Computers in Biology and Medicine 181 (2024) 109047

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Polar contrast attention and skip cross-channel aggregation for efficient learning in U-Net

Mohammed Lawal, Dewei Yi *

Department of Computing Science, University of Aberdeen, Aberdeen, AB24 3UE, United Kingdom

ARTICLE INFO

Keywords:
Medical image processing
Lightweight models
Polar transformation
Polyp segmentation
Skin lesion

ABSTRACT

The performance of existing lesion semantic segmentation models has shown a steady improvement with the introduction of mechanisms like attention, skip connections, and deep supervision. However, these advancements often come at the expense of computational requirements, necessitating powerful graphics processing units with substantial video memory. Consequently, certain models may exhibit poor or non-existent performance on more affordable edge devices, such as smartphones and other point-of-care devices. To tackle this challenge, our paper introduces a lesion segmentation model with a low parameter count and minimal operations. This model incorporates polar transformations to simplify images, facilitating faster training and improved performance. We leverage the characteristics of polar images by directing the model's focus to areas most likely to contain segmentation information, achieved through the introduction of a learning-efficient polar-based contrast attention (PCA). This design utilizes Hadamard products to implement a lightweight attention mechanism without significantly increasing model parameters and complexities. Furthermore, we present a novel skip cross-channel aggregation (SC^2A) approach for sharing cross-channel corrections, introducing Gaussian depthwise convolution to enhance nonlinearity. Extensive experiments on the ISIC 2018 and Kvasir datasets demonstrate that our model surpasses state-of-the-art models while maintaining only about 25K parameters. Additionally, our proposed model exhibits strong generalization to cross-domain data, as confirmed through experiments on the PH^2 dataset and CVC-Polyp dataset. In addition, we evaluate the model's performance in a mobile setting against other lightweight models. Notably, our proposed model outperforms other advanced models in terms of IoU and Dice score, and running time.

1. Introduction

Skin cancer and its various types are responsible for thousands of deaths every year with reported number of cases and deaths rising yearly [1–3]. Systems built to automatically detect and diagnose the disease need to be able to identify the area of interest before any further processing can take place. This is where semantic segmentation comes in, which is where a model goes through an image and labels all the pixels.

Training a segmentation model requires substantial training annotated data. This might not be possible especially in the field of medical image processing when acquiring training images can be challenging. Difficulties like experts individually annotating each image which is not only labour intensive and an inefficient application of already sparse medical human resources, and privacy concerns in utilizing patient data [4–6]. This is where the notion of data augmentation comes in where existing data are altered. This alteration can be in the form of random angle rotations, resizes, and flipping [7,8]. Another form of data augmentation is polar transformations.

Polar transformation is a geometric conversion that takes a point from a Cartesian coordinate system and maps it into a polar coordinate system. In the case of an image, polar transformation involves repositioning each pixel in the Cartesian image according to its distance from a central point and its angular orientation. Concerning data augmentation in the field of medical image segmentation, polar transformation can offer distinct advantages when dealing with datasets that contain images of a circular or radial nature [9,10]. This transformation method can yield substantial benefits in tasks like polyp, lesion, and liver segmentation. By applying polar transformation to medical images with circular attributes, such as lesions, these circular images can become much less complex as a polar transformation of a perfect circle is a straight line in polar coordinates [10]. This transformation can facilitate the learning process for neural networks, potentially enhancing their performance. Polar transformation expedites the training of models, allowing networks to converge more rapidly. This, in turn, can reduce training time and lead to decreased computational complexity and associated costs.

* Corresponding author.
E-mail address: dewei.yi@abdn.ac.uk (D. Yi).

https://doi.org/10.1016/j.compbiomed.2024.109047
Received 21 March 2024; Received in revised form 31 July 2024; Accepted 18 August 2024
Available online 24 August 2024

The proliferation of artificial intelligence has brought many benefits to the field of medicine, from decision support to health data processing and medical image processing like semantic segmentation [11,12]. Artificial intelligence driven systems usually require bespoke hardware like a graphics processing unit to run which takes them out of the reach of edge devices like mobile phones. This makes the area of edge computing ripe for exploration for the application of machine learning in edge devices to improve both the well-being of patients and healthcare practitioners. This is because models can be designed to be lightweight enough to run on inexpensive point of care devices or even smartphones. Some of the benefits of medical applications in edge devices are improved access to healthcare, especially in parts of the world where physicians are in short supply and cost savings [13,14]. Mobile devices also have applications in telemedicine where consultations, examinations, and follow ups can take place entirely online using patients smartphones [15].

Lightweight CNNs have proven to form a basis for efficient and accurate semantic segmentation tasks for medical images [16]. Multilayer perceptrons (MLPs) have been shown to successfully reduce number of parameters and computation all while maintaining and in some cases even beating established segmentation models [17]. Segmentation models can be deployed on modestly powered devices if the reduction in computational requirements and parameter count can be significant enough. In [18], they were able to reduce parameter count and computational cost by utilizing convolutions with fewer feature counts and incorporating Hadamard products as a mechanism of achieving an attention like mechanism to improve the performance of the model. In [19], they were able to show that channel shuffling has the ability to reduce computational requirements without a significant reduction in system accuracy.

In this paper, we propose an efficient low parameter count encoder–decoder model that performs semantic segmentation on skin lesions. We also propose a novel, fast, and lightweight attention mechanism to improve semantic segmentation performance. The attention mechanism divides an image into chunks and draws the attention of the model to the chunk that contains the boundary information between the background and foreground thereby improving segmentation performance. The module is implemented with Hadamard products instead of expensive convolutions and linear transformation operations, this helps in reducing model parameter count and improving inference time. The model also includes a lightweight feature shuffling mechanism utilizing lightweight aggregation operations instead of convolutional operation to improve information flow and further reduce computation without a significant loss in segmentation performance. These contributions results in a model weight file of about 100 kB (kilobytes). The model produces a polar segmentation mask which goes through a Cartesian transformation to get the final prediction segmentation mask. In summary, we make the following contributions:

- We propose a new and novel low parameter count and low computational load model that trains on Cartesian images that go through a polar transformation to produce polar images. The model produces polar masks that go through a Cartesian transformation to generate the final prediction mask.
- We develop a lightweight learning efficient polar contrast attention block that divides an image into chunks and draws the attention of the model to the chunk that contains the boundary information between the background and foreground thereby improving segmentation performance. The module is implemented with Hadamard products instead of expensive convolutional and linear transformation which keeps parameter count low and improves inference time. Moreover, to make feature extraction more efficient, we combine group normalization layer with GELU to speed up training and increase non-linearity.
- To model cross-channel correlations and spatial correlations in a computation-efficient manner, we propose a new skip cross-channel aggregation leveraging aggregation operations instead of expensive convolutions thereby reducing computational load and increasing speed, where we share cross-channel correlations in a skip-connection way along with new Gaussian depthwise convolution.
- We carry out an in-depth evaluation of the model against other state-of-the-art (SOTA) segmentation models, where our proposed model outperforms other models meanwhile being the most lightweight with regard to parameter count and low computational requirements. Moreover, we carry out an extensive ablation study to investigate the effects of each component of the model.
- We also conduct a mobile case study to test the model performance in point of care scenarios, where our proposed model outperforms other SOTA lightweight models both in accuracy and inference time.

2. Related work

2.1. Semantic segmentation

The encoder–decoder architecture is able to improve semantic segmentation performance even in the absence of a lot of training images, as in [20]. They propose upconvolution operations on the decoder phase of the network to recover information from the encoder in conjunction with skip connections that transfer information across blocks on the same level in the network. Attention-based networks can also enhance the process of medial image segmentation, as in [21]. They were also to improve segmentation performance by propagating the segmentation mask from the current epoch to subsequent epochs to help the network refine its output masks and direct its focus towards relevant regions of the image.

Research by Zhang et al. [19] explores the application of channel shuffle in network design. They propose convolution units that combine channel shuffling and pointwise group convolution in an attempt to reduce model complexity. They were able to reduce complexity while maintaining similar accuracy to other segmentation models. This serves as our inspiration for designing the Skip Cross-channel Aggregation block where we improve information flow to improve segmentation performance while keeping model size and computational load low.

Depthwise separable convolutional blocks can improve the performance of U-Net style architectures. In [22], they incorporate a depthwise separable convolution into a U-Net style architecture to capture more information from the image at the head of the encoder. They also design a channel split attention block for capturing feature maps at different scale to improve segmentation performance.

Transformers have self-attention properties allowing them to capture global context information for improving segmentation performance [23]. In [23], they improve lesion segmentation performance by capturing information across different shape and size variations and using that information in generating the final segmentation mask. Attention mechanisms were also shown to improve segmentation performance in [24] where they introduce the dot product, or otherwise known as the Hadamard product, attention.

Combining multiple networks in a model can improve overall performance as shown in [25,26]. In [25], the first network extracts the general area of interest and another network uses that information to perform semantic segmentation. The second network in [27] uses the output of the first network as attention information to help focus on the relevant parts of the image in carotid artery segmentation tasks. Semantic segmentation models can also be employed to improve classification tasks as shown in [28,29].

