Rolling Convolution Filters for Lightweight Neural Networks in Medical Image Analysis

Naveen Paluru[a], **Mehak Arora**[a], **and Phaneendra K. Yalavarthy**[a]
[a]Department of Computational and Data Sciences, Indian Institute of Science, Bengaluru, Karnataka 560012, India

Abstract.
Purpose: To introduce a novel filter design element called rolling convolution filters for developing lightweight convolutional neural networks (CNNs) in medical image analysis, aiming to reduce model complexity and memory footprint without compromising performance.
Approach: Rolling convolution filters were generated by performing a channel-wise rolling operation on a single base filter, creating unique filters while restricting the learnable parameters. The method was applied to various two- and three -dimensional medical image analysis tasks, including reconstruction, segmentation, and classification across MRI, CT, and OCT modalities. The performance was compared with that of standard CNNs and other lightweight architectures.
Results: The proposed rolling convolution filters substantially reduced the number of parameters and model size compared to standard CNNs, with a negligible increase in performance error. For quantitative susceptibility mapping, the rolling filter approach achieved results comparable to those of state-of-the-art methods with $6\times$ fewer parameters. In COVID-19 anomaly segmentation, rolling filters performed on par with existing lightweight models while having approximately $68\times$ fewer parameters. For OCT classification, rolling filters maintained accuracy while significantly reducing the model size ($49\times$).
Conclusions: Rolling convolution filters offer an effective approach for designing lightweight CNNs for medical image analysis tasks, providing substantial reductions in model complexity and memory requirements while maintaining a performance comparable to that of larger models. This method can be easily incorporated into existing architectures and shows promise for deploying efficient deep learning models in resource-constrained medical imaging settings.

Keywords: Lightweight Networks, Parameter Redundancy, and Medical Image Analysis.

*Phaneendra K. Yalavarthy yalavarthy@iisc.ac.in

1 Introduction

Deep learning based medical image analysis has shown promising results in tasks such as segmentation, classification, reconstruction, super-resolution, etc. The standalone design aspects of architectures such as UNet,[1] ResNet[2] and their 3D variants[3] have produced state-of-the-art frameworks for automated medical image analysis. Deep learning based models have surpassed conventional techniques for different tasks across various modalities. However, most existing deep learning models are heavy in terms of the number of parameters and model size, making them difficult to deploy on edge devices, particularly in point-of-care settings. Fast and efficient computer-based diagnosis is crucial in medical imaging to improve instant diagnosis, real-time healthcare solutions, rapid treatment, and substantial cost reduction. Deep learning based computer-aided or point-of-care analysis using efficient and lightweight convolutional neural networks is the need of the hour. Techniques such as network pruning,[4] network quantization,[5] and knowledge distillation[6] for developing lightweight CNNs have been extensively studied.

Pruning, quantization, and distillation methods often require a pretrained heavy model to develop an efficient lightweight model. On the other hand, architectural design based lightweight models have performed on par with existing state-of-the-art heavy models. MobileNets[7–9] utilized depth-wise separable convolutions[3,10,11] along with residual blocks for building lightweight models. As highlighted in Zhang *et al.*[12], interleaved group convolutions proposed a novel building block consisting of primary and secondary group convolutions, thereby promoting limited model complexity and fewer parameters than the previous models. Furthermore, Zhang *et al.*[12] also showed that regular and depth-wise separable convolutions form a special case of interleaved group convolutions. MixConv[13] introduced depth-wise separable convolutions using multiple kernels with different spatial sizes to improve the model performance. Gao *et al.*[14] introduced channel-wise group convolutions (ChannelNets) to promote sparse connectivity among feature maps. Zhang *et al.*[15] and Ma *et al.*[16] introduced channel shuffle operations and point-wise group convolutions to facilitate efficient feature propagation, and proposed several practical guidelines for designing extremely lightweight models, collectively known as ShuffleNets. Tan *et al.*[17 18] have systematically studied the family of EfficientNets for scaling of deep models across the depth, width, and spatial extent of feature maps. Slimmable neural networks[19] train a single network that is switchable to different widths (channels), promoting adaptability to different on-device benchmarks and resource-constrained settings. Structured convolutions with composite kernel structures[20] decompose the convolution operation into sum-pooling components, followed by convolution with fewer weights and less computational requirements. These methods map various convolution operations to reduce the number of trainable parameters and model complexity. However, all these convolution operations need to be carried out independently, still having redundant trainable parameters.

Learning filter bases for reducing model parameters has also shown great promise for building lightweight models. Qiu *et al.*[21] have shown that decomposing the convolution filters using a set of pre-fixed basis and learning the coefficients of the expansion from the data has the potential to reduce the trainable parameters and computation overhead. Yawei *et al.*[22] propose to learn the set of basis filters for reducing the parameters of deep models. Kang *et al.*[23] introduced a deeply shared filter basis for reducing the number of parameters and model complexity. Yang *et al.*[24] proposed lego filters for building a sophisticated module representation using a split-merge-transform strategy leading to efficient convolutional neural networks. The primary bottleneck for building such a filter basis in these methods lies in the choice of the number of basis filters at a given layer, which is an additional hyperparameter.

Parameter re-usability and parameter-sharing methods for developing lightweight models have shown promising results across several tasks. Savarese *et al.*[25] introduced a parameter-sharing scheme for learning feature representations across convolutional layers as a learned linear combination of parameter tensors from a global dictionary. Yang *et al.*[26] introduced filter summary for parameter re-usability/sharing across successive convolutional filters, thus leading to lightweight models. Wang *et al.*[27] and Han *et al.*[28] proposed versatile convolution filters wherein secondary filters have been derived from a primary filter using binary masks, leading to less memory and computation cost. Han *et al.*[29] introduced a series of linear transformations on feature maps to generate more representational features at a minimum cost to reduce the parameters/model complexity in deep models. Cheng *et al.*[30] explored the redundancy of the parameters with the introduction of circular projections instead of linear projections in the fully connected layers. In addition, Refs.[31–34] have also explored circular symmetry for designing neural nets for various applications. However, these methods do not achieve the desired level of reduction in trainable parameters, particularly for tasks in medical image analysis.

This study proposes novel rolling convolution filters that promote parameter reusability/sharing for designing lightweight CNNs. A novel filter design element, called rolling convolution filters, has been introduced, which reduces the number of parameters in convolutional neural networks (CNN), thereby reducing the model complexity and memory footprint. These sets of new filters have been generated by performing a non-parameterized channel-wise rolling (or circular shifting) operation on a single base filter. Each newly developed filter is unique, but the number of learnable parameters is restricted to that of the base filter, which addresses the problem of redundant parameters often observed in deep neural networks. The use case of these rolling convolution filters (both 3D and 2D) in medical image analysis across three different problems, including reconstruction (3D), segmentation (2D), and classification (2D) have been investigated. The proposed filters adequately reduce the number of parameters, accounting for the low model complexity.

The main contributions of this study can be summarized as follows:

1. Development of novel rolling convolutional filters based lightweight CNNs for efficient medical image analysis. The proposed rolling convolution filters promote parameter reusability to reduce model complexity and memory footprint, making them preferable for developing

lightweight CNNs.

2. This is also the first ever channel rolling operation (both in 3D and 2D) utilized to generate a set of new convolution filters from a single base filter. The higher-dimensional equivalence of the feature maps using the proposed filters with the standard convolutional filters is shown in a use case.

3. It was also shown that the proposed rolling convolution filters based lightweight convolutional neural networks (CNNs) perform on par with their heavyweight counterparts across medical image analysis tasks like reconstruction, segmentation, and classification. Specifically, quantitative susceptibility mapping (QSM) reconstruction, COVID-19 anomaly segmentation, and OCT-based retinal disease classification were used to demonstrate the efficacy of the proposed rolling convolution filters.

2 Methods

This section describes various lightweight convolution strategies used in the literature, along with the proposed approach of rolling convolution filters to reduce the model size of CNNs. An overview of the different convolution strategies is presented in Fig. 1. The discussion below details two-dimensional convolutions and can be extended to three-dimensional convolutions.

