IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 1, March 2024, pp. 430~439
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i1.pp430-439

Intra-class deep learning object detection on embedded computer system

Putri Alit Widyastuti Santiary, I Ketut Swardika, Dewa Ayu Indah Cahya Dewi, Ida Bagus Ketut Sugirianta
Department of Electrical Engineering, State Polytechnic of Bali, Bali, Indonesia

Article Info

Article history:

Received Nov 19, 2022
Revised Jan 20, 2023
Accepted Mar 10, 2023

Keywords:

Average precision
Embedded system
Flower dataset
Intra class variation
Object detection

ABSTRACT

Implementation of artificial intelligence tends to be portable, mobile and embeds in embedded computer system (EBD). EBD is a special-purpose computer with limited capacity in a small-form size. Deep learning (DL) had known as cutting edges for object recognition. With DL, object feature extraction analysis is omitted. DL requires large computing resources and capacity. Implement DL algorithm on EBD goal to achieves high detection accuracy and high-efficiency resources. Hence, be able to cope with intra-class variations, and image disturbances. By those challenges and limitations, this study reports the performance of EBD to recognize an object which has high variations in their class, through an optimal raw-input dataset. The raw-input dataset performed optimization process with a supervisor. Yield is the proper optimal input dataset in size. The performance results observed begin from training dataset until evaluation stage of DL. The comparison performs in efficiency resources, loss, validation-loss, timesteps, and detection accuracy by multiclass confusion matrix analysis. This study shows through this purpose method efficient resources are highly archived. Shorter timesteps ensure training stage is successful, and detection accuracy is perfectly archived. In addition, this study proves DL method archived great performances in classifying object that has identical structure.

Corresponding Author:

I Ketut Swardika
Department of Electrical Engineering, State Polytechnic of Bali
Kampus Bukit Jimbaran, Bukit Jimbaran, Kuta Selatan, Badung, 80361 Bali, Indonesia
Email: swardika@pnb.ac.id

1. INTRODUCTION

Single-board architectures (SBAs) for computational (SBCs) and microcontroller (SBMs) purposes began popular and interesting solutions in the last decade, their due to low cost and low consumption, small size, and great flexibility, which makes them an alternative in many applications [1], [2]. Included onboard innumerable integrating sensors and state-of-art of communication technologies that are increasingly used for do-it-yourself (DIY) projects, internet of things (IoT) devices in the field of science, technology, engineering, educational and academic project [3]–[5]. Recently, the SBAs have been significant development toward artificial intelligence (AI) inside capability devices and cluster-computation purposes [6]–[8]. This is possibly due to the significant improvement of capabilities of the advance reduce instruction set computing (ARISC Machine/ARM) CPU of SBAs. Advances application of SBCs are translated into high energy efficiency of giga floating point operations per second/Watt (GFLOPS/W) ARM SBCs-cluster, value for money/$ and space-utilize/M2 (GFLOPS/$, GFLOPS/M2). For example, a 16-node inexpensive and green cloud hosts ARM SBCs run and reaches 60 GFLOPS for computational modeling only consumes of 80-Watt energies [1]. Furthermore, the chips industry had developed embedded systems with onboard powerful computing graphics

processing units (GPU) to provide high processing capabilities. This feature transforms an ARM SBC into a tensor processing unit (TPU) device with fast object recognition capability. The ARM SBCs-based AI is a field of computer science that aims to make functional hardware and software into something that can think like humans. AI is widely used to solve various problems such as business, natural language, perception, diagnosis, engineering, analysis, finance, science, and reasoning [9]–[12].

Machine learning (ML) can be defined as a computer algorithm that works by learning from data and producing predictions in the future. The learning process to acquire intelligence goes through three stages, namely training, testing, and validation. ML is concerned with how to build computer programs so that they can provide solutions based on experience. A common ML topic is a classification based on deep learning (DL). DL has known as a revolutionary method in computer vision. The development of computer vision has increased due to most applications using cameras today. ML algorithm solves the problem by dividing them into several tasks and combined at the final stage, whereas DL solves all tasks in one algorithm. DL learns high-level data features incrementally and eradicates core feature extraction and domain expertise. DL allows the growth of intelligence without a clue (determining feature) but requires large resources [13]–[15].

AI and ML implementation leads to portable, mobile, and embedded system (EBD) devices. EBD systems are small-form special-purpose computers with limited capacities. High efficiency and or high detection accuracy are the goals of the application of EBD system. For those, the algorithm implemented on EBD system must be efficient enough to use low computing capabilities. And achieve high detection accuracy in which objects accomplish with intra-class variations, illuminations, and environmental disturbances. Furthermore, an algorithm must be able to achieve high efficiency on limited memory, speed, and computing capabilities on low-end mobile devices [16]–[18].