2.2. Lightweight semantic segmentation

Research by Valanarasu and Patel [17] explores employing attention style mechanisms while keeping the model light in terms of trainable parameters, they utilize feature tokenization and shifting to achieve this and also employ MLPs in latent space because of their simplicity as a way of reducing parameter count and computational requirements. This combination allows them to perform competitively against other segmentation models while utilizing much fewer parameters and operations.

In [18], they also utilize attention-based mechanisms and low parameter count networks to produce lightweight models that can run on inexpensive devices. They argue that conventional attention-based mechanisms are heavy parameter-wise and require high computational resources because of their quadratic complexity nature. To combat this, they propose using Hadamard products to implement similar attention mechanisms in linear complexity. They also utilize a mechanism that outputs a segmentation mask at each level of the network, and the individual masks are concatenated at the output stage of the network to produce the final segmentation mask. These optimizations produce a model that performs competitively against other segmentation models while drastically reducing parameter count and operations. Lightweight attention mechanisms have also been employed in [30] where they combine elements of CNNs and transformers in a lightweight segmentation architecture. Dilation convolutions together with channel splitting can reduce memory consumption as shown in [16,31]. Dilation convolution is a construct that allows a smaller kernel size to have similar receptive fields to larger kernel sizes. This allows the model to see more while using less memory. Other techniques for lightweight architectures can be found in [32] where they combine a larger kernel with depthwise convolutions for a lightweight model.

Dilation convolutions together with channel splitting can reduce memory consumption as shown in [16,31]. Dilation convolution is a construct that allows a smaller kernel size to have similar receptive fields to larger kernel sizes. This allows the model to see more while using less memory. Depthwise convolutions can also reduce model size as shown in [32] where they combine a larger kernel with depthwise convolutions for a lightweight model.

Our model follows a U-Net encoder–decoder style architecture. We also utilize feature shuffling mechanisms and attention-like mechanisms using Hadamard products to improve segmentation performance while reducing parameter count and computing load.

2.3. Polar transformation in machine learning

Incorporating polar transformations in machine learning applications has been shown to improve model performance [10]. The polar transformations can be automatically performed by the system during training as in [10], or the system can request users to select an origin for the transformation on the image as in [33]. In [10], they apply polar transformation on the input images and train the model with polar images. The model produces polar outputs which the system then converts back to Cartesian images. Their investigation shows the ability of polar transformations to reduce dimensionalities of images and also improve segmentation performance and learning efficiency by having the model use fewer epochs to attain reasonable results. The ability of polar transformations to simplify image data and improve training efficiency has also been proposed in [34] where they found introducing the transformation yields significant performance improvement in classification tasks. Similar conclusions were also drawn in [35–37]. From [10] we see that polar transformations encode boundary information between foreground and background classes of images in the middle, this is the ideal area to have machine learning models draw attention to. This serves as the inspiration for designing our Polar-based Contrast Attention module.

Polar transformations have applications outside of semantic segmentation, they can also be integrated with classification models. In [38], they feed feature maps through a 1×1 convolution block to predict the polar origin of the image before transformation can take place. Polar transformation also has application outside of classification and segmentation, it can also help in data augmentation and object detection. In [39], they utilize polar transformation to help a model detect objects regardless of rotational position. And in [40], they augment the size of dataset by performing numerous polar transformations on a single image using different polar origins. Polar transformations can also work together with Cartesian images for training at the same time. In [41], they improve semantic segmentation performance by having a network with two encoders and a single decoder. One encoder works with polar images while the other encoder works with Cartesian images.

Our model makes use of polar transformations to prepare input images because of their ability to improve training efficiency without a significant increase to computational load and no effect on parameter count.

3. Proposed model

Our model has a polar transformation phase prior to training, and because the model trains with polar images, it produces a polar output. Therefore, there is a Cartesian transformation phase after the model output to produce the final prediction mask. In the training phase, we introduce feature shuffling to reduce parameters and computational load, and a polar optimization block to improve improving segmentation performance by having the model focus on specific points of interests as a result of the polar transformation. We provide the full architecture and descriptions in Fig. 1.

3.1. Network architecture

The network is an encoder–decoder style architecture with twelve blocks. Six blocks are convolutional blocks, and the other six blocks are feature shuffle/Polar optimization blocks. There are also skip connections between the encoder branch and decoder branch to recover spatial information that may be lost during down-convolutions. We employ small features counts to reduce parameter count and model size, the channel list for the whole network utilizing 3×3 convolutional blocks is [8,16,24,32,48,64]. We utilize GELU as the activation function. Max-pooling operations follow the convolution blocks on the encoder phase while a bilinear interpolation operation follows the convolution blocks on the decoder phase.

Utilizing a reduced feature count tends to reduce model performance. To get back some lost performance without a substantial increase to model size, we introduce the Polar Contrast Attention (PCA) together with feature shuffling to improve segmentation performance without a significant increase to computing load. The PCA block exploits the nature of polar images by focusing the model on the area of the polar image most likely to contain segmentation information. For polar lesion images, this region is usually in the middle of the image. We take the idea of channel shuffling from [19] and introduce feature shuffling, this is a mechanism to ensure feature information flow across network levels without increasing parameter count or computational load.

3.2. Learning efficient polar contrast attention

3.2.1. Polar transformation

Polar transformation is a geometric transformation that displays an image around a polar origin. The two points in the polar plane are radius and angle and they determine the point in relation to the polar origin of the image. This is in contrast to Cartesian images that use points (x, y) on a Cartesian plane. Therefore, polar transformation takes

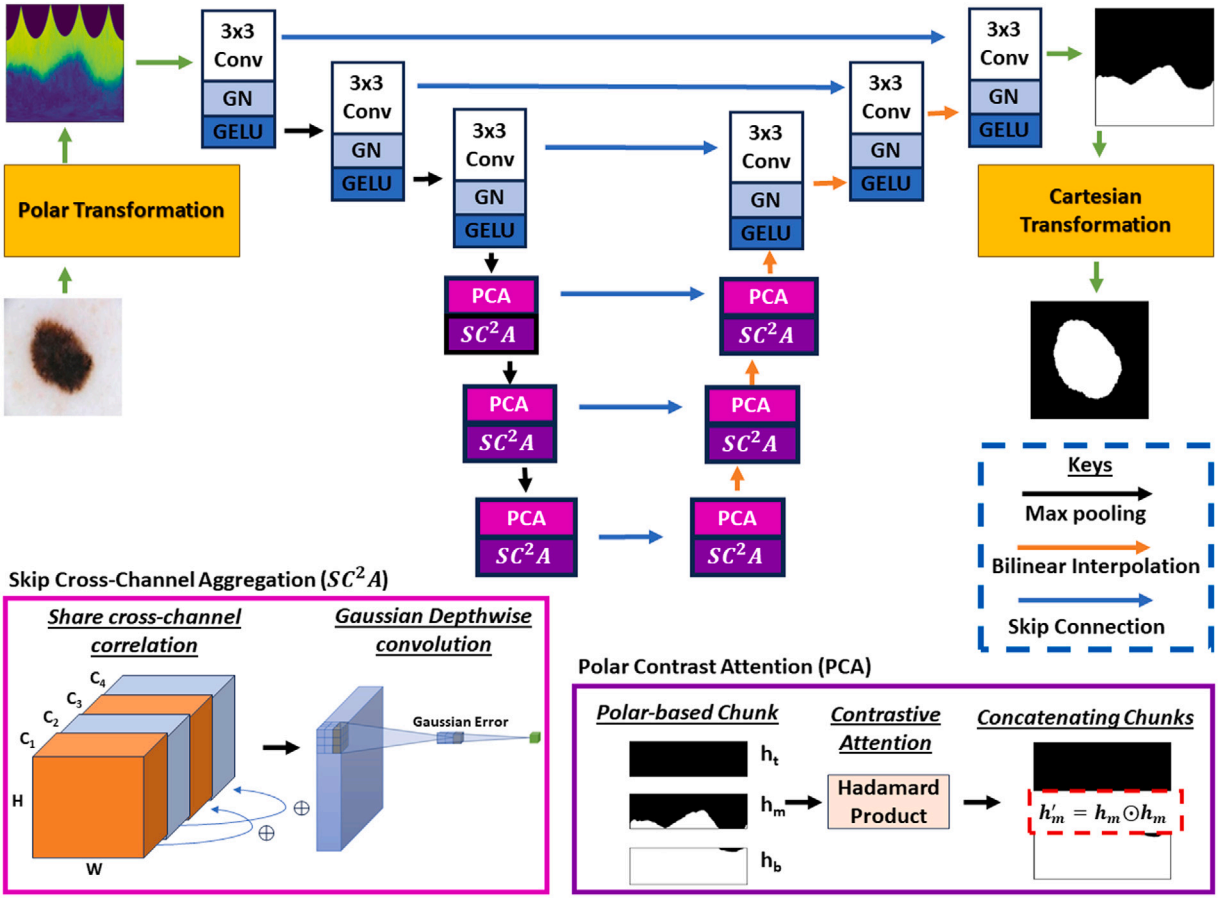

Fig. 1. Overview of our proposed model for skin lesion segmentation. The system performs a polar transformation on the skin lesion image to produce a polar image. This polar image goes through a convolutional layer, group normalization, and GELU activation function. The features are shuffled and model focus is drawn to middle third of the polar image in the polar optimization block. For upconvolutions, the model utilizes bilinear interpolation after convolutional operations to produce a polar mask. The system performs a Cartesian transformation on the polar mask to generate the final prediction.

