Real-Time Deep Learning Framework for Dermatology Image Classification on Low-Power Embedded Devices

Yixin Li
Computer Science
NC State University
Raleigh, USA
yli223@ncsu.edu

Ning Sui
Biochemistry
NC State University
Raleigh, USA
nsui@ncsu.edu

Chengan Guo
Info. and Comm. Eng.
Dalian Univ. of Tech.
Liaoning, China
cguo@dlut.edu.cn

Zhishan Guo
Computer Science
NC State University
Raleigh, USA
zguo32@ncsu.edu

***Abstract*—The utilization of deep learning (DL) in medical research and industry has witnessed substantial growth in recent years. A pivotal application involves employing DL for dermatology image classification tasks. However, the major challenges in such tasks, in terms of the scarcity and bias of high-quality labeled data, significantly hinder further advancement in this domain. Such data insufficiency gives rise to concerns regarding accuracy disparities across different demographic groups, which may ultimately lead to unfair outcomes. Additionally, complex and effective DL models are often unsuitable with low-power embedded devices, which hinders their usability in resource-limited environments. In this paper, we propose a DL framework to address these issues. Our major approach involves augmenting data with Gaussian white noise to generate synthetic data samples and employing knowledge distillation techniques to transfer valuable knowledge from a larger and more complex model to a smaller and more efficient counterpart. Through comprehensive experimentation on an open-access skin disease classification dataset, we demonstrate that our proposed framework significantly enhances the performance of DL models on low-power embedded devices, thereby optimizing the trade-offs among overall accuracy, fairness for different demographic groups, and inference latency on low-power embedded devices.** [1]

***Index Terms*—Deep Learning, Medical Image Classification, Synthetic Data, Knowledge Distillation, Real-time Embedded System**

I. Introduction

With the recent development of artificial intelligence technology, its applications have been penetrating various industries across society. In the medical field, one of the most crucial areas, AI has rapidly developed and engaged. Deep Learning (DL), as a subfield of AI, play an important role in dermatology clinics, assisting doctors in diagnosing skin diseases [1]. The main approach of DL involves learning from a vast amount of annotated skin disease images to understand their underlying features and provide diagnostic outcomes. However, this image classification task faces a common and challenging problem in the medical domain: the lack of high-quality labeled data. Collecting and sharing medical data from different hospitals present various difficulties, such as patient privacy concerns. The scarcity of data can lead to inaccuracies during training process. Additionally, skin disease image classification, as a typical medical imaging task, encounters further challenges. Unlike tasks requiring more specialized and precise measuring instruments, skin disease images are often captured using mobile devices. This introduces uncertainties such as varying image sizes, resolutions, and distractions from factors like hair, scars, or medical markings on the skin, posing greater challenges for deep learning. Moreover, the imbalanced distribution of skin disease data among different ethnicity can result in better model performance for certain skin tones while performing poorly on data-limited ethnic groups. To address the unfairness in medical image classification, Deng et al. [2] proposed a novel method for fair representation learning with respect to multi-sensitive attributes. They formulated this problem mathematically and propose a novel fair representation learning algorithm named FCRO, which pursues orthogonality between sensitive and target representations. Khakurel et al. [3] conduct an empirical study to investigate bias in the image classification domain based on sensitive attribute gender using deep convolutional neural networks (CNN) through transfer learning and minimize bias within the image context using data augmentation to improve overall model performance. However, the study that focuses on improving fairness among different demographic groups and balancing between fairness and accuracy is insufficient. Another challenge arises in real-world scenarios, where local clinics may not have high-power devices to execute the complex DL models. Accessing through a cloud server requires a stable internet connection, and maintaining such powerful devices for remote access is expensive and inefficient. Hence, our study aims to develop a high-performing lightweight DL model that achieves high

[1]Our code and experiments can be reproduced by utilizing the details provided in the Methodology section on image preprocessing and augmentation, model architecture, and training configurations. (https://github.com/yixinli19/Dermatology-image-classification). The MobileNet V3 is available at (https://pytorch.org/vision/stable/models/mobilenetv3.html). The Swin Transformer V2 is available at (https://pytorch.org/vision/stable/models/swin_transformer.html). The dermatology image dataset is available upon request from ESFair 2023. They can be reached at: https://esfair2023.github.io/ESFair/.

overall accuracy, fairness across different demographic groups, and low inference latency on low-power embedded devices.

The remaining sections are organized as follows: Section II explains the targeted problem. Section III elaborates on the employed methodology, mainly comprising image preprocessing, synthetic data generation, and knowledge distillation. Then, Section IV describes our experimental settings, baselines, results, and corresponding discussions. Subsequently, Section V presents the related work in dermatology image classification and real-time DL on low-power devices. Lastly, Section VI concludes the work and points out future research directions.