The object of this research detection is a flower with the Latin name Plumeria (from Charles Plumier 1646-1706, a French botanist). This plant comes from Central America. Flowers with a distinctive fragrance, with five petals of white to purplish red. Easy to cultivate brings up many varieties of flowers that vary in the shape and color of the petals. So that raises problems in naming new varieties. In addition, it has not been recorded in detail in the form of a physical identification database, and the dimensions of the petals and the color of the various flower petals are difficult to classify manually [13]. In this study, performances of an intra-class DL of flower Plumeria detections on EBD are presented. It aims to get the average precision (AP) and speed of frame per second (FPS) of detection high as possible through an optimal row input dataset. By this method, the weight of the network model results become light and runs faster on the EBD.

2. METHOD

This study used the EBD of Raspberry Pi 4 8Gb ram. EBD is a special-purpose computer system, all the necessary parts are integrated into the device. The word embedded indicates this system is a complete set including mechanical and electrical systems. EBD has certain preset capabilities and tasks, unlike general-purpose PCs, EBD systems have limited resources. The EBD system is small-form (credit-card size), making it easy to plant. EBD is implemented including a microcontroller with several general-purpose input-output (GPIO) pins. This system or application is used in medical instrumentation, process control, automated vehicle control, and communication devices [19]–[21].

DL is a development of neural network learning. DL is a specialized field in ML that focuses on the representation of data and adds successive learning layers to improve the representation of input data. DL requires large resources. Some problems in the classification process using the DL method are labeling objects that have high intra-class variations. Figure 1 shows the block diagram of the classification process using the DL method for high intra-class variations [13].

Previous research has tried to run the DL method on the EBD system. As is known DL requires large computing and storage resources, and it becomes a challenge to be able to run well on the EBD system. One researcher combined the offline training predictive model method with the Learnet-model to select the DL model for the new input. Other researchers selected several combinations of architectural hardware to be able to run traffic sign detection and concluded that the tensor processing unit (TPU) gave faster results than the graphic processing unit (GPU). Other researchers have reduced the computational load and memory requirements by compromising the accuracy of DL detection. Another concentrated on the dimensions of the image with a down-up scale using the DL Learnet [20], [21].

The state-of-the-art research approach is to minimize the size of the DL training weight results by minimizing the dimensions of the image dataset that is supervised by an expert (operator) (bold box Figure 2). The original image dataset is scaled down again using the operator-supervised bicubic method. The operator is used here, assuming the operator is a perfect AI engine compared to the distance-median algorithm to check the image quality of down-scaling [22]–[25]. Figure 3 shows the research design of the Plumeria L classification. There are four block stages of the process to be carried out so that the output can classify Plumeria L. flowers with a very good level of precision where the system will run on EBD-TPU. The first

block is the optimization stage of the sample pixel size (1), followed by (2) the dataset preparation process (there are about 34 classes of Plumeria L. flowers, and in each class, there are 200 sample images). The next block (3) is the training process in Google Collaboratory. with a large limiting average precision (AP) of 0.86 (high enough), the difference (error) training-loss with validation-loss 0.1 (low enough) and a minimum network weight are achieved [26].

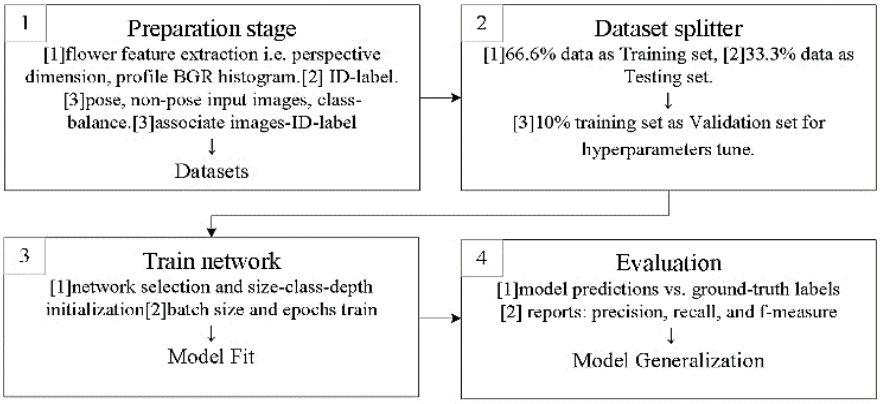

Figure 1. Block diagram of DL classification [13]

