Dual-Rate Alpha Binary Networks: An Energy-Efficient Optimization Approach for Waste Classification

Anonymous Author(s)
Anonymous Affiliation
Anonymous Email Address

***Abstract*—The current global waste management issue continues to grow, with waste segregation becoming an increasingly urgent challenge. In response, AI-based solutions have shown promising potential for waste classification. However, these solutions often encounter performance bottlenecks due to computational constraints. To address these challenges, we propose a novel approach for waste classification, namely the Dual-Rate Alpha Binary Networks: An Energy-Efficient Optimization Approach for Waste Classification. This method introduces a Binary Neural Network with a new trainable parameter using Dual Variable Learning Rates and employs a flexible weight/activation quantization strategy with PReLU. By combining these techniques with early stopping, our method demonstrates improved performance. The proposed approach is evaluated on the CIFAR10 dataset and shows favorable results compared to existing models. Further experiments on CNN and conventional BNN models, utilizing a publicly available Garbage Classification dataset with over 15,000 images, demonstrate high accuracy and low FLOPS. This research opens up new possibilities for applying binary neural networks to resource-constrained devices. Code is publicly available at: https://github.com/ducanh2712/DRA-BNN**

***Index Terms*—binary neural network, dual variable learning rate, waste classification, energy-efficient computing**

I. Introduction

A. Waste Management Problem

With the development of society, waste disposal has become a significant challenge to environmental sustainability, especially in densely populated and developing countries.

Over the years, numerous efforts have been made to address the pollution caused by waste, including recycling, disposal, and treatment. However, due to the manual classification of various types of waste, individuals often encounter challenges in efficiently sorting them.With this error-prone task, the significant consumption of time, effort, and financial resources has become an urgent issue that requires immediate attention.

The question that needs to be addressed is: How can modern technologies be applied to automate the waste classification process, thereby minimizing manual labor, enhancing environmental protection, and promoting global economic development?

B. Problem Statement

Researchers are actively developing methods for automating waste classification using computer vision and deep learning technologies. Convolutional Neural Networks (CNN) and their advanced variants, such as AlexNet [1], VGG [2], Inception [3], and ResNet [16], have shown promising results in sorting tasks. For example, Kang et al. [5] applied a ResNet34-based algorithm for waste classification, incorporating features like multi-feature fusion and residual unit reuse. Sha Meng et al. [8] proposed the X-DenseNet model, which integrated Xception with DenseNet, achieving 94.1% accuracy on the TrashNet dataset.

However, CNNs typically use full-precision (32-bit) weights and activation functions, which often result in a high number of parameters, complex architectures, and lengthy processing times, making them impractical for resource-limited environments like mobile devices or wearables. For instance, YOLOv3 [7] achieved only 60% accuracy in garbage detection while processing at 20 frames per second, which is suitable for real-time applications but not ideal for classification tasks on low-power devices. To address these challenges, one promising solution is the use of quantization techniques, which significantly reduce the model size and computational complexity while maintaining acceptable performance in resource-constrained environments.

C. Proposed Solution

Quantization is a technique used to reduce the computational complexity and memory requirements of deep neural networks by reducing the bit-width of the weights and activations. This process compresses the model, making it more suitable for resource-constrained environments. Quantization techniques can be applied to different parts of the network, including weights, activations, or both, with various optimization approaches to minimize accuracy degradation.

Among these techniques, Binary Neural Networks (BNNs) represent an extreme case of quantization, where parameters are restricted to binary values (+1 or -1). While BNNs reduce computational and memory requirements, they often lead to reduced accuracy due to excessive quantization. As a result, many studies focus on modifying the model, refining binaryization methods for weights and activations, and applying optimization techniques like regularization, learning rate adjustment, and flexible activation functions to mitigate the negative impact of extreme quantization. Notable approaches include BinaryConnect [10], XNOR-Net [11], and differen-

Identify applicable funding agency here. If none, delete this.