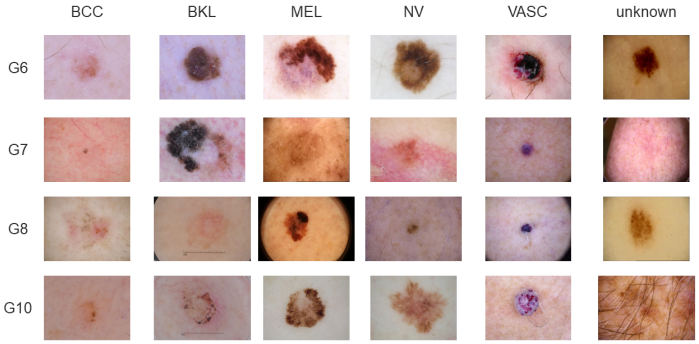

Fig. 1: Sample skin disease images. There are several features shown in this figure: (1) The images have different sizes. (2) Diseases have different sizes, positions, and colors. (3) Some of the images have distracting information, like hair and marks.

II. Problem

Our study draws inspiration from the skin disease classification task presented at the Tiny and Fair ML Design Contest during Embedded Systems Week 2023 [4]. The rapid adoption of machine learning techniques, particularly deep learning, in diverse medical domains owing to the democratization of AI. Dermatology, benefiting from accessible skin lesion datasets, has emerged as a prominent area of application.

A. Dataset

The dataset obtained from the ESFair contest comprises skin disease images categorized into six different classes, namely: (1) Basal-cell carcinoma (BCC), (2) Benign Keratosis-Like Lesions (BKL), (3) Melanoma (MEL), (4) Nevus (NV), (5) Vascular and Anatomic Skin Changes (VASC), and (6) unknown. Additionally, the dataset consists of four subgroups: G6, G7, G8, and G10, based on skin tones. The images within the dataset exhibit diverse sizes, ranging from 640x480 to 6000x4000 pixels. A detailed description of the number of images for each class and subgroup is provided in Table I. The total sample sizes for the four subgroups are 3565, 2350, 3291, and 2007, respectively, while the sample sizes for each disease class (BCC, BKL, MEL, NV, VASC, unknown) are 1231, 982, 1537, 2206, 49, and 5207. Figure 2 illustrates the bias in this dataset, such that the $Unknown$ class in G7 accounts for 20% of the entire dataset, but it takes 59% in G6. Furthermore, Figure 1 presents sample image data, illustrating the variations in image sizes and the presence of distracting information, such as hair and marks. Additionally, skin diseases exhibit diverse shapes, colors, and positions, adding to the complexity of the dataset.

TABLE I: The number of images in each class and group.

	BCC	BKL	MEL	NV	$VASC$	$unknown$
G6	68	63	272	1039	4	2118
G7	641	546	521	138	17	487
G8	303	242	368	708	22	1648
G10	219	131	376	321	6	954
$Total$	1231	982	1537	2206	49	5207

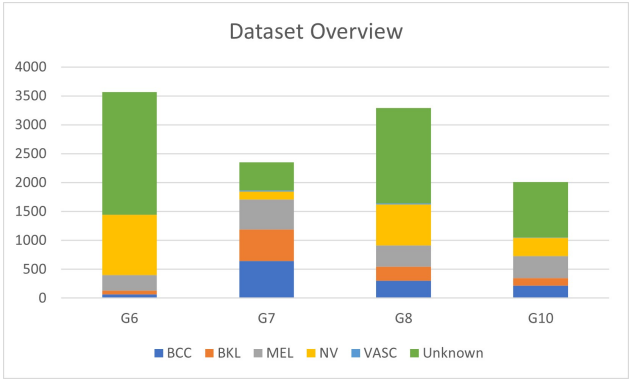

Fig. 2: An overview of the dataset for each subgroup.

B. Main Objectives

Unlike general images, such as those in ImageNet [5], skin lesion datasets often manifest biases due to uneven representation of skin tones. Acknowledging this disparity, Alexander, CEO of First Derm, notes that images of black skin comprise only a small portion (5-10%) of their database, with even fewer samples representing other minority groups, such as Asians and Hispanics [4]. This inherent bias raises significant concerns regarding the use of machine learning in dermatology, as it may yield high overall accuracy but significantly lower accuracy for specific groups. The repercussions of such bias extend widely, ranging from accidents in autonomous driving to linguistic discrimination in language translation, and even life-threatening misdiagnoses in healthcare [4]. In addition, dermatology image classification poses unique challenges compared to medical image analyses of other types, such as CT and MG, which use precise scanning devices to record the data. Skin disease images are typically captured using mobile devices, such as smartphones [6]. The variation in image quality from different mobile devices and under various light conditions can hinder the performance of machine learning-based classification tasks.

