Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 173-200
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Lightweight Convolutional Neural Network with Hierarchical Multi-Scale Feature Fusion for Image Classification

Adama Dembele[1*], Ronald Waweru Mwangi[2], Ananda Omutokoh Kube[3]

[1]Department of Mathematics, Pan African University Institute for Basic Sciences, Technology and Innovation, Nairobi, Kenya
[2]Department of Computing, School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya
[3]Department of Mathematics and Actuarial Sciences, Kenyatta University, Nairobi, Kenya,
Email: *saliaasd96@gmail.com, waweru_mwangi@icsit.jkuat.ac.ke, KUBE.ANANDA@ku.ac.ke

How to cite this paper: Dembele, A., Mwangi, R.W. and Kube, A.O. (2024) A Lightweight Convolutional Neural Network with Hierarchical Multi-Scale Feature Fusion for Image Classification. *Journal of Computer and Communications*, **12**, 173-200.
https://doi.org/10.4236/jcc.2024.122011

Received: November 13, 2023
Accepted: February 26, 2024
Published: February 29, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

Convolutional neural networks (CNNs) are widely used in image classification tasks, but their increasing model size and computation make them challenging to implement on embedded systems with constrained hardware resources. To address this issue, the MobileNetV1 network was developed, which employs depthwise convolution to reduce network complexity. MobileNetV1 employs a stride of 2 in several convolutional layers to decrease the spatial resolution of feature maps, thereby lowering computational costs. However, this stride setting can lead to a loss of spatial information, particularly affecting the detection and representation of smaller objects or finer details in images. To maintain the trade-off between complexity and model performance, a lightweight convolutional neural network with hierarchical multi-scale feature fusion based on the MobileNetV1 network is proposed. The network consists of two main subnetworks. The first subnetwork uses a depthwise dilated separable convolution (DDSC) layer to learn imaging features with fewer parameters, which results in a lightweight and computationally inexpensive network. Furthermore, depthwise dilated convolution in DDSC layer effectively expands the field of view of filters, allowing them to incorporate a larger context. The second subnetwork is a hierarchical multi-scale feature fusion (HMFF) module that uses parallel multi-resolution branches architecture to process the input feature map in order to extract the multi-scale feature information of the input image. Experimental results on the CIFAR-10, Malaria, and KvasirV1 datasets demonstrate that the proposed method is efficient, reducing the network parameters and computational cost by 65.02% and 39.78%, respectively, while maintaining the network performance compared to the MobileNetV1 baseline.

Keywords

MobileNet, Image Classification, Lightweight Convolutional Neural Network, Depthwise Dilated Separable Convolution, Hierarchical Multi-Scale Feature Fusion

1. Introduction

The purpose of computer image classification is to provide a better option for human visual perception of pictures through analysis and categorization of those images. In the field of computer vision, such as image classification [1], target tracking [2], target detection [3], and image segmentation [4], deep convolutional neural networks (CNNs) have gained remarkable success. Most recent CNNs include hundreds of hidden layers and training parameters to improve accuracy, but this comes at a large computational cost. Therefore, it's still challenging to train and deploy large-scale convolutional neural network models, and it needs access to powerful computational and storage infrastructure.

The growth of smart mobile phones, embedded devices, and Internet of Things devices has increased the need for training and deploying convolutional neural networks on such devices. Large convolutional neural networks, such as VGGNet [5] and ResNet [6], are not ideal for training and deployment on such devices due to their restricted computational and storage capacity. As a result, CNNs for real-world applications operating on edge devices must be lightweight and efficient while maintaining high accuracy.

Throughout the years, various research papers have suggested various techniques to construct lightweight networks for performing real-time inference on small hardware. Network pruning is a method for reducing the complexity of a neural network by removing neurons and parameters that have a minimal impact on the model's performance. This approach can be used to address overfitting by eliminating redundant or unnecessary elements from the network [7] [8]. Low-bit representation-based approaches are another strategy that involves utilizing low-precision numbers to represent the parameters and activations of a neural network [9]. Most of the time, these models do not change the structure of the network, and the convolutional operations could be done faster on CPUs by using logical gates. A more recent strategy, known as a compact network, involves factoring a computationally expensive convolution operation [10] [11] [12]. These models are designed to be computationally efficient, which means that the underlying model structure learns fewer parameters and performs fewer floating-point operations (FLOPs).

MobileNetV1 is a type of compact CNN that utilizes depthwise separable convolution [13]. However, this method of convolution can lead to a large number of 1×1 convolutions (pointwise convolutions), which can consume significant computational resources. [14] has developed an optimized version of the Mobi-

leNet baseline called "kMobileNet", which replaces pointwise convolution with grouped pointwise convolution in depthwise separable convolution layer. He has significantly reduced MobileNet network parameters and complexity; however, there are still some drawbacks that need to be addressed. Firstly, as the network deepens, the features extracted by the neural network shift from particular edge features to abstract semantic features. While these deeper layers contain rich semantic information crucial for classification tasks, the model's use of a series of strides of two within its network architecture leads to a substantial issue. This series of large stride operations significantly diminishes the resolution and detail in the final feature map. Consequently, there is a notable reduction in both the detail and spatial information, resulting in a loss of critical fine features essential for accurate image analysis [15] [16] [17]. Secondly, a model with a lot of training parameters can improve the accuracy of classification to some extent, but this will take a lot of time and storage space.

In order to solve the above problems, this study proposed a hierarchical multi-scale feature fusion (HMFF) module to extract features for a lightweight and accurate network. To minimize the impact on network computation costs, the hierarchical multi-scale feature fusion is strategically placed just before the final classification layer in the MobileNetV1 architecture. This module adeptly merges features from different scales. While it is understood that some spatial details are diminished in the earlier layers due to downsampling, the module selectively amplifies residual spatial information still present in the deeper layers. It harnesses the finer nuances embedded within these layers, enhancing the detail resolution that is critical for precise image classification. The hierarchical multi-scale feature fusion module uses depthwise dilated separable convolution, which has the advantage of expanding the kernel size without increasing the number of training parameters. Depthwise dilated separable convolution uses depthwise convolution with dilate rate parameter and combines it with grouped pointwise convolution. Furthermore, the hierarchical multi-scale feature fusion (HMFF) technique is employed to make full use of diverse levels of features, particularly the fusing of shallow edge information and deep semantic information, as well as to avoid gridding artifacts induced by dilated convolution. The main contributions of this study are summarized as follows:

1) The study proposed a lightweight convolutional neural network for image classification with hierarchical multi-scale feature fusion named HMFF-MobileNet, with the goal of building an efficient deep learning network suited for use in cloud computing, mobile vision, and embedding system applications.

2) The study presented a hierarchical multi-scale feature fusion (HMFF) module that is conducive to learning varied-sized input image features and improves prediction performance by enlarging the receptive field and capturing the discriminative multi-scale feature without raising convolution parameters.

3) Experiments on the CIFAR-10, Malaria, and KvasirV1 datasets show that the proposed HMFF-MobileNet network outperforms other state-of-the-art networks, such as MobileNetV1 [13] and kMobileNet [18] variants, despite having few pa-

rameters and low complexity.

The study is divided into five sections: Section 2 reviews the literature, Section 3 details the technique used, Section 4 presents the findings, and Section 5 provides a conclusion to the study.

2. Related Work

In recent years, lightweight CNNs have become a popular way of reducing model sizes. [19] discovered that the deep network parameters contain a lot of redundancy. These parameters were ineffective in improving classification accuracy but had an impact on processing efficiency. [20] greatly enhanced the compressed model by utilizing the combined knowledge of multiple models. The accuracy of classification for this streamlined network was nearly equivalent to that of a more complex network. In 2016, [21] introduced a tiny CNN structure called SqueezeNet that drastically reduced the amount of network parameters by using network compression techniques. However, model compression reduces model accuracy.

MobileNetV1 is one of lightweight CNNs model and its main feature is the use of depthwise and pointwise convolution instead of regular convolution, which is more efficient for mobile devices and embedded applications with limited resources [13]. The authors of MobileNetV1 introduced two hyper-parameters that allow an engineer to choose an appropriate model size depending on the characteristics of the problem. Standard CNN architecture-based models continue to outperform MobileNetV1. Therefore, MobileNetV2 [11] is proposed as a solution to the problem. The model has an inverted residual structure, with shortcut connections between the thin bottleneck layer and the intermediate expansion layer, which uses depthwise convolution to filter features as a nonlinearity source. MnasNet is based on the MobileNet V2 model architecture and incorporates lightweight attention modules into the bottleneck structure via squeeze and excitation [22]. These structures are placed after the depthwise filters feed-forward pass to obtain attention to be applied to the largest image representation. In order to get attention that is applied to the largest image representation, these structures are positioned following the feed-forward pass of the depthwise filters. To address the vanishing gradient issue and guarantee greater accuracy, [23] enhanced the MobileNet V2 and suggested MobileNet V3, which employs modified swish nonlinearities and swaps the original sigmoid function for the hard sigmoid.

ShuffleNet employed a combination of point-group convolution and channel shuffle, which significantly reduced the number of parameters and computation flops while maintaining a high level of performance in tasks such as image classification and object detection [24]. With the introduction of ShuffleNetV2, [12] enhanced the original ShuffleNet even more. The model takes into account both the direct and indirect metrics of computation complexity, such as required memory and device characteristics, as well as indirect metrics like FLOPs.

Dilated convolution is often used for tasks like semantic segmentation and target detection [25] [26], and it is also used in part for image classification tasks [27]. Dilated convolution extends the receptive field while maintaining feature map resolution by inserting holes into the normal convolution (max-pooling or strided convolution reduces feature map resolution) without increasing the amount of complexity. The dilation convolution differs from the original standard convolution in that it contains a hyper-parameter called dilation rate, which corresponds to the number of intervals in the convolution kernel (e.g. standard convolution is dilatation rate 1).

Recently, researchers have begun to combine depthwise separable convolution with dilated convolution [28] [29] [30]. This combination has been found to improve the performance of convolutional neural network, particularly in computer vision. However, depthwise separable convolution uses pointwise convolution, which generates more than 80% of the parameters in the most recent deep convolutional neural architectures, according to [18]. In order to reduce the parameter size and computational complexity, [14] has proposed using grouped pointwise convolution, which can compress the parameters of pointwise convolution while still maintaining a reasonable accuracy. This combination is showing promise in computer vision tasks, it has been demonstrated to reduce the computational complexity of convolutional neural network. This study took advantage of the benefits of dilated convolution, depthwise convolution, and grouped pointwise convolution in order to have a lightweight network. The proposed method, called depthwise dilated separable convolution (DDSC), combines these operations to reduce the number of parameters and increase the accuracy of the network.

Although dilated convolution reduces the disparity between receptive field size and feature map resolution, it still has significant drawbacks. When dilated convolution is used with a single dilation rate, all of the neurons in the feature map have the same receptive field, and the network only uses features on a single scale. This can be a problem when trying to classify objects in an image with different scales because the network might not be able to recognize objects that are much bigger or smaller than the dilation rate. For example, if the dilation rate is set too high, the network may recognize only large objects in the image while missing small objects, and vice versa. However, [31] [32] showed that multi-scale information can aid in the resolution of ambiguous cases and result in more robust classification. [33] proposed atrous spatial pyramid pooling (ASPP) module, which joins together feature maps with different rates of dilation. This way, the output feature map includes semantic information from multiple scales, which can improve classification performance.

