A Study of Image Classifier Combining In-pixel Array Operations and Digital Matrix Operations in Image Sensors

Takeshi ENOMOTO Kota IMAGAWA Kota YOSHIDA Shunsuke OKURA

Research Organization of Science and Engineering, Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu, Shiga 525-8577, Japan

Abstract— Toward the IoT era, where numerous sensors and AI-driven analysis are deployed, we propose an on-chip image classification system that integrates lightweight neural networks within CMOS image sensors. The system combines in-pixel and in-column analog convolution with digital matrix computations. To assess the feasibility of the proposed system, we evaluated the matrix computation circuit area and image classification accuracy through software simulations. The image classification accuracy for the MNIST, Fashion-MNIST, and INRIA-Person datasets reached 88.75%, 79.91%, and 83.79%, respectively.

I. Introduction

In the IoT era, efficient integration of image sensors and artificial intelligence (AI) is essential to reduce cost and power consumption. Edge devices that perform data processing at the sensor level have gained attention for reducing data volume and communication overhead.

A method utilizing stacking technology has been proposed to integrate AI processing capabilities into image sensor chips [1]. Stacking technology requires advanced fabrication technologies, which may lead to increased manufacturing costs.

This study explores a method for implementing an image classifier based on a lightweight neural network (NN) directly on an image sensor chip, combined with in-pixel analog convolution operations.

Fig. 1 shows the difference between conventional on-device image classification systems and the on-chip image classification system discussed in this paper. In on-device image classification systems, a CMOS Image Sensor (CIS) chip captures an image and transmits the digital data to a Microcontroller Unit (MCU). The MCU processes the image data by feeding them into an AI model, such as a Deep Neural Network (DNN), to perform image classification. On the other hand, the on-chip image classification system implements a lightweight matrix computation circuit, specifically a systolic array, within the CIS chip. This configuration enables feature extraction and image classification to be performed entirely within the CIS chip, including pixel processing and analog-to-digital conversion (ADC), without imaging transmission to the MCU. By outputting only the classification results, the CIS chip significantly reduces communication overhead with the MCU and further decreases the overall power consumption of the system by offloading image processing from the MCU.

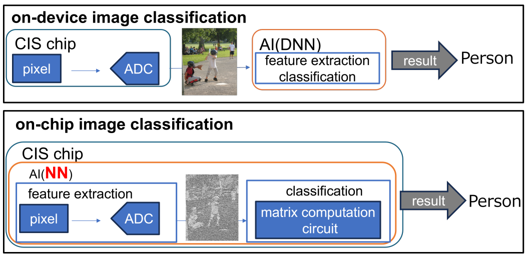

Fig. 1.: the concept of on-chip image classification

This paper presents an evaluation of the matrix computation circuit area and the image classification accuracy of the proposed on-chip image classification system through software simulation. The remainder of this paper is structured as follows. Sec. 2 provides background, introducing the general structure of a CIS and related studies on analog convolution processing circuits. Sec. 3 proposes an on-chip image classification system. Sec. 4 evaluates the proposed system by assessing the area occupied by the additional systolic array circuit, supposing the allocated size less than 10% of the total area of the CIS chip. Sec. 5 examines the classification accuracy of the image classification system using a publicly available data set, assuming the configuration of the systolic array is feasible based on the evaluation of Sec. 4. Finally, Sec. 6 concludes this paper.

II. Background Information

A. *General Structure of CMOS Image Sensors*

Fig. 2 shows the conventional block diagram of the CIS chip. A CIS consists of a pixel array, a vertical scanning

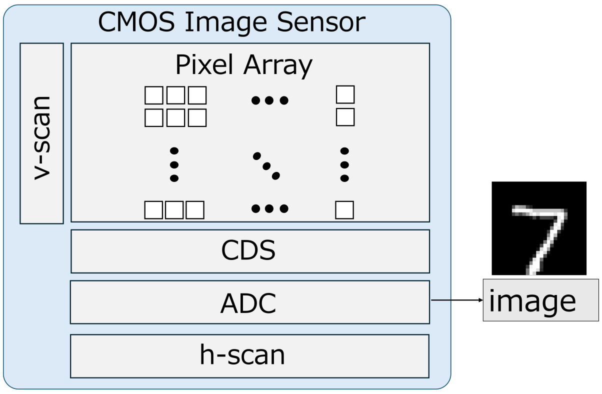

Fig. 2.: Block diagram of the CMOS Image Sensor

(v-scan) circuit, a Correlated Double Sampling (CDS) circuit, an Analog-to-Digital Converter (ADC), and a horizontal scanning (h-scan) circuit.