each point of the Cartesian image and converts them to its equivalent polar radius and angle. After calculating the polar origin, we can calculate the polar coordinates (ρ, θ). We calculate ρ in Eq. (1)

$$\rho(x,y) = \frac{w}{\sqrt{\left(\frac{W}{2.0}\right)^2 + \left(\frac{H}{2.0}\right)^2}} \cdot |(x-\bar{x}, y-\bar{y})| \tag{1}$$

where w is the width of the image. $|(x-\bar{x}, y-\bar{y})|$ is the magnitude of $(x-\bar{x}, y-\bar{y})$. $\bar{x}$ and $\bar{y}$ are the polar origin which is the centroid property of the image. For second polar coordinate θ, we can compute it in Eq. (2),

$$\theta(x,y) = \frac{H}{2\pi} \cdot atan2(x-\bar{x}, y-\bar{y}) \cdot \frac{180}{\pi} \tag{2}$$

where $atan2$ is the arctangent function and H is the height of the image. With the help of Eqs. (1) and (2), the polar transformed image is derived as follows.

$$I_p = f[\rho(x,y), \theta(x,y)] \tag{3}$$

where x and y are the Cartesian coordinators of an image. $f[\cdot,\cdot]$ is the transformation function to convert Cartesian image to polar image and I_p is polar transformed image.

Lastly, we rotate the image 90° anticlockwise. After the polar transformation, the polar image is ready to go through the network for training. The model will, then, produce a polar mask that needs to be converted back to its Cartesian form to produce the final prediction mask. The Cartesian transformation involves rotating the polar mask 90° clockwise and carrying out the polar transformation to retrieve the final Cartesian prediction mask.

3.2.2. Training effective feature extractor

First, a convolution layer with the kernel size of 3 × 3 is performed to learn the inductive biases.

$$F_c = Conv^3(I_p) \tag{4}$$

where $Conv^3$ is a 3 × 3 convolution operation, I_p is the input polar-based image, and F_c is the feature map from convolution layer.

Then, we add a group normalization (GN) which has been shown in [42] to reduce training time, and make it easier for deep networks to converge. Given an image, $i = (i_N, i_C, i_H, i_W)$ is a 4D vector indexing the features in (N, C, H, W). N is the batch axis, C is the channel axis, and H and W are the spatial height and width axes. S_i is the set of pixels for the calculation of mean and std and m is the set size. In a GN layer, the set of S_i is defined as follows.

$$F_{gn} = \sum_{N,C,H,W} S_i = \{k | k_N = i_N, \lfloor \frac{k_C}{C/G} \rfloor = \lfloor \frac{i_C}{C/G} \rfloor\} \tag{5}$$

where F_{gn} is output from GN layer. G is the number of groups, C/G is the number of channels per group. $\lfloor \bullet \rfloor$ is the floor operation so "$\lfloor \frac{k_C}{C/G} \rfloor = \lfloor \frac{i_C}{C/G} \rfloor$" represents that the indexes i and k are in the same group of channels, assuming each group of channels are stored in a sequential order along the C axis.

Finally, a GELU activation is followed by GN layer to form our training effective feature extractor because GELU provides a smooth alternative to ReLU, mitigates problems of dead neurons [43], and performs better in medical segmentation tasks as shown in [44].

$$F_G = G(F_{gn}) = \frac{F_{gn}}{2} + \frac{1}{\sqrt{\pi}} \int_0^{F_{gn}/\sqrt{2}} e^{-t^2} dt \tag{6}$$

where G is GELU activation function and F_G is Gaussian feature map after applying GELU transformation.

3.2.3. Polar-based contrast attention

Here, we propose a novel polar-based contrast attention, where we attempt to find out the region of the image with the most contrast between the background and lesion information. This is inspired by the observation that usually the part with the most contrast between the two is in the middle of the height chunks, which includes the boundary of foreground and background. Specially, a polar-based image is chunked based on the height dimension accordingly into top, middle, and bottom parts.

$$\begin{aligned} S_h(F_G) &= h_t,\ h_m,\ h_b;\quad h'_m = h_m \odot h_m \\ F'_G &= h_t \bigoplus h'_m \bigoplus h_b \end{aligned} \tag{7}$$

where F_G is the input polar-based image and $S_h(\cdot)$ is the height-based splitting function. Due to the fact that the boundary of foreground and background usually shows in the middle part of polar-based image, Hadamard product is applied in the middle part h_m to produce h'_m. Then, we concatenate h_t, h'_m, h_b together to derive the contrast attention map F'_G.

The transformations all have the segmentation information encoded some distance from the middle of the polar image. This is a natural point to have the model focus on during training. This is the motivation of introducing PCA block to help the model pay extra focus to the parts of the image that most likely have lesion boundary information and improve segmentation performance. We perform a layer normalization operation before we split the tensor along the height dimensions into three patches. We identify the patch that is most likely to contain the segmentation information (middle patch) and have the model focus on that patch. We achieve this focus using the Hadamard product on the tensor with itself making it more prominent than other areas of the image. We concatenate the resulting tensor back and feed it to the feature shuffling block to improve feature information flow before convolution.

The feature shuffling involves splitting the features into a different number of patches, we achieve the shuffle by performing a cross addition of the patches to improve information flow. We perform a layer normalization operation on the resulting tensor after it has been concatenated. We, then, pass the tensor through a depthwise convolution block. We present the PyTorch style pseudocode in Algorithm 2

Algorithm 1: Polar-based Contrast Attention (PCA)

Input : Polar-based feature map (x)
Output: Output feature map (x) of PCA

```
// Layer Normalization function
x ← LayerNorm(x);
// Split the batch of tensors along the third
   dimension (height dimension)
h_t, h_m, h_b ← chunk(x, 3, dim = 2);
// Amplify signal at middle patch using the
   Hadamard product
h'_m ← h_m * h_m;
// Concatenate the tensor along third dimension
   (height dimension
x ← cat(h_t, h'_m, h_b, dim = 2);
return x
```

3.3. Skip cross-channel aggregation

3.3.1. Share cross-channel correlations

Depthwise separable convolution (DW) and bilinear interpolation are used to resize high-level features to align with the dimensions of low-level features. Then, to aggregate features cross channels, skip connection is introduced to share features cross channels. The channels of feature map is chunked equally into C_1, C_2, C_3, and C_4. The skip connections are defined as below.

$$\begin{aligned} &S_c(F'_G) = C_1, C_2, C_3, C_4 \\ &F' = C_1 \bigoplus C_2 \bigoplus C_{skip}(C_1, C_3) \bigoplus DW(C_{skip}(C_2, C_4)) \\ &C_{skip}(C_i, C_j) = C_i + C_j,\quad DW(x_{in}) = x_{in} \cdot x_{in} \end{aligned} \tag{8}$$

where $C_{skip(\cdot,\cdot)}$ is the skip connection function to aggregate the feature from different channels. $\bigoplus$ is the feature concatenation and $DW(x_{in})$ is the depthwise separable convolution operation. F' is the new generated feature map. We partly take inspiration from [20] where they implemented skip connections using addition. We also implement improving information flow with addition in this block as we tried implementing this block with multiplication but observed reduced performance.

3.3.2. Gaussian depthwise convolution

In Gaussian depthwise convolution, we factorize the above computation into three steps. The first step applies a 3 × 3 depthwise convolution $\hat{K}$ to each input channel,

$$\hat{O}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m} \tag{9}$$

The second step is to obtain a deterministic decision from a neural network and this gives rise to new non-linearity, the non-linearity is the expected transformation of the stochastic regularizer on an input $\hat{O}_{k,l,m}$ as below. Loosely, this expression states that we scale x by how much greater it is than other inputs. Since the cumulative distribution function of a Gaussian is often computed with the error function, we define the Gaussian Error Linear Unit G as

$$\begin{aligned} G(\hat{O}_{k,l,m}) &= \hat{O}_{k,l,m} P(\hat{O}_{k,l,m} \le \hat{O}_{k,l,m}) \\ &= \Phi(\hat{O}_{k,l,m}) \times I\hat{O}_{k,l,m} + (1 - \Phi(x)) \times 0\hat{O}_{k,l,m} \\ &= \frac{\hat{O}_{k,l,m}}{2} + \frac{1}{\sqrt{\pi}} \int_0^{\hat{O}_{k,l,m}/\sqrt{2}} e^{-t^2} dt \end{aligned} \tag{10}$$

The third step applies 1 × 1 pointwise convolution $\tilde{K}$ to combine the output of depthwise convolution as follows.

$$O_{k,l,n} = \sum_m \tilde{K}_{m,n} \cdot G(\hat{O}_{k-1,l-1,m}) \tag{11}$$

Depthwise convolution and pointwise convolution have different roles in generating new features: the former is used for capturing spatial correlations while the latter is used for capturing channel-wise correlations.