Another problem arises in low-power embedded systems. Due to the massive training parameters, DL models can usually achieve relatively high performances in certain tasks. However, in real-world scenarios, high-power devices are limited

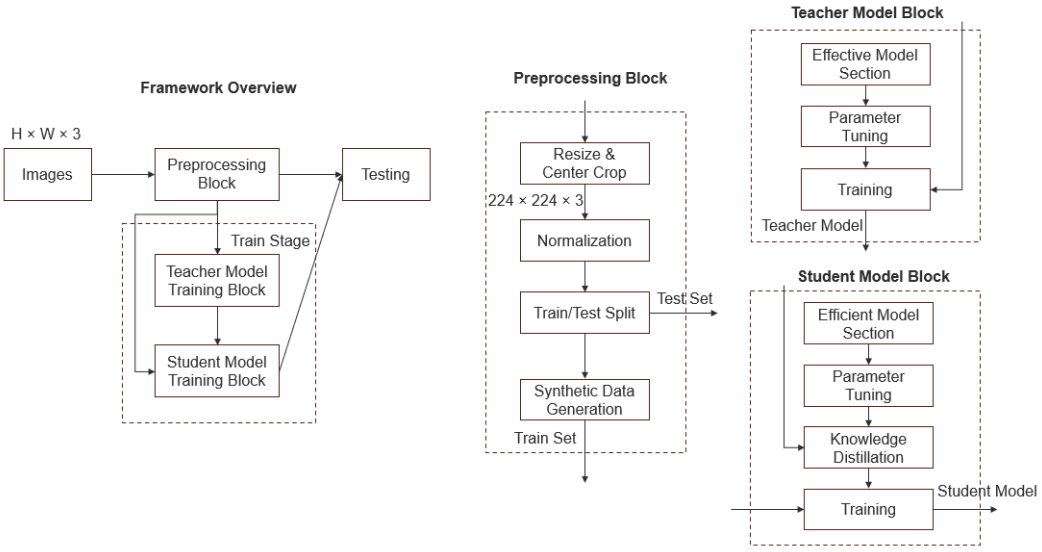

Fig. 3: An overview of the proposed framework.

by costs and availability. In resource-restricted environments where real-time diagnosis is needed, complex DL models are infeasible to accomplish this task. Therefore, to address the limitation of deep learning deployment on low-power embedded devices, efficient and domain-specific neural networks are needed.

To address these issues, we set the DL performance objectives in three critical aspects:

- Overall accuracy, which quantifies the percentage of correct predictions out of the total samples;
- Fairness, measured by examining differences in accuracy across subgroups and employing Statistical Parity Difference (SPD); and
- Inference Latency, representing the time taken for making predictions.

The primary objective is to design DL models capable of achieving high accuracy performance while effectively balancing the accuracy between each subgroup. Moreover, the study aims to execute these neural networks on low-power embedded devices, ensuring relatively fast inference speed.

III. Methodology

In this study, we proposed a novel and comprehensive framework for designing DL models, including preprocessing images, generating synthetic data, tuning parameters, and applying knowledge distillation techniques to optimize overall performances.

A. Framework

The proposed framework is presented in Figure 3, where the initial step involves resizing and normalizing the input raw images to 224x224 pixels in RGB format, considering variations in heights and widths. Next, the dataset is partitioned into training and testing subsets, while the training set is updated and balanced by the generation of synthetic data using white Gaussian noises. During the training phase, a highly effective teacher model, Swin Transformer [7], is employed to establish a foundation of knowledge. This knowledge is then transferred to a more efficient student model using knowledge distillation techniques. Notably, the selection of the Swin Transformer as the teacher model is justified due to its outstanding performance in efficiently modeling long-range dependencies in images while maintaining computational efficiency. It employs a hierarchical design that divides the image into non-overlapping patches and utilizes both local and global attention mechanisms to capture contextual information effectively. This architecture enables Swin Transformer to handle large images and achieve state-of-the-art performance on various computer vision tasks, making it a promising choice for tasks with high-resolution inputs and complex dependencies [7]. As a complement to this, the student model, MobileNet [8], is chosen for its lightweight and high efficiency.

B. Image Preprocessing

Due to the diverse characteristics present in image data, including variations in size, shape, color, and potential distractions, a preprocessing step has been taken to ensure all image data are compatible with the deep learning process. As depicted in Figure 4, we initially resize all images to a standardized dimension of 224x224 pixels. Subsequently, we compute the normalization parameters, encompassing the standard deviation and mean value of the complete dataset. Finally, we transform the normalized images into RGB values for each pixel, facilitating integration into the following stages of our proposed framework. This preprocessing pipeline enhances the effectiveness of our neural network-based approaches for image analysis tasks.

C. Class Weighting

In response to the challenge posed by the imbalanced dataset, we introduced modifications to the class weighting

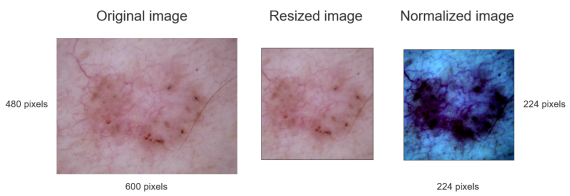

Fig. 4: An example showing how the image data is processed. The image in this example is from G10 with the label BCC. This image size is 600x480 pixels. The first step of preprocessing is to resize the image to 224x224 pixels, and then normalize with the standard deviation and mean values calculated from the entire dataset.

within the DL to afford greater attention to the specific classes. As evidenced in Table I, it becomes apparent that the $Unknown$ class contains a significantly larger number of samples compared to all other classes, while the images in the $VASC$ class only accounts for 0.4% of the entire dataset. Hence, in the context of this scenario, it is imperative to assign a higher class weighting to the minority classes with interests. By adopting this approach to class weighting, we seek to optimize the DL's performance in tackling the imbalanced dataset. In this study, we changed the class weights in the training pipeline to [15, 15, 15, 15, 1, 5] for BCC, BKL, MEL, NV, $Unknown$, and $VASC$. This allows the DL models to put more attention on the minority classes with interests. Due to the limited sample size of $VASC$, we set the weight to 5 and all other class weights to 15 except $Unknown$.