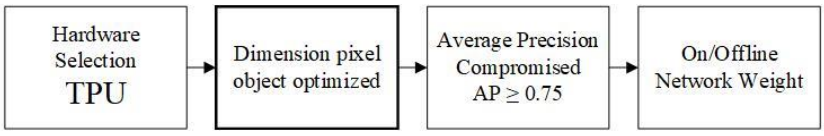

Figure 2. State of the art of this research

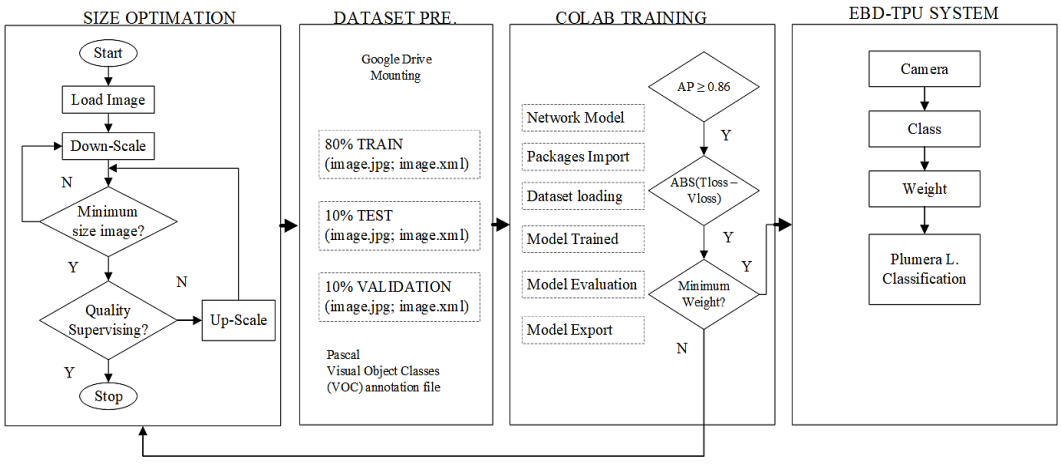

Figure 3. Algorithm of raw-input data optimization

The custom model object detection was developed using Google TensorFlow 2 machine learning framework. The model network selection uses the Efficient-Det mobile version (spec = object_detector. EfficientDetLite2Spec()) and is trained on Google Collaboratory. To simplify, the dataset consists of only five classes with a label (label_map = {1: '0204', 2: '0604', 3: '0704', 4: '0804 ', 5: '3003'}). Each class consists of 200 image files and 200 annotation files, a total of 1,000 image files, and total of 1,000 annotation files [27], [28]. The successfully trained result, the model weight then export to TensorFlow lite version for EBD system. The results of intra-class flower Plumeria L detection in the confusion matrix are compared with or without optimally of the raw-input dataset with or without using tensor processing unit (TPU) acceleration [24].

The original dataset of the flower Plumeria L has been collected with the dimension of 300 by 300 pixels. Then, downscaling to find the optimal dimension of an image using the bicubic interpolation method. Bicubic uses 4 by 4 kernel pixels, where the obtained pixel is interpolated from a summation of four coefficients or weights. Bicubic produces a sharp image, and balances between processing time and output quality [29]–[31].

The dataset preparation consists of two datasets, the original and optimal dimensions of images for comparative study. Each dataset is separated randomly into train, test, and validation folders about 80, 10, and 10 percent of total. For annotation, the image uses the Pascal Visual Object Classes (VOC) on xml file. All dataset is uploaded and mounted as Google drive. In Google Collaboratory, TensorFlow 2 machine learning framework is used to train a custom model. Some parameters are set to obtain high average precision, lowest loss, and minimum model weight. If the results do not satisfy, the process is back to the optimal sizing image process. The minimum model weight result then exports into TensorFlow lite version to run on EBD for the inference system [27], [28].

Measurement of performances by comparison of all process results between with and without optimally of raw-input dataset i.e., the total file size, processing time of upload dataset, output of Google Collaboratory training (APs, Losses, train process time and size of file model.) and results on detection in the confusion matrix. The confusion matrix is used to determine the performances of model classification. This differs from classification accuracy, where it shows ratio in percentage of correct predictions to total predictions made. The misclassification or error rate formula is shown as in (1), where the classification accuracy can be obtained by inverting that formula.

$$error\ rate = (1 - (correct\_predictions/total\_predictions\ )) * 100 \quad (1)$$

Classification accuracy alone can be misleading and might encounter problems in practice because it hides detail for a better understanding of the performance of the classification model. Common problems occur when using multiclass classification accuracy i.e., a high classification score does not reflect true accuracy because one or more classes can be neglected by the model. Or those score model archives always predict the most common class value (imbalanced datasets).

The performances of a classification algorithm summarizing with a confusion matrix [32]. Calculating a confusion matrix can give an overview to understand the model about what types of errors it is making. For multiclass of confusion matrix, there is no positive, or negative actual and predicted class, instead on a square of class matrix each of the predicted rows (P row) contains a cell of true-positive (TP), when the predict and actual class agrees, and other cells are called false-positive (FP) for each other predicted class. On the each of actual columns (A column) classes are called false-negative (FN) for each cell of the other predicted class. The remaining cells are called true-negative (TN). For clarity, Figure 4 shows an illustration of a multiclass confusion matrix of three classes [33]–[36].