Algorithm 2: Skip Cross-channel Aggregation (SC^2A)

Input : Input feature map (x)
Output: Output Feature map (x) of SC^2A

```
C_1, C_2, C_3, C_4 ← chunk(x, 4, dim = 1);
C_3 ← C_1 + C_3;
C_4 ← C_2 + C_4
C_4 ← Depthwise(C_4);
x ← Concat(C_1, C_2, C_3, C_4, dim = 1);
x ← LayerNorm(x);
x ← Depthwise(x);
x ← GELU(x)
return x
```

3.4. Loss function

Binary cross entropy loss is widely used in semantic segmentation and has been shown to perform very well in medical image segmentation [45]. We incorporate Dice loss as well as a way of ensuring the

model not only learns from pixelwise losses but also general segmentation similarities. The combination of both loss functions also helps in mitigating the problem of class imbalance that may arise if we use either of the loss functions alone. The combined loss function is shown in Eq. (12).

$$L = BCE(p, y) + Dice(p, y) \tag{12}$$

where p and y are the prediction and target respectively.

4. Experiments and results

4.1. Dataset

We choose the Skin Imaging Collaboration ISIC (2018) [46] and PH^2 datasets [47] for training and evaluation of our model. The ISIC dataset consists of 2594 skin lesion images together with their corresponding segmentation masks while the PH^2 dataset consists of 200 images and masks. We train the model on the ISIC dataset utilizing a 75-25 split for training and validation and perform the generalization ability test on the PH^2 dataset.

We also utilize the CVC-Polyp dataset [48] which is a dataset of 612 images extracted from colonoscopy videos together with the polyps annotated for segmentation masks and the Kvasir-SEG dataset [49] which is a dataset of gastrointestinal polyp segmentation. The dataset contains 1000 images of polyps in the human intestinal system and their corresponding segmentation masks annotated by experienced gastroenterologists. We split the dataset into 70-30 for training and validation.

We also perform experiments on the dataset of breast ultrasound imagees [50]. The dataset consists of 780 images and their corresponding ground truth masks. The images are taken from women between the ages of 25 and 75. We split the dataset into 75-25 for training and validation.

We perform the following transformations on the training images: random horizontal and vertical flipping with a probability of 0.5, random rotation, and resize to 256 × 256. We normalize the images with a mean of 157.561 and standard deviation of 26.706.

4.2. Metrics

We utilize the popular metrics intersection over union (IoU) and Dice score (DS) for evaluating the performance of the model. We give the definitions of the metrics below:

$$IoU = \frac{t_p}{t_p + f_n + f_p} \tag{13}$$

$$DS = \frac{2t_p}{2t_p + f_n + f_p} \tag{14}$$

where t_p is true positives, f_n is false negatives, and f_p is false positives. We also utilize frames per second (FPS) and inference time in milliseconds (ms) in evaluating how quickly the model produces a prediction.

4.3. Implementation details

We implement our proposed method using the PyTorch framework. More specially, we use the AdamW optimizer with a learning rate of 0.001, and the scheduler for the experiments is the CosineAnnealingLR. All our experiments are implemented by a PC with a GeForce RTX 3070 Mobile graphics card with 8 GB video memory and 16 GB system memory. The lightweight nature of our model means it is also viable to test it in a mobile environment where processing powers are much lower than regular PCs. We optimize the model for mobile execution using PyTorch Lite for android operating system and conduct experiments to measure the inference time of the model. The specifications for the mobile device for this experiment are a Snapdragon 720G System on a Chip (SoC), which is an 8-core processor with Adreno 618 graphics processor, 6 GB system RAM, and android version 12. The model of the device is the Xiaomi Redmi Note 9 pro smartphone.

Table 1
Confidence interval based analysis of the proposed approach.

Dataset	Min (%)	Max (%)	σ	CI
ISIC 2018	92.43	93.72	0.65	93.08 ± 0.13 (0.14%)
Kvasir-SEG	89.17	90.35	0.59	89.76 ± 0.12 (0.13%)
BUSI	81.69	82.13	0.22	81.91 ± 0.05 (0.06%)

Table 2
Quantitative comparison results on the ISIC 2018 dataset.

Method	IoU	DS	FPS	#Param	GFLOPS
Proposed	0.8815	0.9372	95.33	0.026	0.055
EGEUNET [18]	0.7745	0.8730	46.52	0.053	0.081
MADGNet [51]	0.8387	0.9031	9	33.37	11.49
MobileNetV3 [52]	0.5879	0.7258	30	2.9	34.75
Polar U-Net [10]	0.8681	0.9073	42.09	26.08	18.425
MobileNetV4 [53]	0.5230	0.6868	50	19.09	0.38
UNeXT [17]	0.7840	0.8730	4.32	1.47	0.570
RESUNETPP [54]	0.7980	0.8726	42.09	26.08	18.425
DCSAU-Net [22]	0.7741	0.8571	17.62	2.60	6.724
Xbound [23]	0.8447	0.9091	22.88	30.64	6.538
Deeplab [55]	0.8031	0.8758	68.83	22.44	7.903
U-Net [20]	0.7285	0.8135	60.45	7.76	12.103
STDC [56]	0.8549	0.9101	41.02	8.27	33.771
PIDnet [57]	0.7411	0.8493	35.03	7.72	23.723

4.4. Performance comparison

In this section, we compare our model against other segmentation models utilizing different architectures. We compare against lightweight models, transformer inspired models, traditional encoder–decoder inspired networks, and instance segmentation models.

4.4.1. Results on ISIC 2018 dataset

We present the result of our quantitative analysis on ISIC 2018 dataset in Table 2 where we see our model achieve the best Dice score and IoU score. This means our model performs very well in terms of pixelwise predictions as shown by the high Dice score of 0.9372, and it also performs very well in terms of general segmentation performance as shown by the high IoU score of 0.8815. Our model is also the fastest as is evident from the lowest inference time of just 10.49 ms and highest frames per second (FPS) of 95.33. Our model achieves this performance while having the lowest parameter count and lowest number of operations. We also present a qualitative comparison of the models in Fig. 2 and a confidence based analysis in Table 1. We measure the statistical significance of the result using a t-test. We conclude that our model performance improvement is statistically significant because of a combination of t-statistic value of 6.6863 and a p-value of $1.172e^{-4}$.

4.4.2. Results on Kvasir-SEG dataset

In addition, we also compare our method against other advanced segmentation models on Kvasir-SEG dataset [49]. Our proposed method achieves the best performance in terms of Dice Score with a value of 0.9035. Table 3 presents the findings of the experiments. This represents a 12.08% improvement in terms of Dice score and a 21.64% improvement in terms of IoU score in comparison with the next best performing lightweight model tested [17] which has 10x more operations and over 50x trainable parameters. We provide the qualitative comparisons of the methods in Fig. 3

4.4.3. Results on breast ultrasound dataset

To further evaluate the versatility of our approach, we also evaluate our model using the breast ultrasound images dataset [50]. Our proposed method achieves the best performance in terms of Dice Score with a value of 0.8213 as shown in Table 4. This is almost double the Dice Score of the next best performing lightweight model (see Fig. 4).

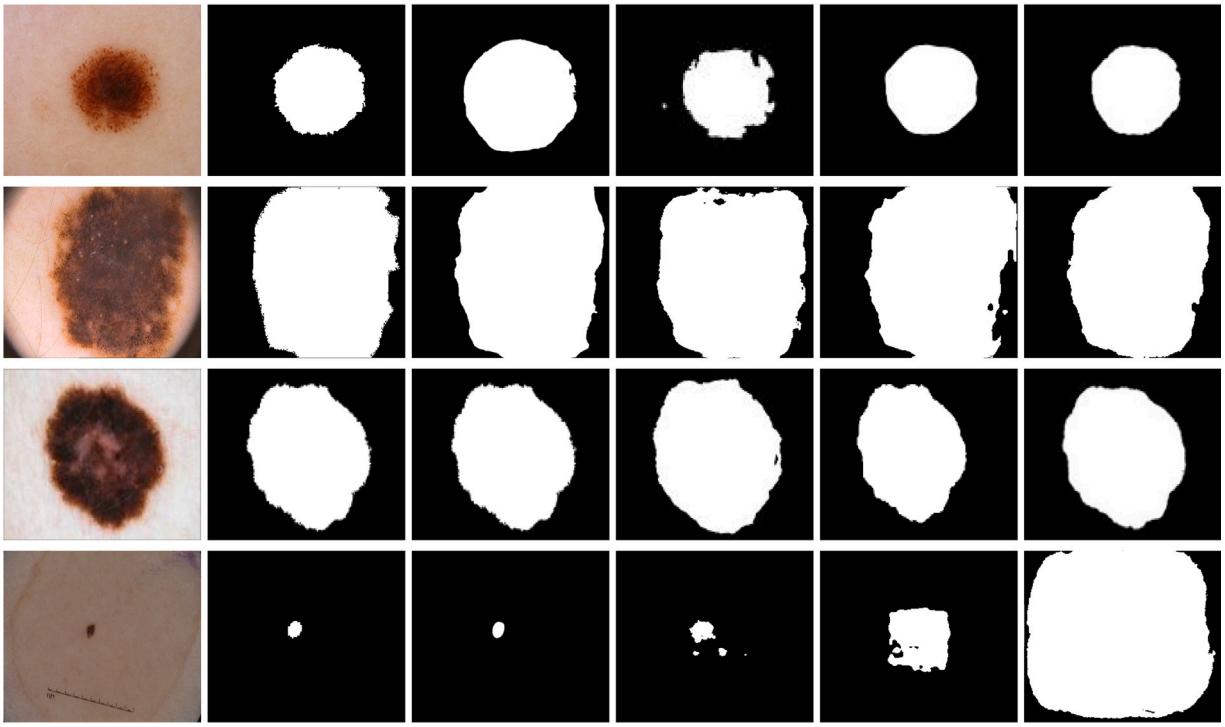

Fig. 2. Qualitative analysis and comparison. The first column is raw lesion image. Second column is the ground truth. The third column is mask generated by our proposed method. The fourth, fifth, and sixth columns are masks generated by PIDnet, DCSAU-Net, and Mask2Former.

