DDSNet: A Lightweight Dense Depthwise Separable Network for Tumor Classification

An Huang
University of Nevada, Las Vegas
Las Vegas, Nevada, USA
huanga7@unlv.nevada.edu

Junggab Son
University of Nevada, Las Vegas
Las Vegas, Nevada, USA
junggab.son@unlv.edu

Zuobin Xiong*
University of Nevada, Las Vegas
Las Vegas, Nevada, USA
zuobin.xiong@unlv.edu

Abstract

Deep learning-based medical image processing plays a significant role in modern computer-aided diagnosis, which facilitates doctors in various disease analysis. However, most researchers focus on the accuracy of medical image classification tasks with ever-increasing model size and the number of parameters but overlook the high diagnostic costs and model efficiency. To reduce such costs and broaden the application scenarios, a low-cost and efficient medical image classification is imperative. To achieve this goal, this paper designs a lightweight model, named Dense Depthwise Separable Network (DDSNet), which combines the merits of Dense Convolution Network and Depthwise Separable Convolution, rendering a low-cost and efficient medical imaging. Moreover, a quantization-based method is invented to deploy the proposed model on real-world IoT devices by converting the original model to an integer-type model while maintaining its classification performance. Extensive experiments are conducted on four cancer image datasets on the IoT device, showing the promising performance of this proposed method against 5 baseline models, including data visualization and interoperability aspects. Notably, compared to DenseNet, the proposed model is about 32× smaller and 5× faster after quantization, with a competitive classification accuracy preserved. Our code is available at https://github.com/OldDreamInWind/DDSNet.

CCS Concepts

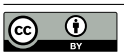
• **Computing methodologies** → **Neural networks**; **Object recognition**; • **Applied computing** → *Bioinformatics*.

Keywords

Neural Networks, Lightweight Model, Medical Image Processing, Tumor Classification

ACM Reference Format:
An Huang, Junggab Son, and Zuobin Xiong. 2025. DDSNet: A Lightweight Dense Depthwise Separable Network for Tumor Classification. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31-April 4, 2025, Catania, Italy*. ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3672608.3707780

*Corresponding Author.



SAC '25, March 31-April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/10.1145/3672608.3707780

1 Introduction

Recently, various deep learning models have been introduced for computer vision, showing remarkable success across different domains. For instance, the classic LeNet [15] demonstrated significant potential in document recognition. The VGG [22] introduced a very deep convolutional network method that excelled in the ILSVRC-2012 dataset. ResNet [4] achieved state-of-the-art performance on the ImageNet dataset with its innovative residual networks. DenseNet [6] approaches to image classification through dense connections showcased high performance while requiring less memory and computational resources. These different methods further promote the development of deep learning in medical image diagnosis. Unlike traditional medical image processing methods that depend on manually designing feature extraction, such as Canny or HOG, deep learning-based methods provide an end-to-end diagnosis paradigm, using automated and efficient feature extraction and achieving outperforming performance. On top of that, many researchers [8, 12, 27, 33] have tried to apply deep learning methods to medical images and gotten exciting results.

However, along with these advancements, deep learning models have become more complex, which requires high computation costs and more energy consumption. This hinders, to some extent, the popularization of related technologies to broader application scenarios. For example, Sitaula et. al. [23] utilized VGG-16 to classify the COVID-19 chest X-ray image, and Wu et. al. [29] combined Transformer and CNN on medical image classification tasks, both achieved high performance. These complex models require several gigabytes of space to store and run, which is hard to apply on many medical hardware with limited computation power and storage. Moreover, depending on the slice size, a single H&E-stained histological image from a patient may consist of thousands of sub-images that require analysis [16]. Consequently, enhancing the detection efficiency for each individual sub-image can lead to a significant cumulative improvement in the overall diagnostic process. From this perspective, reducing the parameters and size of the models and improving their operating efficiency should be a pressing concern, so that these methods and services can be more accessible and more general users can get benefits from them.

In this paper, we focus specifically on improving the efficiency of classification tasks in cancer images using deep neural networks with preserved classification accuracy. Our proposed method Dense Depthwise Separable Network (DDSNet) is inspired by Dense Convolution Networks (DenseNet) and Depthwise Separable Convolution (D-S Conv) that can reach the balance between performance and computational efficiency. Furthermore, we incorporate quantization techniques to transform the original float-number model

into an integer-number model by optimizing the sensitive quantization scale parameter. Additionally, the quantized integer-number model is deployed on real-world IoT devices with limited computational resources for realistic environment testing. Our approach has demonstrated significant improvements on resource-limited devices in benchmarks for cancer image classification, such as Gastrointestinal Cancer (GC) [10, 11], PatchCamelyon (PCam) [2, 26] and other datasets. Moreover, we enhance the interpretability of our model through visual analysis, revealing the underlying decision-making processes of the deep neural networks. The main contributions of our paper are summarized as follows:

- We propose a lightweight model **DDSNet** with much fewer parameters and a smaller model size. It combines dense convolution network and depthwise separable convolution layer, which is fast and effective in cancer image classification tasks.
- We utilize a quantization algorithm to scale the trained model and deploy the quantized model on real-world IoT devices, preserving competitive accuracy.
- We apply the visualization method to illustrate the classification boundary and analyze the model interpretability on top of model performance, which provides insight for domain experts to understand the deep learning-based solution.

The rest of this paper is organized as follows. In Section 2, some existing literature is reviewed based on their common advantages and weaknesses. We introduce the proposed method of this paper in Section 3 and conduct the performance analysis in Section 4. Finally, this paper is concluded in Section 5.