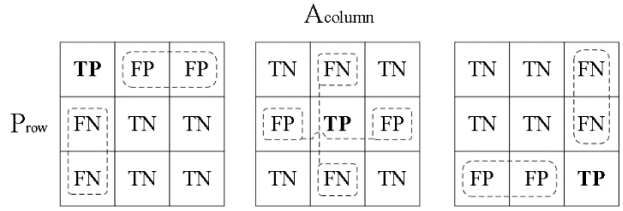

Figure 4. Multiclass confusion matrix of three classes

From the confusion matrix of multiclass at Figure 4, parameters can be calculated to reflect of performance of the classification model i.e. The accuracy describes how accurate the model is to correctly classify the object, as in (2). The precision describes how precise the model predicts requested data, as in (3). The recall or sensitivity describes how the success of the model in retrieving information, as in (4). The F-1 score or F-measure is the harmonic mean of precision and recall. In other words, the statistical measure of model accuracy, as in (5).

$$Accuracy = (TP + TN)/(TP + FP + FN + TN) \quad (2)$$

$$Precision = TP/(TP + FP) \quad (3)$$

$$Recall = TP/(TP + FN) \quad (4)$$

$$F1\ Score = (2 * Recall * Precision)/(Recall + Precision) \quad (5)$$

The EBD-TPU system consists of an HD Webcam mounted on a light emitting diode (LED) ring with a tripod stand, a Raspberry Pi 4 8Gb on a metal chase with a DC power supply, and a USB Google coral TPU accelerator. The Raspberry Pi runs on the Bullseyes OS, python with OpenCV 2 and TensorFlow lite interpreter library, and other dependencies. To simplify, the VNC remote desktop also runs on the EBD system. The EBD-TPU system shows in Figure 5. This study demonstrated only five classes from the total of 28 classes of flower Plumeria L. The multiclass confusion matrix will be five by five class matrix as in Figure 4. The performances of the classification model are computed from (2) through (5) after the results are obtained.

Figure 5. EBD-TPU system uses in this research

3. RESULTS AND DISCUSSION

3.1. Optimal of raw-input dataset

The following are the results of the downscaling of raw images process into an image with a lower dimension than the original shown in Table 1. The scale uses a percentage of 80 to 5, the resulting image becomes smaller in dimension with a smaller kilobyte file size. The down-scale method can be seen in chapter 2, the methodology of this research. Supervision is marked in the marking column, where the resulting image blur cannot be used, then the down-scale process stops. The result is the optimal resolution of the image (marks are marked as optimal), where down-scale is no longer possible. For the results in the Table 1, the optimal down-scale is 16%, with a file size of 6.8 kB, and the image dimension becomes 48 by 48 pixels. Processing and resource efficiency obtained about 84 percent, from the original image. These processes repeat for 28 other classes of flower Plumeria L.

Table 1. Optimal of raw-input dataset for class-0104

Resize (%)	Dimension	Size (kB)	Image	Mark
80	238×238	165.9		Clear
60	179×179	93.9		Clear
40	119×119	41.5		Clear
20	60×60	10.5		Clear
16*	48×48	6.8		Optimal
15	45×45	5.9		Blur

* Indicates the optimal percentage of downscaling

3.2. Evaluation of training processes dataset

Figure 6 shows the result of the training loss and validation loss comparison between the original image (ORI) and the optimal size image (OPT). Figure 6(a) shows the training loss and Figure 6(b) shows the validation loss for both datasets. In general, training and validation loss for the original size image has less than the optimal size image, but the gap does not so much significant. For both datasets, the validation loss has less compared with the training loss. Figure 7 shows the average timestep in seconds for one-step training for both datasets. Here, the optimal size of raw-input data gets the training process four times faster.

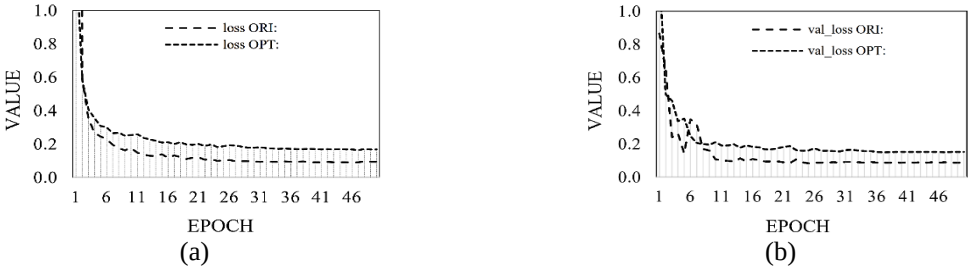

Figure 6. Loss and validation loss comparison between original and optimal image dataset, (a) for loss and (b) for validation loss