2.1 Existing Convolution Filters

2.1.1 Standard Convolution Filters

Given a set of feature maps x of dimensions (batch size = 1, in channels = K, spatial extent = $\mathrm{M \times M}$), a set of convolution filters F of dimensions (out channels (or no. of filters) = K, in

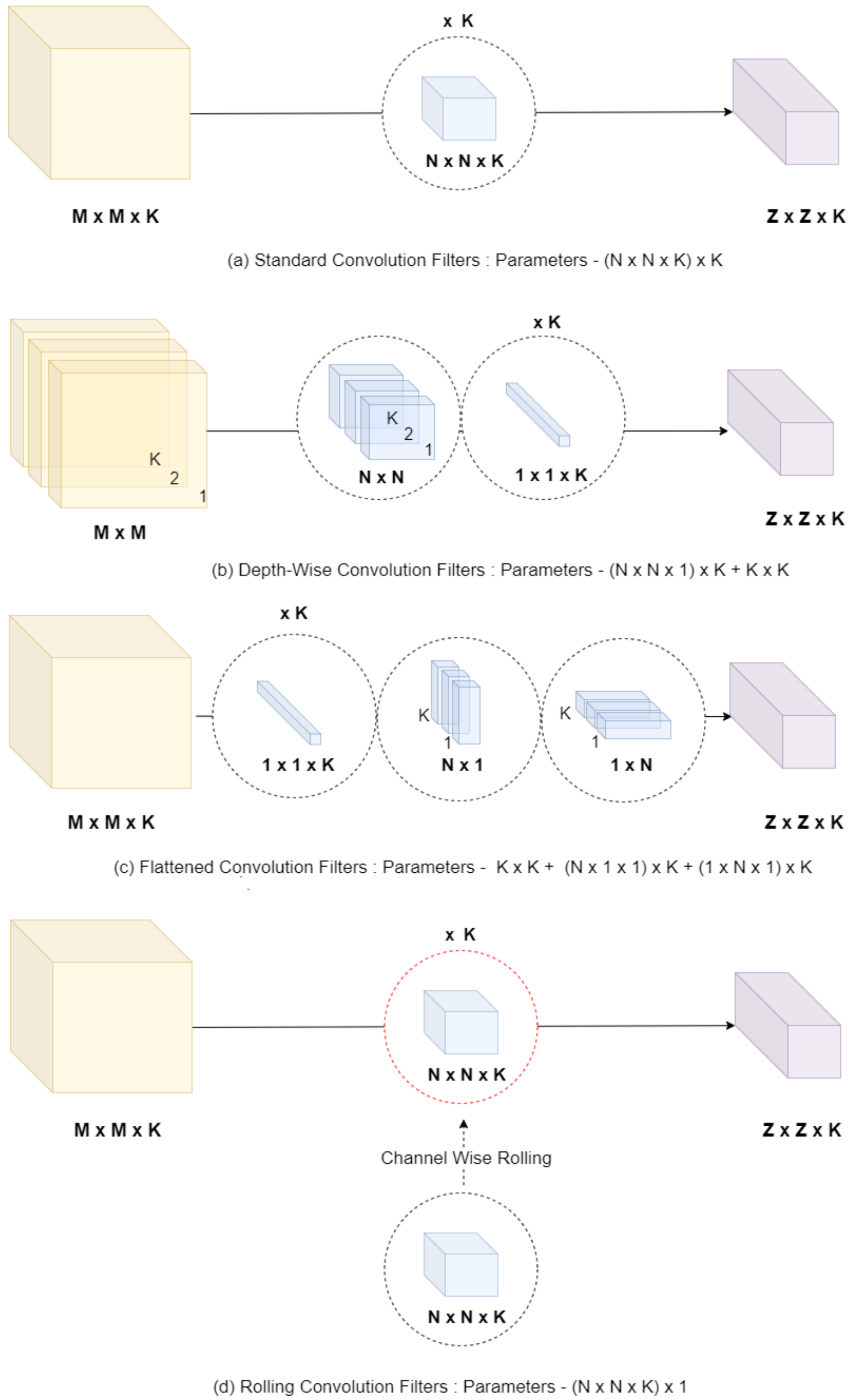


Fig 1 A comparison between existing convolution filters and proposed rolling convolution filters. (a) standard convolution filters, (b) depth-wise separable filters, (c) flattened convolution filters , and (d) rolling convolution filters. Depth-wise separable convolution filters perform independent channel-wise convolutions, and an assimilated representation is formed using 1×1 point-wise convolution. Flattened convolution filters perform the lateral convolution operation across the channels (1×1 point-wise), followed by convolution across the horizontal and vertical dimensions of the feature maps. In contrast, the proposed rolling convolution filters generate sets of new convolution filters by performing a non-parameterized channel-wise rolling (or circular shifting) operation on a single base filter.

channels = K, spatial extent = N x N), the output feature maps $\mathbf{y}$ obtained from the standard 2D convolution filters can be represented as follows:

$$\mathbf{y}_i = \mathbf{x} * F_i \tag{1}$$

where * is the convolution operator, $\mathbf{y}_i$ is the i^{th} feature map in $\mathbf{y}$ and F_i is the i^{th} convolution filter. The number of parameters required for generating a K channel output feature map from a K channel input feature map using a filter with spatial extent N $\times$ N is K^2 $\times$ N^2. During backpropagation, the gradients are computed as

$$\frac{\partial \mathcal{L}}{\partial F_{ij}} = \frac{\partial \mathcal{L}}{\partial \mathbf{y}_i} * \mathbf{x}_j \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_j} = \sum_{i=1}^{\mathrm{K}} \frac{\partial \mathcal{L}}{\partial \mathbf{y}_i} * \psi(F_{ij}) \tag{3}$$

where $\mathcal{L}$ is the computed loss, F_{ij} is the j^{th} kernel of the i^{th} convolution filter, $\mathbf{y}_i$ is the i^{th} feature map in $\mathbf{y}$, $\mathbf{x}_j$ is the j^{th} feature map in $\mathbf{x}$ and ψ is the flipping operator that flips the elements of the kernel both horizontally and vertically with respect to the center. The filters F_1, F_2, ... , F_{K} are independent of each other, and only the i^{th} feature map in $\mathbf{y}$ contributes for updating the parameters of the convolution filter F_i.

2.1.2 Depth-Wise Convolution Filters

Depth-wise separable convolution filters[10] perform convolution independently on each channel of the input feature map. Finally, point-wise $1{\times}1$ convolutions are used to increase or decrease the number of feature maps. The same can be represented as

$$\mathbf{y}_i = \mathbf{x}_i * f_i \tag{4}$$

where $\mathbf{y}_i$ is the i^{th} feature map in $\mathbf{y}$ and f_i is the i^{th} channel of the convolution filter F and, $\mathbf{x}_i$ is the i^{th} feature map in $\mathbf{x}$. This separable convolution is followed by a point-wise convolution using 1×1 convolutions to generate a collective representation. Total parameters required for generating a K channel output feature map from a K channel input feature map using a depth-wise separable filter with spatial extent $\mathrm{N} \times \mathrm{N}$ is $\mathrm{K}^2 + \mathrm{K} \times \mathrm{N}^2$. The former term originates from point-wise convolutions, whereas the latter accounts for channel-wise convolutions.

2.1.3 Flattened Convolution Filters

Flattened convolutions[35] (also known as spatially separable or factorized convolutions) split the standard convolution into three stages: lateral (across channels), followed by convolutions across the horizontal and vertical dimensions. Lateral convolutions are performed using point-wise convolutions. The horizontal and vertical convolutions are followed by lateral convolutions and are factorized across the respective dimensions, which are eventually performed independently (similar to depth-wise separable filters) on each input-feature map. Note that in depth-wise separable filters, point-wise convolutions are performed after separable convolutions, whereas in flattened convolutions, separable convolutions are performed after point-wise (lateral) convolutions. The total number of parameters required to generate a K channel output feature map from a K channel input feature map using a flattened convolution filter are $\mathrm{K}^2 + \mathrm{K} \times \mathrm{N} \times 2$. The former term originates from point-wise convolutions, whereas the latter accounts for flattened convolutions.