[34] developed spatial pyramid pooling (SPP) block to extract multiscale information by running several parallel dilated convolutions. [35] came up with an idea for a multi-scale dilated network with depthwise separable convolution network based on concatenation and summation feature fusion technique for the

prediction of abnormalities in chest radiographs. The concept was based on MobileNet network with an early feature fusion approach, which can result in network redundancy, an increase in the number of parameters to train, and a more complicated model. Their approach differs from ours in that ours uses late feature fusion with hierarchical feature fusion. The hierarchical fusion approach is advantageous in that it utilizes the contextual information of both high-level and low-level features.

[36] has proposed a three-branch hierarchical multi-scale feature fusion network structure called HiFuse for medical image classification. The HiFuse model consists of a parallel hierarchy of local and global feature blocks to extract local features and global representations at various semantic scales. However, the computational complexity of the proposed model is significantly higher compared to traditional lightweight models. This is due to the parallel processing and multiple feature extraction layers involved in the HiFuse model. [37] presented a novel image classification system called CMSFL-Net, which utilizes a consecutive multiscale feature-learning approach. The CMSFL-Net is a combination of consecutive multiscale feature learning (CMSFL) modules, max-pooling operations, and fully connected dense layers for feature extraction and classification. The proposed system addresses the challenges of efficient computation, low generalization on small-scale images, and underfitting with limited data. However, the results suggest that the optimal number of CMSFL modules depends on the specific datasets and the trade-off between accuracy and efficiency.

[38] has developed the ESPNet network for semantic segmentation based on the efficient spatial pyramid (ESP) module. A standard convolution is divided into a point-wise convolution and a spatial pyramid of dilated convolutions using three steps. Firstly, the ESP module performs a 1×1 convolution on input features to project high-dimensional feature maps onto a low-dimensional space. Secondly, it divides pointwise convolution kernel maps into multiple parallel branches and performs dilated convolution operations with different rates on each kernel independently. Indeed, it performs hierarchical feature fusion in order to extract multi-scale features. The ESP module aligns with the idea of this study's multi-scale hierarchical feature fusion module. In reference to the eighth paragraph of this section, it is noteworthy to highlight that the 1×1 convolution accounts for over 80% of the computational cost within the ESP module. This underscores the potential for significant performance enhancements in the multi-scale hierarchical feature fusion module by optimizing the efficiency of the 1×1 convolution. Furthermore, this research suggested replacing the computationally intensive convolution with the grouped pointwise convolution, as proposed by [14]. This substitution could offer the dual advantage of reducing computational overhead while preserving accuracy.

3. Materials and Methods

This section details the different components of the proposed hierarchical mul-

ti-scale feature fusion (HMFF) module, followed by a description of the lightweight network architecture (HMFF-MobileNet).

3.1. Baseline Methods

This subsection briefly introduces depthwise dilated convolution and grouped pointwise convolution in order to develop depthwise dilated separable convolution (DDSC), which is a component of the HMFF module.

3.1.1. Depthwise Dilated Convolution

A depthwise dilated convolution is a depthwise convolution variant in which the kernel is applied to the input feature map with a dilation rate. The dilation rate is a hyperparameter that determines the gap between values in the kernel, which in turn influences the size of the convolution's receptive field [39].

Let us take a depthwise convolution layer as an input $H \times W \times M$ feature map F and produces $H' \times W' \times N$ feature map O, where H and W are respectively the spatial height and width of input feature map, M is the number of input channels (input depth), H' and W' are respectively the spatial width and height of output feature map and N is the number of output channel [13], then a dilated convolutional layer operates on the feature map $F \in \mathbf{R}^{H \times W \times M}$ using a convolutional kernel $K \in \mathbf{R}^{D_K \times D_K \times M \times N}$ with a dilation rate d.

Using the kernel K, we compute the output feature map $O \in \mathbf{R}^{H' \times W' \times N}$ in Equation (1):

$$O_{k',h,w} = \sum_{k,m,n} F_{k,h+dm,w+dn} \times K_{k',k,m,n} \tag{1}$$

Then, depth-wise convolution is applied to the dilated convolution as in Equation (2):

$$F'_{k,h,w} = \sum_{m,n} K^{d}_{k,m,n} \times F_{k,h+dm,w+dn} \tag{2}$$

It should be noted that the process does not perform cross-channel multiplication and instead only performs spatial multiplication, where K^d is the kernel for the depth-wise dilated convolution.

Floating points of operations (Flops) and number of parameters are used to assess the network's computational complexity. Flops denote the network's time complexity, while number of parameters represents the network's spatial complexity.

The cost $Flops_{dw}$ of the depthwise dilated convolution layer with stride one can then be definied as:

$$Flops_{dw} = H \times W \times M \times D_K \times D_K \tag{3}$$

The total number of parameters (P_{dw}) depthwise dilated convolution generates is:

$$P_{dw} = D_K \times D_K \times M \tag{4}$$

As can be seen, the cost multiplicatively depends on the number of input, the number of output channels, the kernel size and the input channel size.

3.1.2. Grouped Pointwise Convolution (Gconv)

Grouped pointwise convolution use the idea of group convolution and use advantage of parallelization computing on pointwise convolution. The architecture as defined in Figure 1, starts with a pointwise grouped convolution layer R, which is made up of filter groups R_1 and R_2 paths. This is followed by a channel interleaving layer, which combines channels for the next pointwise grouped convolution layer L (constituted by filter groups L_1 and L_2 paths). All of the channels in groups R_1 and R_2 are added together to make one path. The same process is performed for the L layer. For layers R and L, the number of filter groups and filters per group is calculated by exact divisions of the original number of input channels and filters by Ch [14].

Grouped pointwise convolution has been proposed to replace the computational expensive standard pointwise convolution, which generates $M \times N$ parameters, and costs $H' \times W' \times M \times N$ computation.

Now, let us consider the group convolution introduced in [14] with Ch number of channels per group. The total number of parameters P_{gconv} of grouped pointwise convolution is then calculated by multiplying the number of original filters by the number of input channels per filter:

$$P_{gconv} = 2\left(N \times Ch\right) \tag{5}$$

The cost $Flops_{gconv}$ of the grouped pointwise convolution layer with stride one can then be definied as:

$$Flops_{dw} = H' \times W' \times 2\left(N \times Ch\right) \tag{6}$$

Hence, given the number of parameters and the costs of grouped pointwise convolution, it is clear that Ch must be significantly less than $M/2$ in order to optimize a regular pointwise convolutional layer.

3.2. Proposed Method

This section explains the HMFF-MobileNet architecture by describing its depthwise

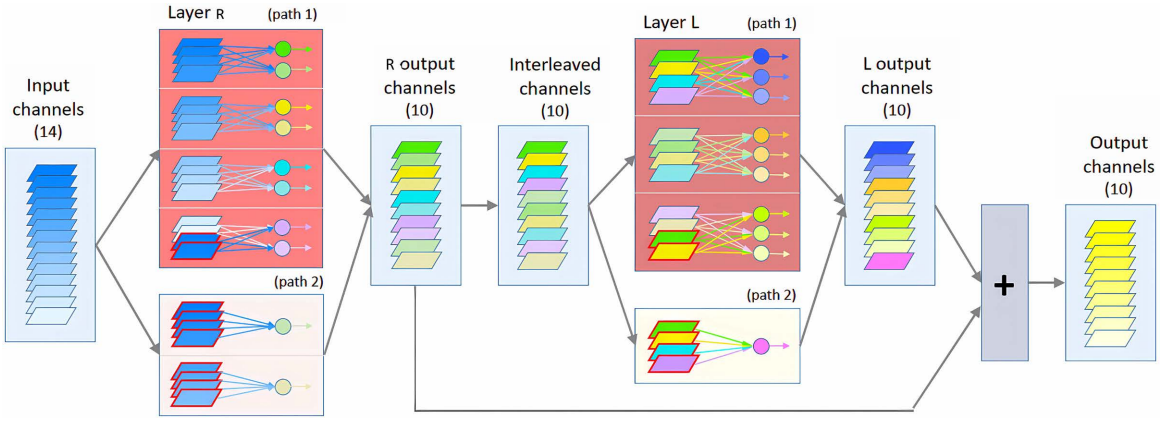

Figure 1. Schematics show the grouped pointwise convolution architecture and the pointwise convolution replacement. This example takes the place of a pointwise convolution by using 14 input channels and 10 filters. It has two convolutional layers, R and L, and it also includes one interleaving layer and one summation layer. Channels that have been replicated have a red line around them.

dilated separable convolutions, which allow the network to efficiently learn representations from a large effective receptive field. A description of how to use the hierarchical multi-scale feature fusion (HMFF) module is given, which enables the proposed network to effectively capture features at different scales and combine them for improved performance.

3.2.1. Depthwise Dilated Separable Convolution (DDSC)

Many different ways have been suggested to replace the standard convolution layers in a deep CNN architecture with different types of convolution layers, such as the depthwise separable convolution layer [13].

Depthwise dilated separable convolution, use the idea of depthwise separable convolution (DSC). It combines depthwise dilated convolution with grouped pointwise convolution. After each layer in the modified depthwise separable convolution, batch normalization [40] and Swish [41] as the activation function are applied.

From Equations (4) and (5), the total number of parameters (P) of DDSC layer is defined as:

$$P_{DDSC} = D_K \times D_K \times M + 2\left(N \times Ch\right) \tag{7}$$

The cost $Flops_{DDSC}$ of DDSC convolution layer with stride one can then be definied as:

$$Flops_{DDSC} = H \times W \times M \times D_K \times D_K + H' \times W' \times 2\left(N \times Ch\right) \tag{8}$$

As a result, the computational cost ratio of depthwise separable convolution to depthwise dilated separable convolution can be expressed as:

$$\frac{Flops_{DDSC}}{Flops_{DSC}} = \frac{H \times W \times M \times D_K \times D_K + H' \times W' \times \left(N \times 2Ch\right)}{H \times W \times M \times D_K \times D_K + H' \times W' \times \left(N \times M\right)} \tag{9}$$

3.2.2. Hierarchical Multi-Scale Feature Fusion (HMFF) Module

To increase the accuracy of the model of imaging classification, local characteristics and global representations from different levels can be fused, this study proposed an hierarchical multi-scale feature fusion (HMFF) module based on the split-transform-merge strategy. The HMFF module starts by splitting the output of grouped pointwise convolution into K parallel branches (Step1 (Split) in **Figure 2**). Then, each branch uses depthwise dilated separable convolution with different dilation rates given by $2^{p-1}, p \in \{1, \cdots, K-1\}$ to process these feature maps at the same time in parallel without changing the network's parameters or complexity (Step2 (Transform) in **Figure 2**). The HMFF module can learn representations from a large effective receptive field by using different dilation rates in each branch. However, using dilated convolution as a feature extractor directly will lead to some information being lost because it suffers from the gridding artifact problem. Thus, before concatenating the feature maps obtained with different dilation rates, they are hierarchically added (HFF). To strengthen information flow, a skip-connection between input and output is added (Step3 (HFF) in **Figure 2**). The residual connection is used when the number of

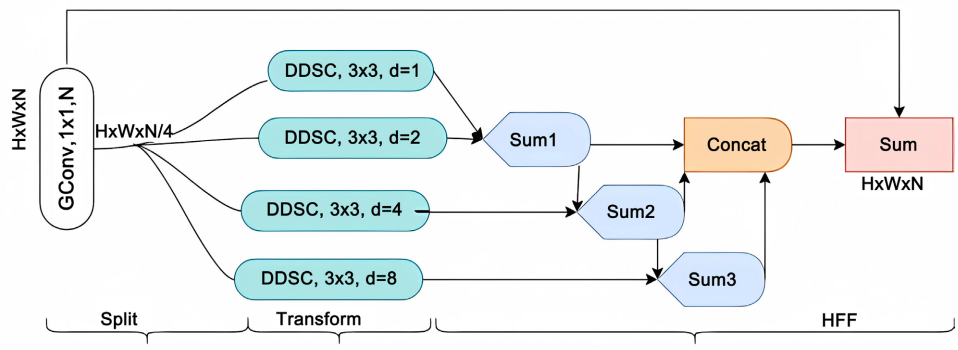

Figure 2. Hierarchical multi-scale feature fusion (HMFF) module.

channels going into the block and coming out of the block is the same. We use batch normalization and activation function after every convolutional layer.