2 Related Work

2.1 Deep Neural Network in Cancer Image

Cancer diagnosis has seen rapid development, powered by the advancement of deep learning technology. These new technologies provide promising strategies for cancer image classification and detection with high accuracy. For instance, Sarwinda et. al. [21] use ResNet to detect colorectal cancer with a classification accuracy of above 80%. Gao et. al. [3] utilize 3D-CNN in lung nodule detection, which improves the performance of this special task. Other than image classification tasks, deep neural networks also work well on segmentation tasks for tumor/cancer images. Milletari et. al. [19] use V-net for volumetric (3D) image segmentation, which has proven effectiveness in segmenting tumors from surrounding tissues in 3D MRI and CT scans. Huang et. al. [7] use UNet 3+ for medical image segmentation task. Lou et. al. [18] rethink the architecture of UNet [30, 31] and optimize the convolution block with a dual channel to improve the efficiency of the segmentation task. As seen, using neural network-related learning algorithms to extract image features and classify them can effectively and conveniently improve the accuracy of classification, to assist doctors in the diagnosis of cancer.

However, these methods lack efficiency, due to the large number of parameters, and thus pose issues in deployment on resource-limited medical devices. Such challenges inspire the current trends in reducing computational costs for deployment on constrained environments and the advanced models in the precise analysis and diagnosis of cancer through medical imaging.

2.2 Lightweight Neural Network

Due to the large memory and computation requirements of deep neural networks, model lightweight has become a significant research direction. Inception v2 [24] rethinks its predecessor and optimizes the inception block, followed by Inception v3 that changes to the larger convolution kernel to expand the receptive field. Moreover, SqueezeNet [9] designs squeeze layer and expand layer to reduce the parameters of the network model, which makes the model more efficient. MobileNet [5] adds depthwise separable convolution to its model architecture and can adjust the model by width multiplier and resolution multiplier.

On the other hand, ShuffleNet [34] employs a channel shuffle operation that allows for the effective use of group convolutions, reducing the number of parameters while maintaining competitive performance on visual recognition tasks. EfficientNet [25] systematically scales all dimensions of the network with a set compound coefficient, achieving state-of-the-art accuracy with significantly fewer parameters and lower computational cost.

2.3 Deep Neural Network Quantization

Model quantization is a technique commonly used to reduce the size of a deep learning model and speed up inference time, particularly beneficial for deployment on IoT devices with limited computational resources. Park et. al. [20] presented a novel value-aware quantization while separately handling a small amount of large data in high precision. Kluska et.al. [13] performed a comprehensive study on post-training quantization that quantizes every single layer to the smallest bit width. Then, Liu et. al. [17] proposed Nonuniform-to-Uniform Quantization (N2UQ), which can maintain the strong representation ability of nonuniform methods while being efficient. Xiong et al. [32] designed a quantized federated learning method for distributed learning scenario. Yet, those methods are theoretical-oriented without considering the realistic deployment. Instead, in this work, we propose a quantization-based method and implement it on real IoT devices for comparative performance evaluation.

3 Method

3.1 Preliminaries and Notations

In our method, we formulate the cancer image classification problem as follows. We define the training dataset as $D_t = \{x_i, y_i\}_{i=1}^{N}$, consisting of inputs cancer images $x_i \in X$ and correct class labels $y_i \in Y$. For the classification model, it is formulated as $M_\theta : X \Rightarrow Y$, where θ is the model parameters for trained on dataset D_t, which is a float type model. Moreover, in order to deploy this model on the IoT device, we use the quantization function $F(\theta, \Delta, N_l)$, by searching for a suitable scale index Δ, which quantizes float parameters of the model θ as integer parameters. The quantized model M_{θ_q} is the final model to be deployed on the real device.

3.2 Methodology

As shown in Fig. 1, the proposed DDSNet framework contains four parts. In part 1, the Dense Convolution Network is used as the base architecture for classification. In part 2, we use depthwise separable convolution as the lightweight convolution algorithm, which replaces the convolution layer in DenseNet for efficient training and

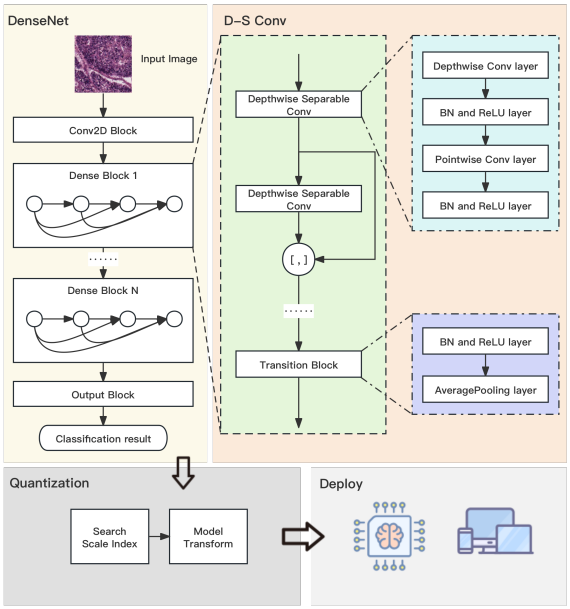


Figure 1: The DDSNet framework of this paper, including 4 parts: Dense Convolution Network, Depthwise Separable Convolution, Quantization of Model Parameters, and Model Deployment.