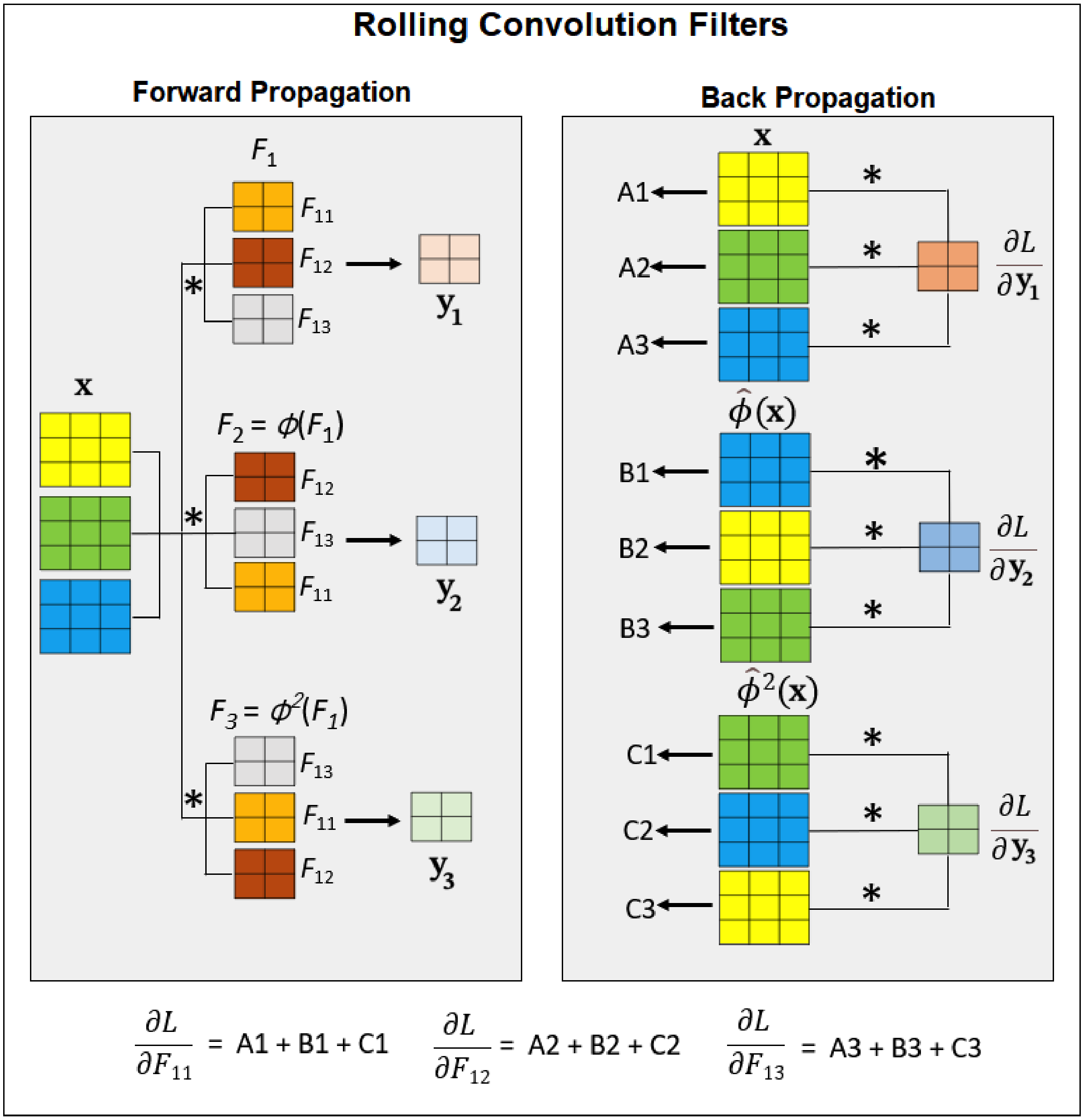


Fig 2 Forward propagation and backpropagation through 2D rolling convolution filters. $\mathbf{x}$ is a 3-channel feature map, $*$ is the convolution operator, ϕ is the channel-rolling operator, $\mathbf{y}$ is the output feature map, and $\hat{\phi}$ is the channel-rolling operator that operates in the opposite direction of ϕ.

2.2 Proposed Rolling Convolution Filters

This study proposes generating a set of convolution filters from a single base filter using the channel rolling operation, as shown in Fig. 2. Let F_1 be a convolution filter of dimension (out channels = 1, in channels = K, and spatial extent = N x N). Subsequently, the set of filters generated from F_1

are as follows

$$
\begin{aligned}
F_2 &= \phi(F_1) \\
F_3 &= \phi(F_2) = \phi^2(F_1) \\
&\vdots \\
F_{\mathrm{K}} &= \phi(F_{\mathrm{K}-1}) = \phi^{\mathrm{K}-1}(F_1)
\end{aligned}
\tag{5}
$$

where, ϕ is the channel rolling operator shown in Fig. 2. The channel rolling operator ϕ cyclically shifts the filter channels by a single position. ϕ^k performs k cyclic shifts, and ϕ^0 performs identity mapping. The output feature maps $\mathbf{y}$ obtained from the rolling 2D convolutional filters are represented as follows:

$$
\mathbf{y}_i = \mathbf{x} * \phi^{i-1}(F_1) \tag{6}
$$

In standard convolution, the convolutions are performed among the corresponding channels of the input and the filter. In rolling convolution, the channel-rolling operation on the filter convolves each channel of the filter with every other channel of the input to generate a salient representation of the feature space. The number of parameters required for generating a K channel output feature map from a K channel input feature map using a rolling convolution filter with a spatial extent $\mathrm{N} \times \mathrm{N}$ is limited to $\mathrm{K} \times \mathrm{N}^2$. During backpropagation, the gradients of the rolling convolution are computed as follows:

$$
\frac{\partial \mathcal{L}}{\partial F_{1j}} = \sum_{i=1}^{\mathrm{K}} \frac{\partial \mathcal{L}}{\partial \mathbf{y}_i} * \left(\hat{\phi}^{i-1}(\mathbf{x})\right)_j \tag{7}
$$

$$
\frac{\partial \mathcal{L}}{\partial \mathbf{x}_j} = \sum_{i=1}^{\mathrm{K}} \frac{\partial \mathcal{L}}{\partial \mathbf{y}_i} * \psi\left(\phi^{i-1}(F_1)\right)_j \tag{8}
$$

where $\hat{\phi}$ is the channel rolling operator that operates in the opposite direction of ϕ and ψ is the flipping operator that flips the kernel elements both horizontally and vertically with respect to the

Algorithm 1 Rolling Convolution Operation (2D)

Input: $\mathbf{x} \leftarrow$ Input Feature Map `(B, C, H1, W1)`
    $\mathbf{w} \leftarrow$ Convolution Filter `(1, C, K, K)`
    $s \leftarrow$ Stride
    $p \leftarrow$ Padding
Output: $\mathbf{y} \leftarrow$ Output Feature Map `(B, C, H2, W2)`
for $i = 1$ to C **do**
  if $i == 1$ **then**
    $\mathbf{f} \leftarrow \mathbf{w}$
  else
    $\mathbf{w} \leftarrow$ CircularShift($\mathbf{w}$, dim $= 1$, shift $= 1$)
    $\mathbf{f} \leftarrow$ Concatenate($\mathbf{f}$, $\mathbf{w}$, dim $= 0$)
  end if
end for
{$\mathbf{f}$ `is the Rolled Filter with Dimensions (C, C, K, K)`}
$\mathbf{y} \leftarrow$ Convolution($\mathbf{x}$, $\mathbf{f}$, stride $= s$, padding $= p$)
$\mathbf{y} \leftarrow$ ReLU($\mathbf{y}$)
return $\mathbf{y}$

center. As the rolling convolution generates all output feature maps from a single base filter, the gradients accumulate during backpropagation. The channel rolling operator $\hat{\phi}$ cyclically shifts the filter channels by a single position. $\hat{\phi}^{\mathrm{K}}$ performs K cyclic shifts, and $\hat{\phi}^0$ performs identity mapping. The forward and backward propagations through the 2D rolling convolution filters are shown in Fig. 2. Further, a detailed pseudocode explaining the rolling convolution operation (2D) is shown in Algorithm 1. Similarly, the rolling convolution operation can be extended to the 3D case as well. Given a K-channeled filter, at most a K-channeled feature map can be generated using rolling convolutions. To increase the number of feature maps, say to 2K, two independent filters, each with K channels, must be used.

2.3 Rolling Filters: Effective for Medical Image Analysis

The proposed rolling convolution filters can be deployed in any deep learning model. However, the proposed methodology is more effective for medical image analysis in the following ways: First, the complexity/variability in modeling input-to-output mapping is limited in medical image

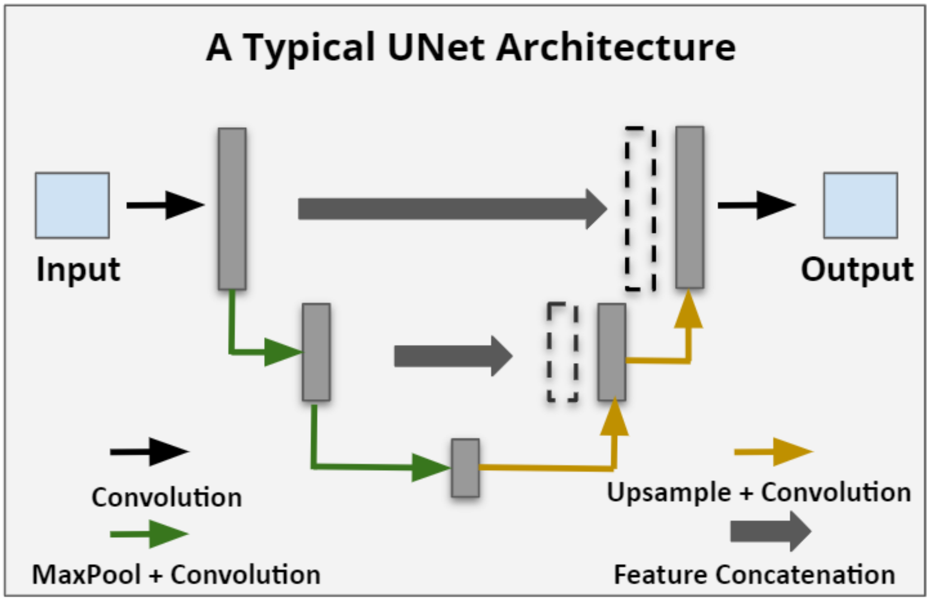


Fig 3 A typical design aspect of a UNet architecture. The encoder-decoder structure embedded with feature concatenations forms a method for salience representation. Feature maps (convolution layer 1) obtained from UNet and Rolling UNet on a sample chest CT slice for the task of COVID-19 anomaly segmentation are shown in Fig. 7.

analysis compared with computer vision tasks. Hence, the extreme compression offered by the rolling convolution filter has little effect on the performance of deep learning models in automating medical image analysis. Second, the design aspect of UNet has laid a strong foundation for the development of various image-to-image (and volume-to-volume)-based deep learning models for automated medical image analysis. A typical UNet architecture is shown in Fig. 3. The model consists of an encoder-decoder structure embedded with feature concatenation. Feature concatenation provides a salient representation and enables a hierarchical gradient flow for efficient training. As shown in Fig. 3, it is worth noting that the UNet model's final output is strongly dependent on the features extracted from the initial layers. The proposed rolling convolution filters with fewer parameters can capture similar semantics from the initial layers compared to standard convolution filters (refer to Fig. 7).