Let $\hat{F}^i$, the output of grouped pointwise convolution with N^i filters in layer i, d dilated convolution rate, and ($K=4$) number of branches.

Because $\frac{N^i}{K}$ is not a perfect division in general, combining K and $\frac{N^i}{K}$-dimensional feature maps would not generate an N^i-dimensional output. To deal with this, we use n_1 kernels with a dilation rate of 1 and n_2 kernels with a dilation rate of 2^p, $p \in \{1, K-1\}$. Depthwise dilated separable convolution is applied to each branch in order to extract multi-scale features. The output of each convolution in K-partitions is defined as:

$$\hat{O}^p_{H\times W\times \hat{N}^i} = \begin{cases} DDSC_{H\times W\times n_1, d=1}\left(F^i\right) & \text{if } p=1, \\ DDSC_{H\times W\times n_2, d=2^{p-1}}\left(F^i\right) & \text{if } p\in\{1,\cdots,K-1\} \end{cases} \tag{10}$$

where

$$\hat{N}^i = \begin{cases} n_1 = \left(N^i - (K-1)\left\lfloor \frac{N^i}{K} \right\rfloor\right) & \text{if } d=1 \\ n_2 = \left\lfloor \frac{N^i}{K} \right\rfloor & \text{if } d=2^p, p\in\{1,\cdots,K-1\} \end{cases} \tag{11}$$

Output of hierarchical feature fusion (HFF) step is defined as:

$$\begin{cases} S^1_{H\times W\times n_2} = \hat{O}^2_{H\times W\times n_2} + \hat{O}^4_{H\times W\times n_2} \\ S^2_{H\times W\times n_2} = \hat{O}^8_{H\times W\times n_2} + S^1_{H\times W\times n_2} \end{cases} \tag{12}$$

$$concat = Activation\left(BN\left(\left[\hat{O}^2_{H\times W\times n_1}, S^1_{H\times W\times n_2}, S^2_{H\times W\times n_2}\right]\right)\right) \tag{13}$$

$$O_{H\times W\times N^i} = Activation\left(concat + \hat{F}^i\right) \tag{14}$$

In Equation (13), the three matrices (or tensors) denoted by $\hat{O}^2_{H\times W\times n_1}$, $S^1_{H\times W\times n_2}$, and $S^2_{H\times W\times n_2}$ are concatenated (combined) into a single matrix, which is subsequently processed through an Activation function and a batch normalization (BN) function. The outcome of these operations is stored in the variable concat as defined in Equation (14).

3.2.3. Lightweight Network Architecture

Based on the proposed hierarchical multi-scale feature fusion (HMFF) module and depthwise dilated separable convolution, and kMobileNet baseline architecture [18], the architecture of the proposed network (HMFF-MobileNet) is shown in **Table 1**. The proposed method involves two modifications to kMobileNet. Firstly, the network architecture contains five layers of separable filters with 512 filters, compared to kMobileNet's architecture which has six such layers. This reduction in the number of layers improves the efficiency of the network, resulting in a decrease in memory consumption while maintaining accuracy. Secondly, the final layer of the KMobileNet architecture has been swapped out for the HMFF module. This modification adds the ability to capture multi-scale features, enhancing the network's discriminative power.

The proposed network increases the depth of the network by repeating depthwise dilated separable convolutions with a dilation rate of 1 at each spatial level. In addition, batch-normalisation and Swish activation function [41] are used after every convolution layer. The discriminative features obtained from a series of DDSCs are sent to HMFFM to learn multi-scale imaging features. Then, a fully connected layer is used to make predictions based on the features that were extracted.

4. Experiments

This section evaluates and compares the performance of the HMFF-MobileNet network on CIFAR-10, malaria, and Kvasir datasets for image classification. First

Table 1. HMFF-MobileNet architecture, where Conv stands for standard convolution, FC stands for full connected layer.

Operator Layer	No. of Filter	Kernel Size	Stride
Conv	32	3 × 3	2
DDSC	64	3 × 3, 1 × 1	1
DDSC	128	3 × 3, 1 × 1	2
DDSC	128	3 × 3, 1 × 1	1
DDSC	256	3 × 3, 1 × 1	2
DDSC	256	3 × 3, 1 × 1	1
DDSC	512	3 × 3, 1 × 1	2
DDSC	512	3 × 3, 1 × 1	1
DDSC	512	3 × 3, 1 × 1	1
DDSC	512	3 × 3, 1 × 1	1
DDSC	512	3 × 3, 1 × 1	1
DDSC	1024	3 × 3, 1 × 1	2
HMFF	1024	3 × 3, 1 × 1	1
Global-Avg Pool	-	7 × 7	1
FC	1000	-	1

of all, a highlight of the performance of various HMFF-MobileNet architecture design is given. By varying the values of hyperparameters Ch, trade-offs between accuracy and computational cost in terms of model size and floating-point operations (FLOPS) is investigated.

4.1. Datasets

- **CIFAR-10** [42]: It is a popular dataset in computer vision and machine learning. It consists of 60,000 labeled images, each measuring 32 × 32 pixels and featuring RGB color. These images are categorized into ten different classes. These classes represent commonplace objects and scenes such as planes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. It is considered a relatively small dataset, but it is still hard to work with because the images are small and have low resolution, and the objects in the images often look the same. This dataset has been divided into 50,000 training images and 10,000 testing images. We chose 5000 images for validation and left 45,000 in the training set.
- **Malaria dataset** [43]: The dataset consists of 27,558 cell images, divided into two categories of infected and healthy cells, each with an equal number of images. 10% of the images, or 2756 images, were set aside for validation, and an additional 10% were designated for testing. In all training, validation, and testing subsets, half of the images depict healthy cells.
- **KvasirV1** [44]: The dataset consists of 4000 images of endoscopic gastrointestinal diseases, categorized into 8 classes with each class containing 500 images. It includes images showcasing anatomical landmarks such as Z-line, pylorus, or cecum, as well as pathological findings like esophagitis, polyps, and ulcerative colitis. The dataset contains images of varying resoluti ons, ranging from 720 × 576 to 1920 × 1072 pixels. To standardize the data, all images are downsized to 224 × 224 pixels. The dataset is then split into 3200 training images and 800 testing images. 800 images were picked for validation and left 2400 in the training set.

4.2. Implementation Details

The proposed model is implemented using Python with K-CAI [14], Tensorflow/Keras [45], deep learning framework, trained and tested on machine with CPU Intel Core i7-1760H @2.2GHz, GPU Nvidia GTX 1060, RAM 12GB DDR4, and CUDA 10.1 with cuDNN back-ends. For optimization, Adam optimizer [46] is used and decay schedule learning rate [47]. Decay schedule learning rate is a learning rate schedule that reduces the learning rate by a factor every few epochs, where epoch count is a hyperparameter. At each epoch, the learning rate Lr is evaluated as:

$$Lr = lr_0 \times \left(decayfactor * \left[\frac{epoch}{stepdecay} \right] \right) \tag{15}$$

where lr is initial learning rate, decayfactor is to drop the learning rate by half

every *stepdecay* epoch.

The experiment set ($lr_0 = 1e-3$ for Cifar-10 and malaria data), ($lr_0 = 1e-3$ for Kvasir data), *stepdecay by 10*, $decayfactor = 0.75$ as initial values of the parameters.

Data augmentation techniques such as random cropping, flipping, and rotation were applied to improve the robustness of model. To fit the settings for the baseline network's origin, the size of the various datasets is changed to 224 × 224. With cifar-10 and malaria data, the differents networks have been trained with a batch size of 64 for 50 epochs, and a batch size of 32 for 150 epochs with Kvasir data by optimizing the cross-entropy loss. The technique outlined in [48] was used to initialize the weights in networks.

The minimum number of input channels per group is appended to the end of each implementation's name. HMFF-MobileNet 32Ch, for instance, has a minimum of 32 input channels per group. This naming convention allows to easily differentiate between different versions of implementations.

4.2.1. Performance Metrics

To evaluate the performance of HMFF-MobileNet, the number of Flops, and the number of parameters have been employed. Accuracy, Precision, and Recall are used as classification indicators. Equation (16) indicates the accuracy of the model, which is the metric used to measure the total number of correct predictions. However, the accuracy rate of the model does not guarantee the model's ability to classify the classes if the dataset has an unequal distribution with class imbalance; therefore, it is necessary to generalize to all classes when classifying medical images. Precision and recall play a crucial role in providing valuable information about the model's performance under these conditions. All of these metrics are calculated using the confusion matrix. The symbols in the confusion matrix are defined as follows: true positive (TP), true negative (TN), false positive (FP), and false negative (FN) (FN). As a result, Equation (16) calculates accuracy to obtain the percentage of correctly identified samples:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

To reflect the accuracy of the model prediction for binary classification, Equation (17) has been used to calculate the precision rate, which is the proportion of samples with correct, true values among the samples predicted to be correct:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{17}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{18}$$

To reflect the comprehensiveness of the model prediction for binary classification task, Equation (18) has been used to determine the recall rate, which is the number of positive samples identified in the data for which all true values are properly predicted.

In a multi-class classification setting, micro-averaged precision and recall was

used, and defined in Equations (19) and (20) respectively:

$$\text{Micro Precision} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i} \tag{19}$$

$$\text{Micro Recall} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FN_i} \tag{20}$$

4.2.2. Baseline Models

In order to illustrate the performance of the proposed network, two categories of state-of-the-art methods have been selected: high-accuracy methods and effective, lightweight methods. To further highlight the superior balance of complexity and performance offered by HMFF-MobileNet, we juxtaposed its performance against these models. To further highlight the superior balance of complexity and performance offered by HMFF-MobileNet, this experiment juxtaposed its performance against these models. Models such as Res2Net, HiFuse, Deformable Registration of Medical Images with Anatomical ShuffleNet V2, MnasNet, and MobileNetV3 were utilized as references. The details of these techniques are not mentioned because they have already been covered in Section 2. The following subsection describes the identical training and evaluation conditions that were applied to both the reference models and our proposed method.

4.3. Results on Cifar-10 Dataset

Table 2 compares multiple versions of the MobileNetV1 model and their

Table 2. After 50 epochs, the Cifar-10 dataset showed the following results.