- The vertical scanning circuit is connected to the row shared lines of the pixel array, controlling both selected row for readout and exposure time. The selected row for readout is swept sequentially from the top down.
- The horizontal scanning circuit is connected to the column-shared lines and sequentially selects columns to readout the digitized pixel signals, where one line of pixel signal selected by the vertical scanning circuit is digitized. The readout scan is taken place from left to right, column by column.
- The CDS circuit is connected to the column-shared lines and serves to reduce reset noise and fixed pattern noise present in the pixel signal.
- The ADC converts the output voltage from the CDS circuit into a digital signal. In a conventional CIS, a column-parallel ADC architecture is commonly employed, where an ADC is laid out in each column.

B. *Related Studies*

For the CIS in image processing applications, research has focused on extracting image features through in-pixel or in-column analog processing during image capture. This approach aims to reduce the computational load on subsequent digital processing circuits and improve overall system power efficiency.

Nakagawa et al. reported an in-pixel architecture capable of performing analog MAC (Multiply-Accumulate) operations using crystalline IGZO (In–Ga–Zn–O) FET [2]. Additionally, Jeong et al. proposed an in-column architecture where analog MAC (Multiply-Accumulate) circuits are arranged along columns, enabling convolution operations on captured signals [3].

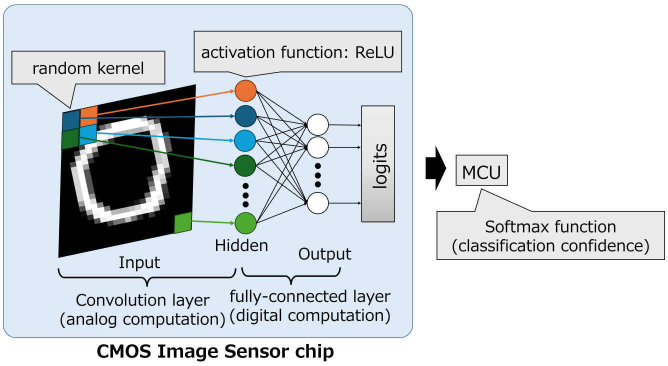

Fig. 3.: Proposed Neural Network Structure

III. Proposed System

A. *Overall Architecture of the Three-Layer Convolutional Neural Network (CNN)*

Fig. 3 shows the structure of the proposed neural network, which consists of three layers: an input layer, a hidden layer, and an output layer. First, image features are extracted in the hidden layers by convolution processing of the input image with random kernels. The extracted feature vectors are then processed with the ReLU activation function. A fully connected network is employed between the hidden and output layers, where pre-trained weights are applied to the hidden layer output to obtain a non-normalized classification results (logits). The CIS chip transmits logits to the MCU. Finally, the MCU, the Softmax function normalizes the logits to produce the classification confidence.

B. *Circuit Configuration of the On-Chip Image Classifier*

Fig. 4 shows the circuit configuration of the on-chip image classifier. This design incorporates matrix computation circuit and filter computation circuit in addition to the conventional CIS architecture.

- Convolution processing with random kernel is taken place in both the pixel array and the column parallel CDS circuit. Weights in row and column directions are respectively controlled by the exposure time of each pixel row and the gain of the each CDS circuit.
- The filter computation circuit sums the ADC output values from neighboring columns using an adder. Furthermore, the ReLU activation function is realized by subtracting a threshold value.
- The matrix computation circuit performs fully connected layer operations on the intermediate feature vector, which is the filter computation circuit output.