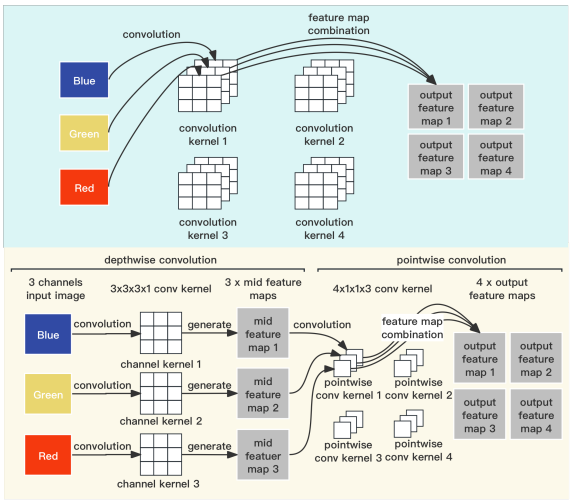


Figure 2: The traditional convolution (top) vs. depthwise separable convolution (bottom)

inference. For part 3, we search the scale index to apply the model quantization algorithm. Finally, we deploy the quantized model to an IoT device, which provides efficient and effective utility for medical imaging purposes.

3.2.1 Dense Convolution Network. The Dense Convolution Network (DenseNet) architecture introduces a unique approach to deep learning models, primarily designed to enhance the propagation and reuse of features within the network. This is achieved through its dense connectivity pattern, a fundamental disparity from traditional convolutional neural network (CNN) designs. The design of DenseNet addresses several issues prevalent in deep networks. By ensuring that each layer receives input from all preceding layers, the network facilitates feature reuse, which significantly mitigates the vanishing-gradient problem. Additionally, this characteristic reduces the model's susceptibility to overfitting and diminishes the total number of parameters, leading to a more robust model.

Generally, the ResNet model uses a shortcut connection to do the residual learning, and it can be described as a formula like follows:

$$o_l = H_l(o_{l-1}) + o_{l-1}\,, \tag{1}$$

where o_l is the output feature map of the l-th layer and H_l means a convolution block of layer l, which can include different operations such as Convolution, Pooling, Batch Norm, or Activation Unit.

In the DenseNet model, the significant difference from other CNN models is the dense connections pattern from any layer to all subsequent layers, which can improve the information flow among different layers. In this way, the l-th layer accepts the feature maps from all preceding layers, which is denoted as follows:

$$o_l = H_l(o_0 \oplus o_1 \oplus \cdots \oplus o_{l-1})\,, \tag{2}$$

where $o_0 \oplus o_1 \oplus \cdots \oplus o_{l-1}$ denotes the concatenation of the feature maps from layers $0, 1, \cdots, l-1$.

Then, to train this cancer image classification model, we use Cross Entropy Loss to optimize the parameters of the DenseNet model as follows:

$$\mathcal{L}(D_t;\theta) = \mathcal{L}_{CE}(M_\theta(x), y;\theta)\,, \tag{3}$$

where $M_\theta(x)$ is the output of our model with input x, y is the target label, and $\mathcal{L}_{CE}$ is the cross entropy of $M_\theta(x)$ and y.

3.2.2 Depthwise Separable Convolution. To further reduce the parameter of the neural network model and improve the speed of inference time, we combine the ideas of separable convolution with the DenseNet backbone. Compared with traditional convolution, separable convolution separates the convolution process into depthwise convolution and point-wise convolution as denoted by Fig. 2. More generally, we consider such a convolution process, where the convolution kernel has the same dimension D_K of height and width and the input image has the same dimension D_x of height and width. Then the parameters of traditional convolution C_t is denoted as follows:

$$C_t = D_K \times D_K \times I_c \times O_c \times D_x \times D_x\,, \tag{4}$$

where I_c and O_c are the channels of input and output.

To reduce the parameters of traditional convolution, our Depthwise Separable Convolution (D-S Conv) considers splitting the process into two steps: depth-wise convolution and point-wise convolution. The depth-wise convolution can be viewed as feature filters for each input channel and then produce intermediate feature maps. And the point-wise convolution is used to merge all intermediate feature maps to the final feature maps. This two-step convolution can reduce the parameters of the entire model.

The amount of parameters for separable convolution can be calculated as follows. Firstly, depth-wise convolution defines the convolution process at the channel level. As shown in Fig. 2, the colored images have 3 channels as RGB and depth-wise convolution generates the intermediate feature maps for each channel. Assume that the number of input channels is I_c, the parameters of depth-wise convolution C_d is denoted as follows:

$$C_d = D_K \times D_K \times I_c \times D_x \times D_x \ . \tag{5}$$

Secondly, the point-wise convolution defines convolution using 1×1 convolution kernel and filters the input intermediate feature maps for O_c output channels. In this way, the point-wise convolution combines the intermediate feature maps from depth-wise convolution and adjusts the feature map dimension from the input channel to the output channel. The parameters of point-wise convolution C_p are represented as follows:

$$C_p = I_c \times O_c \times D_x \times D_x \ . \tag{6}$$

We can compare the ratio between traditional convolution and depthwise separable convolution based on the above amount of parameters. The ratio of their parameters reduction is calculated as follows:

$$\begin{aligned} &\frac{D_K \times D_K \times I_c \times D_x \times D_x + I_c \times O_c \times D_x \times D_x}{D_K \times D_K \times I_c \times O_c \times D_x \times D_x} \\ &\qquad = \frac{C_d + C_p}{C_t} = \frac{1}{O_c} + \frac{1}{D_K^2} \ . \end{aligned} \tag{7}$$

The ratio indicates that we can shrink the number of parameters for the model by D_K^2 times via the depthwise separable convolution and thus improve the efficiency of the deep learning model during the training and inference process.