Models	Params (Million)	Reduction	FLOPs (Billion)	Reduction	Accuracy
Res2Net	23.53	-	1.75	-	0.788
ShuffleNetV2	1.13	-	1.24	-	0.755
MnasNet	3.12	-	0.93	-	0.611
MobileNetV3	2.24	-	0.73	-	0.731
MobilenetV1	3.21	0%	0.567	0%	0.926
CMSFL-Net	0.86	-	0.65	-	0.794
kMobileNet16Ch	0.244	96.40%	0.092	83.79%	0.885
kMobileNet32Ch	0.403	87.47%	0.153	72.90%	0.910
kMobileNet64Ch	0.718	77.66%	0.251	55.65%	0.920
kMobileNet128Ch	1.32	58.83%	0.370	34.76%	0.923
HMFF-MobileNet16Ch	0.245	92.35%	0.090	84.11%	0.898
HMFF-MobileNet32Ch	0.375	88.31%	0.148	73.89%	0.913
HMFF-MobileNet64Ch	0.633	80.29%	0.238	57.98%	0.921
HMFF-MobileNet128Ch	1.12	65.02%	0.341	39.78%	**0.927**

adjustments in terms of trainable parameters, FLOPs reduction, and test accuracy.

With 3.21 million trainable parameters and 0.567 billion FLOPs, and a test accuracy of 92.62%, the MobileNetV1 model serves as a baseline for comparison. The changes are done in order to reduce the number of trainable parameters and FLOPs while maintaining test accuracy. Compared to the MobileNetV1 model, the kMobileNet models with 16Ch, 32Ch, 64Ch, and 128Ch reduced the number of trainable parameters and FLOPs by 83.79%, 72.90%, 55.65%, and 34.76%, respectively. Nevertheless, this comes at the expense of reduced test accuracy, with the kMobileNet 16Ch model having the lowest accuracy of 88.51%.

In comparison to the MobileNetV1 model, the HMFF-MobileNet models with 16Ch, 32Ch, 64Ch, and 128Ch have reduced the number of trainable parameters, and FLOPs by 84.11%, 73.89%, 57.98%, and 39.78%, respectively. These models' test accuracy, however, are slightly higher than that of the kMobilenet models, ranging from 89.83% to 92.78%.

After the initial decline, the loss curves for the HMFF-MobileNet-128Ch, HMFF-MobileNet-64Ch, and MobileNetV1 models show a small discrepancy between the training and validation losses, with the validation loss plateauing and the training loss continuing to decrease, indicating a moderate level of overfitting. This is most likely mitigated by the figure's effective early stopping implementation (Figure 3).

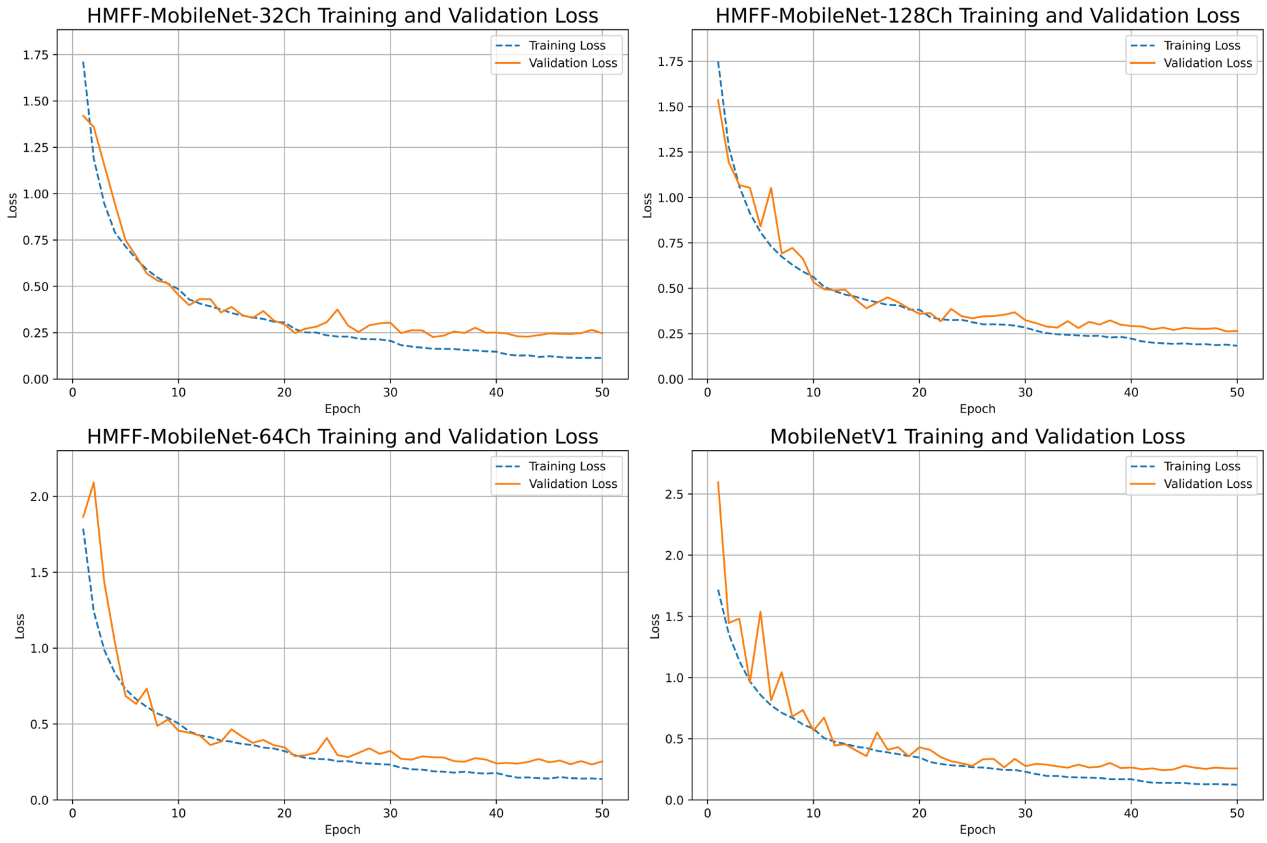

Figure 3. Experimental observation of (training and validation) loss vs. total number of epochs on Cifar-10 dataset.

In addition, According to Figure 4 and Figure 5, the various models appear to have performed well for most classes, with high precision and recall values. However, these models perform slightly worse on recall for the classes cat, bird, and dog, indicating that these models have a higher number of false negatives for these classes.

Overall, the findings indicate that altering the MobileNetV1 model can result in models with fewer parameters and FLOPs while still maintaining excellent accuracy. In terms of retaining accuracy while reducing parameters and FLOPs, the HMFF-MobileNet models tend to outperform the kMobileNet models.

4.4. Results on Malaria

For the Malaria dataset, Table 3 compares the performance of multiple variants of HMFF-MobileNet and the MobileNetV1 model. The results show that the HMFF-MobileNet models obtain the highest accuracy of all models, ranging from 96.92% to 97.24% accuracy. Figure 6, which shows the loss curves for each

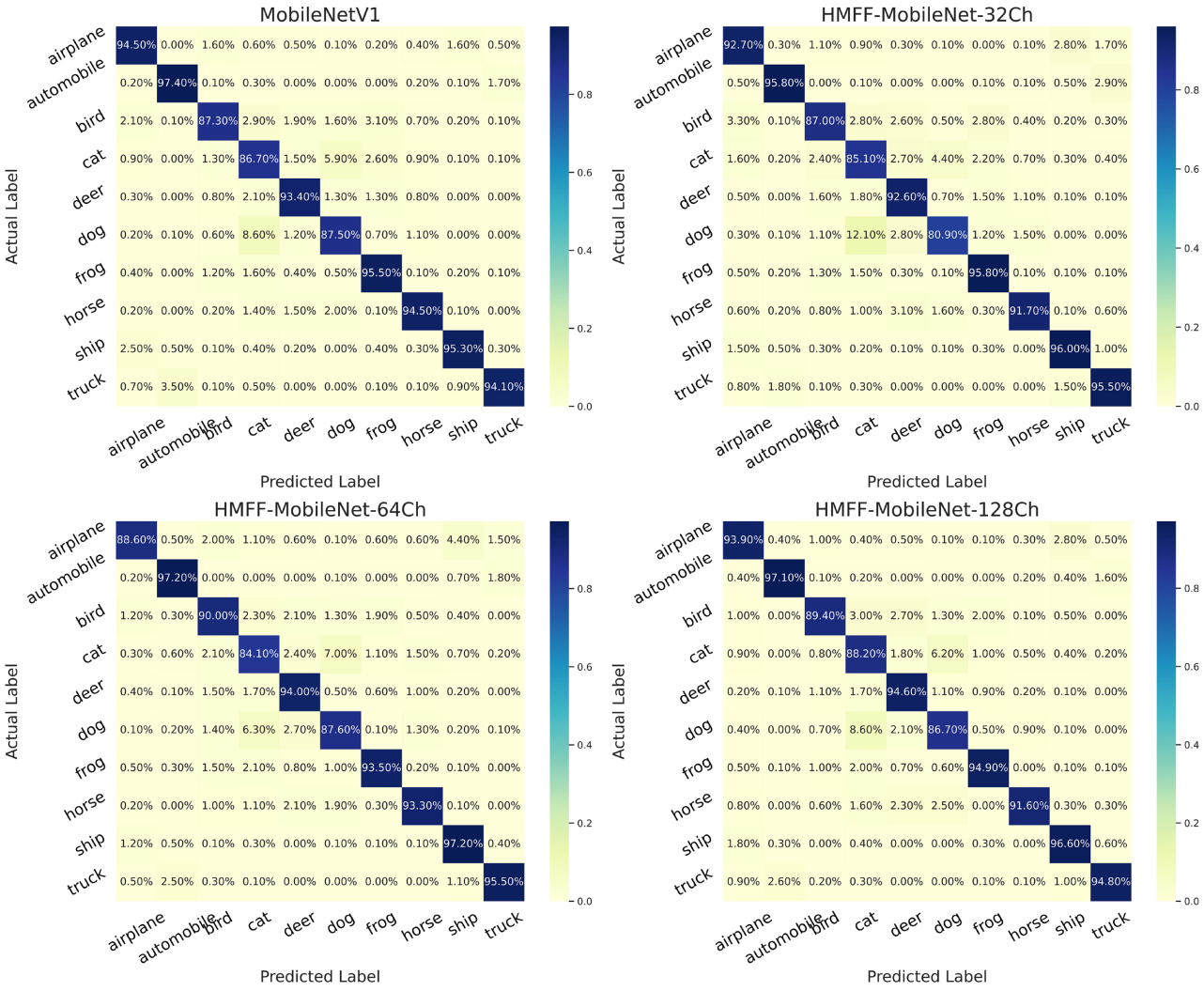

Figure 4. Confusion matrix showcasing the classification performance of the proposed methods on Cifar-10 dataset.