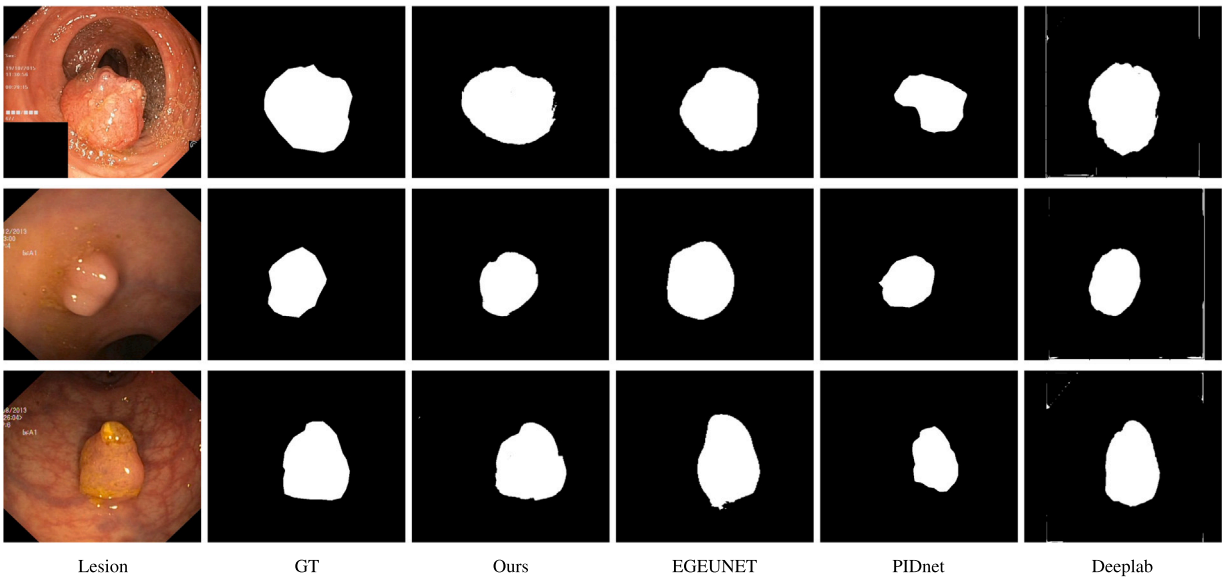

Fig. 3. Qualitative analysis and comparison. The first column is raw lesion image. Second column is the ground truth. The third column is mask generated by our proposed method. The fourth, fifth, and sixth columns are masks generated by EGEUNET, PIDnet, and Deeplab.

4.5. Generalization ability

The generalization ability of artificial intelligence models is very important especially when the models are for medical image processing and have the ability to run on point of care devices and smartphones. It is not plausible to capture every possible variant of segmentation training data in the training dataset. This is the motivation for performing experiments to determine the generalization ability of our proposed model.

4.5.1. Train on ISIC 2018 generalize on PH^2 dataset

We perform testing on the PH^2 dataset to see how well how model can generalize on a different dataset. We train the model on the ISIC 2018 dataset and test the model on all the images of the PH^2 dataset. We present the findings of this experiment in Table 5, where we see from the results that our model performs very well on unseen data as seen by the high Dice score of 0.9423 and IoU score of 0.8909. The next best performing model is Valanarasu and Patel [17] with a Dice score of 0.9189 and IoU of 0.8504. Our model is reports a 2.55% improvement in terms of Dice score over this method and a 4.76% improvement

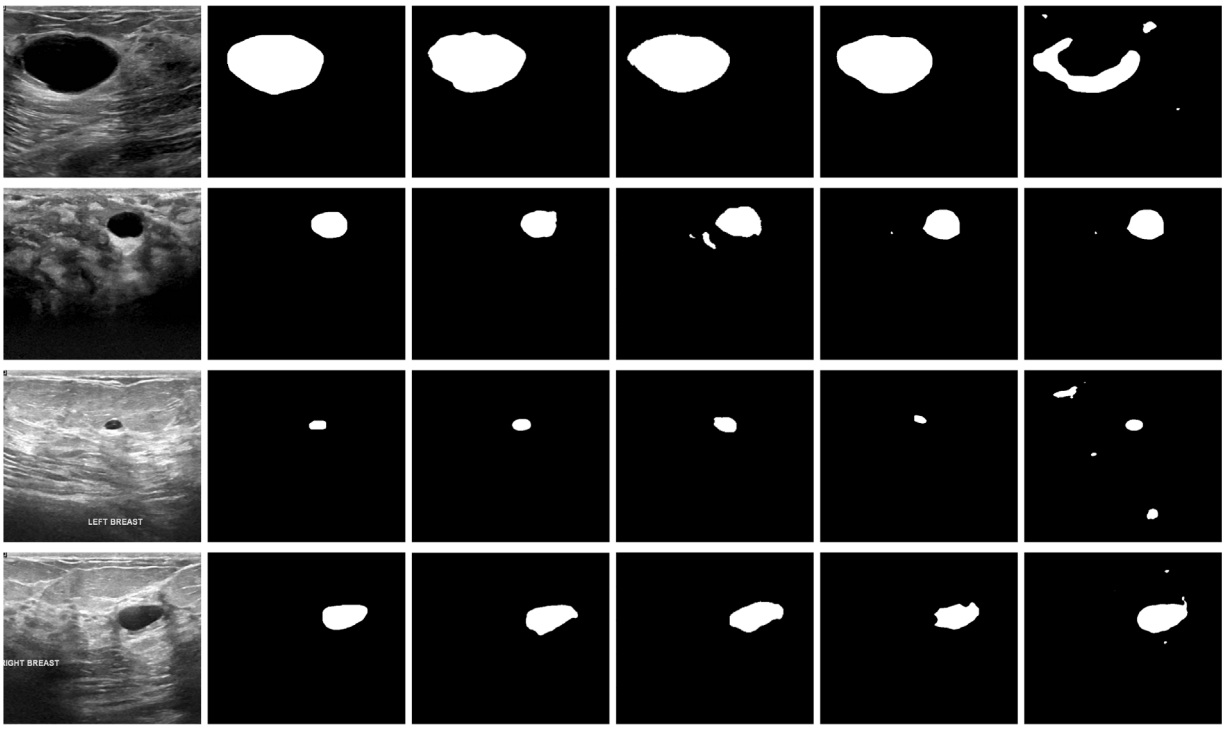

Fig. 4. Qualitative analysis and comparison. The first column is raw lesion image. Second column is the ground truth. The third column is mask generated by our proposed method. The fourth, fifth, and sixth columns are masks generated by EGEUNET, PIDnet, and PoolFormer.

Table 3
Quantitative comparison results on Kvasir-SEG dataset.

Method	IoU	DS	FPS	#Param	GFLOPS
Proposed	0.8240	0.9035	95.33	0.026	0.055
UNeXT [17]	0.6774	0.8061	4.32	1.42	0.570
PIDnet [57]	0.7154	0.8205	35.03	7.72	23.723
MobileNetV4 [53]	0.2304	0.3745	50	19.09	0.38
MADGNet [51]	0.8357	0.9028	9	33.37	11.49
SAN [58]	0.4905	0.6525	11	8.4	64.3
EGEUNET [18]	0.5217	0.6904	46.52	0.053	0.081
DDRNET [59]	0.4152	0.4610	43.7	5.73	18.24
SegNeXt [60]	0.5364	0.6343	23	4.23	25.18
RESUNETPP [54]	0.5847	0.6781	42.09	26.08	18.425
Deeplab [55]	0.7527	0.8348	68.83	22.44	7.903
U-Net [20]	0.5208	0.6435	60.45	7.76	12.103

Table 4
Quantitative comparison results on breast ultrasound dataset [50].

