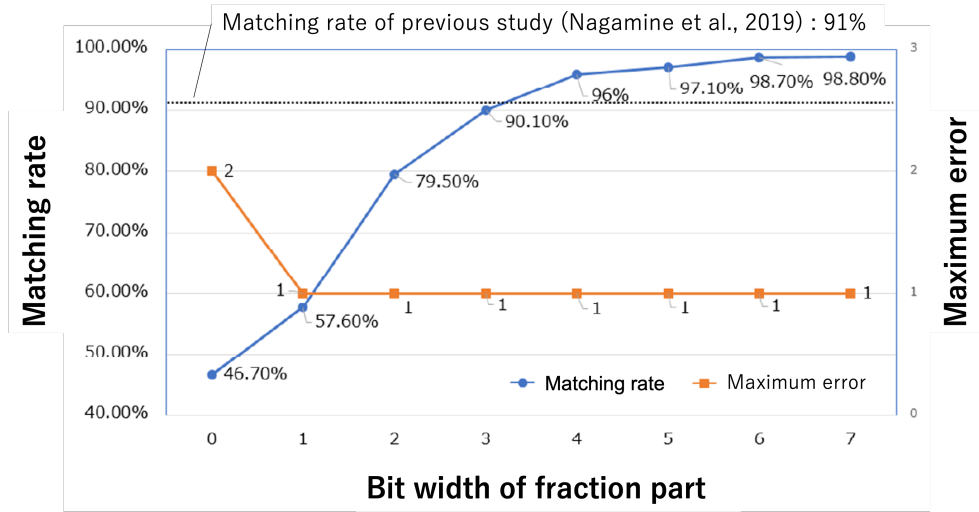


Figure 11: Angle Matching Rate and Maximum Error between Angles by the Proposed Method and atan2 Function

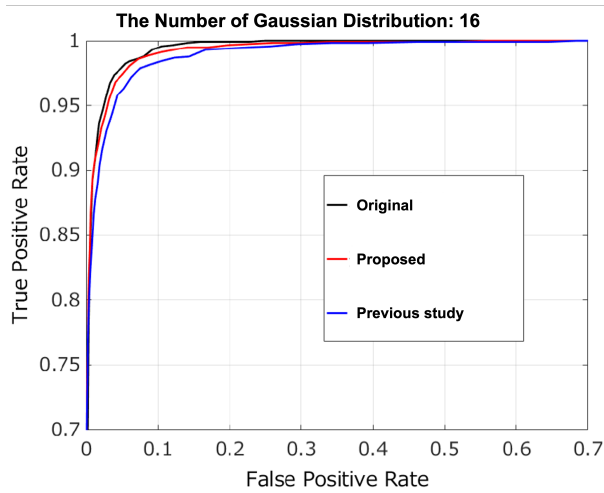


Figure 12: Human Recognition Accuracy in the case of 16 Mixtures

Table 5: Human Recognition Accuracy of Binarized VGG-16

	Accuracy rate
train	77.4 [%]
test	44.3 [%]

## 5.2 Human Recognition Circuit Integrating Hardware-Oriented GMM-MRCoHOG and BNN

Table 6 presents the estimated processing time of human recognition. The proposed hardware was approximately 118 times faster than the software implementation and approximately 123 times faster than the hardware implementation of the binarized VGG-

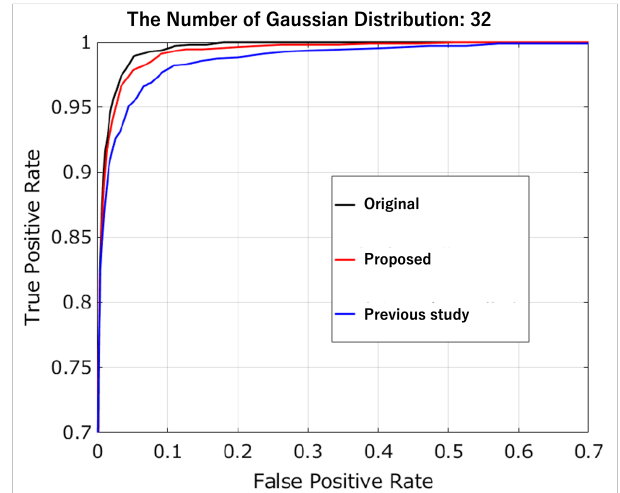


Figure 13: Human Recognition Accuracy in the case of 32 Mixtures

16.

Table 6: Processing Time of Human Recognition

Methods	Time[ms]
Proposed (software)	5.2
Proposed (hardware)	0.044
Binarized VGG-16 (hardware)	5.4

Table 7 presents the estimated circuit resource utilization of the proposed human recognition circuit and Table 8 shows the estimated circuit resource utilization of the binarized VGG-16. As presented in Table 7, the proposed circuit can be implemented in the XCZU9EG FPGA, whereas the circuit could not be implemented in the XC7Z020 FPGA owing to a lack of resources. The dominant resource in the circuit was

the block random access memory (BRAM), which was determined by the number of center coordinates and width of the mixture Gaussian distribution, and the synaptic weights of the BNN. Compared with the binarized VGG-16, the proposed human recognition circuit consumed fewer FFs and LUTRAMs, but more BRAMs and LUTs.