D. Synthetic Data Generation

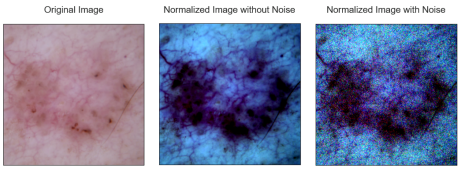

Fig. 5: An example comparing the normalized images with and without Gaussian white noises. In this example, the standard deviation is 0.2.

We add the Gaussian white noises to the original RGB values to generate new synthetic samples. To be specific, Gaussian white noise is a fundamental concept used in statistical signal processing and models for communication channels [9]. Incorporating Gaussian white noise into image data involves several steps. Initially, the RGB value at each pixel is determined. Then, the noise is calculated, adhering to a Gaussian distribution with zero mean and constant variance. The computed noise is then added to the original pixel value. This process is performed for all pixels in the image, resulting in a new sample with the added Gaussian white noise. The formula is shown in Equation 1 [9].

$$n(x, y) = \eta(x, y) \cdot \sigma \tag{1}$$

where $n(x, y)$ is the Gaussian white noise value at the pixel position (x, y), $\eta(x, y)$ is a random value drawn from a standard Gaussian distribution with a mean of zero and a variance of one, and σ is the standard deviation of the desired Gaussian white noise. It controls the magnitude of the noise to be added to the image.

Figure 5 provides a visual representation of the impact of Gaussian white noise on image processing. The addition of noises to the image enables us to regulate the size of the training data, thereby mitigating the adverse effects of biased data.

E. Knowledge Distillation

Knowledge distillation, an essential technique for transferring knowledge from a larger neural network to a smaller one, plays an important role in optimizing the balance between effectiveness and efficiency. In the typical architecture of neural networks, class probabilities are generated using a "softmax" output layer, which converts the logit value, represented as z_i, corresponding to each class, into a probability denoted as q_i. This transformation involves a comparison between z_i and other logits [10]. The general formula is shown as:

$$q_i = \frac{exp(\frac{z_i}{T})}{\Sigma_j exp(\frac{z_j}{T})} \tag{2}$$

where T is a temperature that is normally set to 1. The higher value of the temperature T produces a softer probability distribution over classes.

Two built-in loss functions from PyTorch are used in our knowledge distillation technique.

- CrossEntropyLoss [11]: The traditional cross-entropy loss (ce_loss): This measures the difference between the student's predictions and the ground-truth labels. It aligns with the logistic loss employed on the neural network outputs when the softmax function is utilized. [12].
- KLDivLoss [13]: The Kullback-Leibler divergence loss (kl_loss). This loss function captures the similarity between the softened predictions of the student and teacher models. The softened predictions are obtained by applying softmax to the model logits divided by the temperature parameter.

The overall loss is then computed as a weighted sum of the cross-entropy loss and the KL divergence loss. The weight of each loss is determined by alpha. And the temperature parameter, T, is squared in the KL loss term ($T^2 \cdot$ kl_loss) to balance the magnitudes of the two loss components. The detailed knowledge distillation loss function is shown below.

$$ce\_loss = cross\_entropy(student\_pred, target) \tag{3}$$

$$softmax(z_i) = \frac{exp(z_i)}{\Sigma_j exp(z_j)} \tag{4}$$

$$soft\_t = softmax(\frac{teacher\_pred}{T}) \tag{5}$$

$$soft\_s = log(softmax(\frac{student\_pred}{T})) \quad (6)$$

$$kl\_loss = kl\_divergence(soft\_s, soft\_t) \quad (7)$$

$$loss = (1 - alpha) \cdot ce\_loss + alpha \cdot T^2 \cdot kl\_loss \quad (8)$$

where

- student_pred: Student model predictions
- teacher_pred: Teacher model predictions
- target: Ground-truth labels
- T: Temperature to soften the probabilities
- alpha: A hyperparameter controlling the trade-off between cross-entropy and knowledge distillation losses

In summary, this knowledge distillation loss function allows the student model to learn from both its own predictions (cross-entropy loss) and the predictions of a larger teacher model (kl divergence loss) to improve its performance on classification tasks, particularly when dealing with imbalanced datasets.

F. Evaluation Metric

Three key performance indicators (KPIs) have been considered for evaluating the performance of our proposed work.