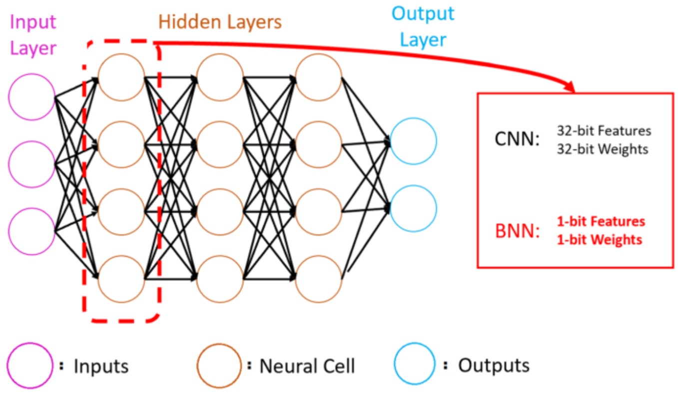

Fig. 1. Binary Neural Networks (Source: Yuan and Agaian [15]).

tiable surrogates like the straight-through estimator (STE) [17] or Soft Binary Activation [13].

In the mentioned research, despite significant progress, gradient loss remains a major challenge. Our method addresses this by using learnable parameters (α_{DVLR}) for each parameter, bypassing the traditional approach and effectively optimizing gradient loss, resulting in:

- Proposing a weight and activation quantization model without using STE.
- Applying an efficient activation function, minimizing computational cost (FLOPS) without sacrificing accuracy.
- Demonstrating the method's effectiveness through experiments on the CIFAR-10 and a 12-class public garbage classification dataset.

II. Related Work

A. Binary Neural Networks

BNN is a specialized type of neural network in which some or all of the weights and activations are constrained to 1-bit values, except for the input and output layers as shown in Fig. 1. The process of quantizing from 32-bit precision to 1-bit, known as binarization, is expected to reduce complexity and computation costs. Rastegari et al. [14] demonstrated that XNOR-Net reduces memory usage by approximately 32 times compared to conventional CNNs while maintaining competitive performance. Moreover, the use of BNNs introduces a novel approach to deep learning computations by leveraging bitwise operations such as XNOR and bit-counting, which significantly enhance computational efficiency. These operations allow BNNs to replace conventional matrix multiplications with lightweight binary operations, making them highly suitable for deployment in resource-constrained environments such as edge devices and mobile applications.

In Binary Neural Networks (BNNs), the binarization function is typically the sign function, which converts the output into either +1 or -1, as shown in (1).

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \quad (1)$$

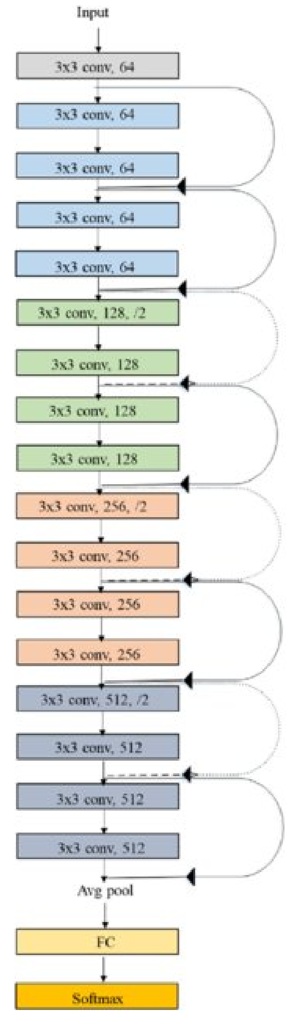

Fig. 2. ResNet18 architecture

While this binarization simplifies computation, it introduces a significant challenge during the backpropagation phase. Specifically, the sign function is non-differentiable, which means that the gradient of the activation with respect to the weights is zero almost everywhere, making it impossible to update the weights using traditional gradient descent.To address this issue, STE is commonly used to approximate the gradient of the sign function by treating it as an identity function during the forward pass and allowing the gradient to pass through as if the activation were continuous during the backward pass. This enables the use of gradient-based optimization techniques despite the non-differentiability of the sign function.An illustration of the process using both the sign function and STE can be seen in Fig. 3. This approach has been widely adopted in BNNs to enable efficient training while maintaining binary activations.