3 Experiments

The related works, datasets, and implementation details of various tasks considered in this study are detailed in each subsection.

3.1 Quantitative Susceptibility Mapping

Quantitative susceptibility mapping (QSM) is a magnetic resonance (MR) imaging-based parametric imaging method that measures magnetic susceptibility and has applications in the assessment of several brain disorders, such as brain hemorrhage, multiple sclerosis, and Parkinson's disease.[36] A vital step in QSM is reconstruction (dipole inversion), which is a classical ill-posed inverse problem in medical imaging. This involves deconvolving the susceptibility distribution from the relative difference field (also known as the local field or tissue phase) obtained from the phase information in the MR image. The relationship between the local field (also known as the relative difference field or tissue phase) $\delta_{\mathrm{B}}(r)$ and susceptibility $\chi(r)$ is as follows:[37]

$$\delta_{\mathrm{B}}(r) = \frac{1}{4\pi}\int_{\mathbf{r}'\neq\mathbf{r}}^{R^3} \chi(r')\frac{3\cos^2\Theta - 1}{|r - r'|^3}dr' \tag{9}$$

where, r is a spatial location [x,y,z] in the three-dimensional (3D) MR volume and Θ is the angle between the unit vector along $r - r'$ and the unit vector along the direction of the main magnetic field during MR acquisition. The same relation can be written as a convolution operation ($*$) between $\chi(r)$ and d(r) (known as the dipole kernel[38]), as follows:

$$\delta_{\mathrm{B}}(r) = \chi(r) * d(r) \tag{10}$$

where,

$$d(r) = \frac{3\cos^2\Theta - 1}{4\pi|r|^3} \tag{11}$$

Given $d(r)$ and $\delta_{\mathrm{B}}(r)$, Eq. (10) must be deconvolved to reconstruct $\chi(r)$. Conventionally, deconvolution is performed in the Fourier domain. The deconvolution problem becomes ill-posed when $\Theta \sim 54.7356°$, leading to a division by zero in the Fourier domain. As a result, several streaking artifacts get induced in the reconstructed susceptibility maps. Liu *et al.*[39] have introduced the COSMOS algorithm (that serves as the gold standard for QSM), which utilizes multiple head orientations data for effective susceptibility mapping. This requires data from multiple orientations, which makes the data acquisition time prohibitively long. Recently, deep-learning methods such as DeepQSM,[40] QSMnet,[41] QSMnet+,[42] and xQSM[43] were more effective than traditional methods for solving ill-posed dipole deconvolution. In the existing literature,[44] deep learning methods for QSM are heavy models based on 3D-UNet (for example, QSMnet has $\sim$ 99 million parameters). As highlighted in Jung *et al.*[45], efficient designs (or frameworks) for deep learning based susceptibility mapping is the need of the hour.

A total of 12 healthy volunteers' MRI data from five different head orientations were used in this study.[41] In total, there were 60 $(= 12 \times 5)$ scans (three-dimensional (3D) data), of which 25 scans were utilized for training, 5 scans for validation, and 30 scans for testing. The data acquisition details are provided in.[41] All the deep models in this study were trained using 3D patches of dimensions $64 \times 64 \times 64$. There were 16800 3D patches in the training set and 1680 3D patches in the validation set. Following Yoon *et al.*,[41] the data pairs, that is, the input local field and label COSMOS maps matching the input local-field orientation, were utilized to train the proposed framework. Experiments using the proposed method of rolling kernels were conducted using QSMnet[41] as the baseline. The peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), high-frequency error component (HFEN), and normalized mean square error (NMSE) were used to quantify the performance of the quantitative susceptibility mapping methods.[46]

As shown in Eq. (13), a linear combination of ℓ_1 error and the regularization loss $\mathcal{L}_{reg}$ Eq. (12) was backpropagated to train the Rolling QSMnet, where χ is the predicted susceptibility, χ_{C} is the COSMOS-generated susceptibility, and ∇ is the gradient extraction kernel that returns the x, y, and z gradients. Following Yoon *et al.*,[41] the values of w_1 and w_2 were set to 0.5 and 0.1, respectively. The model was trained for 25 epochs with a mini-batch size of 16. The initial learning rate was $5e^{-3}$ and was gradually decayed by 0.1 once for every 15 epochs. The model parameters were optimized using the Adam[47] optimizer.

$$\mathcal{L}_{\text{reg}} = w_1 \big|\big|\, d * \chi - d * \chi_{\mathrm{C}} \,\big|\big|_1 + w_2 \big|\big|\, \nabla * \chi - \nabla * \chi_{\mathrm{C}} \big|\big|_1 \tag{12}$$

$$\mathcal{L}_{\text{total}} = \big|\big|\, \chi - \chi_{\mathrm{C}} \,\big|\big|_1 + \mathcal{L}_{\text{reg}} \tag{13}$$

3.2 COVID-19 Anomalies Segmentation

Findings from chest computed tomography (CT) images are beneficial for screening COVID-19.[48] Deep-learning-based data-driven approaches have been proposed for instant COVID-19 diagnosis.[49–52] The imaging features of interest from the CT images of COVID-19 patients were ground-glass opacities (GGOs), Consolidations, and Pleural effusions[52–54]. Among these, the GGO was the predominant feature. The segmentation of GGOs from chest CT images has been extensively studied well in the literature. Anamorphic depth embedding-based lightweight CNN,[55] called Anam-Net, has been proposed for efficient segmentation of COVID-19 anomalies in a point-of-care setting. Fan *et al.*[56] proposed a multi-attention semi-supervised approach for segmenting COVID-19 anomalies from chest CT images. Wang *et al.*[57] addressed the problem of high-level annotations by proposing a robust COVID-19 segmentation framework trained from low-level (noisy) annotations.

The dataset[58] consists of 3410 axial CT slices obtained from 20 patients. These slices were divided into four folds at the patient level. Three-fold cross-validation on folds F1, F2, and F3 was performed. The fold F4 (with 545 slices) was explicitly used for testing. The datasets were split, and the preprocessing of the CT slices was performed as described in Ref.[55] The annotations consisted of three labels: abnormal region (class-0, having GGOs, consolidations, or pleural effusions), the normal region (class-1), and the background (class-2, non-lung region). Experiments with the proposed method of rolling kernels were conducted to segment COVID-19 anomalies using UNet[1] as a baseline. Standard figures of merit, Specificity, Sensitivity, Accuracy, and Dice score were utilized to quantify the performance of the segmentation models considered in this study.

The CT COVID dataset exhibited class imbalance among the background, normal, and abnormal regions. To address this, we employed a weighted cross-entropy loss across all models in our experiments.[55] The weighted cross-entropy loss is shown in Eq. (14), where $\mathbf{y}'$ has the predicted softmax probabilities, $\mathbf{y}$ is the one-hot encoded annotation, and w_{ij} is the weight given to the corresponding label at ij^{th} location. These weights were computed as $w(t) = 1/p(t)$, where $p(t)$ is the fraction of voxels having label $t \in \{0{,}1{,}2\}$ in the training set. The model was trained for 50 epochs with a mini-batch size of 5. The initial learning rate was $5e^{-4}$ and gradually decayed by a factor of

0.1 once every 33 epochs. The model parameters were optimized using Adam[47] optimizer.

$$\mathcal{L} = -\sum_{i=1}^{rows}\sum_{j=1}^{cols}(w_{ij})\sum_{t=0}^{2}\mathbf{y}_{ijt}\log\left(\mathbf{y}'_{ijt}\right) \tag{14}$$

3.3 OCT B-scans Classification

The most common retinal diseases that can be diagnosed using OCT B-scan images include Choroidal Neovascularization (CNV), Diabetic Macular Edema (DME), and Drusen (DRU). A small amount of retinal fluid formed near the retinal layer characterizes CNV; DME accounts for the formation of fluid-filled cysts, and Drusen results in irregular retinal boundaries. Given these distinctions, computer-aided automated detection and classification of these abnormalities is of vital interest, which one wishes to perform in real time. Several deep-learning-based attempts[59–62] have been made to fully automate this detection/classification without the need for expert or clinician intervention. The current state-of-the-art method is an ensemble of deep residual networks proposed by Feng *et al.*[63] This framework instantiates the responses from four ResNet-50[2] architectures and makes an ensemble decision for its prediction. Given the task of automated classification of retinal diseases using OCT images (a small problem compared with vision-based general object classification), mini convolutional neural networks with proper hyperparameters perform on par with current deep (heavy) models. Experiments with the proposed method of rolling filters were conducted to classify OCT B-scans using ResNet-18[2] as the baseline. The University of California San Diego (UCSD) OCT dataset[59] was used for modeling retinal disease classification.