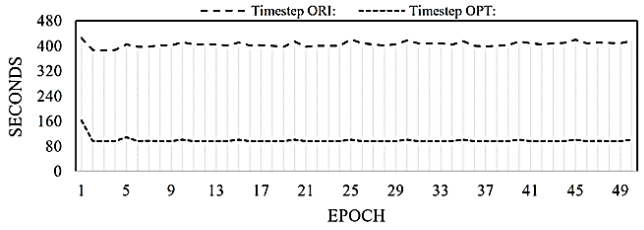

Figure 7. Timestep (s) comparison between original and optimal image dataset

3.3. Average precision (AP)

Table 2 shows the result of AP for various test modes for TensorFlow (TF) and TensorFlow lite (TF lite) versions for the original image dataset (ORI) and optimal image dataset (OPT). In column AP, there are general AP, AP for 50% or 70% of test data, AP_/ for each class, and other any specific hardware architecture. In general, there is a significantly improves in AP values for the optimal dataset and AP for each class and other any specific hardware architecture. There is a trade between the TF and TF lite. AP decreases but not significantly, and trades with a lighter framework of TF, hence it can be run on EBD system.

Table 2. Comparison of AP between original and optimal image dataset

AP	TF ORI	TF Lite ORI	TF OPT	TF Lite OPT
AP	0.66019803	0.66019803	0.92493236	0.91545844
AP50	1.0	1.0	1.0	1.0
AP75	1.0	1.0	1.0	1.0
AP_/0504	0.7	0.7	0.92318875	0.9106436
AP_/0604	0.7	0.7	0.92989224	0.9262494
AP_/1104	0.6	0.6	0.9153264	0.90673953
AP_/2404	0.6	0.6	0.89737105	0.89200497
AP_/3103	0.7009901	0.7009901	0.9588834	0.9416546
Apl	-1.0	-1.0	-1.0	-1.0
Apm	0.66019803	0.66019803	0.92495656	0.91545844
Aps	-1.0	-1.0	-1.0	-1.0
Arl	-1.0	-1.0	-1.0	-1.0
Arm	0.68	0.68	0.96	0.929
ARmax1	0.68	0.68	0.933	0.929
ARmax10	0.68	0.68	0.96	0.929
ARmax100	0.68	0.68	0.96	0.929
Ars	-1.0	-1.0	-1.0	-1.0

3.4. Object Detection on EBD system

Object detection of various variants of the flower Plumeria L was carried out as in Figure 5. For the preliminary study, only five classes from 28 total classes of flower Plumeria L have been prepared i.e., 0204, 0604, 0704, 0805, and 3003. Class naming or labeling of this collection of datasets of flower Plumeria L following the result of [13]. There are two processes identical for object detection i.e., detection using the original dataset and using optimally dataset. This is done by only changes of custom model weight respectively according to the dataset. The interested class is chosen randomly from five actual class objects and placed on a green screen with LED lighting. The number of replication (repetition of detection of one class) is 25 times. By statistically, adequate to draw reliability of the conclusions. Then, the EBD system predicts the class (predicted class object) in real-time. The results on the camera are a bounding box of object detection and class label predictions along with average precision (%) and frame movement speed (FPS) in seconds.

Figure 8 shows the results detection of class flower Plumeria L runs on EBD system for both datasets. Figure 8(a) shows result for the original dataset, while Figure 8(b) show result for the optimal dataset. There are 25 images as the result of detection of one class, in Figure 8(a) and Figure 8(b) showing only one representative. The object of class of flower Plumeria L detects on the screen with a green bounding box, a caption with the predicted class label, and accuracy. The screen also shows the FPS value. In general, for both datasets results show a highly accurate prediction of above 95% for all classes.

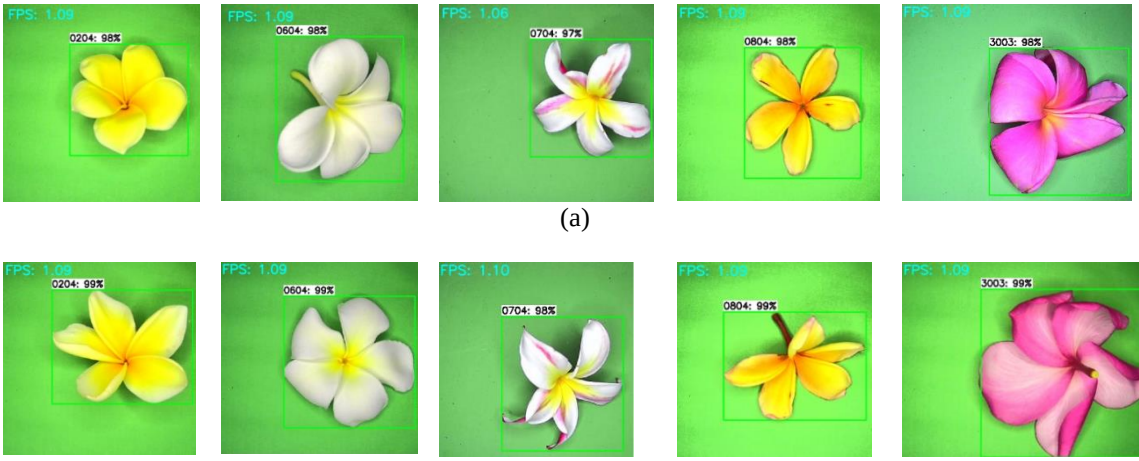

Figure 8. Detection results on EBD system, (a) the original dataset, for classes 0204, 0604, 0704, 0804, and 3003 respectively and (b) the optimal dataset, for classes 0204, 0604, 0704, 0804, and 3003 respectively