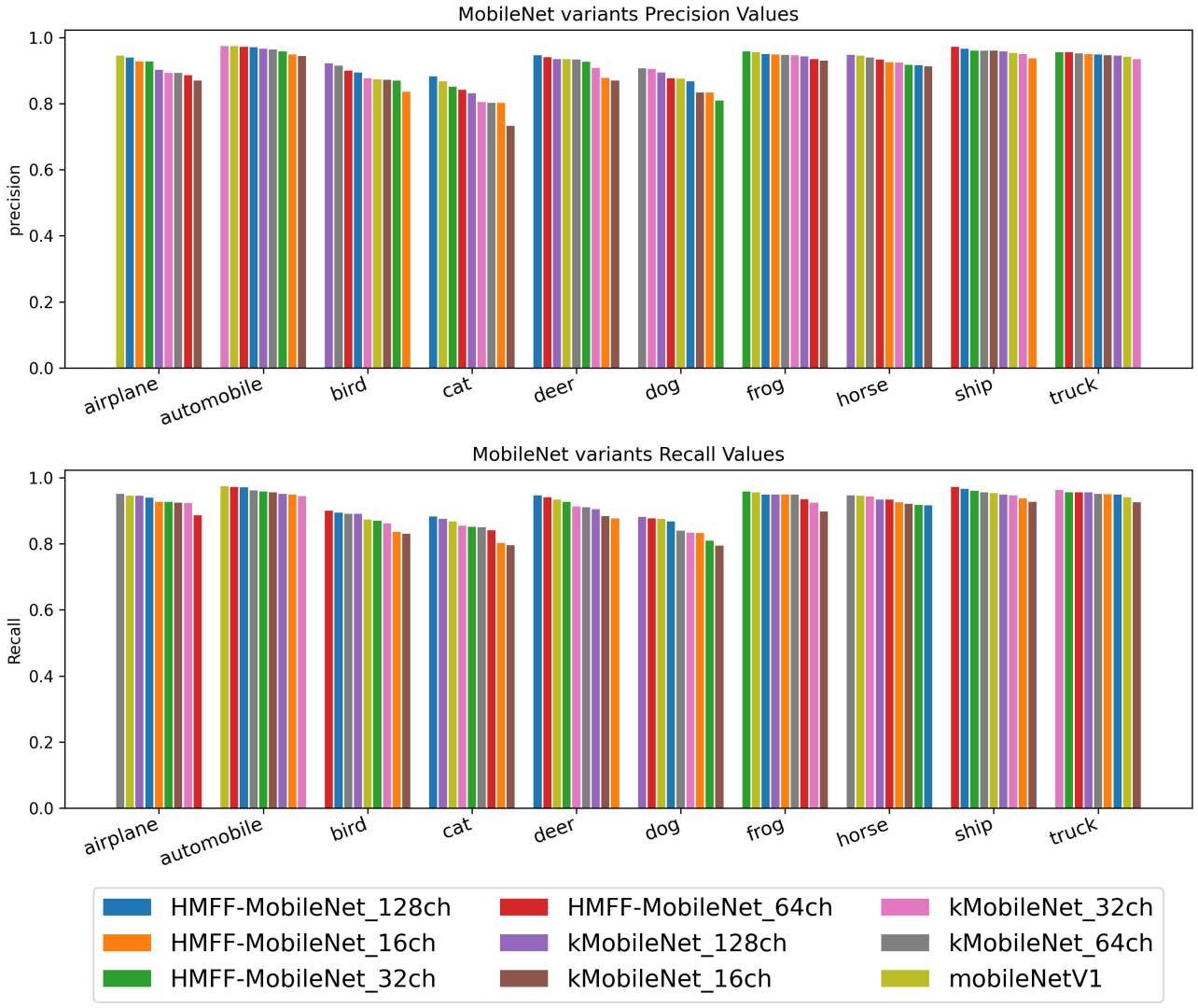

Figure 5. Performance analysis of proposed HMFF-MobileNet with the different baseline deep learning models for Cifar-10 dataset.

Table 3. After 50 epochs, the Malaria dataset showed the following results.

Models	Params (Million)	Reduction	FLOPs (Billion)	Reduction	Accuracy
MobileNetV1	3.21	100%	0.567	0%	0.966
kMobileNet16Ch	0.244	96.40%	0.092	83.79%	0.9652
kMobileNet32Ch	0.403	87.47%	0.153	72.90%	0.9673
kMobileNet64Ch	0.718	77.66%	0.251	55.65%	0.9670
kMobileNet128Ch	1.32	58.83%	0.370	34.76%	0.9706
HMFF-MobileNet16Ch	0.245	92.35%	0.090	84.11%	0.9692
HMFF-MobileNet32Ch	0.375	88.31%	0.148	73.89%	0.9706
HMFF-MobileNet64Ch	0.633	80.29%	0.238	57.98%	0.9721
HMFF-MobileNet128Ch	1.12	65.02%	0.341	39.78%	0.9724

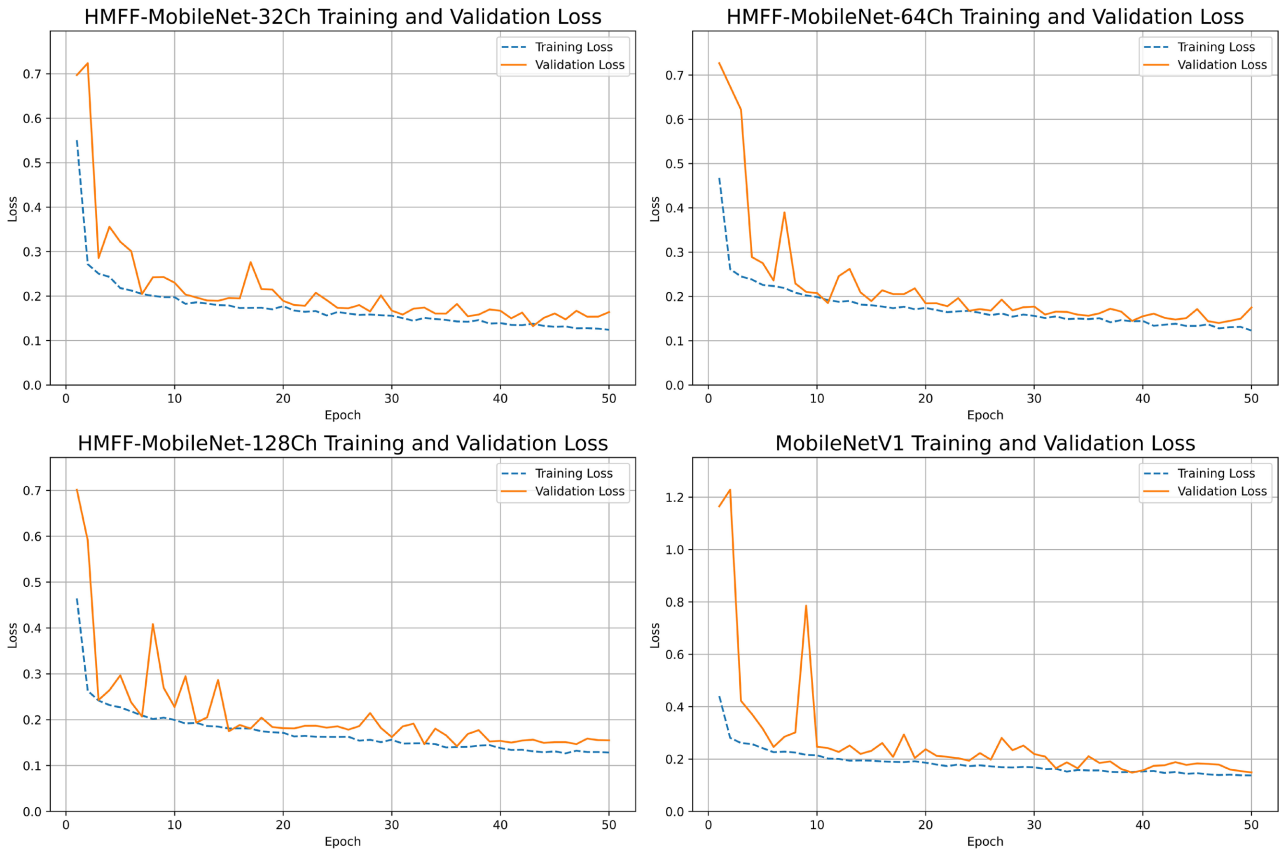

Figure 6. Experimental observation of (training and validation) loss vs. total number of epochs on Malaria dataset.

proposed model and shows how they perform on the training and validation datasets over the training epochs, illustrates the models' capacity for generalization. **Figure 7** and **Figure 8** show that these models achieve high precision and recall scores for all the classes, indicating that they have accurately classified the images in the malaria datasets while minimizing false positives and false negatives. This is an extremely promising outcome, as high accuracy on medical datasets like the malaria dataset is critical for proper diagnosis and treatment.

The table also indicates that reducing the number of channels in the convolutional layers (16Ch, 32Ch, 64Ch, and 128Ch) reduces the number of trainable parameters and FLOPs while having no effect on accuracy. This is a significant funding because decreasing the computational complexity of deep learning models is critical for applications that demand real-time processing or have limited computational resources.

Additionally, the results show that the proposed HMFF-MobileNet models outperform the standard MobileNetV1 model while requiring far fewer trainable parameters and FLOPs. This shows that the suggested model's HMFF module is effective at lowering computational complexity while maintaining good accuracy. Furthermore, the HMFF-MobileNet models' use of hierarchical feature fusion allows them to collect both low-level and high-level aspects in the data, which can lead to higher accuracy.

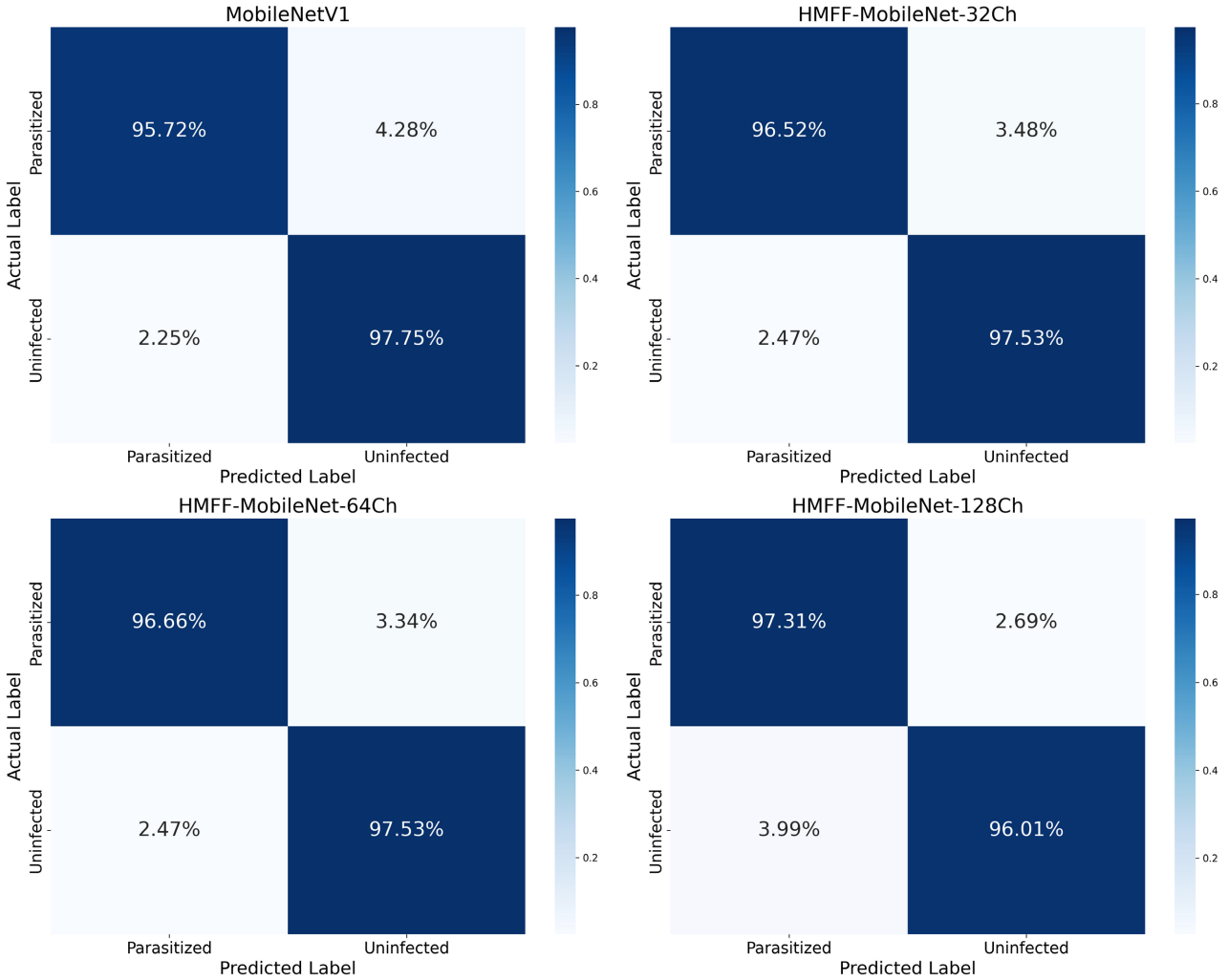

Figure 7. Confusion matrix showcasing the classification performance of the proposed methods on Malaria dataset.