B. ResNet Model

Residual Networks (ResNets), introduced by He et al. [16], revolutionized deep learning by addressing the vanishing gradient problem through residual connections. These connections

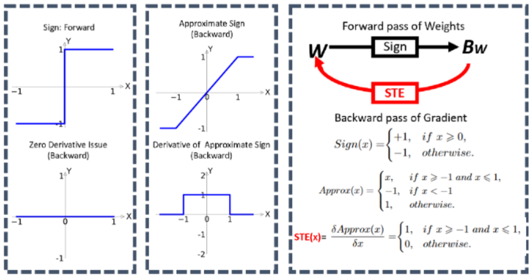

Fig. 3. The Sign function and STE used in Binary Neural Networks (BNNs) (Source: Yuan and Agaian [15]).

allow gradients to propagate more effectively, enabling the training of very deep networks. Specifically, ResNet-18 [18], which consists of 18 layers, provides a simpler and more efficient architecture. ResNet-18 showed significant improvements in image classification tasks, outperforming previous architectures on benchmarks like ImageNet. For BNN, ResNet-18 has been adapted, allowing for effective training with binary weights and activations.

C. Dual Variable Learning Rates

Dual Variable Learning Rates (DVLR) [19] is a method designed to optimize the training process of neural networks by adjusting the learning rate for different variables based on their importance. The method helps achieve better convergence by applying distinct learning rates to different parameter groups, ensuring that critical parameters are updated faster, while less important ones are updated more slowly. This approach has been shown to improve training efficiency and model performance across a variety of tasks, including image classification and other common machine learning benchmarks.

The effectiveness of DVLR has been validated through extensive experiments on standard datasets, demonstrating its ability to accelerate training and achieve higher accuracy. Specifically, it has been shown to outperform traditional methods with a single learning rate, highlighting its potential for optimizing neural network training.

In terms of implementation, the update rule for parameters W in a network with corresponding gradients ∇W under DVLR can be expressed as:

$$W^{(t+1)} = W^{(t)} - \eta_i \nabla W^{(t)}, \quad i \in \{1, 2\} \tag{2}$$

where η_1 and η_2 are the learning rates applied to different parameter groups. This differential approach to learning rate application ensures that more important parameters are updated with a higher rate, while less significant parameters are adjusted more conservatively.

By incorporating DVLR, it becomes possible to stabilize training and improve performance in BNNs. The method can be used to apply different learning rates to binary weights, enhancing convergence and preventing the model from getting stuck in suboptimal solutions. This makes DVLR a promising approach for future research in BNNs, potentially leading to more efficient and accurate binary models.

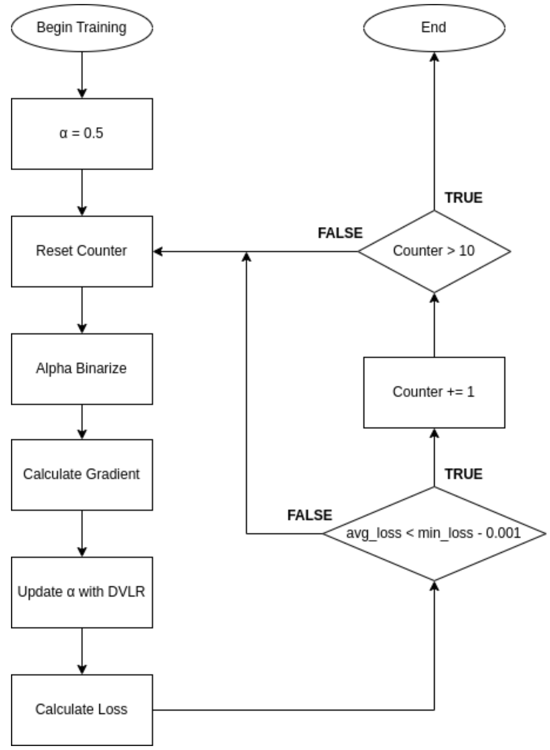

Fig. 4. Illustration of the proposed DRA-BNN framework, highlighting key components and data flow.