3.2.3 Quantization of Model Parameters. After the model size reduction via DS-conv, the next step is to deploy our trained model to the IoT device. We need to quantize convolution neural network parameters for inference with integer weights and activate functions. However, during the process of model quantization, the type changes from float to integer will inevitably bring performance loss to the model. Therefore, we have to deploy a suitable method for searching optimal quantization scales to minimize the accuracy drop.

The quantization algorithm contains two steps: (i) transforming parameters from float type to integer type; and (ii) restricting the parameters to the suitable area. As introduced in [14], a general quantization uses scale (Δ) and zero-point bias (z) to define the transformation process. To simplify this process, symmetric quantization is a potential affine quantization we can start with. By restricting zero-point bias to 0, entire quantization process can be denoted as follows:

$$F(\theta, \Delta, N_l) = \begin{cases} clamp(-\frac{N_l}{2}, \frac{N_l}{2} - 1, round(\frac{\theta}{\Delta})) & \text{if } signed \\ clamp(0, N_l - 1, round(\frac{\theta}{\Delta})) & \text{if } unsigned \end{cases}, \tag{8}$$

where $clamp(b_l, b_u, n)$ means the clamp operation on value n with lower bound b_l and upper bound b_u. Here, N_l is the range bound of a given data type, e.g., $N_l = 256$ for an 8-bit integer type. θ is the origin model parameter, Δ is the quantization scale and $round(*)$ means rounding operation.

Next, the problem is to search for an optimal scale parameter Δ that can minimize the accuracy drop caused by quantization. We use a two-step search algorithm: (i) a coarse-grained search by KL divergence; and (ii) a fine-grained search by cosine similarity. Particularly, we select a subset of the training dataset as calibration dataset D_c and all subsequent optimizations are based on the calibration dataset. In the first step, we focus on the activation layer output. For each potential scale, we generate a potential weight parameter θ'_q, and infer the model with this parameter. We collect the histogram of activation layer output values H_o by the original model parameters θ and H_q by the quantized model parameters θ'_q. These histograms store the distribution of activation values in many discrete bins. And Eq. (9) shows how to calculate the KL divergence and how to optimize the optimal scale parameter Δ by minimizing the KL divergence.

$$\min_{\Delta} D_{KL}(P, Q) = \sum_{h \in H} P(h) \ \log\left(\frac{P(h)}{Q(h)}\right), \tag{9}$$

where h is the number of activation values in the bin of histograms, $P(h)$ is the discrete probability distribution of H_o, and $Q(h)$ is the discrete probability distribution of H_q.

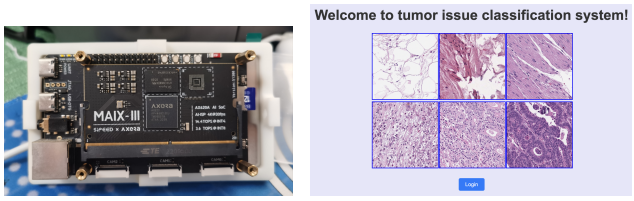
In the second step, we fine-grain the scale Δ based on the previous step's output by minimizing the cosine similarity between the convolution layer output and activation layer output of the original model and the quantized model. As introduced in EasyQuant [28], we set the scale Δ from KL divergence as the start scale, and search its neighborhood. For each potential scale, we generate a potential weight θ'_q and infer the model with this weight parameter. Accordingly, the output of the original model parameter θ is denoted as I_o, and the output of the quantized model parameter θ'_q is I_q. The following Eq. (10) shows how to compute the cosine similarity and maximize the cosine similarity to find the best Δ:

$$\max_{\Delta} \cos(I_o, I_q) = \frac{I_o \cdot I_q}{|I_o||I_q|} \ . \tag{10}$$

After the two-step scale search algorithm, we can find the empirically optimal scale Δ to minimize the accuracy drop caused by model quantization.

3.2.4 Model Deployment. The last part of our method is more about the software engineer, which utilizes a model inference engine on the IoT devices. In this part, we deploy our model to the develop board, named Maix-III, which is shown in Fig. 3(a). This is an IoT terminal device equipped with the AX620 chip, an AI SoC chip with a NPU that has a computing power of 3.6TOPs@INT8, a high energy efficiency ratio, and low power consumption. The device also integrates a quad-core Cortex A7 @ 1Ghz CPU with a floating-point operation unit and supports NEON, which can build an AI operating environment at a lower cost.

Then we use the Pulsar [1] to apply the quantized model inference. Pulsar is an efficient tool to deploy models on the device with AX chips. The entire inference process is built based on C++ code, which will return the classification results of given medical image data. Finally, we developed a website for terminal users to easily use the classification system just through simple UI to get the classification results, as shown in Fig. 3(b).



(a) IoT board (b) UI of the tumor issue classification system

Figure 3: The Maix-III device with AX-620 chip and the developed system.

4 Experimental Results

4.1 Experiment Settings

Datasets. We apply our model on Gastrointestinal Cancer (GC), PatchCamelyon (PCam), Skin Cancer (SC) and Cervical Cancer (SipakMed) datasets.

1) **GC**: It consists of 11,977 histological images (512×512) in gastrointestinal cancer, which has six classes, such as adipose tissue (ADI), mucus (MUC), stroma (STR), muscle (MUS), colorectal cancer epithelial tissue (TUM), and stomach cancer epithelial tissue (STU).