The UCSD dataset had 37455 B-scans for CNV, 11598 for DME, 8866 for Drusen, and 51390 for Normal. There were approximately 109308 images that were split into the training set (70%), validation set (15%), and testing set (15%). The models were trained using the cross-entropy shown in Eq. (15), where $\mathbf{y}'$ has the predicted softmax probabilities and $\mathbf{y}$ is the one-hot encoded annotation. The models were trained for 30 epochs with a mini-batch size of 32. The initial learning rate was $5e^{-4}$ and gradually decayed by 0.1 once every 7 epochs. The parameters were optimized using the Adam[47] optimizer.

$$\mathcal{L} = -\sum_{t=0}^{3} \mathbf{y}_t \log\left(\mathbf{y}'_t\right) \tag{15}$$

Across all experiments, the data were carefully partitioned into training, validation, and testing sets, as well as across different folds, with all splits performed at the patient level, promoting fair evaluation practices and preventing data leakage. All experiments were performed using PyTorch[64] on a Linux workstation with i9 9900X (CPU) with 128 GB RAM and an NVIDIA Quadro RTX 8000 GPU card with a capacity of 96 GB.

4 Results

4.1 Quantitative Susceptibility Mapping

The representative reconstruction results of the dipole deconvolution methods DeepQSM,[40] xQSM,[43] QSMnet,[41] FINE,[65] LPCNN[66] and the proposed Rolling QSMnet on a sample test volume (sagittal view) are shown in Fig. 4. The averaged figures of merit over the test volumes across all orientations, utilizing all dipole deconvolution methods considered in this study, are listed in Table 1. Furthermore, comparisons with depth-wise separable filters and flattened convolution filters (with QSMnet as the baseline) are detailed in Table 1. Despite having $\sim 6\times$ fewer parameters and $\sim 7\times$ lighter in terms of model size than existing lightweight designs (Table 1), the proposed Rolling QSMnet performed on par with depth-wise QSMnet, flattened QSMnet, and QSMnet (Ta-

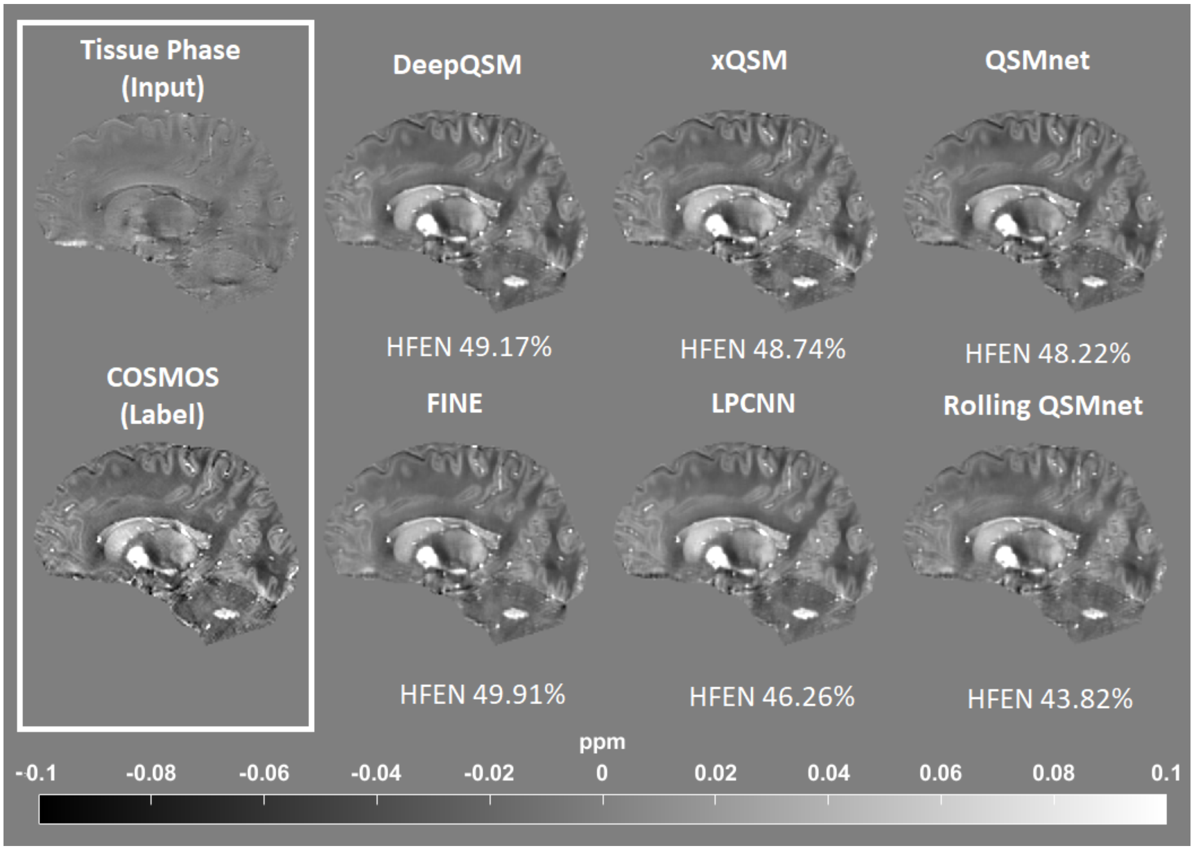


Fig 4 Representative susceptibility reconstruction results (sagittal view) across orientation 1 of considered methods (column-wise) on a sample test volume. The high-frequency error component (HFEN(%) ↓) with reference to the gold standard COSMOS was provided for each slice. Overall quantitative metrics of considered methods across the test volumes are shown in Table 1.

ble 1). As shown in Fig. 4, a promising perceptual similarity of the proposed Rolling QSMnet to QSMnet and to COSMOS was observed, and the same is evident from the metrics quantified in Table 1. Voxel-wise difference (error) images between Rolling QSMnet and COSMOS, as well as corresponding error maps for the other QSM methods were shown in Fig. 5. These maps show that discrepancies are mainly localized to regions with strong susceptibility gradients. Overall, the Rolling QSMnet exhibited a strong spatial agreement with COSMOS. The “promising perceptual similarity” refers to the close visual correspondence in anatomical contrast, preservation of venous structures, and accurate depiction of susceptibility variations in deep gray matter, with only minor edge related differences and no obvious systematic bias.

Table 1 Averaged figures of merit over 30 patient volumes (test cases) from all dipole deconvolution methods considered in this work. Note that the inference (s) reported were for the (GPU and CPU) implementation. Susceptibility reconstruction results (sagittal view) across orientation 1 of considered methods are shown in Fig. 4.