Method	IoU	DS	FPS	#Param	GFLOPS
Proposed	0.6968	0.8213	95.33	0.026	0.055
PIDNET [57]	0.6244	0.7261	35.03	7.72	23.723
UNeXT [17]	0.2226	0.3396	4.32	1.42	0.570
EGE-UNET [18]	0.2971	0.4582	46.52	0.053	0.081
MADGNet [51]	0.7340	0.8130	9	33.37	11.49
PoolFormer [61]	0.6400	0.7417	28	15.65	48.02
DEEPLAB [55]	0.4005	0.4753	68.83	22.44	7.903
RESUNETPP [54]	0.1822	0.2395	42.09	26.08	18.425
U-NET [20]	0.2258	0.2994	60.45	7.76	12.103
STDC [56]	0.4497	0.4741	41.02	8.27	33.771
DDRNET [59]	0.3917	0.5202	43.7	5.73	18.24

in terms of IoU score. Our model achieves this improvement while using over 50x fewer parameters than [17] and 10x fewer operations. The other lightweight model [18] has 2x more parameters that our proposed model and is able to report a Dice score of 0.8905 and IoU score of 0.8026. This represents a 5.82% improvement in terms of Dice score and an 11% improvement in terms of IoU score. This proves our model outperforms the two SOTA lightweight models in the task of generalization ability using the PH^2 dataset.

Table 5
Generalization ability on PH^2 dataset.

Method	IoU	DS	FPS	#Param	GFLOPS
Proposed	0.8909	0.9423	95.33	0.026	0.055
UNeXT [17]	0.8504	0.9189	4.32	1.47	0.570
RESUNETPP [54]	0.7711	0.8560	42.09	26.08	18.425
Deeplab [55]	0.8042	0.8863	68.83	22.44	7.903
U-Net [20]	0.6925	0.7977	60.45	7.76	12.103
EGEUNET [18]	0.8026	0.8905	46.52	0.053	0.081
DCSAU-Net [22]	0.8255	0.9002	17.62	2.60	6.724
PIDNet [57]	0.6152	0.7585	35.03	7.71	23.723
STDC [56]	0.7345	0.8459	41.02	8.27	33.771
Xbound [23]	0.5925	0.6963	22.88	30.64	6.538

4.5.2. *Train on Kvasir-SEG generalize on CVC-Polyp dataset*

In order to carry out the generalization test on the polyp segmentation task, we utilize the CVC-Polyp dataset [48]. For the generalization ability test, we take the model trained on the Kvasir-SEG dataset and test on the images of the CVC-Polyp dataset. We present the findings of the test in Table 6. We see our model is able to generalize the best among the models tested with a Dice score of 0.8437 and IoU of 0.7297. The next best performing model in terms of generalization ability on the polyp datasets is the deeplab [55] model with a Dice score of 0.6406 and an IoU score of 0.5432. This is a 31.70% improvement in terms of Dice score and a 34.33% improvement in terms of IoU score all utilizing much fewer parameters and operations.

4.6. *Mobile performance*

Semantic segmentation models can have immense applications in the field of medical image processing, but these models can suffer from some problems when they pursue accuracy without much regard for

Table 6
Generalization ability on CVC-Polyp dataset.

Method	IoU	DS	FPS	#Param	GFLOPS
Proposed	0.7297	0.8437	95.33	0.026	0.055
UNeXT [17]	0.2837	0.4399	4.32	1.47	0.570
PIDnet [57]	0.5229	0.6328	35.03	7.72	23.723
SAN [58]	0.2405	0.3445	11	8.4	64.3
EGEUNET [18]	0.3005	0.4621	46.52	0.053	0.081
MADGNet [51]	0.6970	0.7750	9	33.37	11.49
DDRNET [59]	0.4367	0.4686	43.7	5.73	18.24
SegNeXt [60]	0.0363	0.0682	23	4.23	25.18
RESUNETPP [54]	0.3494	0.4527	42.09	26.08	18.425
Deeplab [55]	0.5432	0.6406	68.83	22.44	7.903
U-Net [20]	0.4168	0.5221	60.45	7.76	12.103

Table 7
Performance benchmark on mobile device (Xiaomi Redmi Note 9 pro), the unit of inference time is millisecond.

Method	Inference time	FPS	#Param (M)	GFLOPS
Proposed	55	18.182	0.026	0.055
EGEUNET [18]	112	8.929	0.053	0.081
UNeXT [17]	132	7.576	1.47	0.570
RESUNETPP [54]	910	1.099	26.08	18.425
DCSAU-Net [22]	1762	0.578	2.60	6.724
Xbound [23]	1091	0.917	30.64	6.538
Deeplab [55]	524	1.908	22.44	7.903
U-Net [20]	693	1.443	7.76	12.103

model complexity and computational requirements. They may suffer with long inference time, high latency, and increased energy consumption. This is where low parameter and low computational requirement models have an added advantage. If model can be engineered to be lightweight enough, it can run locally on the user's device. This immediately solves the problem of high latency, and it allows the model to produce output with low inference times. It also helps in preserving the users privacy by not sending any information to the cloud for processing as all processing happens locally. It is in view of all these reasons that we test the model performance on mobile devices.

To test the performance of the system in a low powered device, we developed an android application and conduct experiments measuring inference time on the device. Inference time is measured as the time taken in milliseconds after the input is passed to the model and a prediction is produced. Table 7 shows the results of the experiments with a lower inference time being better. Our model achieves the lowest inference time with a reported time of just 55 ms and by consequence the highest FPS of 18.182. The next performing model is more than 2x slower in producing a prediction with an inference time of 112 ms while have twice as many parameters and 1.5x more operations than our proposed model.

The experiments show our model to be the best performing as evidenced by the high Dice score and lowest inference time. The next best performing model has 1000x more parameters and over 100x operations and took 19x longer in producing a prediction on the mobile device.

4.7. Ablation study

We follow the methods in [26] to conduct comprehensive ablation experiments to determine what every component of the model contributes to performance in terms of Dice score and IoU score. Table 10 shows the results of the experiments with the baseline serving as an encoder–decoder model from Ruan et al. [62]. The introduction of skip cross-channel aggregation (SC2 A) improves information flow which enables capturing of higher-level spatial information and incorporating them with lower-level semantic information. This results in an overall improved semantic segmentation as shown by the increase of Dice score from 0.8656 to 0.8808 which represents a 1.76% improvement, and

Table 8
Performance metrics for different values of PCA and SC2A blocks on ISIC 2018 dataset.

	k = 1	k = 2	k = 3	k = 4
IoU	0.8821	0.8833	0.8834	0.8834
Dice	0.9382	0.9391	0.9393	0.9393

Table 9
Performance metrics for different values of PCA and SC2A blocks on Kvasir-SEG dataset.

	k = 1	k = 2	k = 3	k = 4
IoU	0.8240	0.7907	0.7502	0.7395
Dice	0.9085	0.8831	0.8573	0.8503

Table 10
Ablation study on different Component of our proposed method on the ISIC 2018 Dataset, where SC^2A: Skip Cross-Channel Aggregation, Polar-Transform: Polar Transformation, PCA: Polar-based Contrast Attention.

Baseline	SC^2A	Polar-Transform	PCA	IoU	Dice score
✓				0.7630	0.8656
✓	✓			0.7870	0.8808
✓	✓	✓		0.8609	0.9253
✓	✓	✓	✓	0.8815	0.9372

an increase of IoU from 0.7630 to 0.7870 which represents a 3.15% improvement. We introduce polar transformation as a way reducing image complexities allowing for a model with low number parameters to efficiently learn the necessary segmentation information, this results in an increase of Dice score to 0.9253 which is a 5.05% improvement, and an increase of IoU to 0.8609 which is a 9.39% improvement. This also serves as a pre-processing step for the polar contrast attention block which helps the model focus on the region most likely to contain relevant segmentation information by amplifying the signal at the point. The introduction of the polar contrast attention block results in an increase in Dice score of 0.9371 which is a 1.29% improvement, and an increase in IoU to 0.8815 representing a 2.39% improvement. To further investigate the impact of the polar PCA and SC2 A blocks, we added up to four blocks to the network. We report the findings in Table 8 for the ISIC 2018 dataset and Table 9 for the Kvasir-SEG dataset with k referring to the number of the blocks in the network. We also test the impact of the Gaussian kernel by replacing it with a ReLU kernel, we record a segmentation performance loss of 0.7%.

Taking all the components into account translates into an overall improvement of 8.27% in Dice score and 15.53% in IoU score. We also explored the training efficiency of the approach, we set the epoch to 10 with a batch size of 4 and we got a Dice score of 0.9246. To achieve the best result on the IoU of 0.8815 and Dice score of 0.9372, the number of epoch is only 20 with batch size of 4 as well.

5. A case study on skin care Internet of Medical Things (IoMT)

Although the performance of lesion segmentation models has been steadily improving over time, they usually improve performance with increasing computational requirements and parameter count. This leads to a much more complex model with high inference time making them inappropriate for point of care or mobile applications. This leaves a gap in real world applications because some point of care devices are not powerful devices able to run inference on complex models in a timely manner. This observation is the motivation for designing a low computational requirement and low parameter count model. Lightweight models have the potential to save costs on expensive hardware and reduce the load on physicians and patients.