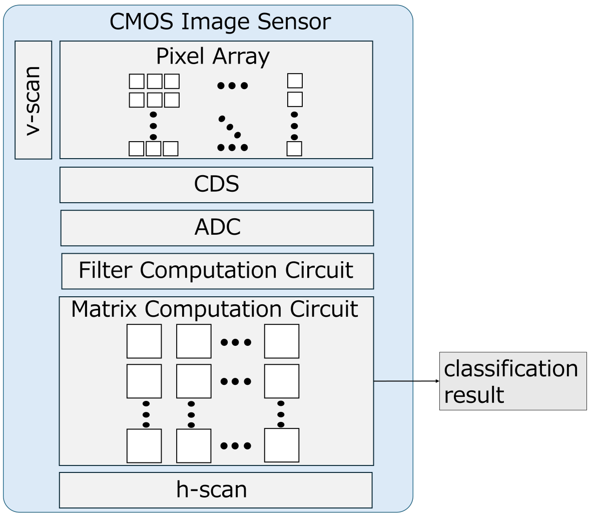

Fig. 4.: Block diagram of the Image Classifier

C. *In-Pixel Analog Convolution Using Random Kernels*

Convolution operations are widely used in image classification tasks to extract spatial features from local regions within an image. In this paper, the values of the convolutional kernel are randomly determined and fixed. Exposure time control per row and readout gain control per column are combined to realize random weighting. Since the typical CIS supports destructive readout of pixel signal charge, the pixel signal can be used only once for convolution. Therefore, the number of output rows obtained after convolution is given by division of the number of pixels by the kernel size. While pixel-wise weighting is realized within the pixel array and in the column parallel CDS, the addition operations are performed in the filter computation circuit.

D. *Filter Computation Circuit*

The weighted pixel signals can be read as output voltages from each pixel. These signals are then digitized using an ADC, and summed to obtain the digital values of the intermediate feature vector. Furthermore, threshold subtraction is applied to perform activation processing using the ReLU function. The ReLU function is widely used in image classification tasks and is defined for an input x as follows:

$$\mathrm{ReLU}(x) = \begin{cases} 0 & (x < 0) \\ x & (x \geq 0). \end{cases} \tag{1}$$

E. *Matrix Computation for the Fully Connected Layer Using Systolic Arrays*

The fully connected layer operation in neural networks is realized with a matrix inner product. Consequently, various matrix computation circuits have been proposed as hardware accelerators for inference with DNN.

In this study, the fully connected layer operation is applied to the intermediate feature vector output from the filter computation circuit, using a matrix computation circuit. Since the pixel signal is read out row by row, the intermediate feature vector is fed into the matrix computation circuit row by row. To accommodate this row-by-row sequence, a matrix computation circuit based on a systolic array architecture [4] was designed.

Fig. 5 shows the matrix computation circuit based on a systolic array where Processing Elements (PEs)—each consisting of an adder, a multiplier, and a register—are aligned in a array. First, pre-trained weight parameters are supplied to each PE individually. The intermediate feature values output from the filter computation circuit, are sequentially transferred from top to bottom through the registers within the PE array. Simultaneously, intermediate results from the Multiply-Accumulate (MAC) operations are transferred left to right through the PE registers. It is noted that the leftmost PE receives zero as the first intermediate result. The PE array output is accumulated in each row of the matrix computation circuit to generate the logits as the non-normalized classification results.

Detailed PE array operations in convolution processing are as follows. While the intermediate feature vector is sequentially output from the filter computation circuit, each PE in the systolic array simultaneously performs vertical data propagation and horizontal MAC operations in parallel. Specifically, the filter output is fed into each PE in the first row, where each PE performs multiplication with the corresponding weight. The PE then adds the multiplication result to the partial sum generated by the left-adjacent PE. The addition result is stored in a register. The register is connected to the PE input of the next column, enabling sequential horizontal propagation of intermediate results. This configuration ensures progressive accumulation in the horizontal direction. Simultaneously, the filter outputs are forwarded to registers in vertically connected PEs, propagating from top to bottom. The first-row PEs continue processing until the intermediate feature vector from all pixel rows is read out.

PEs in the second and following rows remain idle until they receive the intermediate feature vector from the row above. Once the PEs receive the feature vector, the same computational process as the first row is taken place in each PEs.

Detailed accumulator operation is as follows. At the right end of the PE array, accumulators composed of an adder and a register are allocated to sum the PE array output propagated in the horizontal direction. The accumulator then outputs the logits in row-parallel, which represents the non-normalized classification result.