	DeepQSM [40]	**xQSM** [43]	**QSMnet** [41]	**FINE** [65]	**LPCNN** [66]	**Depth Wise QSMnet**	**Flattened QSMnet**	**Rolling QSMnet** (proposed)
Parameters	$\sim$ 5.64 M	$\sim$ 5.21 M	$\sim$ 99 M	$\sim$ 5.64 M	$\sim$ 470 K	$\sim$ 2.5 M	$\sim$ 2.2 M	$\sim$ **445 K**
Size (MB)	21.54	19.89	379.38	21.54	1.71	9.75	8.59	**1.70**
Inference (s)	**0.85, 8.25**	1.25, 13.6	1.32, 14.1	90.25, 285.35	3.72, 24.35	0.62, 16.5	1.25, 17.6	1.41, 15.2
SSIM	**0.910** $\pm$ **0.01**	0.908 $\pm$ 0.01	0.910 $\pm$ 0.01	0.909 $\pm$ 0.01	0.905 $\pm$ 0.01	0.900 $\pm$ 0.01	0.900 $\pm$ 0.01	0.900 $\pm$ 0.01
PSNR	**41.10** $\pm$ **0.93**	40.98 $\pm$ 1.03	41.04 $\pm$ 1.01	41.07 $\pm$ 0.91	40.88 $\pm$ 1.06	40.72 $\pm$ 0.98	40.79 $\pm$ 0.98	40.71 $\pm$ 1.03
NMSE (%)	**51.19** $\pm$ **3.39**	52.23 $\pm$ 3.78	51.39 $\pm$ 3.81	51.40 $\pm$ 3.37	52.93 $\pm$ 4.01	53.31 $\pm$ 4.20	52.89 $\pm$ 3.61	53.34 $\pm$ 3.72
HFEN (%)	49.21 $\pm$ 3.82	50.43 $\pm$ 4.66	**48.41** $\pm$ **4.42**	49.43 $\pm$ 3.81	49.71 $\pm$ 4.78	51.22 $\pm$ 5.16	50.64 $\pm$ 4.53	51.31 $\pm$ 4.54

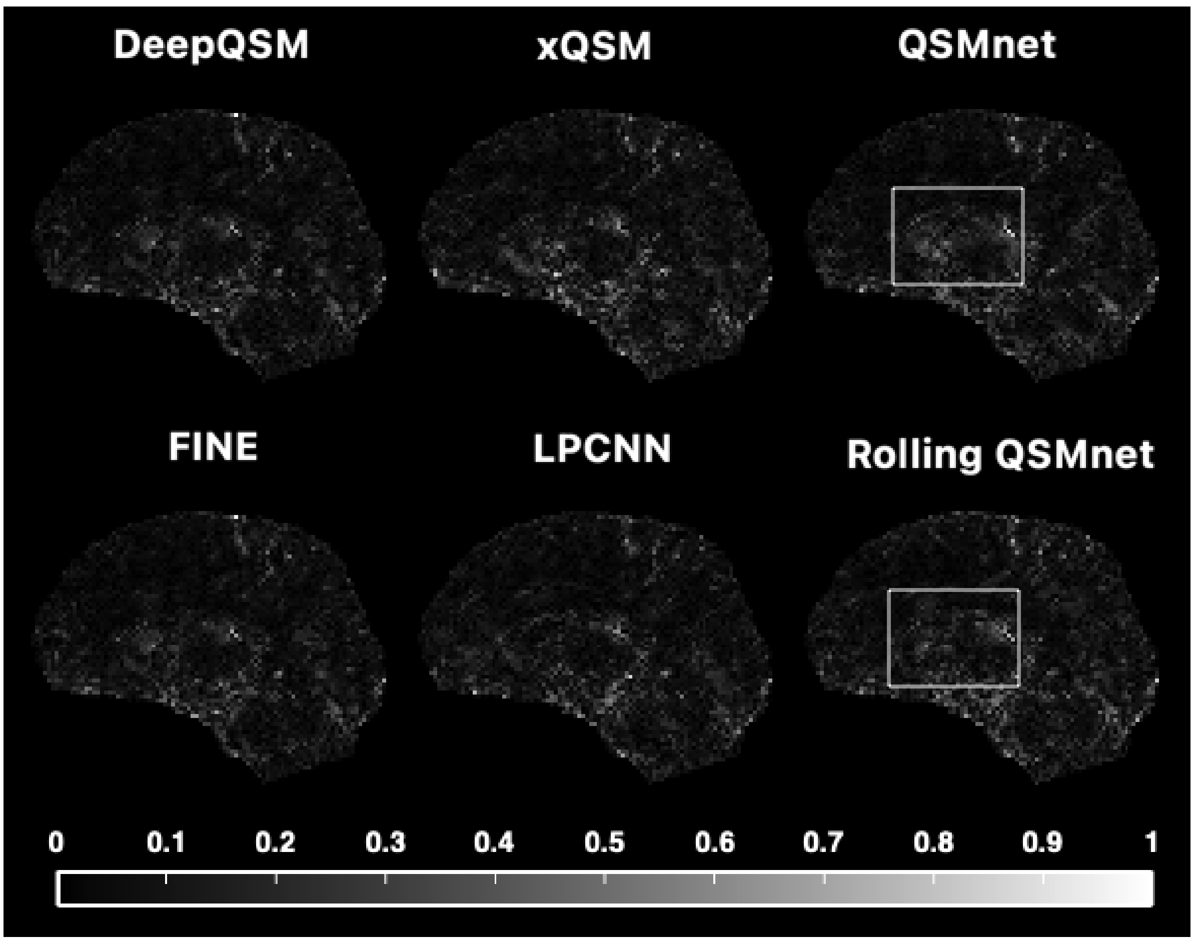


Fig 5 Error maps of the QSM reconstructions shown in Fig. 4, computed with respect to the COSMOS reconstructions as reference. Error maps are shown for all the compared QSM methods. The comparison between the baseline QSMnet and the proposed Rolling QSMnet is highlighted, illustrating differences in spatial error distribution and overall agreement with COSMOS.

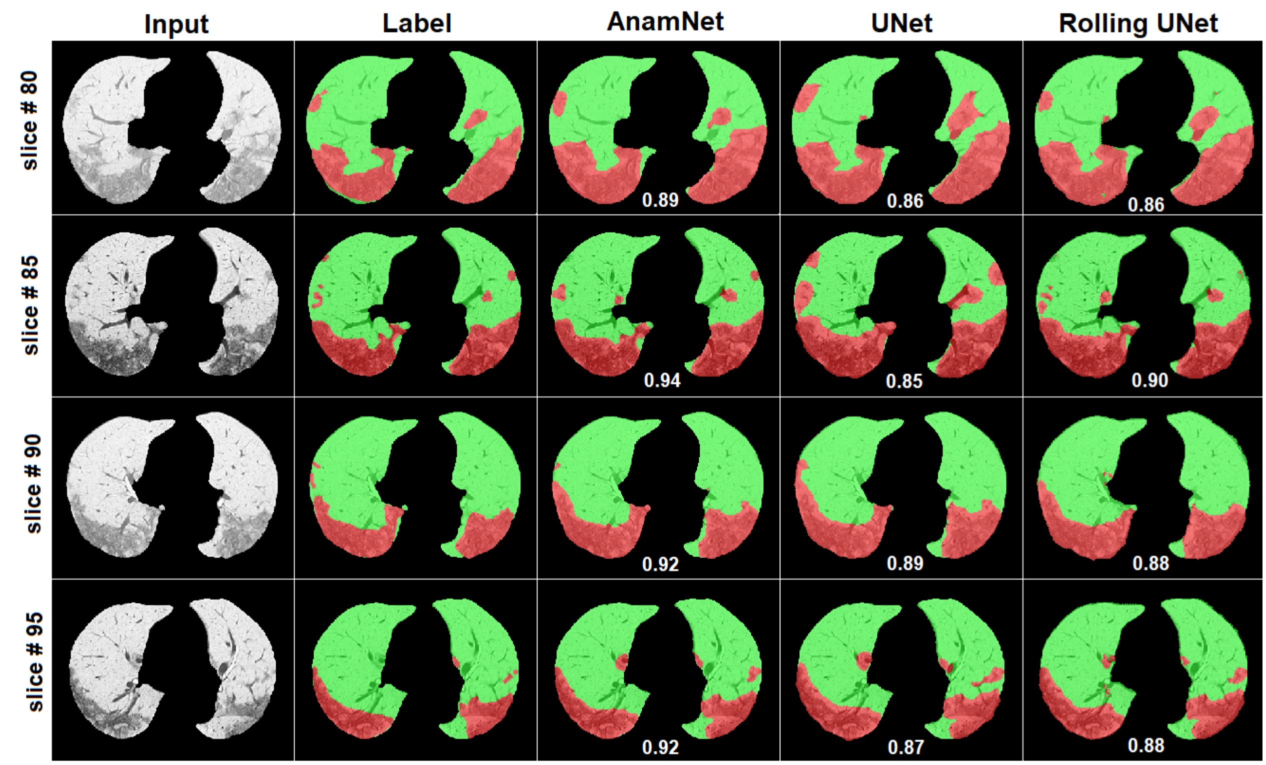


Fig 6 Representative segmentation results on a patient volume (test cases). The input slices (test cases) are shown in the first column. The respective annotations (ground truth) are presented in the second column. The predictions of the UNet and Rolling UNet are presented in the fourth and fifth columns, respectively. Abnormalities in the lung region are indicated in red, and the normal lung region is indicated in green. The average Dice Similarity scores are given below the corresponding slices. Overall quantitative metrics of considered methods across the test cases are shown in Table 2.

4.2 COVID-19 Anomalies Segmentation

Representative COVID-19 anomaly segmentation results were shown in Fig. 6, and the corresponding figures of merit were provided in Table 2. The UNet and the proposed rolling counterpart performed equally well, with comparable Dice similarity scores for COVID-19 anomaly segmentation. The proposed Rolling UNet (for COVID-19 segmentation) performed on par (refer to Table 2 and Fig. 6) with the existing state-of-the-art lightweight model AnamNet[55] (~ 4.4M parameters). Rolling UNet had $\sim 68\times$ fewer parameters and $\sim 64\times$ lighter model size than existing lightweight designs (Depth-Wise UNet and Flattened UNet).