- **Overall accuracy**: the number of correct predictions out of the total samples.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (9)$$

- **Fairness**: Based on the accuracy for each subgroup and Statistical Parity Difference (SPD) [14]. The formula of SPD is:

$$P(\tilde{Y} = 1|A = \text{minority}) - P(\tilde{Y} = 1|A = \text{marjority}) \quad (10)$$

where $\tilde{Y}$ are the model predictions, and A is the group of sensitivity attributes.
For the minority group (the group that has the minimum samples), the SPD is 0. After that operation, we will have the table of the SPD for each group. For the unfairness score, the formula is:

$$Fairness = 1 - \frac{(0.2 - \sum \forall g_i \in G\{|SPD_i|\}/N)}{0.2} \quad (11)$$

where N is the number of subgroups.

- **Inference Latency**: We run 10 iterations to get the average latency L (in s) on Raspberry Pi 4 with 100 randomly selected images from testing dataset.

$$L = \frac{1}{10}\sum_{i=1}^{10} L_i \quad (12)$$

where L_i means the latency at i-th iteration. The average inference latency L will be recorded. The latency score will be normalized by the following formula.

$$L_n = 1 - \frac{L - L_{min}}{L_{max} - L_{min}} \quad (13)$$

where L_{min}=0s, and L_{max}=100s.

The overall performance is calculated by the sum of these three scores with equal weights. Note that such uniformly balanced performance calculation of accuracy, fairness, and latency was proposed by the ESFair competition and adopted directly in this work.

$$\text{Performance} = \frac{1}{3}(Accuracy + Fairness + Latency) \quad (14)$$

IV. Experiment

A. Experiment Setting

In our experiments, we used one NVIDIA GeForce GTX 1080 Ti for training purposes, and a Raspberry Pi 4 Model B with 4GB of onboard RAM to test our neural networks. Initially, we divided the dataset into training and testing sets using the 80-20 Train/Test split technique. Then, applying a 5-fold cross validation technique to verify the performance. The epoch size is 400 and the batch size is set to 16 due to the limitation of the hardware used in this experiment.

B. Baselines

Two state-of-art deep learning models in the image classification domain are adopted in our experiments, in terms of Swin Transformer [7] and MobileNet [8]. The baselines are the following:

- Swin Transformer V2 [7]: A hierarchical vision transformer for computer vision tasks that utilizes shifted windows and transformer blocks to efficiently process high-resolution images.
- MobileNet V3 [8]: A lightweight and efficient convolutional neural network architecture designed for mobile and embedded devices, optimized to perform image recognition tasks with minimal computational resources.
- Our proposed approach: Training Swin Transformer as a teacher model, then transferring knowledge to the student model, MobileNet, with our proposed knowledge distillation loss function.

Additionally, we conduct a comprehensive evaluation of synthetic data generation across five different settings shown in Table II. In specific terms, we investigated two different variations: training data sizes and standard deviations for Gaussian white noises. The baseline scenario involves solely employing the training set without any synthetic data generation. Conversely, the subsequent experimental settings explore alternative configurations. For instance, in the context of [2500, 2500, 2500, 2500, 2500, 2500], this shows that the data sizes for all six classes within each subgroup, comprising both original and synthetic data, is set to 2500.

C. Result

Table III presents the comparisons between our proposed work with the other two baselines. After training each model, the size of MobileNet is 6.041MB, Swin Transformer is 113.53MB, and our proposed work is 6.039MB. Due to the small adjustment on parameters in each setting shown in

TABLE II: The experiment settings for training purposes

Setting	Data size for each subgroup	STD
S1	Original training data	N/A
S2	[2500, 2500, 2500, 2500, 2500, 2500]	0.03
S3	[2500, 2500, 2500, 2500, 2500, 2500]	0.05
S4	[4000, 4000, 4000, 4000, 2500, 2500]	0.03
S5	[4000, 4000, 4000, 4000, 2500, 2500]	0.05

Table II, the model sizes remain the same for each experiment setting. The scores in Table III are calculated by the average scores using the five cross-validation technique. All the experiments are conducted on a Raspberry Pi 4 Model B with 4GB of onboard RAM.

TABLE III: Experiment Results Overview

Model	Setting	Accuracy	Fairness	Latency	Overall
MobileNet (6.041MB)	S1	0.723	0.636	**0.924**	0.761
	S2	0.725	0.641	0.920	0.762
	S3	0.708	0.629	0.916	0.751
	S4	0.795	0.768	0.910	0.824
	S5	0.780	0.721	0.911	0.804
Swin Transformer (113.53MB)	S1	0.827	0.744	0.093	0.555
	S2	0.834	0.758	0.091	0.561
	S3	0.818	0.721	0.091	0.543
	S4	**0.867**	**0.846**	0.093	0.602
	S5	0.843	0.799	0.090	0.577
Ours (6.039MB)	S1	0.801	0.692	0.909	0.8
	S2	0.812	0.716	0.903	0.810
	S3	0.798	0.687	0.911	0.799
	S4	0.843	0.810	0.906	**0.853**
	S5	0.811	0.787	0.909	0.836