The size of the systolic array corresponds to the number of columns in the intermediate feature vector being

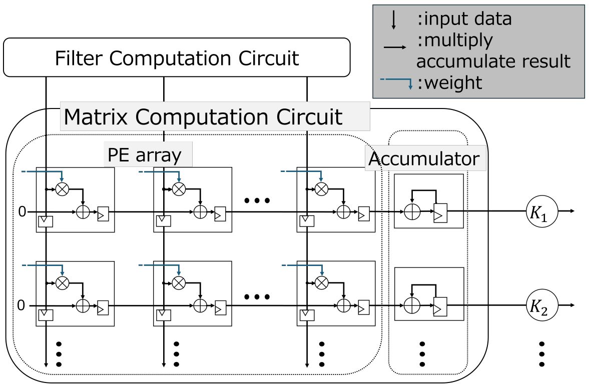
:input data
:multiply accumulate result
:weight
Filter Computation Circuit
Matrix Computation Circuit
PE array
Accumulator
0
0
K_1
K_2

Fig. 5.: Matrix Operation Circuit

read out and to the number of rows which is given by the classification class count. Let's suppose that the number of readout pixel columns is W, the kernel size for both rows and columns is F, and the number of classification classes is K. The size of the systolic array is determined as W/F columns (where W is a multiple of F) and K rows. For example, when a readout pixel array is 96×96 and the convolutional kernel size is 3×3, the number of columns and rows in the intermediate feature vector are both $96/3 = 32$. The systolic array operation finishes in $K + W/F$ clock cycles to process all intermediate feature vectors. The classification results, logits, are then outputted from the accumulator.

IV. Circuit Area Evaluation

This section reports the area evaluation results of the matrix computation circuit, which is as a major additional circuit to the conventional CIS. In this study, since we constrain the additional circuit area for the entire CIS chip, resulting from the matrix computation circuit, to be less than 10%, the on-chip NN is designed for small-scale image classification tasks.

As the small-scale image, we suppose three open datasets: MNIST [5], Fashion-MNIST [6], and INRIA-Person [7]. The MNIST dataset consists of a 10-class handwritten digit classification task (digits 0–9), containing 60,000 training images and 10,000 evaluation images. The Fashion-MNIST dataset consists of a 10-class fashion item classification task, containing 60,000 training images and 10,000 evaluation images. Both datasets have a 1:1 aspect ratio of images. The INRIA-Person dataset is a binary classification task, distinguishing between images with and without human presence. The dataset contains 4,760 training images and 1,579 evaluation images, whose aspect ratio is 1:2.

For the MNIST and Fashion-MNIST with 1:1 aspect ratio images, 36 matrix computation circuits were designed in Verilog HDL. The column counts of the PE array were 8, 16, 32, and 64, while the row counts, representing the classification class count, ranged from 2 to 10. The ROHM 0.18 μm Kyoto University/Kyoto Institute of Technology library [8] was used, and logic synthesis was performed with Synopsys Design Compiler Q-2019.12-SP1 to evaluate the circuit area. The bit widths of the intermediate feature values and weights were set to 8-bit signed integers. Table I presents the circuit area of the matrix computation circuit, where the number of gates obtained through logic synthesis is expressed in terms of NAND gates. When a small-sized CIS, such as a 1/2.5" optical format with an area of 25 mm^2, is assumed, the circuit area of the matrix computation circuit is constrained to be less than around 2.5 mm^2. Since the NAND gate size in the standard cell library is 12.9 μm^2, the gate count of the matrix computation circuit is constrained to be less than 194 kGE. The gate counts less than 194 kGE are shaded in gray in Table I. As long as the column count of the PE array is 8 or 16, the matrix computation circuit can support up to 10 classification classes. With a column count of 32, the circuit can support up to 5 classification classes. However, the circuit area exceeds the constraint when the column count reaches 64. Additionally, it is noted that classification accuracy decreases as the PE column count decreases, since the NN becomes shallower and fails to extract complex features.

For the INRIA-Person dataset with 1:2 aspect ratio images, 3 matrix computation circuits were designed in Verilog HDL. The column counts of the PE array were 8, 16, and 32, while the row counts, representing the classification class count, are 2.

The gate counts less than 194 kGE are also shaded in gray in Table II. Since the INRIA person dataset is designed for binary classification, column counts of up to 32 can be supported within the area constraint. This paper focuses on evaluating the matrix computation circuit area for the fully connected layer. The area of additional blocks such as the filter computation circuit will be considered in future work.