4.5. Results on KvasirV1

HMFF-MobileNets were tested on the Kvasir dataset to evaluate its generalization capabilities. Table 4 showcases the performance of different models on the heterogeneous KvasirV1 dataset, highlighting HMFF-MobileNet models in particular The HMFF-MobileNet models include 16, 32, 64, and 128 channels, with varying levels of channel reduction. The results indicate that as the channel reduction increases, the trainable parameters and FLOPs decrease, while the test accuracy decreases. However, the HMFF-MobileNet models perform better in terms of accuracy than the kMobileNet models, despite having a slightly smaller reduction in parameters and FLOPs. This indicates that the HMFF module is effective at enhancing the model's performance by integrating multi-level features from different scales and resolutions. The results indicate that the HMFF module can be a useful technique for enhancing the performance of CNNs in a variety of computer vision tasks.

It is demonstrated that the HMFF-MobileNet 64Ch model outperforms the HMFF-MobileNet 128Ch model in terms of test accuracy. This is somewhat

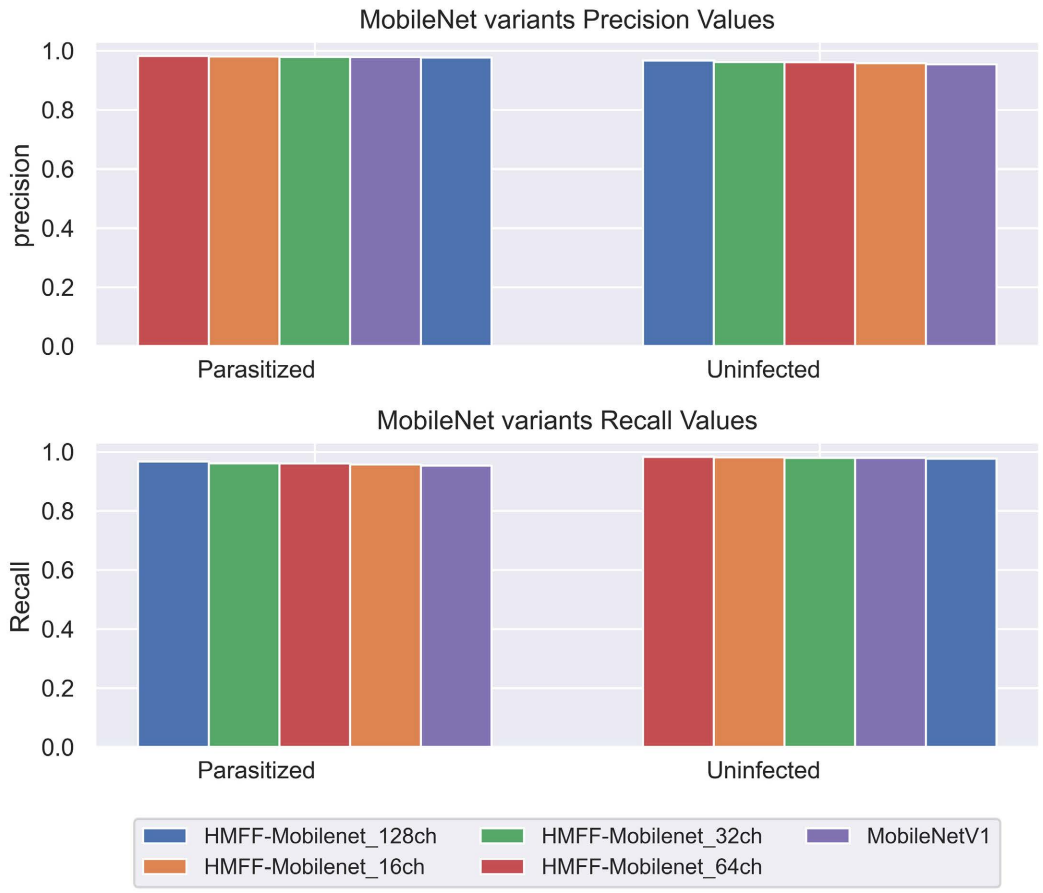

Figure 8. Performance analysis of proposed HMFF-MobileNet with the different baseline deep learning models for Malaria dataset.

Table 4. After 150 epochs, the KvasirV1 dataset showed the following results.

Model	Params (Million)	Reduction	FLOPs (Billion)	Reduction	Test Accuracy
VGG-19	143.68	-	19.67	-	0.6089
HiFuse_Tiny	82.49	-	8.13	-	0.7287
HiFuse_Small	93.82	-	8.84	-	0.7314
HiFuse_Base	127.80	-	10.97	-	**0.7474**
MobileNetV1	3.21	0%	0.567	0%	0.7175
kMobileNet16Ch	0.244	96.40%	0.092	83.79%	0.6587
kMobileNet32Ch	0.403	87.47%	0.153	72.90%	0.6800
kMobileNet64Ch	0.718	77.66%	0.251	55.65%	0.7013
kMobileNet128Ch	1.324	58.83%	0.370	34.76%	0.7288
HMFF-MobileNet16Ch	0.245	92.35%	0.090	84.11%	0.6925
HMFF-MobileNet32Ch	0.375	88.31%	0.148	73.89%	0.7288
HMFF-MobileNet64Ch	0.633	80.29%	0.238	57.98%	**0.7438**
HMFF-MobileNet128Ch	1.12	65.02%	0.341	39.78%	0.7425

surprising because one would expect a model with more channels to outperform a model with fewer channels. Again, this indicates that the complexity of the images influences the optimal number of channels per group selection. In other words, images that are less complex may require fewer channels per group.

To validate the effectiveness of the proposed methods, the training and validation loss curves were plotted. As shown in **Figure 9**, all models show a significant improvement at the start of training, which is typical because initial weights are adjusted significantly to reduce loss. The most stable and closely aligned training and validation loss curves are found in the HMFF-MobileNet-64Ch model, indicating effective learning and generalization without significant overfitting. There is some divergence between training and validation loss in the MobileNetV1 and HMFF-MobileNet-32Ch models, but it is not as pronounced, indicating mild overfitting. The HMFF-MobileNet-128Ch model, on the other hand, exhibits quite a bit overfitting, as indicated by the persistent gap between training and validation losses.

In addition, According to **Figure 10**, the performance of HMFF-MobileNet models have consistently performed in identifying "polyps" and "ulcerative-colitis", with the latter class consistently showing high true positive rates. That is confirmed with high precision and recall values in **Figure 11**. Moreover, it underperformed in other classes, which including "dyed-resection margins" and "polyps", with

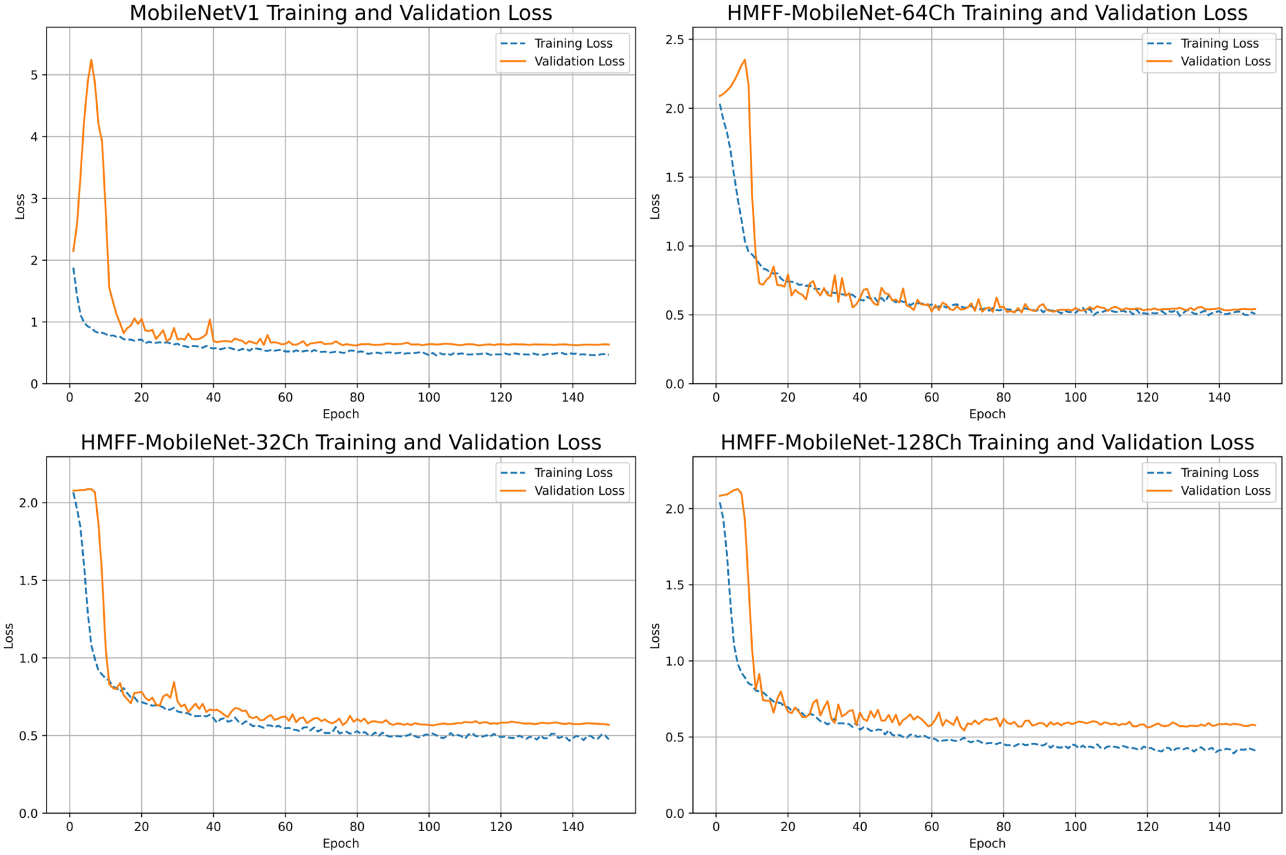

Figure 9. Confusion matrix showcasing the classification performance of the proposed methods on KvasirV1 dataset.