Intuitively, these results indicate that the rolling convolution operation effectively preserves the

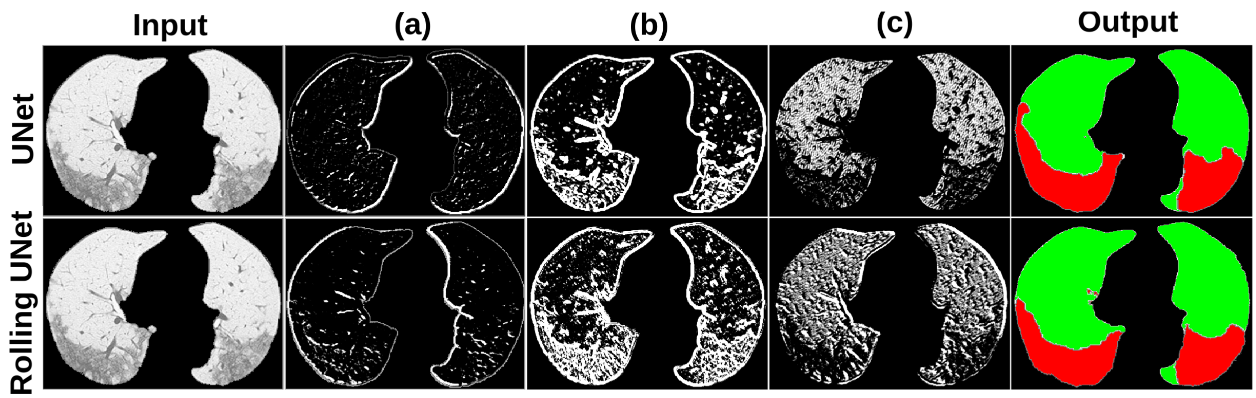


Fig 7 Feature representations of CT images from the initial layers of UNet driven architectures while segmenting COVID-19 anomalies. The input CT images are presented in the first column. The final predictions are presented in the fifth column. The features obtained with standard convolutions (UNet) and from rolling convolutions (Rolling UNet) are shown in the first and second rows, respectively. (a) clearly distinguishes the lung region from the background (through edge extraction), (b) most of the neurons fired for the abnormal region (shown in red color), and (c) activated neurons focused on the normal/healthy tissue (shown in green color) of the lung.

Table 2 Average figures of merit across three cross folds for the COVID-19 anomalies segmentation. Note that the reported inference (s) were for GPU and CPU. Representative segmentation results on sample chest CT slices (test cases) are shown in Fig. 6.

	AnamNet [55]	**UNet** [67]	**Depth Wise UNet**	**Flattened UNet**	**Rolling UNet** (proposed)
Parameters	∼4.4 M	∼31 M	∼6.01 M	∼6.01 M	∼**88 K**
Size (MB)	17.21	118.5	22.90	22.90	**0.34**
Inference (s)	0.36, 1.5	0.52, 1.2	**0.21, 0.99**	0.21, 1.15	0.45, 2.3
Sensitivity	**0.914**	0.910	0.906	0.884	0.909
Specificity	**0.993**	0.992	0.990	0.991	0.989
Accuracy	**0.988**	0.987	0.984	0.984	0.983
Dice Score	**0.869**	0.864	0.829	0.829	0.831

critical spatial and contextual information required for accurate anomaly segmentation, even with a substantially reduced number of parameters. By efficiently reusing and shifting filter responses, the rolling design minimizes redundancy inherent in conventional convolutions, enabling compact models to maintain strong representational capacity without compromising segmentation performance. Despite using significantly fewer parameters, the proposed rolling convolution filters are able to capture semantic representations in the early layers that are comparable to those learned

Table 3 Average figures of merit (over $\sim$ 14500 OCT B-scans, Test Cases) for OCT classification task. Note that inference (s) reported were for GPU, CPU.

	MobileNetV2 [8]	**ShuffleNetV2** [16]	**SqueezeNet** [68]	**ResNet18** [2]	**Rolling ResNet18** (proposed))
Parameters	$\sim$2.2 M	$\sim$1.2 M	$\sim$724 K	$\sim$11 M	$\sim$ **226 K**
Size (MB)	8.50	4.79	2.76	42.25	**0.86**
Inference (s)	0.18, 0.2	0.19, 0.2	0.18, 0.1	**0.15, 0.2**	0.18, 0.5
Precision	0.96 $\pm$ 0.01	0.95 $\pm$ 0.01	**0.96** $\pm$ **0.01**	0.96 $\pm$ 0.01	0.95 $\pm$ 0.01
F1 Score	0.94 $\pm$ 0.01	0.93 $\pm$ 0.01	**0.95** $\pm$ **0.01**	0.94 $\pm$ 0.01	0.92 $\pm$ 0.01
Accuracy	0.97 $\pm$ 0.01	0.96 $\pm$ 0.01	**0.97** $\pm$ **0.01**	0.96 $\pm$ 0.01	0.96 $\pm$ 0.01

by standard convolution filters (see Fig. 7). This indicates that the rolling mechanism effectively reuses filter weights to extract similar low level features without sacrificing representational capacity.

4.3 OCT B-scans Classification

The performance of the rolling convolution filters was consistent with that of retinal disease classification. The average results in terms of precision, F1 score, and accuracy on the UCSD dataset are listed in Table 3. Comparisons with existing lightweight models, such as MobileNetV2,[8] ShuffleNetV2,[16] and SqueezeNet,[68] were also detailed in Table 3. The abnormal regions and Grad-Cam[69] visualizations are shown in Fig. 8. The differences observed in some of the Grad-CAM visualizations can be attributed to the reduced parameterization of the Rolling ResNet18. While the rolling convolution filters reuse weights to generate multiple effective filters, this constrained parameter space can lead the network to rely on slightly different feature combinations and activation pathways compared to the standard ResNet18. Consequently, the regions emphasized by Grad-CAM may vary in some cases, even though the overall predictive performance remains comparable. Despite this, the Grad-CAM visualizations for Rolling ResNet18 consistently focus on the regions of interest highlighted by the bounding boxes, capturing the relevant visual semantics associated with the abnormalities. Further Fig. 9 presents the One-vs-Rest (OVR) ROC curves for OCT image classification across CNV, DME, Drusen, and Normal classes using (a) the baseline ResNet18 and (b) the proposed Rolling ResNet18. The similar ROC profiles across all classes indicate that the proposed method performs on par with the baseline model in terms of class-wise discrimination.

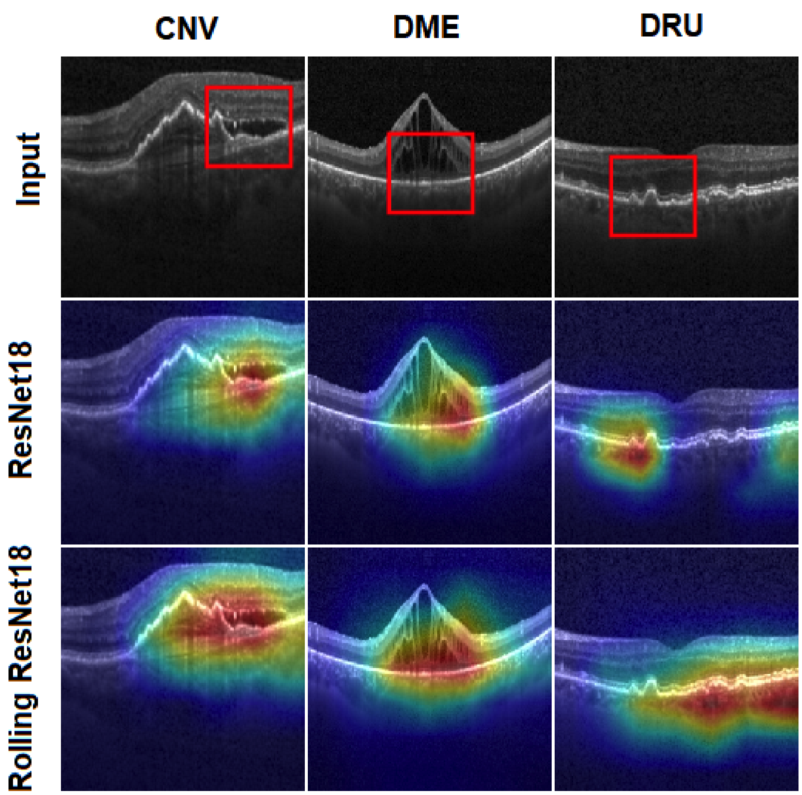
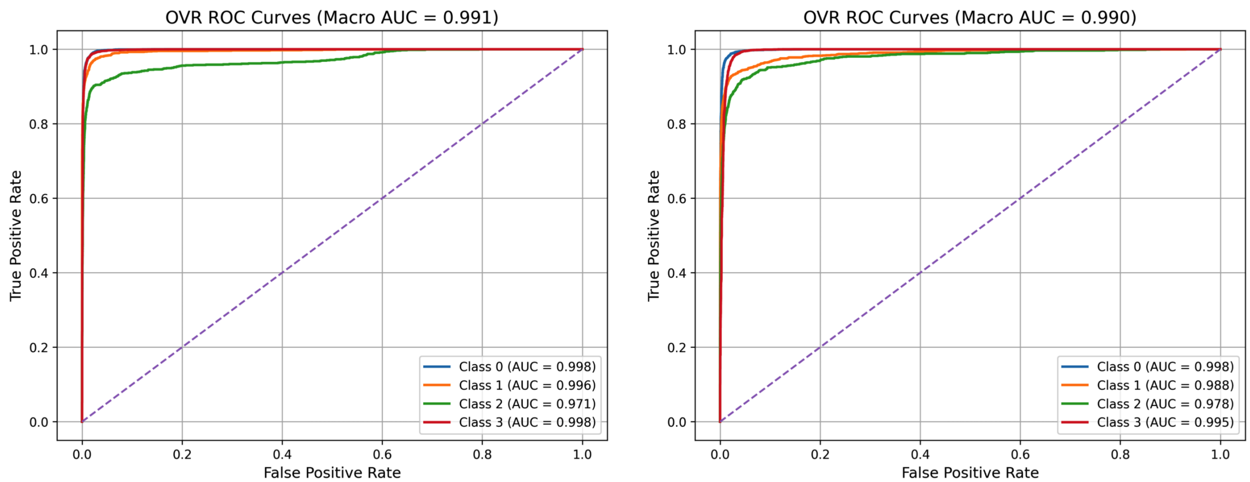