To benchmark the performance of the model in a low powered device, we develop an android application to run inference on and record the time taken. The application presents an interface shown in

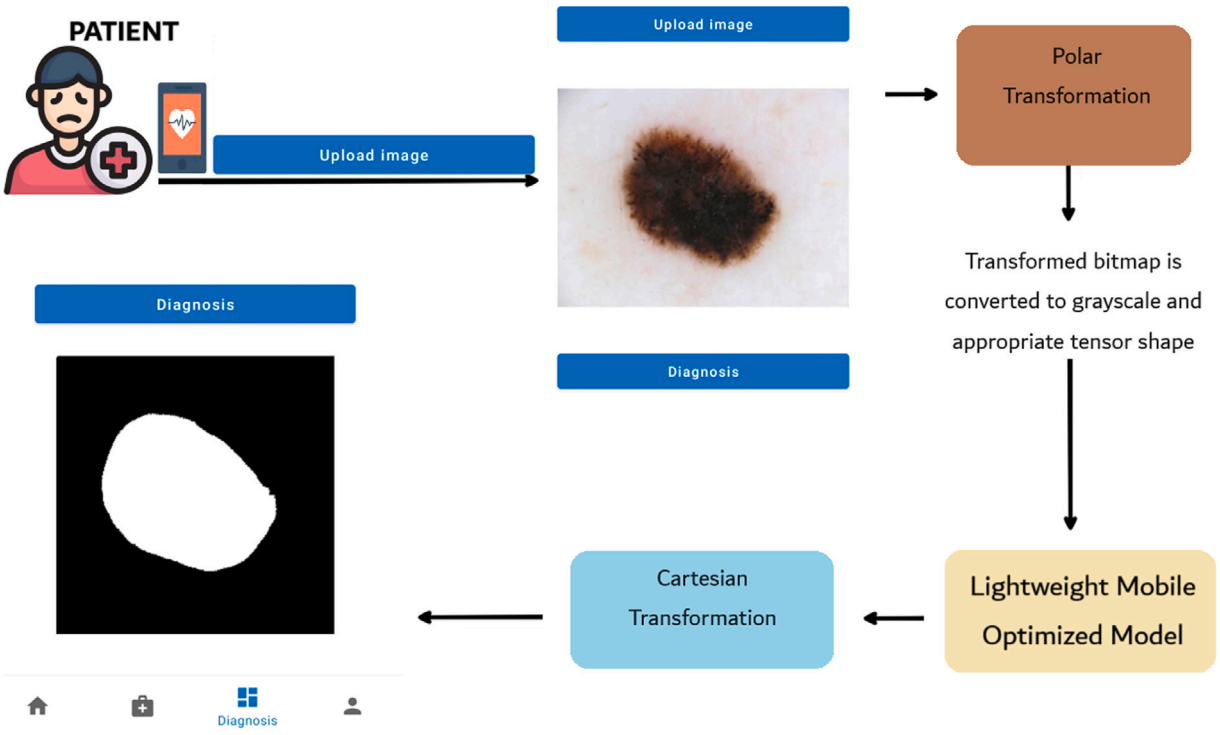

Fig. 5. Overall architecture and data flows of mobile App on edge unit.

Fig. 5 allowing the patient to select an image from the phone gallery . The application then converts the image into its polar form and resizes and normalizes it accordingly. The polar form gets converted to a PyTorch tensor and goes through the model. The model produces the polar mask which is then converted back to a cartesian image and resized appropriately for display to the user.

6. Discussion and conclusion

This paper proposes a novel lightweight lesion segmentation model with low computational requirements by leveraging polar transformation in both training and prediction phases. These transformations simplify the complexity of learning lesion segmentation tasks, enabling the model to perform exceptionally well with minimal parameters and operations. The model trains on polar images and produces polar masks that are then converted to Cartesian masks. Moreover, it incorporates skip cross-channel aggregation to enhance information flow without increasing the number of parameters. Furthermore, a polar-based contrast attention block is designed to concentrate on the boundary information of the mask areas within the polar image. Experimental results demonstrate that these enhancements improve segmentation performance while maintaining a parameter count of approximately 25k making the lightest segmentation model with a model weight file size of just 130KB and the fastest lesion semantic segmentation model. This means the model can automatically segment 10k images in about 100 s on consumer level hardware freeing the medical practitioner's valuable time. The lightweight model can also be utilized as a preprocessor to improve classification tasks and ease memory requirements [63]. The attention mechanism proposed here only work on polar images limiting their functionality to models that incorporate polar transformation, and polar transformation when applied in semantic segmentation models work best in spherical lesions like skin and polyps.

We have demonstrated the ability of the proposed lightweight polar attention blocks to improve segmentation performance in medical images. These blocks are very fast because they use quick operations, but a limitation of these blocks is they are designed to work on polar images exclusively. This means they cannot be dropped into models that are not designed to work with polar images. Another limitation of this approach is because this model extracts features by encoding images around a polar origin, this means when presented with images with many polar origins as in the case with many masks in one image, pixel mislabelling can occur. This opens an avenue for future work, to improve the performance of the model in the face of challenging images.

The model is trained on the ISIC 2018 and Kvasir datasets and evaluated on both the ISIC 2018, PH^2, Kvasir, and CVC-Polyp datasets to assess its performance and generalization ability. The model performs well on the ISIC 2018 dataset and demonstrates robust generalization on the PH^2 dataset, which achieve the highest Dice score and IoU score on both datasets. We also conduct experiments in IoMT based case study in a mobile environment to assess potential benefits of a lightweight model in a point-of-care setting, where our model can achieve the best performance in both segmentation accuracy and inference time compared to other advanced models.

CRediT authorship contribution statement

Mohammed Lawal: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Dewei Yi:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Dewei Yi reports financial support was provided by Cancer Research UK. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Cancer Research UK (CRUK) under Grant EDDPJT-May23/100001 and was also supported in part by the Petroleum Technology Development Fund (PTDF) Nigeria.