2) **PCam**: It consists of 327,680 color images (96×96) extracted from histopathologic scans of lymph node sections. Each image is annotated with a binary label (tumor or normal) indicating the presence of metastatic tissue.

3) **SC**: It consists of 10,015 dermatoscopic images (600×450) with seven diagnostic categories: Actinic keratoses and intraepithelial carcinoma (akiec), basal cell carcinoma (bcc), benign keratosis-like lesions (bkl), dermatofibroma (df), melanoma (mel), melanocytic nevi (nv) and vascular lesions (vasc).

4) **SipakMed**: It consists of 4,049 images of isolated cells of cervical cancer. The cell images are divided into five categories: dyskeratotic (DYS), koilocytotic (KOI), metaplastic (MET), parabasal (PAR) and superficial-intermediate (SUP).

Baselines. VGG, ResNet, Inception, SqueezeNet, and DenseNet are used as baselines to evaluate our method.

Training setting. These models are trained on an NVIDIA A30 GPU using PyTorch version 2.2.2. We use a batch size of 32 and train the models for 50 epochs. The adopted loss function is cross-entropy loss, and the optimizer is SGD with a suitable initial learning rate for different models, and momentum is 0.9.

4.2 Model Parameters Comparison

In order to demonstrate the performance in terms of model efficiency, we compared different models in various facets including the number of parameters, the number of operations, and the size of the model itself. Table 1 summarizes the model parameters difference of different deep learning models based on specific input image size. Due to the model limitation, the image size for Inception-V3 is 299×299. And for other models, the input image size is 224×224.

The Params column lists the total number of trainable parameters in millions (M) for each model. Parameters are indicative of the model's potential capacity to learn from data. The VGG model has 128.79 M parameters, which is the largest among all models' parameters. Our method shows the smallest number of parameters with only 0.19 M, suggesting it is the most compact model in terms of trainable parameters.

Table 1: Model parameters comparison with specific input image size.

Method	Params (M)	Flops (G)	Size (MB)
VGG	128.79	7.64	491.32
ResNet	11.18	1.82	42.72
SqueezeNet	0.73	0.75	2.84
Inception-v3	24.36	5.75	93.26
DenseNet	6.96	2.90	27.13
DDSNet	**0.19**	**0.43**	**0.84**

The Floating point operations per second (Flops) are given in Gigaflops and represent the number of operations needed for a single forward pass, which is usually a measure of computational costs. Again, our method has the lowest computational complexity at 0.43 G Flops, implying it requires the least computation during the inference stage.

The size of the model in megabytes (MB) reflects the amount of memory needed to store the model. The model size of DenseNet is about 27.13 MB, while our method is only 0.84 MB, which is about 32 times smaller model size than DenseNet, indicating it is likely the most storage-efficient and possibly suitable for deployment in environments with limited storage capacity, such as small IoT or edge devices.

From this table, it's evident that our method is designed to be significantly more lightweight than traditional heavy-weight models like VGG, ResNet, Inception-v3, and DenseNet. This speaks out that when our model can offer competitive accuracy, it is far more efficient, requiring fewer computational resources and less memory.

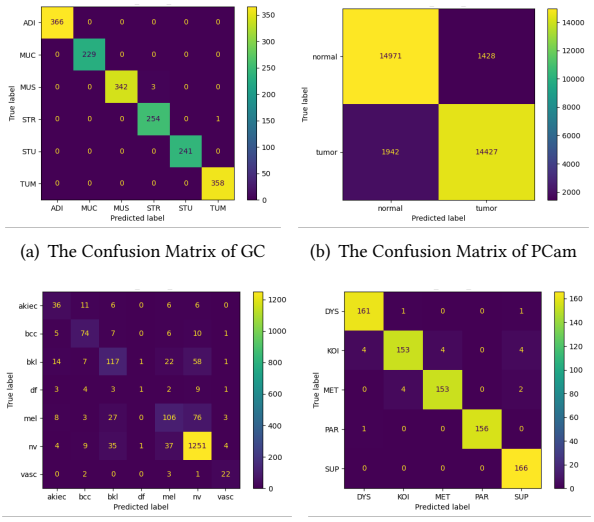
4.3 Classification Performance Comparison

Table 2 presents the classification performance of various deep learning models on four datasets. Here we only analyze the accuracy performance, and more details can be found in the table. Our proposed method shows the best accuracy of 99.6% on the GC dataset and 97.1% on the SipakMed dataset. On the PCam and SC dataset, the accuracy of our model is also competitively high, with 89.6% and 80.6% accuracy. DenseNet model performs very well on several datasets. This model gets the best accuracy on GC and SC. The results suggest that DenseNet is effective at capturing relevant features from all datasets for classification. ResNet shows a strong performance with 99.4% accuracy on GC, 89.7% on PCam and 95.5% on SipakMed. However, the accuracy performance of SqueezeNet indicates that this light-weight model is harder to train from scratch than other models, as always has the lowest accuracy. In summary, our method performs well on several tumor datasets, and we find a balance between classification accuracy and efficiency of lightweight models.