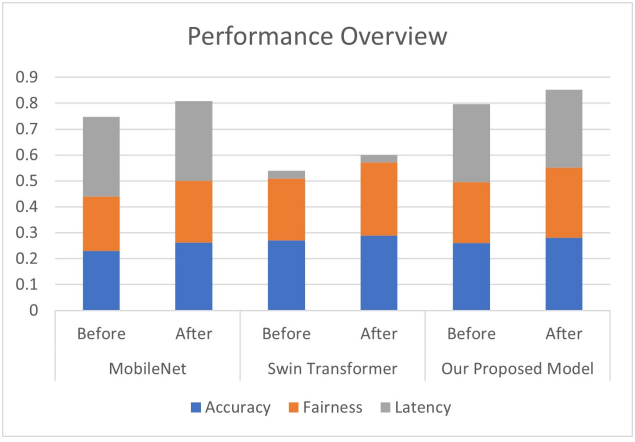

Fig. 6: Baselines include MobileNet, Swin Transformer, and our proposed work. The overall performance is calculated by the sum of accuracy, fairness, and inference latency. In this figure, each model produces two results, including the performance before and after adding synthetic data.

Figure 6 depicts that our proposed work with synthetic data generation technique reaches the highest overall performance. In the aspects of accuracy and fairness, Swin Transformer achieves the highest scores due to its complex and effective architecture, where the shifted window mechanism can analyze each portion of the images and learn the deeper features than the normal convolutional neural network architectures. However, this design significantly slows down the inference speed on Raspberry Pi 4. MobileNet, on the other hand, has a lightweight architecture that can run fast on low-power devices but is weak at accuracy and fairness. Compared to these two baselines, our proposed work well-balanced the trade-offs between effectiveness and efficiency. In addition, Figure 7 shows the confusion metrics for our proposed work in S1 and S4. Each matrix shows the results for a subgroup. The accuracy for the subgroups in S1 are 0.844 (G6), 0.716 (G7), 0.826 (G8), and 0.794 (G10). For S4, the accuracy are 0.882 (G6), 0.753 (G7), 0.868 (G8), and 0.848 (G10), and the overall accuracy increased from 0.801 to 0.843.

D. Discussions

Based on the experimental results shown in Figure 6, our proposed work achieved the highest performance compared to the other baselines. We analyze these results in three aspects: **Comparison in Knowledge Distillation.** Without any adjustment in training the DL models, the model with the best performance in accuracy and fairness is Swin Transformer. But this model is incapable of running on low-power embedded devices. The model size is 113.53MB, which is 18.8 times larger than MobileNet with a size of 6.041MB. Without high-power computing resources supported, it takes around 95 seconds to classify 100 images. In contrast, MobileNet executes with the fastest inference speed in our experiment, and it can classify 100 images in less than 10 seconds. However, these two baselines still cannot balance well between effectiveness and efficiency. Our proposed work, instead, has a 10% increase in accuracy, an 8.8% increase in fairness, and a 1.6% loss in inference latency compared to MobileNet. As for the comparison with Swin Transformer, even though the accuracy of our work decreased by 3%, and the fairness score decreased by 7%, the inference latency score is 9.8 times higher. Our proposed work shows the best ability of balancing these three KPIs, in terms of accuracy, fairness, and inference latency.

Comparison in Synthetic Data Generation. Compared to S1, the performances in S2 increased in all aspects, but the average increases in the three KPIs are around 1%. However, when comparing S1 with S4 for all baselines, the overall performance increased by 5%, the accuracy increased by 4%, and the fairness increased by 11%. This improvement points out a direction on how to deal with biased datasets. Another finding shows in the comparison between S2 with S3, and S4 with S5, where the only difference is the standard deviation (STD) in Gaussian white noise generations. With the higher STD, the noises in the image data increase, which leads to a loss in image features. As shown in the result, with an STD of 0.05, the performances decreased significantly in all KPIs.

Comparison in Each Subgroup. As shown in Figure 7, the left confusion metrics represent the result for our proposed work in S1. The right metrics are the results from S4. Before applying our DL mechanisms, the overall accuracy is