III. Methodology

In this section, we address the limitations of traditional BNNs during the training process, such as information loss, reduced accuracy, and limited flexibility, especially when the input data varies significantly. To address these issues, we propose the Dual-Rate Alpha Binary Networks (DRA-BNN). This approach enhances quantization while maintaining meaningful gradient flow during backpropagation, effectively preventing gradient vanishing.The flowchart in Figure 4 illustrates the flow of the proposed method.

A. Alpha Quantization

In DRA-BNN, the coefficient α_{DVLR} serves as a learnable parameter used to scale the binarized weights after applying the sign$(\cdot)$ function. The training of α_{DVLR} is performed via gradient descent with the assistance of the DVLR method.

In general, the binarized values are computed as follows:

$$\text{binary\_values} = \text{sign}(\text{value}) \cdot \alpha_{DVLR} \tag{3}$$

Where:

- sign(weight): The sign function, returning $+1$ if weight > 0, -1 if weight < 0, and 0 if weight $= 0$.
- α_{DVLR}: A learnable scaling factor, initialized to 0.5.

By adjusting the α_{DVLR} coefficient based on the data distribution, the binarization coefficients are better characterized and distributed according to the task.

B. Gradient of α_{DVLR} in the Backward Pass

The gradient of α_{DVLR} is calculated based on the gradient of the loss with respect to the output (grad_output) and the binarized weight values:

$$\text{grad}_{\alpha_{DVLR}} = \sum(\text{grad\_output} \cdot \text{sign}(\text{weight})) \tag{4}$$

Where:

- grad_output: The gradient propagated back from the subsequent layer.
- $\sum$: Summation over all elements in the weight tensor.

C. Updating α in DVLR

The primary goal is to adaptively adjust α with DVLR during training to improve the binarization process while ensuring that the gradient flow remains meaningful.

First, α is constrained within the range $[0.1, 2.0]$ for stability:

$$\alpha_{t+1} = \text{clamp}(\alpha_{t+1}, 0.1, 2.0) \tag{5}$$

To update α, we apply a gradient descent rule:

$$\alpha_{t+1} = \alpha_t - \eta_\alpha \cdot \nabla_\alpha L \tag{6}$$

where α_t represents the α coefficient at time step t, η_α is the learning rate for α, and $\nabla_\alpha L$ is the gradient of the loss function with respect to α.

The gradient $\nabla_\alpha L$ ensures that the scaling factor α contributes to preserving the critical information during binarization. This allows the model to adaptively adjust the scaling factor based on the characteristics of the data, thus improving the overall performance without relying on the traditional STE approach.

By updating α in this manner, DVLR enables the model to maintain computational efficiency while improving the expressiveness and accuracy of the binarized model. The independent and adaptive learning of both weights and α results in a more robust network that can handle imbalanced distributions and vanishing gradients effectively.

D. PReLU Activation Function

In this study, we explore the use of a more effective activation function to enhance the model's training performance. Specifically, we consider using the Parametric ReLU (PReLU) [22] activation function as an alternative to the traditional ReLU. The PReLU function is defined as (7):

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ \alpha_{\text{PReLU}} x & \text{if } x \leq 0, \end{cases} \tag{7}$$

where α_{PReLU} is a learnable parameter during training.

Unlike ReLU, which outputs 0 for negative inputs, PReLU allows the slope of the negative part of the activation function to be learned via the parameter α_{PReLU}. This addresses the "dying ReLU" problem, where certain neurons may become inactive if their inputs are consistently negative, causing them to stop updating and resulting in poor learning. By learning the parameter α_{PReLU}, PReLU ensures that neurons can still learn from negative inputs, improving model performance compared to ReLU.

IV. Experiments

A. Dataset

In this experiment, we first validated our model using the CIFAR-10 dataset [20], a widely used benchmark for image classification tasks. Subsequently, we applied our model to a more specific dataset, the Garbage Classification Dataset [21], which consists of 15,150 images categorized into 12 classes of household garbage: paper, cardboard, biological, metal, plastic, green-glass, brown-glass, white-glass, clothes, shoes, batteries, and trash.