A confusion matrix is an error matrix that can be used to visualize results to provide an objective assessment of the classification performance of the classification model. The depth of the color in the diagonal represents the number in the cell, and the number

Table 2: Classification performance on different datasets and different models. This table contains accuracy (acc), precision (pre), recall (rec), and f1-score (f1). The bold values show the best one, and the underline values show the second.

method	dataset															
	GC				PCam				SC				SipakMed			
	acc	pre	rec	f1	acc	pre	rec	f1	acc	pre	rec	f1	acc	pre	rec	f1
VGG	0.973	0.973	0.973	0.973	0.853	0.854	0.853	0.853	0.754	0.729	0.754	0.738	0.923	0.923	0.923	0.923
ResNet	0.994	0.994	0.994	0.994	**0.897**	**0.899**	**0.897**	**0.897**	0.774	0.760	0.774	0.764	0.955	0.955	0.955	0.955
SqueezeNet	0.895	0.897	0.895	0.895	0.853	0.853	0.853	0.853	0.736	0.676	0.736	0.702	0.713	0.578	0.713	0.635
Inception-v3	0.990	0.990	0.990	0.990	0.878	0.887	0.878	0.877	0.757	0.724	0.757	0.735	0.926	0.936	0.925	0.925
DenseNet	**0.996**	**0.996**	**0.996**	**0.996**	0.886	0.894	0.886	0.886	**0.810**	**0.794**	**0.810**	**0.800**	0.963	0.962	0.963	0.962
DDSNet	**0.996**	**0.996**	**0.996**	**0.996**	0.896	0.900	0.896	0.895	0.806	0.792	0.806	0.796	**0.971**	**0.971**	**0.971**	**0.971**



(a) The Confusion Matrix of GC

(b) The Confusion Matrix of PCam

(c) The Confusion Matrix of SC

(d) The Confusion Matrix of SipakMed

Figure 4: The confusion matrix of our method in different datasets. The x-axis is predicted label, and the-y axis is true label. The sub-figures show the result on specific dataset.

concentrated on the diagonal represents the number of correct classifications. As shown in Fig. 4, our method achieves high classification performance on the GCand SipakMed datasets, while the prediction accuracy of SC and PCam is relatively low. Fig. 4(c) shows that the distribution of labels in the SC dataset is imbalanced, as the sample size of 'akiec', 'df' and 'vasc' are much smaller than the 'nv'. This reminds us that additional data augmentation methods should be applied to imbalanced datasets. As for PCam, this dataset has a small origin image size (96×96), which contains less feature information.

4.4 Quantization Performance on IoT Device

Our quantization performance test is based on the device shown in Fig. 3(a). The pre-quantization inference performance is measured by an onboard CPU, quad-core Cortex-A7, which focuses on floating-point arithmetic. The post-quantization inference performance is measured by an onboard NPU, AX620A, which does integer arithmetic. And the quantization process is deployed in a virtual machine, with a 6-core CPU and 16 GB memory.

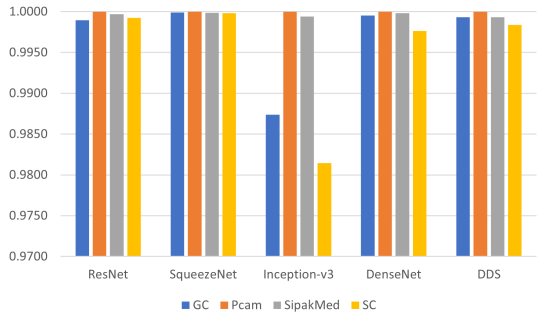


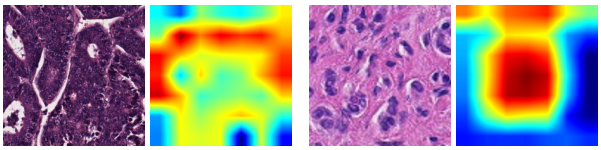
Figure 5: The column chart of cosine similarity. This chart compares cosine similarity of the original output and quantized output for different models on four datasets. The x-axis shows the name of each model, the y-axis shows the value of cosine similarity.

First, cosine similarity is the most common metric to illustrate the difference in prediction output between the original model and the quantized model. We compare the cosine similarities of different models on different datasets, and the results show that all the cosine similarities are higher than 0.98, which means the quantization method only changes the model output slightly and the quantized model is nearly unaffected. Fig. 5 shows the detail of cosine similarity. The complex architecture of Inception-v3 results in lightly worse performance. Due to the hardware limitation, we can not quantize the VGG model successfully.

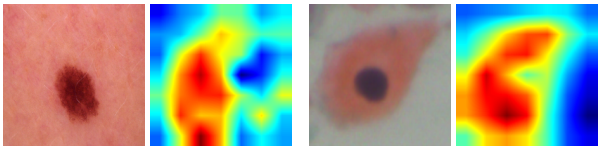
Table 3 presents the inference time performance of the baseline models and our proposed method. The inference time measured in milliseconds indicates how long each model takes to make an inference on CPU, GPU, and NPU. We apply the original models on the CPU and quantized models on the NPU. The "Initializer" column shows the preparation time required to load the quantized models. It's worth noticing that our DDS method achieves the best time efficiency in three metrics, i.e., shortest CPU, NPU, and Initialization time. The results demonstrate that the quantized model has a drastic speed improvement compared to the original model. After quantization, the inference time is approximately 118 times faster (2951.35ms/25.02ms) for DenseNet and about 106 times faster

Table 3: Inference time performance for different models. We compare the CPU inference time and NPU inference time specifically for origin model and quantized model.