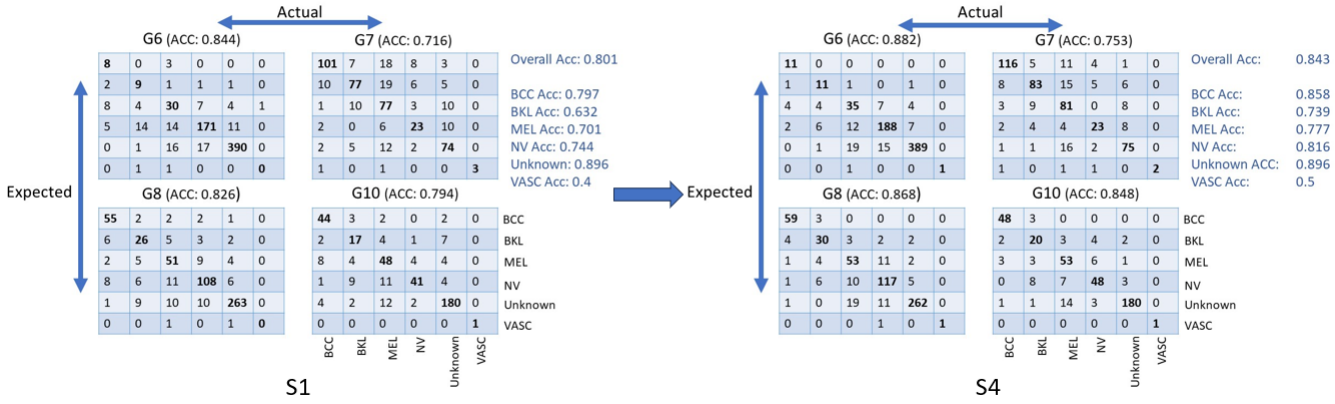

Fig. 7: The comparison between S1 and S4 with our proposed work shown in confusion metrics.

0.801. The major obstacles are the low performances in the classes other than $Unknown$. Due to the images in $Unknown$ taking a large portion of the entire dataset, the DL models trained on this data have a biased higher performance in classifying $Unknown$ images but are weak at other classes. With our proposed work, the overall performance increases to 0.843, and each class's accuracy has significant improvements. Specifically, the average improvement in BCC, BKL, MEL, and NV is from 0.718 to 0.798. This result safely proves that our work can be beneficial in balancing the biased dataset as well as improving the overall performance.

According to the experiment results, it is obvious that training DL models with our proposed framework can significantly improve the performances in multiple dimensions. Our research is one of the first studies to analyze the performances combining accuracy, fairness, and inference latency, especially in AI healthcare systems. By effectively balancing the trade-offs among these three KPIs, our proposed framework has the potential to be adapted to any other DL tasks in real-world scenarios.

V. Related Work

Over the past decades, the healthcare system has experienced a significant rise in the demand for medical image classification services, encompassing various imaging modalities such as Radiography, Computed Tomography (CT), Mammography Images (MG), Ultrasound images, Magnetic Resonance Imaging (MRI), Magnetic Resonance Angiography (MRA), and Positron Emission Tomography (PET) [15]. To meet these demands, Deep Learning technology has emerged as a powerful tool. However, medical image classification faces a common challenge, the scarcity of high-quality labeled data due to limited samples and expensive labeling processes. This challenge is particularly critical in dermatology image classification, where skin diseases, affecting nearly one-third of the world's population, are often underestimated despite their visibility [16]. Additionally, due to the complexity of powerful neural networks, the practicality of their execution on low-power embedded devices is typically limited. Therefore, researchers have put efforts into addressing these issues in the domain of dermatology image classification and real-time DL execution on low-power devices.

A. Dermatology image classification

Yanagisawa et al. [17] developed a convolutional neural network (CNN) model for skin image segmentation, leading to a skin disease image dataset suitable for the classification of multiple skin diseases. The CAD system achieved approximately 90% sensitivity and specificity in distinguishing atopic dermatitis from malignant diseases and complications. However, the authors claimed that the constraint lies in the restricted number of images, which may have introduced a bias in the machine learning-based attribute extraction for the identified skin diseases. In another study, He et al. [18] proposed SEECNN, a Genetic Algorithm (GA) with a simple encoding scheme for evolving both the architectures and weight initialization values of CNNs to address image classification problems. The limitation includes the lack of research on handling imbalanced dataset. Mijwil et al [19] worked on analyzing more than 24,000 skin cancer images using three ConvNet architectures (InceptionV3, ResNet, and VGG19). Unfortunately, this study did not evaluate the performance in terms of fairness and inference speed.

B. Real-time DL execution on low-power devices

Li et al. [20] introduced EfficientFormer, which enhances ViT-based models by identifying inefficient designs, introducing a dimension-consistent pure transformer, and applying latency-driven slimming. Goel et al. [21] introduced the Modular Neural Network Tree architecture to improve accuracy and reduce redundancy and energy consumption in DL models. This architecture utilizes multiple smaller DL modules for image classification, based on a novel visual similarity metric. Experimental results on Raspberry Pi 3 and Raspberry Pi Zero demonstrated significant reductions in memory requirements, inference time, energy consumption, and operations compared to existing DL architectures. Additionally, Chang et al. [22] proposes a low-power, memory-efficient, and high-speed ML

algorithm for classifying smart home activity data in resource-constrained environments. Jafari et al. [23] introduce SensorNet, a scalable and low-power embedded deep convolutional neural network (DCNN) designed for classifying multimodal time series signals. When implemented on NVIDIA Jetson TX2 SoC (CPU + GPU) and compared to TX2 single-core CPU and GPU implementations, FPGA-based SensorNet achieves a 15 and 4 improvement in energy consumption. However, little effort has been made in balancing the trade-offs between accuracy and inference speed.