This dataset was created with the goal of improving the recycling process by classifying waste into more specific categories, which allows for better sorting and higher efficiency in recycling. Unlike many other datasets that focus on fewer classes (2–6 categories), the inclusion of 12 categories can significantly enhance recycling efforts by allowing for more precise identification of waste types.

B. Model Setup and Training

The experiments were conducted on a system with an NVIDIA GeForce GTX 1660 GPU, Intel Core i5-10400 CPU, and 8GB RAM, running Ubuntu 22.04. The software environment includes Python 3.9.21 and PyTorch 2.5.1.

The model used is BinaryResNet18, which is created by binarizing all layers except for the input and output layers to reduce computational cost. The initial value of the α_{DVLR} parameter was set to 0.5, and the learning rate was initialized at 0.001. The Adam optimizer was used with the CrossEntropyLoss function for multi-class classification, and the learning rate was adjusted using the ReduceLROnPlateau scheduler, which decreases the rate by 50% if the loss doesn't improve after 5 epochs, with a minimum LR of 0.0001.

Training was performed for up to 400 epochs, with early stopping applied if there was no significant improvement in the loss after 10 epochs (delta = 0.001). The dataset was split into 80% for training and 20% for validation, with data augmentation techniques such as rotation, flipping, and color adjustment applied to the training set. The validation set was resized and normalized.

V. Result

The model's performance was evaluated based on accuracy, and the computational efficiency was measured by calculating FLOPS.

A. Model Performance Evaluation on CIFAR-10

In Table I, our approach, using PReLU activation, outperforms ReLU in terms of accuracy, with DRA-BNN (PReLU) achieving 89.80% accuracy compared to 85.47% for DRA-BNN (ReLU). This demonstrates that PReLU allows the model to better adapt to negative inputs, leading to improved overall performance.

Moreover, while CNN-ResNet18 use higher bit-widths for both weights and activations, our DRA-BNN model still achieves competitive results with significantly reduced computational cost (237M FLOPS). This highlights the feasibility

TABLE I
PERFORMANCE COMPARISON ON CIFAR-10

Model	Accuracy (%)	FLOPS (M)
CNN-ResNet18	93.3	1820
XNOR-Net	89.83	410
BinaryNet	88.7	700
BinaryConnect	85	550
LAB-Net	87.7	700-1000
DRA-BNN (ReLU)	85.47	237
DRA-BNN (PReLU)	**89.80**	**237**

TABLE II
PERFORMANCE COMPARISON ON PUBLIC GARBAGE DATASET

Model	FLOPS (M)	Accuracy
CNN-ResNet18	1820	91.62% (epoch 167)
BNN-STE	237.63	42.89% (epoch 61)
DRA-BNN	237.63	**82.66%** (epoch 97)

of our approach, providing a strong trade-off between performance and computational efficiency, making it suitable for resource-constrained environments.

B. Dataset-Specific Result

Based on the promising results from CIFAR-10, our method was evaluated for the task of waste classification. As shown in Table II, the results demonstrate that DRA-BNN achieves an accuracy of 82.66% after 97 epochs, outperforming BNN-STE, which only achieves 42.89% after 61 epochs. Although CNN-ResNet18 achieves a higher accuracy (91.62%), the DRA-BNN method, with much lower FLOPS (237.63M compared to 1820M of CNN-ResNet18), shows the feasibility of the model in computationally constrained environments while maintaining good performance compared to other methods.

To further evaluate the model's performance, the loss and accuracy plots are shown in Figures 6 and 8, while the confusion matrix, displayed in Figure 7, provides additional insight into the classification results.

VI. CONCLUSION

In this paper, we propose Dual-Rate Alpha Binary Networks (DRA-BNN) for weight/activation quantization. In DRA-BNN, the trainable DVLR alpha parameter improves performance more effectively compared to traditional BNN models. By binarizing weights relative to the mean value and computing gradients without relying on STE, the approach provides a compelling trade-off between model size, computational efficiency, and accuracy.