Detection results are summarized into a multiclass confusion matrix as seen in Table 3 for both datasets. Table 3(a) show result for the original dataset, while Table 3(b) show result for the optimal dataset. In general, all actual classes (column-header name) perfectly predicted row-header name classes for both datasets. The 25 value is a repetition of detection of one class. The zero value means no other class is predicted. In this section's result, no differences are affected by the optimal of the raw-input dataset. Except for a very small difference in average FPS and accuracy.

Table 3. Comparison of summarized confusion matrix results between both datasets

(a). Original dataset

FPS: 1.0908		Actual Class				
Accuracy: 0.9698		0204	0604	0704	0804	3003
Predicted Class	0204	25	0	0	0	0
	0604	0	25	0	0	0
	0704	0	0	25	0	0
	0804	0	0	0	25	0
	3003	0	0	0	0	25

(b). Optimal dataset

FPS: 1.08795		Actual Class				
Accuracy: 0.9786		0204	0604	0704	0804	3003
Predicted Class	0204	25	0	0	0	0
	0604	0	25	0	0	0
	0704	0	0	25	0	0
	0804	0	0	0	25	0
	3003	0	0	0	0	25

Table 4 shows part of the confusion matrix and performance results for both datasets. The true-positive (TP) number for the actual class 0204 (the first row in Table 4) is 25. Because all the replication (repetition of

detection of one class is 25 times) results agree between the predicted and actual class. The true-negative (TN) number for the actual class 0204 is 100 (total repetition detection of five classes). Hence, false-positive (FP) and false-negative (FN) numbers become zero. For other classes, (remain rows in Table 4) give the same result as class 0204. The detection performances are computed using (2) through (5) and results are also shown in Table 4. All detection performances are one because all replication detection perfectly predicts TP. In Table 4 precision and recall numbers for all classes are one, this is the goal of object detection. Unfortunately, hard to get a coral USB TPU accelerator due to chips outage. By using this, the FPS number will be double.

Table 4. Part of confusion matrix and performances result for both datasets

CLASS	TP	TN	FP	FN	ACCURACY	PRECISION	RECALL	F1-SCORE
0204	25	100	0	0	1	1	1	1
0604	25	100	0	0	1	1	1	1
0704	25	100	0	0	1	1	1	1
0804	25	100	0	0	1	1	1	1
3003	25	100	0	0	1	1	1	1

4. CONCLUSION

Detection of the intra-class of flower Plumeria L which has a high variation between classes has been carried out with good results. By optimizing the size of the raw-input dataset, resources and processing obtained more efficiency. The average precision obtained dramatically improved. Most useful in the training process, where train timesteps are shorter and loss becomes converges faster. This can avoid training process failure which took a long-time process. Performances of object detection on the EBD system are perfectly archived. This shows that artificial intelligence with deep learning methods on recognition between intra-classes is not only based on object structure but can recognize differences from specific object features, such as color. This proposed method is novel and can be used to resource efficiency and improve detection results.

ACKNOWLEDGEMENTS

This study part of the State Polytechnic of Bali budget implementation registration form with contract number is SP DIPA-023.18.2.677608/2022 Rev.03 (February 15, 2022).

REFERENCES

[1] P. J. Basford *et al.*, "Performance analysis of single board computer clusters," *Future Generation Computer Systems*, vol. 102, pp. 278–291, 2020, doi: 10.1016/j.future.2019.07.040.

[2] S. J. Johnston *et al.*, "Commodity single board computer clusters and their applications," *Future Generation Computer Systems*, vol. 89, pp. 201–212, 2018, doi: 10.1016/j.future.2018.06.048.

[3] J. Cheng, "Evaluation of physical education teaching based on web embedded system and virtual reality," *Microprocessors and Microsystems*, vol. 83, 2021, doi: 10.1016/j.micpro.2021.103980.

[4] S. Khant and A. Patel, "COVID19 remote engineering education: Learning of an embedded system with practical perspective," *Proceedings of International Conference on Innovative Practices in Technology and Management, ICIPTM 2021*, pp. 15–19, 2021, doi: 10.1109/ICIPTM52218.2021.9388360.

[5] X. Li, L. Sun, and C. A. Rochester, "Embedded system and smart embedded wearable devices promote youth sports health," *Microprocessors and Microsystems*, vol. 83, 2021, doi: 10.1016/j.micpro.2021.104019.