Models	CPU	GPU (A30)	NPU	Initializer
VGG	3096.49	1.77	-	-
ResNet	1210.28	1.48	5.26	547
SqueezeNet	642.99	**1.25**	7.44	195.27
Inception-v3	6165.96	6.69	24.35	2031.94
DenseNet	2951.35	9.47	25.02	1397.85
DDS	**488.25**	2.75	**4.59**	**73.88**



(a) A TUM sample in GC (b) A tumor sample in PCam



(c) A nv sample in SC (d) A DYS sample in SipakMed

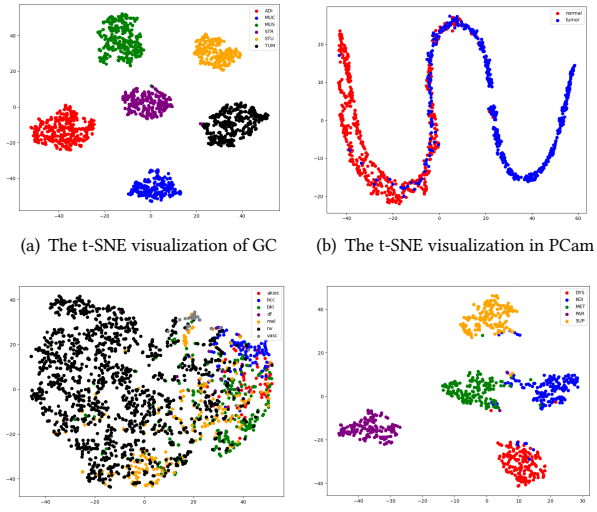
Figure 6: The Gradient Class Activation Map (GCAM) of cancer images. For each pair of images, the left is the original image, and right is heatmap

(488.25 ms/4.59ms) for our method. Additionally, our method is about 6 times faster than DenseNet before quantization (488.25ms vs. 2951.35ms) and about 5.5 times faster after quantization (4.59ms vs. 25.02ms), indicating that our lightweight model method runs much more efficiently with significantly less computation cost.

In summary, compared to other models, our method stands out by having a faster quantization inference time, making it a desirable choice for computing resources-limited applications.

4.5 Visualization and Explainability

In medical applications, providing predictions along with explanations is expected. We adopted the Gradient Class Activation Map (GCAM) to generate a heat map for the input image, representing the contribution distribution to the predicted output. A higher score indicates that the corresponding area of the original image has a strong response to the network and contributes significantly to the prediction. The different color areas in the gradient heat map represent varying impact levels (gradient scores), with red indicating higher contributions and blue indicating lower ones, as shown in Fig. 6. We used four image examples to demonstrate the explainability of our method: a TUM image from the GC dataset, a tumor image from the PCam dataset, an nv image from the SC dataset, and a DYS image from the SipakMed dataset. For each sub-figure, the left image is the original test image, and the right image is the corresponding GCAM heat map. These visualizations help illustrate



(a) The t-SNE visualization of GC (b) The t-SNE visualization in PCam

(c) The t-SNE visualization in SC (d) The t-SNE visualization in SipakMed

Figure 7: The t-SNE data visualization results. The x-axis and y-axis represent the numerical range of the t-SNE compression. The sub-figures (a) to (d) illustrate t-SNE results of different results.

which areas of the image the model is focusing on when making predictions, providing insights into the decision-making process. For example, in the Fig. 6(c) and 6(d), the red areas highlight the regions of the image most relevant to their respective classes.

Besides the gradient-based method, we also visualize the dataset in low-dimensional space using t-distributed Stochastic Neighbor Embedding (t-SNE). The t-SNE randomly embeds high-dimensional image features into a lower-dimensional space through an algorithm, allowing them to be visually represented. Here, we extract the prediction output of our proposed method with test dataset, and use t-SNE to compress the features into two dimensions, which can then be plotted as a 2D figure, as shown in Fig. 7. The Fig. 7(a) and 7(d) are the visualizations of GC and SipakMed datasets. Our proposed method achieves high accuracy on these two datasets, with clear decision boundaries, as sample points with the same class label cluster together. The PCam dataset is a binary classification task, and the Fig. 7(b) illustrates the t-SNE plot forming a "w" shape, with the decision boundary twisted near the middle of the manifold. The SkinCancer dataset has lower accuracy compared to the others. From Fig. 7(c), we can see that the sample points in this dataset are imbalanced, which is the main reason for the lower accuracy. In summary, t-SNE helps facilitate the analysis of neural network performance and the identification of potential issues.

4.6 Limitation

Our model has achieved competitive results compared to the baseline across multiple datasets. However, its performance is slightly lower than that of best complex models, due to the inevitable precision loss (float->int) during the quantization process. Additionally, our model's performance on imbalanced datasets still requires improvement. This highlights the need for special attention when

handling datasets during the training of lightweight models, as this can significantly impact the final outcomes.

5 Conclusion

In this paper, we investigate the problem of tumor tissue image classification and realize the proposed method in a real resource-limited IoT device. Specifically, considering the large size and slow computation of current deep learning models, we adopt the dense convolutional network as the model framework and then use depth-wise separable convolution to lightweight the network model for reducing the size and parameters of the proposed model. In addition, based on the trained tumor tissue image classification model, we apply the model quantization algorithm by optimizing the scale parameter. Finally, the quantized model is deployed and evaluated on the development device and an end-to-end platform for tumor tissue classification is developed. Extensive experiments based on the original model and the quantized model are conducted on the device, showing the outstanding performance of our proposed method in accuracy and efficiency after quantization deployment.

Acknowledgments

This work was partially supported by the National Science Foundation under grants No. 2429960, 2434899, 2117941, and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the South Korea Government (MSIT) (No. RS 2023-00261068, Development of Lightweight Multimodal Anti-Phishing Models and Split-Learning Techniques for Privacy-Preserving Anti-Phishing) and (No. RS-2024-00431388, the Global Research Support Program in the Digital Field program).