The results presented on general datasets such as CIFAR-10 and specifically on the Garbage Dataset show that the DRA-BNN model, utilizing PReLU, achieves stable performance without complicating the model. Therefore, we conclude that this solution is feasible and opens up new directions for utilizing high-accuracy BNN models on embedded systems or specialized FPGAs for real-time applications.

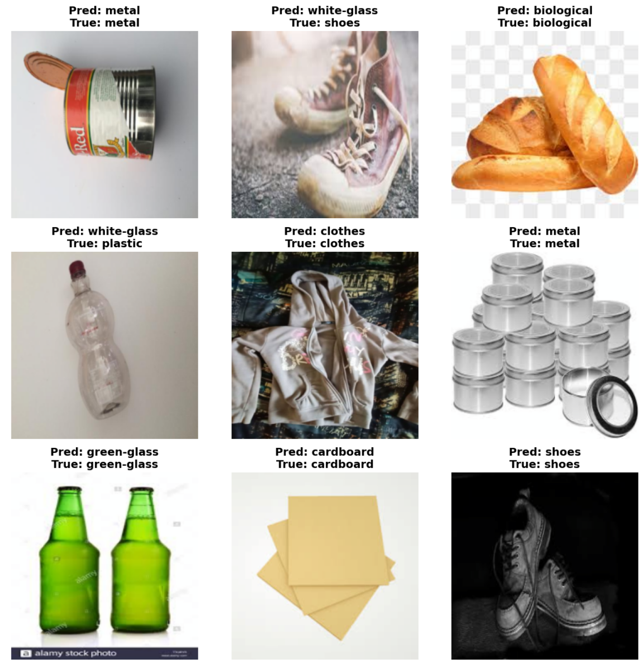

Fig. 5. Garbage Classification Testing Result.

ACKNOWLEDGMENT

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[5] Z. Kang, J. Yang, G. Li and Z. Zhang, "An Automatic Garbage Classification System Based on Deep Learning," in IEEE Access, vol. 8, pp. 140019-140029, 2020, doi: 10.1109/ACCESS.2020.3010496. keywords: Classification algorithms;Convolution;Feature extraction;Machine learning;Servomotors;Image classification;Hardware;Artificial intelligence;deep learning;image classification;neural networks

[6] B. D. Carolis, F. Ladogana and N. Macchiarulo, "YOLO TrashNet: Garbage Detection in Video Streams," 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Bari, Italy, 2020, pp. 1-7, doi: 10.1109/EAIS48028.2020.9122693. keywords: Waste management;video stream analysis;computer vision;deep learning

[7] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, arXiv:1804.02767, doi: 10.1109/AAAI.2018.01928

[8] Sha Meng, Ning Zhang, and Yunwen Ren, "X-DenseNet: Deep Learning for Garbage Classification Based on Visual Images," *Journal of Physics: Conference Series*, vol. 1575, 5th Annual International Conference on Information System and Artificial Intelligence [ISAI2020], 22-23 May 2020, Zhejiang, China, doi: 10.1088/1742-6596/1575/1/012139.

[9] M. Courbariaux and Y. Bengio, "BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," CoRR, vol. abs/1602.02830, 2016. Available: http://arxiv.org/abs/1602.02830.

[10] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2425–2433.

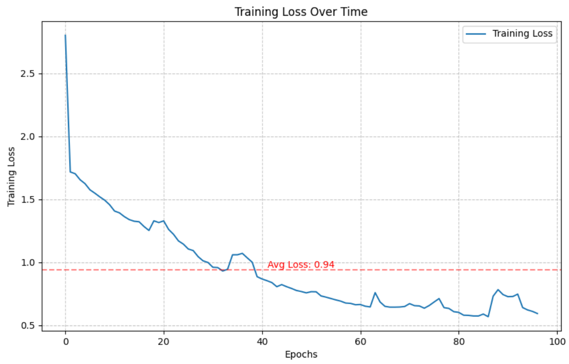

Fig. 6. Loss curves on Public Garbage Dataset training phase.