[6] C. Alippi, S. Disabato, and M. Roveri, "Moving convolutional neural networks to embedded systems: The AlexNet and VGG-16 case," *Proceedings - 17th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2018*, pp. 212–223, 2018, doi: 10.1109/IPSN.2018.00049.

[7] M. Lopez-Montiel, U. Orozco-Rosas, M. Sanchez-Adame, K. Picos, and O. H. M. Ross, "Evaluation method of deep learning-based embedded systems for traffic sign detection," *IEEE Access*, vol. 9, pp. 101217–101238, 2021, doi: 10.1109/ACCESS.2021.3097969.

[8] B. Taylor, V. S. Marco, W. Wolff, Y. Elkhatib, and Z. Wang, "Adaptive deep learning model selection on embedded systems," *ACM SIGPLAN Notices*, vol. 53, no. 6, pp. 31–43, 2018, doi: 10.1145/3299710.3211336.

[9] W. Rahmaniar and A. Hernawan, "Real-time human detection using deep learning on embedded platforms: A review," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 462-468Y, 2021, doi: 10.18196/jrc.26123.

[10] A. Subbotin, N. Zhukova, and T. Man, "Architecture of the intelligent video surveillance systems for fog environments based on embedded computers," *2021 10th Mediterranean Conference on Embedded Computing, MECO 2021*, 2021, doi: 10.1109/MECO52532.2021.9460270.

[11] J. L. Álvarez, J. D. Mozo, and E. Durán, "Analysis of single board architectures integrating sensors technologies†," *Sensors*, vol. 21, no. 18, 2021, doi: 10.3390/s21186303.

[12] I. K. Swardika, P. A. W. Santiary, I. B. I. Purnama, and I. W. Suasnawa, "Development of green zone energy mapping for community-based low carbon emissions," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 10, no. 6, pp. 2472–2477, 2020, doi: 10.18517/ijaseit.10.6.12642.

[13] P. A. W. Santiary, I. K. Swardika, I. B. I. Purnama, I. W. R. Ardana, I. N. K. Wardana, and D. A. I. C. Dewi, "Labeling of an intra-class variation object in deep learning classification," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 1, pp. 179–188, 2022, doi: 10.11591/ijai.v11.i1.pp179-188.