References

[1] B. Hu, P. Zhou, H. Yu, Y. Dai, M. Wang, S. Tan, Y. Sun, LeaNet: Lightweight U-shaped architecture for high-performance skin cancer image segmentation, Comput. Biol. Med. 169 (2024) 107919.
[2] K. Urban, S. Mehrmal, P. Uppal, R.L. Giesey, G.R. Delost, The global burden of skin cancer: A longitudinal analysis from the Global Burden of Disease Study, 1990–2017, JAAD Int. 2 (2021) 98–108.
[3] S. Nivedha, S. Shankar, Melanoma diagnosis using Enhanced Faster Region convolutional neural networks optimized by artificial gorilla troops algorithm, Inf. Technol. Control 52 (4) (2023) 819–832.
[4] R. Venugopal, N. Shafqat, I. Venugopal, B.M.J. Tillbury, H.D. Stafford, A. Bourazeri, Privacy preserving generative adversarial networks to model electronic health records, Neural Netw. 153 (2022) 339–348.
[5] D. Yi, Y. Hua, P. Murchie, P.K. Sharma, Label-free medical image quality evaluation by semantics-aware contrastive learning in IoMT, IEEE J. Biomed. Health Inform. (2023).
[6] F. Garcea, A. Serra, F. Lamberti, L. Morra, Data augmentation for medical imaging: A systematic literature review, Comput. Biol. Med. (2022) 106391.
[7] M.A. Khan, S. Kwon, J. Choo, S.M. Hong, S.H. Kang, I.-H. Park, S.K. Kim, S.J. Hong, Automatic detection of tympanic membrane and middle ear infection from oto-endoscopic images via convolutional neural networks, Neural Netw. 126 (2020) 384–394.
[8] M. Hussain, M.A. Khan, R. Damaševičius, A. Alasiry, M. Marzougui, M. Alhaisoni, A. Masood, SkinNet-INIO: multiclass skin lesion localization and classification using fusion-assisted deep neural networks and improved nature-inspired optimization algorithm, Diagnostics 13 (18) (2023) 2869.
[9] Q. Wu, T.M. McGinnity, L. Maguire, A. Belatreche, B. Glackin, 2D co-ordinate transformation based on a spike timing-dependent plasticity learning mechanism, Neural Netw. 21 (9) (2008) 1318–1327.
[10] M. Benčević, I. Galić, M. Habijan, D. Babin, Training on polar image transformations improves biomedical image segmentation, IEEE Access 9 (2021) 133365–133375.
[11] V. Kaul, S. Enslin, S.A. Gross, History of artificial intelligence in medicine, Gastrointest. Endosc. 92 (4) (2020) 807–812.
[12] G. Ren, Monkeypox disease detection with pretrained deep learning models, Inf. Technol. Control 52 (2) (2023) 288–296.
[13] Y. Wu, Y. Wang, L. Wang, P. Yin, Y. Lin, M. Zhou, Burden of melanoma in China, 1990–2017: Findings from the 2017 global burden of disease study, Int. J. Cancer 147 (3) (2020) 692–701.
[14] D. Yi, P. Baltov, Y. Hua, S. Philip, P.K. Sharma, Compound scaling encoder-decoder (cosed) network for diabetic retinopathy related bio-marker detection, IEEE J. Biomed. Health Inform. (2023).
[15] E. Tensen, J. Van Der Heijden, M. Jaspers, L. Witkamp, Two decades of teledermatology: current status and integration in national healthcare systems, Curr. Dermatol. Rep. 5 (2016) 96–104.
[16] N. Awasthi, L. Vermeer, L.S. Fixsen, R.G. Lopata, J.P. Pluim, LVNet: Lightweight model for left ventricle segmentation for short axis views in echocardiographic imaging, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 69 (6) (2022) 2115–2128.
[17] J.M.J. Valanarasu, V.M. Patel, Unext: Mlp-based rapid medical image segmentation network, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2022, pp. 23–33.
[18] J. Ruan, M. Xie, J. Gao, T. Liu, Y. Fu, EGE-UNet: an efficient group enhanced UNet for skin lesion segmentation, 2023, arXiv preprint arXiv:2307.08473.
[19] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.
[20] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.
[21] N.K. Tomar, D. Jha, M.A. Riegler, H.D. Johansen, D. Johansen, J. Rittscher, P. Halvorsen, S. Ali, Fanet: A feedback attention network for improved biomedical image segmentation, IEEE Trans. Neural Netw. Learn. Syst. (2022).
[22] Q. Xu, Z. Ma, H. Na, W. Duan, DCSAU-Net: A deeper and more compact split-attention U-Net for medical image segmentation, Comput. Biol. Med. 154 (2023) 106626.
[23] J. Wang, J. Zhang, Z. Wu, G. Qi, Y. Tian, P.-A. Heng, J. Zhu, XBound-Former: Toward cross-scale boundary modeling in transformers, IEEE Trans. Med. Imaging 42 (6) (2023) 1735–1745.
[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[25] M. Jian, C. Tao, R. Wu, H. Zhang, X. Li, R. Wang, Y. Wang, L. Peng, J. Zhu, HRU-Net: A high-resolution convolutional neural network for esophageal cancer radiotherapy target segmentation, Comput. Methods Programs Biomed. 250 (2024) 108177.
[26] M. Jian, H. Chen, C. Tao, X. Li, G. Wang, Triple-DRNet: A triple-cascade convolution neural network for diabetic retinopathy grading using fundus images, Comput. Biol. Med. 155 (2023) 106631.
[27] Q. Huang, L. Zhao, G. Ren, X. Wang, C. Liu, W. Wang, NAG-Net: Nested attention-guided learning for segmentation of carotid lumen-intima interface and media-adventitia interface, Comput. Biol. Med. 156 (2023) 106718.
[28] S. Maqsood, R. Damaševičius, Multiclass skin lesion localization and classification using deep learning based features fusion and selection framework for smart healthcare, Neural Netw. 160 (2023) 238–258.
[29] S. Bibi, M.A. Khan, J.H. Shah, R. Damaševičius, A. Alasiry, M. Marzougui, M. Alhaisoni, A. Masood, MSRNet: multiclass skin lesion recognition using additional residual block based fine-tuned deep models information fusion and best feature selection, Diagnostics 13 (19) (2023) 3063.
[30] Z. Lu, C. She, W. Wang, Q. Huang, LM-Net: A light-weight and multi-scale network for medical image segmentation, Comput. Biol. Med. 168 (2024) 107717.
[31] Y. Yin, Z. Han, M. Jian, G.-G. Wang, L. Chen, R. Wang, AMSUnet: A neural network using atrous multi-scale convolution for medical image segmentation, Comput. Biol. Med. 162 (2023) 107120.
[32] Z. Han, M. Jian, G.-G. Wang, ConvUNeXt: An efficient convolution neural network for medical image segmentation, Knowl.-Based Syst. 253 (2022) 109512.
[33] B.-S. Kim, J.-Y. Sun, S.-W. Kim, M.-C. Kang, S.-J. Ko, CNN-based UGS method using cartesian-to-polar coordinate transformation, Electron. Lett. 54 (23) (2018) 1321–1322.
[34] Q. Paletta, A. Hu, G. Arbod, P. Blanc, J. Lasenby, SPIN: Simplifying polar invariance for neural networks application to vision-based irradiance forecasting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5182–5191.
[35] P. Ghasemzadeh, S. Banerjee, M. Hempel, H. Sharif, A novel deep learning and polar transformation framework for an adaptive automatic modulation classification, IEEE Trans. Veh. Technol. 69 (11) (2020) 13243–13258.
[36] J. Chen, Z. Luo, Z. Zhang, F. Huang, Z. Ye, T. Takiguchi, E.R. Hancock, Polar transformation on image features for orientation-invariant representations, IEEE Trans. Multimed. 21 (2) (2018) 300–313.
[37] D. Bhattacharjee, Adaptive polar transform and fusion for human face image processing and evaluation, Hum.-Cent. Comput. Inf. Sci. 4 (2014) 1–18.
[38] C. Esteves, C. Allen-Blanchette, X. Zhou, K. Daniilidis, Polar transformer networks, 2017, arXiv preprint arXiv:1709.01889.
[39] J. Kim, W. Jung, H. Kim, J. Lee, CyCNN: A rotation invariant CNN using polar mapping and cylindrical convolution layers, 2020, arXiv preprint arXiv:2007.10588.
[40] H. Salehinejad, S. Valaee, T. Dowdell, J. Barfett, Image augmentation using radial transform for training deep neural networks, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2018, pp. 3016–3020.
[41] Q. Liu, X. Hong, W. Ke, Z. Chen, B. Zou, DDNet: cartesian-polar dual-domain network for the joint optic disc and cup segmentation, 2019, arXiv preprint arXiv:1904.08773.
[42] Y. Wu, K. He, Group normalization, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 3–19.
[43] M. Lee, et al., Mathematical analysis and performance evaluation of the gelu activation function in deep learning, J. Math. Univ. Tokushima 2023 (2023).
[44] Y. Peng, X. Yi, D. Zhang, L. Zhang, Y. Tian, Z. Zhou, ConvMedSegNet: A multi-receptive field depthwise convolutional neural network for medical image segmentation, Comput. Biol. Med. 176 (2024) 108559.
[45] S. Jadon, A survey of loss functions for semantic segmentation, in: 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB, IEEE, 2020, pp. 1–7.
[46] B. Cassidy, C. Kendrick, A. Brodzicki, J. Jaworek-Korjakowska, M.H. Yap, Analysis of the ISIC image datasets: Usage, benchmarks and recommendations, Med. Image Anal. 75 (2022) 102305.
[47] T. Mendonça, P.M. Ferreira, J.S. Marques, A.R. Marcal, J. Rozeira, PH 2-A dermoscopic image database for research and benchmarking, in: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, IEEE, 2013, pp. 5437–5440.
[48] J. Bernal, F.J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, F. Vilariño, WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians, Comput. Med. Imaging Graph. 43 (2015) 99–111.
[49] D. Jha, P.H. Smedsrud, M.A. Riegler, P. Halvorsen, T. de Lange, D. Johansen, H.D. Johansen, Kvasir-seg: A segmented polyp dataset, in: International Conference on Multimedia Modeling, Springer, 2020, pp. 451–462.
[50] W. Al-Dhabyani, M. Gomaa, H. Khaled, A. Fahmy, Dataset of breast ultrasound images, Data Brief 28 (2020) 104863.
[51] J.-H. Nam, N.S. Syazwany, S.J. Kim, S.-C. Lee, Modality-agnostic domain generalizable medical image segmentation by multi-frequency in multi-scale attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 11480–11491.

[52] B. Koonce, B. Koonce, MobileNetV3, in: Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, Springer, 2021, pp. 125–144.
[53] D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, F. Yang, W. Wang, C. Banbury, C. Ye, B. Akin, et al., MobileNetV4-universal models for the mobile ecosystem, 2024, arXiv preprint arXiv:2404.10518.
[54] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested u-net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4, Springer, 2018, pp. 3–11.
[55] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 801–818.
[56] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, X. Wei, Rethinking bisenet for real-time semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9716–9725.
[57] J. Xu, Z. Xiong, S.P. Bhattacharyya, PIDNet: A real-time semantic segmentation network inspired by PID controllers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19529–19539.
[58] M. Xu, Z. Zhang, F. Wei, H. Hu, X. Bai, Side adapter network for open-vocabulary semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2945–2954.
[59] Y. Hong, H. Pan, W. Sun, Y. Jia, Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes, 2021, arXiv preprint arXiv:2101.06085.
[60] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, S.-M. Hu, Segnext: Rethinking convolutional attention design for semantic segmentation, Adv. Neural Inf. Process. Syst. 35 (2022) 1140–1156.
[61] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, S. Yan, Metaformer is actually what you need for vision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10819–10829.
[62] J. Ruan, S. Xiang, M. Xie, T. Liu, Y. Fu, MALUNet: A multi-attention and light-weight unet for skin lesion segmentation, in: 2022 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, IEEE, 2022, pp. 1150–1156.
[63] S. Takahama, Y. Kurose, Y. Mukuta, H. Abe, M. Fukayama, A. Yoshizawa, M. Kitagawa, T. Harada, Multi-stage pathological image classification using semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 10702–10711.