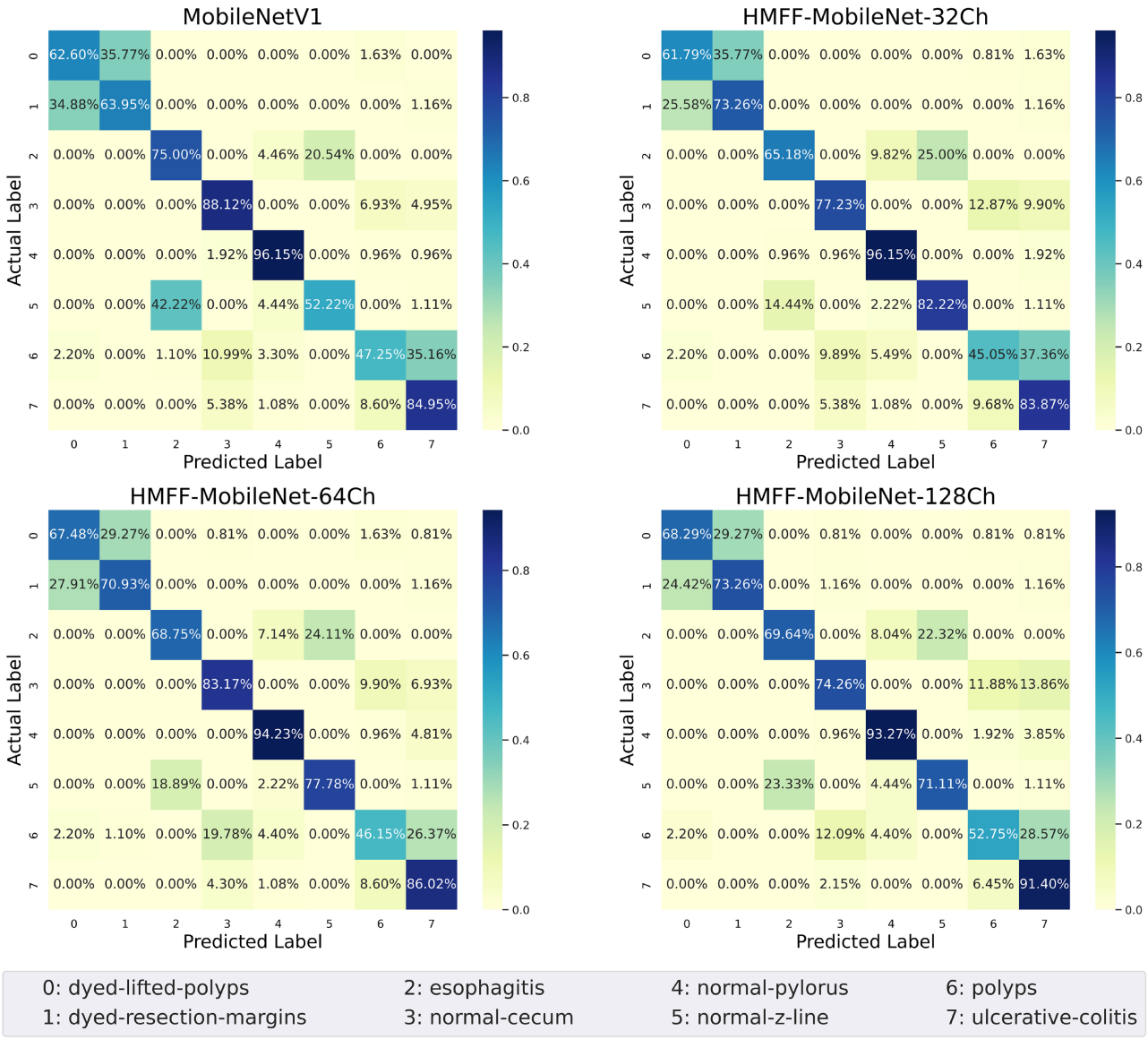

Figure 10. Experimental observation of (training and validation) loss vs. total number of epochs on KvasirV1 dataset.

low recall and precision values. The differences in true positive rates between models for "dyed-lifted-polyps" and "normal-pylorus" suggest that there may be room for improvement in the feature extraction and model training processes to achieve more balanced classification performance across all classes.

4.6. Models' Inference on Hardware

A neural network's throughput is defined as the maximum number of input instances that the network can process in a unit of time (e.g. a second) [49]. Unlike latency, which requires the processing of a single instance, we would like to process as many instances as possible in parallel to achieve maximum throughput. The effective parallelism is obviously dependent on data, model, and device. To accurately measure throughput, we first performed the following two steps:

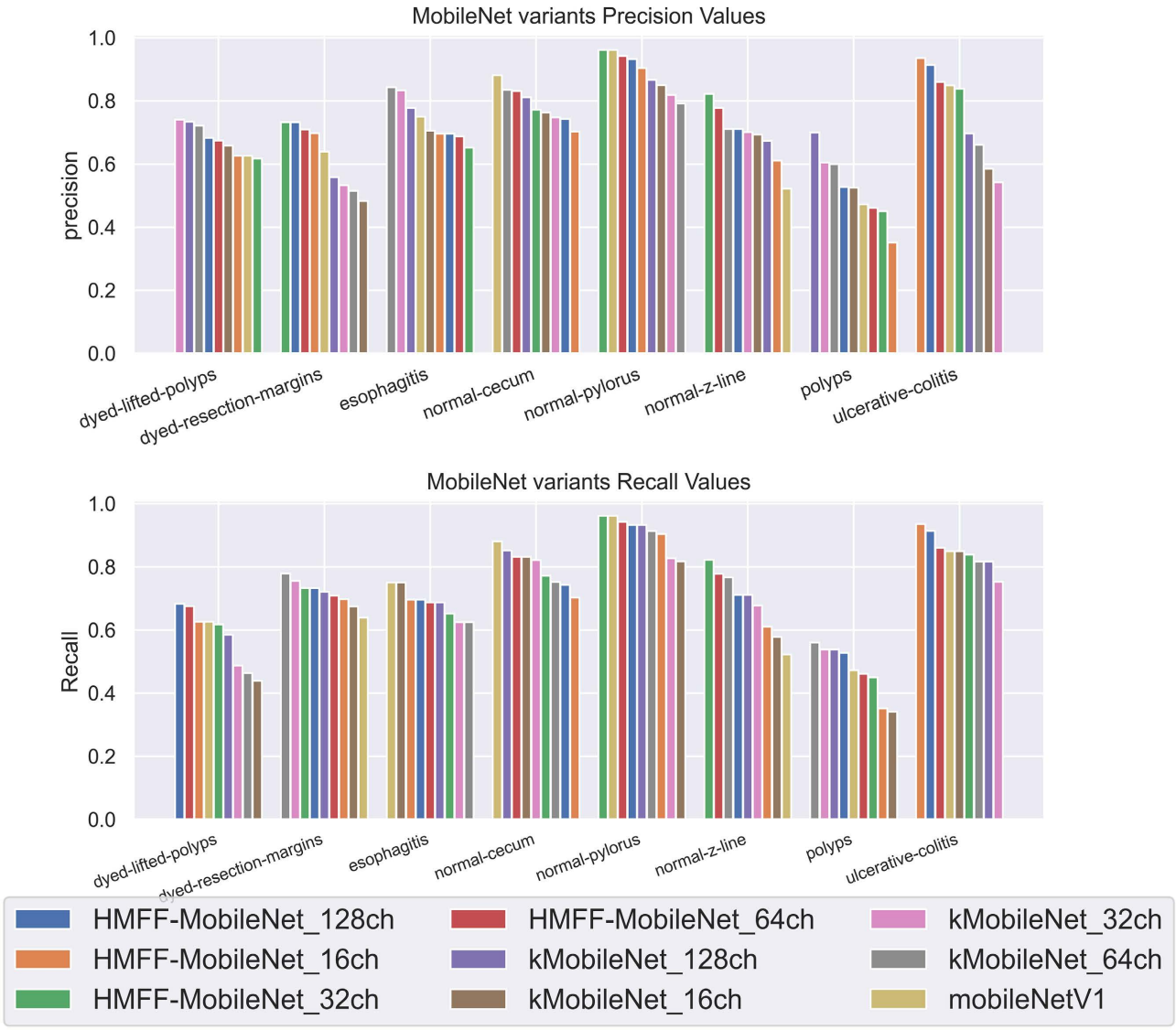

Figure 11. Performance analysis of proposed HMFF-MobileNet with the different baseline deep learning models for KvasirV1 dataset.

1) We calculated the optimal batch size for maximum parallelism, and 2) given this optimal batch size, we calculated the number of instances the network can perform in one second. To determine the optimum batch size, which depends on the hardware type and network size, we used a for loop, increasing the batch size iteratively until the run-time error is achieved. The largest batch size is 256 that the GPU can process for the different neural network models and the input data it processes.

The latency and throughput of the compared models were measured using an image malaria dataset with an image size of 224 × 224. **Table 5** shows the average latency and throughput for the different models that run on the cloud machine of paperspace (A4000, 45GB RAM, 8CPU, 16GB GPU Nvidia). All experiments have been done without compressing the different models.

Table 5. The average latency and throughput for the different models.

Model	Latency(s)	Throughput(s)
MobileNetV1	0.0647	358.048
kMobileNet16Ch	0.0491	284.72
kMobileNet32Ch	0.0499	290.54
kMobileNet64Ch	0.0554	284.78
kMobileNet128Ch	0.0626	299.66
HMFF-MobileNet16Ch	0.0481	276.53
HMFF-MobileNet32Ch	0.04918	286.49
HMFF-MobileNet64Ch	0.0551	279.53
HMFF-MobileNet128Ch	0.0620	285.12

In terms of latency, the average latency of the HMFF-MobileNet models is slightly better than the regular MobileNetV1 model, specifically the HMFF-MobileNet models with 16 channels, as shown in table. These findings imply that the HMFF-MobileNet models with 128 channels could be an appropriate choice for applications requiring high accuracy, low latency, and high throughput. Because it trades off low latency, fewer parameters, and complexity.

5. Conclusion

In this study, a lightweight convolution neural network named hierarchical multi-scale feature fusion MobileNet (HMFF-MobileNet), an improved variant of MobileNet with multi-scale feature fusion method is proposed. The aim of this research was to develop an image classification model with fewer parameters and low computational complexity. To achieve this, this study developed a hierarchical multi-scale feature fusion (HMFF) module that can effectively extract important cross-dimensional features and spatial information at various scales in images by learning representations from a large effective receptive field. The experimental results demonstrate that our proposed network achieves state-of-the-art performance on several benchmark datasets while reducing the number of parameters and computational complexity, making it a promising solution for real-world applications. For further research, the proposed HMFF module can be integrated into various deep learning architectures to improve their performance in tasks, such as image classification, object detection, and semantic segmentation.

Authors' Contributions

Conceptualization: A.D., R.W.M. and A.O.K.; formal analysis: A.D.; methodology: A.D., R.W.M. and A.O.K.; software: A.D.; supervision: R.W.M. and A.O.K.; validation: A.D.; visualization: A.D.; writing—original draft preparation: A.D.; writing—review and editing: R.W.M. and A.O.K. All authors have read and agreed to the published version of the manuscript.