[14] Y. M. Tabra and F. N. Tawfeeq, "Reduced hardware requirements of deep neural network for breast cancer diagnosis," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 4, pp. 1362–1372, 2022, doi: 10.11591/ijai.v11.i4.pp1362-1372.
[15] S. Karunaratna and P. Maduranga, "Artificial intelligence on single board computers: An experiment on sound event classification," *5th SLAAI - International Conference on Artificial Intelligence and 17th Annual Sessions, SLAAI-ICAI 2021*, 2021, doi: 10.1109/SLAAI-ICAI54477.2021.9664746.
[16] B. Qureshi and A. Koubaa, "On energy efficiency and performance evaluation of single board computer based clusters: A hadoop case study," *Electronics (Switzerland)*, vol. 8, no. 2, 2019, doi: 10.3390/electronics8020182.
[17] R. Rodriguez-Zurrunero and A. Araujo, "Adaptive frequency scaling strategy to improve energy efficiency in a tick-less Operating System for resource-constrained embedded devices," *Future Generation Computer Systems*, vol. 124, pp. 230–242, 2021, doi: 10.1016/j.future.2021.05.038.
[18] S. Lin *et al.*, "FFT-based deep learning deployment in embedded systems," *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*, vol. 2018-January, pp. 1045–1050, 2018, doi: 10.23919/DATE.2018.8342166.
[19] A. Erbsen, S. Gruetter, J. Choi, C. Wood, and A. Chlipala, "Integration verification across software and hardware for a simple embedded system," *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pp. 604–619, 2021, doi: 10.1145/3453483.3454065.
[20] M. Verma, S. K. Vipparthi, G. Singh, and S. Murala, "LEARNet: Dynamic imaging network for micro expression recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 1618–1627, 2020, doi: 10.1109/TIP.2019.2912358.
[21] J. Prados-Garzon, T. Taleb, and M. Bagaa, "LEARNET: Reinforcement learning based flow scheduling for asynchronous deterministic networks," *IEEE International Conference on Communications*, vol. 2020-June, 2020, doi: 10.1109/ICC40277.2020.9149092.
[22] K. Chen, K. Franko, and R. Sang, "Structured model pruning of convolutional networks on tensor processing units," Jul. 2021, [Online]. Available: http://arxiv.org/abs/2107.04191.
[23] M. Hauru, A. Morningstar, J. Beall, M. Ganahl, A. Lewis, and G. Vidal, "Simulation of quantum physics with Tensor Processing Units: brute-force computation of ground states and time evolution," 2021, [Online]. Available: http://arxiv.org/abs/2111.10466.
[24] N. Jouppi, C. Young, N. Patil, and D. Patterson, "Motivation for and evaluation of the first tensor processing unit," *IEEE Micro*, vol. 38, no. 3, pp. 10–19, 2018, doi: 10.1109/MM.2018.032271057.
[25] Z. Pan and P. Mishra, "Hardware acceleration of explainable machine learning using tensor processing units," 2021, [Online]. Available: http://arxiv.org/abs/2103.11927.
[26] S. A. Sanchez, H. J. Romero, and A. D. Morales, "A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework," *IOP Conference Series: Materials Science and Engineering*, vol. 844, no. 1, 2020, doi: 10.1088/1757-899X/844/1/012024.
[27] W. Vallejo, C. Díaz-Uribe, and C. Fajardo, "Google Colab and virtual simulations: Practical e-learning tools to support the teaching of thermodynamics and to introduce coding to students," *ACS Omega*, vol. 7, no. 8, pp. 7421–7429, 2022, doi: 10.1021/acsomega.2c00362.
[28] Q. Wang, M. Ihme, Y. F. Chen, and J. Anderson, "A TensorFlow simulation framework for scientific computing of fluid flows on tensor processing units," *Computer Physics Communications*, vol. 274, 2022, doi: 10.1016/j.cpc.2022.108292.
[29] Y. Zhu, Y. Dai, K. Han, J. Wang, and J. Hu, "An efficient bicubic interpolation implementation for real-time image processing using hybrid computing," *Journal of Real-Time Image Processing*, vol. 19, no. 6, pp. 1211–1223, 2022, doi: 10.1007/s11554-022-01254-8.
[30] Z. Huang and L. Cao, "Bicubic interpolation and extrapolation iteration method for high resolution digital holographic reconstruction," *Optics and Lasers in Engineering*, vol. 130, 2020, doi: 10.1016/j.optlaseng.2020.106090.
[31] Y. Li, F. Qi, and Y. Wan, "Improvements on Bicubic image interpolation," *Proceedings of 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2019*, pp. 1316–1320, 2019, doi: 10.1109/IAEAC47372.2019.8997600.
[32] S. Wang, H. Lu, A. Khan, F. Hajati, M. Khushi, and S. Uddin, "A machine learning software tool for multiclass classification [Formula presented]," *Software Impacts*, vol. 13, 2022, doi: 10.1016/j.simpa.2022.100383.
[33] H. Asil and J. Bagherzadeh, "A new approachto image classification based on adeep multiclass AdaBoosting ensemble," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 4872–4880, 2020, doi: 10.11591/ijece.v10i5.pp4872-4880.
[34] B. Jeong *et al.*, "Comparison between statistical models and machine learning methods on classification for highly imbalanced multiclass kidney data," *Diagnostics*, vol. 10, no. 6, 2020, doi: 10.3390/diagnostics10060415.
[35] I. Shiri *et al.*, "High-dimensional multinomial multiclass severity scoring of COVID-19 pneumonia using CT radiomics features and machine learning algorithms," *Scientific Reports*, vol. 12, no. 1, 2022, doi: 10.1038/s41598-022-18994-z.
[36] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass confusion matrix reduction method and its application on net promoter score classification problem," *Technologies*, vol. 9, no. 4, 2021, doi: 10.3390/technologies9040081.

BIOGRAPHIES OF AUTHORS

Putri Alit Widyastuti Santiary works as a lecture at the Department of Electrical Engineering, the State Polytechnic of Bali. She holds a master's degree in computer engineering from Udayana University and teaching computer programming language for more than twenty years at the State Polytechnic of Bali. She wrote many teaching materials about programming in C++, Java, and Python. She had to author more than eleven journal papers. Her current research interest is on botanical classification with deep learning. She can be contacted at email at: putrialit@pnb.ac.id.

I Ketut Swardika is an associate professor at the Department of Electrical Engineering, the State Polytechnic of Bali. He received his Ph.D. degree from Yamaguchi University in 2012. He had experience in Japanese's satellite remote sensing program. He wrote many teaching materials from the topic in RS, electrical to computer programming. He had the authoring of more than twenty journal papers. His current research interests include nighttime RS observation for the sustainability of energy and the environment. He can be contacted at email: swardika@pnb.ac.id.

Dewa Ayu Indah Cahya Dewi work as a lecture at Department of Electrical Engineering, the state Polytechnic of Bali. She received her master's degree in computer engineering from Udayana University in 2018. Her current research interest in data mining. She can be contacted at email: ayuindahcahyadewi@pnb.ac.id.

Ida Bagus Ketut Sugirianta is an associate professor at the Department of Electrical Engineering, the State Polytechnic of Bali. He received his master's degree from Udayana University and teaching electrical installation for more than thirty years. He wrote many teaching materials on the topic of electrical installation and project management. He had the authoring of more than twenty journal papers. His current research interests are in fault detection, classification, and the application of solar PV. He can be contacted at email: ibksugirianta@pnb.ac.id.