TABLE I
: synthesis area [kGE]
(aspect ratio = 1:1)

Number of classes	PE columns			
	8	16	32	64
2	14.4	35.2	96.1	298.0
3	20.4	48.3	126.7	376.9
4	26.5	61.6	157.4	455.7
5	32.6	75.0	188.1	534.6
6	38.8	88.2	218.9	613.5
7	44.9	101.5	249.6	692.5
8	51.0	114.8	280.3	771.4
9	57.1	128.1	311.1	850.3
10	63.3	141.4	341.8	929.3

V. Simulation-Based Accuracy Evaluation

A. Simulation Setup

This section presents the software simulation results to evaluate the image classification accuracy based on matrix computation circuit sizes. Image classification accuracy was evaluated using three open datasets for small-scale image classification tasks: MNIST, Fashion-MNIST, and INRIA-Person. The input image size (readout pixel size) depends on the systolic array size. The number of PE columns is given by W/K, where it is reminded that W and K represent the readout pixel columns and the convolution filter size, respectively. In the simulation, the filter size is set to 3×3. When the input image size exceeds that of the original dataset, it is upscaled using bilinear interpolation while preserving the aspect ratio.

In the simulation, the weights of the convolutional layer are given with random number and kept fixed. Meanwhile, the weights of the fully connected layer are trained using gradient descent with the training dataset. Tensorflow [9] was used for the simulation, and the classification accuracy was calculated as the average of 10 independent runs for each configuration.

B. Simulation Results

Fig. 6 shows the dependency of image classification accuracy on the matrix computation circuit area, where the datasets are MNIST, Fashion-MNIST, and INRIA-Person. The vertical axis represents the classification accuracy, while the horizontal axis represents the circuit area. The constraint on the maximum area of the matrix computation circuit, 194 kGE, is indicated by a dashed line. It is confirmed that image classification accuracy improves as the matrix computation circuit gate size increases. In the MNIST and Fashion-MNIST 10-class classification, the image classification accuracy reaches 88.75% and 79.91%, respectively, with a PE column count of 16 see Table I. In the INRIA-Person 2-class classification, the image classification accuracy reaches 83.79%, with a PE column count of 32 see Table II. These results suggest that the proposed three-layer neural network is robust to variations in the input image aspect ratio but exhibits a dependency on the matrix computation circuit area. Table III summarizes the comparison of on-chip image classification systems. the proposed system achieves an image classification accuracy of 88.75% for the MNIST dataset, which is lower than that of existing CNN-based implementations such as Jeong et al. However, unlike their face detection-focused analog CNN, our system targets general-purpose small-scale classification tasks and is designed for compact implementation. In the proposed system, the convolution weights are randomly fixed, which contributes to a reduction in the classification accuracy. However, this design simplifies implementation because analog circuits do not easily support weight reconfiguration. In contrast, the digital fully connected layer allows flexibility, such as using trained weights and changing the number of classes. To evaluate the power efficiency of the proposed system, a logic-level power estimation was conducted using Synopsys Design Compiler. The result shows that the matrix computation circuit consumes approximately 5.97 μW when operating at 6.67 kHz. This estimation is based on 50 μs of one pixel row readout period and 150 μs of one row of intermediate feature vector processing period generated from three pixel row signals. These results suggest that the proposed system is highly feasible for low-power, small-scale image classification tasks.

TABLE II
: synthesis area [kGE]
(aspect ratio = 1:2)

Number of classes	PE columns		
	8	16	32
2	16.9	41.5	122.0

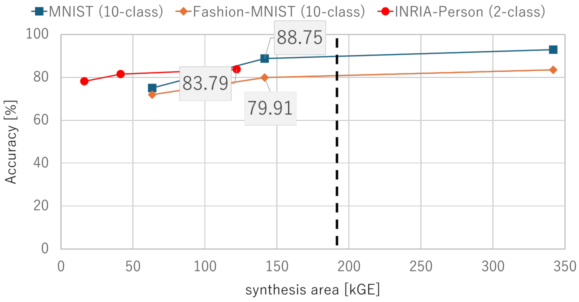

Fig. 6.: Trade-off between classification accuracy and matrix computation circuit area