VI. Conclusion and Future Work

In this research, we present a comprehensive Deep Learning (DL) framework for the real-time dermatology image classification task on low-power embedded systems. Compared to baseline models, our approach combines image preprocessing, data augmentation, and knowledge distillation, resulting in the highest overall performance which consists of overall accuracy, fairness for different demographic groups, and inference latency on low-power embedded devices. The versatility of our framework extends its potential to various other DL tasks, enhancing feasibility on low-power embedded devices across diverse applications. Our future work aims to further refine the framework's performance by exploring and developing more effective models and techniques, fine-tuning the parameters in knowledge distillation and synthetic data generation to find the optimal performances, and implementing a product-level assistant tool for dermatology clinics.

Acknowledgment

We thank the organizers of the Tiny and Fair ML Design Contest during Embedded Systems Week 2023 for providing the labeled dataset and their guidance in the early phase of our study. This work is partially supported by startup funding from NC State University.

References

[1] Y.-W. Chen and L. Jain, *Deep Learning in Healthcare Paradigms and Applications: Paradigms and Applications*, 01 2020.

[2] W. Deng, Y. Zhong, Q. Dou, and X. Li, "On fairness of medical image classification with multiple sensitive attributes via learning orthogonal representations," *arXiv:2301.01481*, 2023.

[3] U. Khakurel and D. B. Rawat, "On the performance of machine learning fairness in image classification," in *Defense + Commercial Sensing*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258392941

[4] L. Yang, Q. Lou, Y. Sheng, and J. Yang, "Fair and intelligent embedded system challenge at esfair 2023. tiny and fair ml design - embedded systems week," Embedded Systems Week, accessed: Aug. 2023. [Online]. Available: https://ESFair.org/tiny-and-fair-ml-design/

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[6] K. T. Ashique, F. Kaliyadan, and S. J. Aurangabadkar, "Clinical photography in dermatology using smartphones: An overview," *Indian dermatology online journal*, vol. 6, no. 3, pp. 158–163, 2015. [Online]. Available: https://doi.org/10.4103/2229-5178.156381

[7] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin transformer v2: Scaling up capacity and resolution," *arXiv:2111.09883*, 2022.

[8] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," *arXiv:1905.02244*, 2019.

[9] A. V. Balakrishnan and R. R. Mazumdar, "On powers of gaussian white noise," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7629–7634, nov 2011. [Online]. Available: https://doi.org/10.1109%2Ftit.2011.2158062

[10] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[11] "Crossentropyloss pytorch 2.0 documentation," PyTorch, accessed: Aug. 2023. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

[12] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," *arXiv:2304.07288*, 2023.

[13] "Kldivloss pytorch 2.0 documentation," PyTorch, accessed: Aug. 2023. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.KLDivLoss.html

[14] "Explore fairness," accessed: Aug. 2023. [Online]. Available: https://www.mathworks.com/help/risk/explore-fairness-

[15] M. Puttagunta and S. Ravi, "Medical image analysis based on deep learning approach," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 24 365–24 398, 2021.

[16] C. Flohr and R. Hay, "Putting the burden of skin diseases on the global map," *The British journal of dermatology*, vol. 184, no. 2, pp. 189–190, 2021. [Online]. Available: https://doi.org/10.1111/bjd.19704

[17] Y. Yanagisawa, K. Shido, K. Kojima, and K. Yamasaki, "Convolutional neural network-based skin image segmentation model to improve classification of skin diseases in conventional and non-standardized picture images," *Journal of Dermatological Science*, vol. 109, no. 1, pp. 30–36, 2023. [Online]. Available: https://doi.org/10.1016/j.jdermsci.2023.01.005

[18] X. He, Y. Wang, X. Wang, W. Huang, S. Zhao, and X. Chen, "Simple-encoded evolving convolutional neural network and its application to skin disease image classification," *Swarm and Evolutionary Computation*, vol. 67, p. 100955, 2021. [Online]. Available: https://doi.org/10.1016/j.swevo.2021.100955

[19] M. M. Mijwil, "Skin cancer disease images classification using deep learning solutions," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26 255–26 271, 2021. [Online]. Available: https://doi.org/10.1007/s11042-021-10952-7

[20] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "Efficientformer: Vision transformers at mobilenet speed," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 12 934–12 949.

[21] A. Goel, S. Aghajanzadeh, C. Tung, S. Chen, G. K. Thiruvathukal, and Y. Lu, "Modular neural networks for low-power image classification on embedded devices," *ACM Transactions on Design Automation of Electronic Systems*, vol. 26, no. 1, pp. 1–35, 2021. [Online]. Available: https://doi.org/10.1145/3408062

[22] J. Chang, M. Kang, and D. Park, "Low-power on-chip implementation of enhanced svm algorithm for sensors fusion-based activity classification in lightweighted edge devices," *Electronics (Basel)*, vol. 11, no. 1, p. 139, 2022. [Online]. Available: https://doi.org/10.3390/electronics11010139

[23] A. Jafari, A. Ganesan, C. S. K. Thalisetty, V. Sivasubramanian, T. Oates, and T. Mohsenin, "Sensornet: A scalable and low-power deep convolutional neural network for multimodal data classification," *IEEE Transactions on Circuits and Systems. I, Regular Papers*, vol. 66, no. 1, pp. 274–287, 2019. [Online]. Available: https://doi.org/10.1109/TCSI.2018.2848647