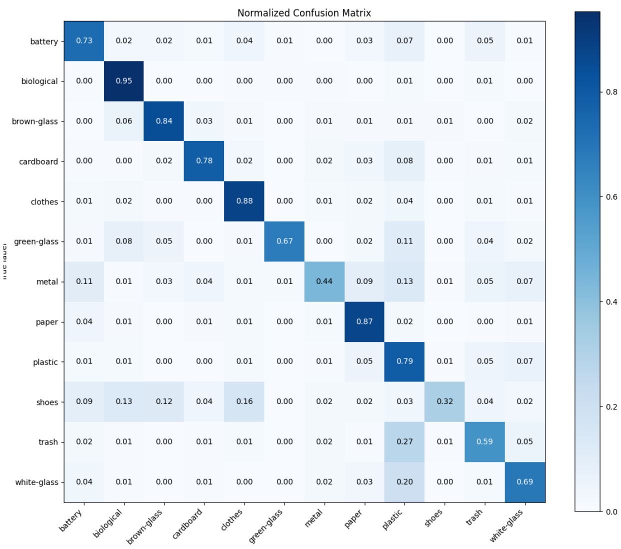

Fig. 7. Confusion matrix result on Public Garbage Dataset.

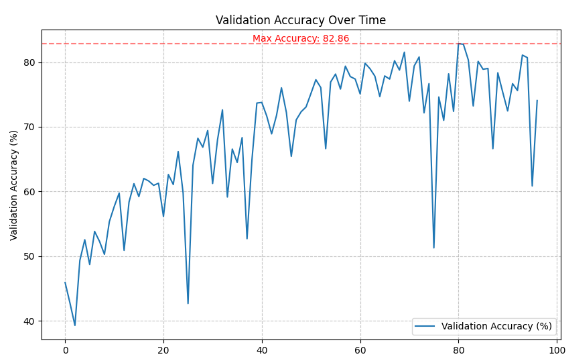

Fig. 8. Accuracy on Public Garbage Dataset training phase.

[11] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2425–2433.

[12] Y. Bengio, P. Simard, and P. Frasconi, "Learning Deep Architectures with a Softmax Activation Function," in *Journal of Machine Learning Research*, vol. 3, pp. 157–191, 2013.

[13] M. Courbariaux, Y. Bengio, and J. David, "BinaryConnect: Training Deep Neural Networks with Binary Weights During Propagation," in *Proc. of Neural Information Processing Systems (NeurIPS)*, 2015.

[14] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer, Cham, 2016, pp. 525–542. doi: 10.1007/978-3-319-46493-0_32.

[15] C. Yuan and S. S. Agaian, "A comprehensive review of Binary Neural Network," *Artificial Intelligence Review*, vol. 56, pp. 12949–13013, 2023. [Online]. Available: https://doi.org/10.1007/s10462-023-10464-w

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[17] Y. Bengio, Y. LeCun, and G. E. Hinton, "Estimating or propagating gradients through stochastic neurons for classification and generation," *arXiv preprint arXiv:1305.5207*, 2013.

[18] F. Ramzan, M. U. Khan, A. Rehmat, S. Iqbal, T. Saba, A. Rehman, and Z. Mehmood, "A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks," *Journal of Medical Systems*, vol. 44, Dec. 2019, doi: 10.1007/s10916-019-1475-2.

[19] E. Liner and R. Miikkulainen, "Improving neural network learning through dual variable learning rates," *arXiv preprint arXiv:2002.03428*, 2020. doi: 10.48550/arXiv.2002.03428.

[20] A. Krizhevsky, "CIFAR-10 dataset," 2009. Available at: https://www.cs.toronto.edu/ kriz/cifar.html.

[21] M. Abla, "Garbage Classification Dataset," Kaggle, 2023. Available at:https://www.kaggle.com/datasets/mostafaabla/garbage-classification/data.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," arXiv preprint arXiv:1502.01852, 2015.