VI. Conclusion

This paper proposed an on-chip image classification system that integrates lightweight neural networks within a CIS. The system combines in-pixel and in-column analog convolution with digital matrix computations, enabling direct feature vector processing within the CIS. Direct feature vector computation within the CIS reduces communication overhead and system power consumption compared to MCU-based classification. To evaluate the feasibility of the proposed on-chip image classification system, we evaluated the matrix computation circuit area and the image classification accuracy through software simulation. In the circuit area evaluation, the matrix computation circuit supports up to 10 classification classes for a 1:1 aspect ratio image when the PE array column count is 8 or 16, and up to 5 classes with 32 columns. For a 1:2

TABLE III
: Comparison of On-Chip Image Classification Systems

	Nakamura et al. (2024)[1]	**Jeong et al. (2023)[3]**	**Proposed (PE=16, MNIST)**
Digital Area (mm^2)	N/A	N/A	≈ 2.5 mm^2 ‡
Accuracy (%)	N/A	Up to 98.75 (Face Detection, Measured)	88.75 (MNIST, Simulated)
Power (mW)	90.4 @ 200fps (Estimated, DNN inference)	4.02 @ 120fps (Measured, analog CNN inference)	Not evaluated (only logic-level estimation for fully connected layer)
Process	N/A	110 nm CMOS	180 nm CMOS
Architecture	3-wafer stacked	Monolithic	Monolithic (not fabricated)
CNN type	MobileNet V2	Fully analog 3-layer	3-layer (analog conv. + digital fully connected)
Hardware Configuration	Dedicated digital wafer† (DNN + SRAM)	In-column analog MAC	Logic synthesis only (fully connected layer)

Note: "N/A" indicates that the information is not available or not disclosed in the referenced paper.
† DNN circuit is integrated in the bottom wafer of a 3-wafer-stacked CIS, but its area is not separately disclosed.
‡Estimated from logic synthesis of the fully connected layer only. Not representative of total area.

aspect ratio image and 2-class classification, 32 columns PE can be supported under a given area constraint. In the image classification accuracy evaluation, the accuracy reached 88.75% and 79.91% for the MNIST and Fashion-MNIST 10-class classifications, with a PE column count of 16. The image classification accuracy reached 83.79% for the INRIA-Person 2-class classifications, with a PE column count of 32. It was also confirmed that increasing the number of columns in the matrix computation circuit improves the classification accuracy. In future work, we will improve the circuit configuration to reduce the matrix computation circuit. Additionally, we will explore classification tasks suited for small-scale neural networks on the CIS chip.

References

[1] R. Nakamura, H. Tsugawa, H. Yamagishi, Y. Fujisaki, Y. Suda, Y. Tatsumi, K. Shimizu, Y. Kagawa, K. Ono, Y. Horie, R. Koganei, H. Nakano, K. Kobayashi, T. Kamibavashi, N. Araki, K. Saito, R. Suzue, W. Otsuka, and H. Iwamoto, "A novel 1/1.3-inch 50 megapixel three-wafer-stacked cmos image sensor with dnn circuit for edge processing," in *2024 IEEE International Electron Devices Meeting (IEDM)*, pp. 1–4, 2024.

[2] T. Nakagawa, Y. Negoro, S. Yoneda, H. Shishido, H. Kobayashi, M. Oota, T. Kawata, T. Ikeda, and S. Yamazaki, "Image sensor with in-pixel calculation using crystalline igzo fet in display," *Japanese Journal of Applied Physics*, vol. 59, no. SG, p. SGGE01, 2020.

[3] B. Jeong, J. Lee, J. Choi, M. Song, Y. Son, and S. Y. Kim, "A 0.57 mw@1 fps in-column analog cnn processor integrated into cmos image sensor," *IEEE Access*, vol. 11, pp. 61082–61090, 2023.

[4] H. T. Kung and C. E. Leiserson, "Systolic arrays for vlsi," Tech. Rep. CMU-CS-79-103, Carnegie-Mellon University, Department of Computer Science, Pittsburgh, PA, 1978. A seminal work proposing systolic array architectures for VLSI implementation of matrix computations.

[5] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, pp. 141–142, 2012.

[6] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," 2017.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005.

[8] H. Onodera, A. Hirata, T. Kitamura, and K. Tamaru, "P2lib: process-portable library and its generation system," in *Proceedings of CICC 97 - Custom Integrated Circuits Conference*, pp. 341–344, 1997.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.