References

[1] Axera-tech. 2023. Pulsar user document. https://github.com/AXERA-TECH/pulsar-docs-en

[2] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. 2017. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama* 318, 22 (2017), 2199–2210.

[3] Jun Gao, Qian Jiang, Bo Zhou, and Daozheng Chen. 2021. Lung nodule detection using convolutional neural networks with transfer learning on CT images. *Combinatorial Chemistry & High Throughput Screening* 24, 6 (2021), 814–824.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[5] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR* abs/1704.04861 (2017). arXiv:1704.04861 http://arxiv.org/abs/1704.04861

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. 2020. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 1055–1059.

[8] Shih-Cheng Huang, Anuj Pareek, Malte Jensen, Matthew P Lungren, Serena Yeung, and Akshay S Chaudhari. 2023. Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *NPJ Digital Medicine* 6, 1 (2023), 74.

[9] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *CoRR* abs/1602.07360 (2016). arXiv:1602.07360 http://arxiv.org/abs/1602.07360

[10] Jakob Nikolas Kather. 2019. *Histological images for tumor detection in gastrointestinal cancer.* https://doi.org/10.5281/zenodo.2530789

[11] Jakob Nikolas Kather, Alexander T Pearson, Niels Halama, Dirk Jäger, Jeremias Krause, Sven H Loosen, Alexander Marx, Peter Boor, Frank Tacke, Ulf Peter Neumann, et al. 2019. Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer. *Nature medicine* 25, 7 (2019), 1054–1056.

[12] Hee E Kim, Alejandro Cosa-Linan, Nandhini Santhanam, Mahboubeh Jannesari, Mate E Maros, and Thomas Ganslandt. 2022. Transfer learning for medical image classification: a literature review. *BMC medical imaging* 22, 1 (2022), 69.

[13] Piotr Kluska and Maciej Zięba. 2020. Post-training quantization methods for deep learning models. In *Intelligent Information and Database Systems: 12th Asian Conference, ACIIDS 2020, Phuket, Thailand, March 23–26, 2020, Proceedings, Part I 12*. Springer, 467–479.

[14] Raghuraman Krishnamoorthi. 2018. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342* (2018).

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[16] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. 2017. A survey on deep learning in medical image analysis. *Medical image analysis* 42 (2017), 60–88.

[17] Zechun Liu, Kwang-Ting Cheng, Dong Huang, Eric P Xing, and Zhiqiang Shen. 2022. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4942–4952.

[18] Ange Lou, Shuyue Guan, and Murray Loew. 2021. DC-UNet: rethinking the U-Net architecture with dual channel efficient CNN for medical image segmentation. In *Medical Imaging 2021: Image Processing*, Vol. 11596. SPIE, 758–768.

[19] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*. Ieee, 565–571.

[20] Eunhyeok Park, Sungjoo Yoo, and Peter Vajda. 2018. Value-aware quantization for training and inference of neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 580–595.

[21] Devvi Sarwinda, Radifa Hilya Paradisa, Alhadi Bustamam, and Pinkie Anggia. 2021. Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer. *Procedia Computer Science* 179 (2021), 423–431.

[22] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[23] Chiranjibi Sitaula and Mohammad Belayet Hossain. 2021. Attention-based VGG-16 model for COVID-19 chest X-ray image classification. *Applied Intelligence* 51, 5 (2021), 2850–2863.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[25] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[26] Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. 2018. Rotation Equivariant CNNs for Digital Pathology. *CoRR* abs/1806.03962 (2018). arXiv:1806.03962 http://arxiv.org/abs/1806.03962

[27] Weibin Wang, Dong Liang, Qingqing Chen, Yutaro Iwamoto, Xian-Hua Han, Qiaowei Zhang, Hongjie Hu, Lanfen Lin, and Yen-Wei Chen. 2020. Medical image classification using deep learning. *Deep learning in healthcare: paradigms and applications* (2020), 33–51.

[28] Di Wu, Qi Tang, Yongle Zhao, Ming Zhang, Ying Fu, and Debing Zhang. 2020. Easyquant: Post-training quantization via scale optimization. *arXiv preprint arXiv:2006.16669* (2020).

[29] Xin Wu, Yue Feng, Hong Xu, Zhuosheng Lin, Tao Chen, Shengke Li, Shihan Qiu, Qichao Liu, Yuangang Ma, and Shuangsheng Zhang. 2023. CTransCNN: Combining transformer and CNN in multilabel medical image classification. *Knowledge-Based Systems* 281 (2023), 111030.

[30] Zuobin Xiong, Zhipeng Cai, Qilong Han, Arwa Alrawais, and Wei Li. 2020. ADGAN: Protect your location privacy in camera data of auto-driving vehicles. *IEEE Transactions on Industrial Informatics* 17, 9 (2020), 6200–6210.

[31] Zuobin Xiong, Wei Li, Qilong Han, and Zhipeng Cai. 2019. Privacy-preserving auto-driving: a GAN-based approach to protect vehicular camera data. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 668–677.

[32] Zuobin Xiong, Wei Li, Yingshu Li, and Zhipeng Cai. 2023. Exact-Fun: An Exact and Efficient Federated Unlearning Approach. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1439–1444.

[33] Honghui Xu, Zhipeng Cai, Zuobin Xiong, and Wei Li. 2023. Backdoor Attack on 3D Grey Image Segmentation. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 708–717.

[34] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6848–6856.