Data Availability

Data available: CIFAR-10 [42], Malaria [43], KvasirV1: https://datasets.simula.no/kvasir/.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017) Imagenet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, **60**, 84-90. https://doi.org/10.1145/3065386

[2] Wang, N. and Yeung, D.Y. (2013) Learning a Deep Compact Image Representation for Visual Tracking. *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Nevada, 5-10 December 2013, 809-817.

[3] Wei, W., Can, T., Xin, W., Yanhong, L., Yongle, H. and Ji, L. (2019) Image Object Recognition via Deep Feature-Based Adaptive Joint Sparse Representation. *Computational Intelligence and Neuroscience*, **2019**, Article ID: 8258275. https://doi.org/10.1155/2019/8258275

[4] Li, F., Wang, C., Liu, X., Peng, Y. and Jin, S. (2018) A Composite Model of Wound Segmentation Based on Traditional Methods and Deep Neural Networks. *Computational Intelligence and Neuroscience*, **2018**, Article ID: 4149103. https://doi.org/10.1155/2018/4149103

[5] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv: 1409.1556.

[6] Targ, S., Almeida, D. and Lyman, K. (2016) Resnet in Resnet: Generalizing Residual Architectures. arXiv: 1603.08029.

[7] Li, C. and Shi, C.R. (2018) Constrained Optimization Based Low-Rank Approximation of Deep Neural Networks. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018, Springer, Cham, 746-761. https://doi.org/10.1007/978-3-030-01249-6_45

[8] Wen, W., Wu, C., Wang, Y., Chen, Y. and Li, H. (2016) Learning Structured Sparsity in Deep Neural Networks. arXiv: 1608.03665.

[9] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. and Bengio, Y. (2017) Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. *The Journal of Machine Learning Research*, **18**, 6869-6898.

[10] Huang, G., Liu, S.X., van der Maaten, L. and Weinberger, K.Q. (2018) CondenseNet: An Efficient Dense Netusing Learned Group Convolutions. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 2752-2761. https://doi.org/10.1109/CVPR.2018.00291

[11] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. AND Chen, L.C. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[12] Ma, N., Zhang, X., Zheng, H.T. and Sun, J. (2018) ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018, Springer, Cham, 122-138.

https://doi.org/10.1007/978-3-030-01264-9_8

[13] Howard, A.G., Zhu, M., Chen, B., *et al.* (2017) Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv: 1704.04861.

[14] Schwarz Schuler, J.P., Also, S.R., Puig, D., Rashwan, H. and Abdel-Nasser, M. (2022) An Enhanced Scheme for Reducing the Complexity of Pointwise Convolutions in CNNs for Image Classification Based on Interleaved Grouped Filters without Divisibility Constraints. *Entropy*, **24**, Article 1264. https://doi.org/10.3390/e24091264

[15] Sunkara, R. and Luo, T. (2022) No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. In: Amini, M.R., Canu, S., Fischer, A., Guns, T., Kralj Novak, P. and Tsoumakas, G., Eds., *ECML PKDD* 2022: *Machine Learning and Knowledge Discovery in Databases*, Springer, Cham, 443-459. https://doi.org/10.1007/978-3-031-26409-2_27

[16] Ku, T., Yang, Q. and Zhang, H. (2021) Multilevel Feature Fusion Dilated Convolutional Network for Semantic Segmentation. *International Journal of Advanced Robotic Systems*, **18**, 1-12. https://doi.org/10.1177/17298814211007665

[17] Li, X., Song, D. and Dong, Y. (2020) Hierarchical Feature Fusion Network for Salient Object Detection. *IEEE Transactions on Image Processing*, **29**, 9165-9175. https://doi.org/10.1109/TIP.2020.3023774

[18] Schuler, J.P.S., Romani, S., Abdel-Nasser, M., Rashwan, H. and Puig, D. (2022) Grouped Pointwise Convolutions Reduce Parameters in Convolutional Neural Networks. *Mendel*, **28**, 23-31. https://doi.org/10.13164/mendel.2022.1.023

[19] Denil, M., Shakibi, B., Dinh, L., Ranzato, M. and De Freitas, N. (2013) Predicting Parameters in Deep Learning. arXiv: 1306.0543.

[20] Hinton, G., Vinyals, O. and Dean, J. (2015) Distilling the Knowledge in a Neural Network. arXiv: 1503.02531.

[21] Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J. and Keutzer, K. (2016) SqueezeNet: AlexNet-Level Accuracy with 50× Fewer Parameters and <0.5 MB Model Size. arXiv: 1602.07360.

[22] Tan, M., Chen, B., Pang, R., *et al.* (2019) Mnasnet: Platform-Aware Neural Architecture Search for Mobile. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 2815-2823. https://doi.org/10.1109/CVPR.2019.00293

[23] Qian, S., Ning, C. and Hu, Y. (2021) MobileNetV3 for Image Classification. 2021 *IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering* (*ICBAIE*), Nanchang, 26-28 March 2021, 490-497. https://doi.org/10.1109/ICBAIE52039.2021.9389905

[24] Zhang, X., Zhou, X., Lin, M. and Sun, J. (2018) ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 6848-6856. https://doi.org/10.1109/CVPR.2018.00716

[25] Xia, H., Sun, W., Song, S. and Mou, X. (2020) Md-Net: Multi-Scale Dilated Convolution Network for CT Images Segmentation. *Neural Processing Letters*, **51**, 2915-2927. https://doi.org/10.1007/s11063-020-10230-x

[26] Liu, S., Huang, D. and Wang, Y. (2018) Receptive Field Block Net for Accurate and Fast Object Detection. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018, Springer, Cham, 404-419. https://doi.org/10.1007/978-3-030-01252-6_24

[27] Lei, X., Pan, H. and Huang, X. (2019) A Dilated CNN Model for Image Classification. *IEEE Access*, **7**, 124087-124095.

https://doi.org/10.1109/ACCESS.2019.2927169

[28] Wang, W., Hu, Y., Zou, T., Liu, H., Wang, J. and Wang, X. (2020) A New Image Classification Approach via Improved MobileNet Models with Local Receptive Field Expansion in Shallow Layers. *Computational Intelligence and Neuroscience*, **2020**, Article ID: 8817849. https://doi.org/10.1155/2020/8817849

[29] Sun, W., Zhang, X. and He, X. (2020) Lightweight Image Classifier Using Dilated and Depthwise Separable Convolutions. *Journal of Cloud Computing*, **9**, Article No. 55. https://doi.org/10.1186/s13677-020-00203-9

[30] Drossos, K., Mimilakis, S.I., Gharib, S., Li, Y. and Virtanen, T. (2020) Sound Event Detection with Depthwise Separable and Dilated Convolutions. 2020 *International Joint Conference on Neural Networks* (*IJCNN*), Glasgow, 19-24 July 2020, 1-7. https://doi.org/10.1109/IJCNN48605.2020.9207532

[31] Xie, W., Jiao, L. and Hua, W. (2022) Complex-Valued Multi-Scale Fully Convolutional Network with Stacked-Dilated Convolution for PolSAR Image Classification. *Remote Sensing*, **14**, Article 3737. https://doi.org/10.3390/rs14153737

[32] Kaddar, B., Fizazi, H., Hernández-Cabronero, M., Sanchez, V. and Serra-Sagristà, J. (2021) DivNet: Efficient Convolutional Neural Network via Multilevel Hierarchical Architecture Design. *IEEE Access*, **9**, 105892-105901. https://doi.org/10.1109/ACCESS.2021.3099952

[33] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. (2017) Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**, 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

[34] Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. (2017) Pyramid Scene Parsing Network. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 6230-6239. https://doi.org/10.1109/CVPR.2017.660

[35] Wang, G., Yuan, G., Li, T. and Lv, M. (2018) An Multi-Scale Learning Network with Depthwise Separable Convolutions. *IPSJ Transactions on Computer Vision and Applications*, **10**, Article No. 1. https://doi.org/10.1186/s41074-017-0037-0

[36] Huo, X., Sun, G., Tian, S., *et al.* (2024) HiFuse: Hierarchical Multi-Scale Feature Fusion Network for Medical Image Classification. *Biomedical Signal Processing and Control*, **87**, Article ID: 105534. https://doi.org/10.1016/j.bspc.2023.105534

[37] Olimov, B., Subramanian, B., Ugli, R.A.A., Kim, J.S. and Kim, J. (2023) Consecutive Multiscale Feature Learning-Based Image Classification Model. *Scientific Reports*, **13**, Article No. 3595. https://doi.org/10.1038/s41598-023-30480-8

[38] Lian, X., Pang, Y., Han, J. and Pan, J. (2021) Cascaded Hierarchical Atrous Spatial Pyramid Pooling Module for Semantic Segmentation. *Pattern Recognition*, **110**, Article ID: 107622. https://doi.org/10.1016/j.patcog.2020.107622

[39] Mehta, S., Rastegari, M., Shapiro, L. and Hajishirzi, H. (2019) ESPNetv2: A Light-Weight, Power Efficient, and General Purpose Convolutional Neural Network. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 9182-9192. https://doi.org/10.1109/CVPR.2019.00941

[40] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv: 1502.03167.

[41] Ramachandran, P., Zoph, B. and Le, Q.V. (2017) Searching for Activation Functions. arXiv: 1710.059417.

[42] Krizhevsky, A. and Hinton, G. (2009) Learning Multiple Layers of Features from Tiny Images.

https://scholar.google.com/scholar?q=Learning+Multiple+Layers+of+Features+from+Tiny+Images&hl=zh-CN&as_sdt=0&as_vis=1&oi=scholart

[43] Rajaraman, S., Antani, S.K., Poostchi, M., *et al.* (2018) Pre-Trained Convolutional Neural Networks as Feature Extractors toward Improved Malaria Parasite Detection in Thin Blood Smear Images. *PeerJ*, **6**, e4568. https://doi.org/10.7717/peerj.4568

[44] Pogorelov, K., Randel, K.R., Griwodz, C., *et al.* (2017) Kvasir: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection. Association for Computing Machinery, New York. https://doi.org/10.1145/3193289

[45] Abadi, M., Agarwal, A., Barham, P., *et al.* (2015) TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://tensorflow.org/

[46] Kingma, D.P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. arXiv: 1412.6980.

[47] Ge, R., Kakade, S.M., Kidambi, R. and Netrapalli, P. (2019) The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure for Least Squares. *Proceedings of the* 33*rd International Conference on Neural Information Processing Systems*, Vancouver, 8-14 December 2019, 14977-14988.

[48] Glorot, X. and Bengio, Y. (2010) Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Journal of Machine Learning Research*, **9**, 249-256.

[49] Hanhirova, J., Kamarainen, T., Seppala, S., Siekkinen, M., Hirvisalo, V. and Yla-Jaaski, A. (2018) Latency and Throughput Characterization of Convolutional Neural Networks for Mobile Computer Vision. *Proceedings of the* 9*th ACM Multimedia Systems Conference*, Amsterdam, 12-15 June 2018, 204-215. https://doi.org/10.1145/3204949.3204975

Abbreviations

The following abbreviations are used in this manuscript:

CNNs	Convolutional neural networks
DDSC	Depthwise dilated separable convolution
HMFF	Hierarchical multi-scale feature fusion
CMSFL	Consecutive multiscale feature learning
ESP	Efficient spatial pyramid
Gconv	Grouped pointwise convolution
HFF	Hierarchical feature fusion
FLOPs	Floating-point operations
TP	True positive
TN	True negative
FP	False positive
FN	False negative