Table 7: Circuit Resource Utilization of the Proposed Human Recognition Circuit

	Used	Available	Utilization [%]
BRAM	154	912	16.9
DSP48E	0	2,520	0
FF	11,529	548,160	2.1
LUT	27,331	274,080	10.0
LUTRAM	111	144,000	0.1

Table 8: Circuit Resource Utilization of the Binarized VGG-16

	Used	Available	Utilization [%]
BRAM	148	912	16.2
DSP48E	0	2,520	0
FF	21,751	548,160	3.9
LUT	21,765	274,080	7.9
LUTRAM	1,934	144,000	1.3

Table 9 lists the estimated power consumption of the circuit. As shown in the table, the power consumption of the proposed circuit is 0.923 [W]. It is noteworthy that this power was for only the programmable logic on the XCZU9EG chip, not for the entire FPGA board, including the processing system on the chip and dynamic RAMs on the board.

Table 9: Estimated Power Consumption of the Circuit

	Power [W]
Proposed circuit	0.923
Binarized VGG-16	0.949

## 6 DISCUSSION

### 6.1 Coarse Angle Calculation Method by Using Fixed-Point Tangent Table

As shown in the experimental results (Figs. 9 and 10), the number of LUTs and FFs increased linearly while the fraction part of fixed-point numbers was in range from zero to six bits. In the case of the seven-bit model for the fraction part, the number of LUTs and FFs decreased, and the number of DSPs increased be-

cause the high-level synthesis compiler estimated using the DSP was more efficient than using LUTs and FFs to represent multiplications.

Table 10 is a summary of the comparison of FFs and LUTs utilization for the  $\tan^{-1}$  function between the high-level synthesis of atan2 function, the method of the previous study (Nagamine et al., 2019), and the proposed method. As presented in the table, the proposed method, even with six bits for the fraction part, which was the most resource-intensive method among the proposed method, required approximately 1/30 of the circuit resources for both FF and LUT of the high-level synthesis of the atan2 function. Moreover, the number of LUTs in the proposed circuit was significantly smaller than that in the previous study. Therefore, the proposed method succeeded in reducing the size of the circuit.

Table 10: Circuit Resource Utilization of the Original Algorithm, Previous Study, and Proposed Method

	FF	LUT
$\tan^{-1}$	6,000	10,000
Previous study	76	3,087
Proposed (0 bit)	52	97
Proposed (1 bit)	75	112
Proposed (2 bit)	100	167
Proposed (3 bit)	119	197
Proposed (4 bit)	130	236
Proposed (5 bit)	137	266
Proposed (6 bit)	183	297

The accuracy of the proposed method for the human recognition task was better than that of the binarized VGG-16, as well as in a previous study (Nagamine et al., 2019). According to a previous study, the accuracy for the same task was 92.4% whereas, the accuracy of the proposed method was 97.1%. Additionally, a discrepancy in the angle calculation of the previous method was 9°. Therefore, the proposed method extracted more precise features, resulting in better performance in the human recognition task.

### 6.2 Human Recognition Circuit Integrating Hardware-Oriented GMM-MRCoHOG and BNN

Although there was no significant difference between the proposed circuit and binarized VGG-16 in terms of circuit size and power consumption, the proposed circuit outperformed the binarized VGG-16 for the human recognition task, and the processing time of the proposed circuit was significantly faster than that of the binarized VGG-16 because the proposed cir-



cuit computed the algorithm in parallel using an effective pipeline architecture with line buffers. Therefore, we concluded that the proposed circuit is more suitable for a human detection system than the binarized VGG-16.

## 7 CONCLUSIONS

For robots and self-driving cars operating near humans, a high-accuracy, high-speed, and low-power human detection function is required. In this study, we designed a dedicated circuit of GMM-MRCoHOG with high human recognition performance and implemented it in an FPGA to realize a high-speed and low-power human recognition system. Using the  $\tan\theta$  table, the proposed hardware-oriented algorithm simplifies the calculation of luminance gradients, which is a high-cost operation in the original algorithm. The experimental results show that the proposed method improves the accuracy and processing speed of the human recognition task while reducing the circuit resources.

In future work, we plan to implement a human detection system on an FPGA by feeding multiple regions of interest from an image to the proposed circuit for human recognition. Because the processing speed of the circuit is high, the realization of a real-time human detection system can be expected.

## REFERENCES

- Cristianini, N. and Shawe-Talor, J. (2000). An introduction to support vector machines. In *Cambridge University Press*.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38.
- Gavrila, D. M. and Enzweiler, M. (2008). Monocular pedestrian detection: Survey and experiments. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, volume 31, pages 2179–2195.
- Higashi, S., Michishita, Y., Enokida, S., Shibata, M., and Yamada, H. (2018). Pedestrian detection based on gaussian mixture model multiresolution cohog. In *Proc. 4th World Congress on Electrical Engineering and Computer Systems and Sciences*.
- Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29, pages 4107–4115.
- Iwata, S. and Enokida, S. (2014). Object detection based on multiresolution cohog. In *Proc. 10th International Symposium on Visual Computing*, pages 427–437.
- Michishita, Y., Higashi, S., Shibata, M., Muramatsu, R., Yamada, H., and Enokida, S. (2018). Autonomous state space construction method based on mixed normal distributions for pedestrian detection. In *IEEJ Transactions on Electronics, Information and Systems*, volume 138, pages 1100–1107.
- Nagamine, Y., Yoshihiro, K., Enokida, S., M. Shibata, H. Y., and Tamukoh, H. (2019). Human detection using hardware oriented gmm-mrcohog. In *35th Fuzzy System Symposium*, pages 715–719.
- Nakahara, H., Yonekawa, H., Fujii, T., Shimoda, M., and Sato, S. (2019). Guinness: A gui based binarized deep neural network framework for software programmers. In *IEICE Transactions on Information and Systems*, volume E102.D, pages 1003–1011.
- Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proc. International Conference on Learning Representations (ICLR)*.