Fig 8 Example OCT images (first row) from UCSD dataset for each class of retinal disease given correspondingly on top of each image (column-wise), with abnormal region shown by a bounding box. The subsequent rows show the overlaid gradients on the input images, specifically the Grad-CAM visualizations. Overall quantitative metrics of considered methods across the test cases are shown in Table 3

5 Discussion

This study introduced a novel method for designing lightweight convolutional networks using a nonparameterized channel-rolling operation. The proposed method of generating a new set of



(a) ResNet18 (b) Rolling ResNet18

Fig 9 One-vs-Rest (OVR) Receiver Operating Characteristic (ROC) curves for OCT image classification across the four categories - Class 0 (CNV), Class 1 (DME), Class 2 (Drusen), and Class 3 (Normal) corresponding to (a) ResNet18 and (b) Rolling ResNet18.

convolutional filters from a single base filter resulted in efficient forward and backward flows for training deep learning models. Channel rolling filters facilitate the design of deep models without increasing their complexity. The shared weights across the filters reduce the classical overfitting problem in CNNs. The proposed rolling filters were easily incorporated into popular state-of-the-art architectures to analyze the performance of the proposed rolling convolution filters. As shown in Fig. 3, the UNet model's final output strongly depends on the features extracted from the initial layers. The proposed rolling convolution filters with fewer parameters can capture similar semantics from the initial layers compared to standard convolution filters (refer to Fig. 7). The relationship between the number of input and output channels in the baseline models was either $1\times$, $2\times$, or $0.5\times$. In the proposed design of the rolling filters, to generate output channels equal to the input channels ($1\times$), the rolling operations of the single-base filter must be equal to the number of input channels. For generating output channels of $2\times$ input channels, two base filters were used. The independent rolling operations performed on the two base filters were equal to the number of corresponding input channels. Similarly, to generate output channels of $0.5\times$ input channels, half the number of rolling operations were performed on a single-base filter.

Achieving a balance between model size, computational efficiency, and accuracy is a significant challenge in the design of lightweight CNNs. Although lightweight models aim to reduce resource requirements, they often sacrifice accuracy compared to larger and more complex models. The filters utilized in these models play an important role in developing tailor-made lightweight models. Existing architectural modifications, such as depth-wise separable convolutions, skip connections, and efficient building blocks, have focused on decreasing the number of operations rather than on an efficient filter design. Thus, this work is a significant step towards efficient filter design for medical imaging tasks that are less complex, with the added advantage of seamless incorporation of the proposed rolling filters into existing deep learning architectures to convert them into lightweight models. Moreover, the proposed rolling-filter-based architectures are as accurate as larger/complex models, thus addressing the major limitations in designing these lightweight models.

The performance of the proposed rolling convolution filters was statistically compared with other lightweight design approaches, namely depth-wise separable filters and flattened convolution filters, across multiple tasks using two-tailed Welch's t-tests. For all evaluation metrics and tasks, the resulting p-values were greater than 0.05, indicating no statistically significant differences between the proposed and existing lightweight designs. Importantly, this lack of statistical difference suggests performance equivalence rather than inferiority, demonstrating that the proposed approach achieves comparable performance while requiring fewer parameters. To further improve the accuracy and performance of the proposed rolling filters, the following detailed studies will be considered in future work: (1) increasing the number of independent filters at a given convolution layer, (2) introducing attention modules within the layer activations that can further boost performance, and (3) determining the optimal number of independent filters in a given layer.

6 Conclusion

This study presented a novel approach for designing lightweight convolutional neural networks using rolling convolution filters, demonstrating significant reductions in model size and parameters while maintaining comparable performance across various medical image analysis tasks. The proposed method outperforms other lightweight CNN designs in terms of parameter efficiency and can be easily integrated into existing architectures. By addressing the challenge of developing efficient deep learning models in medical imaging, the proposed method showed promise for deploying CNNs in resource-constrained settings. The successful application of rolling convolution filters to quantitative susceptibility mapping reconstruction, COVID-19 anomaly segmentation, and retinal disease classification from OCT images, highlighted its generalizability and applicability for automated medical image analysis.

Disclosures

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Code and Data Availability

COVID-19 dataset is publicly available at http://medicalsegmentation.com/covid19/. The QSM data from QSMnet[41] was made available to the authors by Prof. Lee (e-mail: jonghoyi@snu.ac.kr) of the Department of Electrical and Computer Engineering, Seoul National University. The OCT dataset[59] is publicly available at https://www.kaggle.com/datasets/paultimothymooney/kermany2018. The proposed lightweight convolution filter design code was made available as an open source for enthusiastic users at https://github.com/NaveenPaluru/Rolling-Filters

Acknowledgment

The authors thank Prof. Jongho Lee, Seoul National University, Seoul, South Korea, for providing the QSM data.

Funding Information

This work was supported by the Prime Minister's Research Fellowship (PMRF) and also the WIPRO–GE Collaborative Laboratory on Artificial Intelligence in Healthcare and Medical Imaging.

References

1 O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 234–241, Springer (2015).

2 K. He, X. Zhang, S. Ren, *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).

3 M. Gridach, "Pydinet: Pyramid dilated network for medical image segmentation," *Neural Networks* **140**, 274–281 (2021).

4 D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, *et al.*, "What is the state of neural network pruning?," *Proceedings of machine learning and systems* **2**, 129–146 (2020).

5 A. Gholami, S. Kim, Z. Dong, *et al.*, "A survey of quantization methods for efficient neural network inference," in *Low-Power Computer Vision*, 291–326, Chapman and Hall/CRC.

6 J. Gou, B. Yu, S. J. Maybank, *et al.*, "Knowledge distillation: A survey," *International Journal of Computer Vision* **129**(6), 1789–1819 (2021).

7 A. G. Howard, M. Zhu, B. Chen, *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861* (2017).

8 M. Sandler, A. Howard, M. Zhu, *et al.*, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520 (2018).

9 A. Howard, R. Pang, H. Adam, *et al.*, "Searching for mobilenetv3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1314–1324 (2019).

10 V. Vanhoucke, "Learning visual representations at scale," *ICLR invited talk* **1**(2) (2014).

11 F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258 (2017).

12 T. Zhang, G.-J. Qi, B. Xiao, *et al.*, "Interleaved group convolutions," in *Proceedings of the IEEE international conference on computer vision*, 4373–4382 (2017).

13 M. Tan and Q. V. Le, "Mixconv: Mixed depthwise convolutional kernels," in *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, 74, BMVA Press (2019).

14 H. Gao, Z. Wang, and S. Ji, "Channelnets: Compact and efficient convolutional neural networks via channel-wise convolutions," *Advances in neural information processing systems* **31** (2018).

15 X. Zhang, X. Zhou, M. Lin, *et al.*, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6848–6856 (2018).

16 N. Ma, X. Zhang, H.-T. Zheng, *et al.*, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 122–138 (2018).

17 M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 6105–6114 (2019).

18 M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*, 10096–10106, PMLR (2021).

19 J. Yu, L. Yang, N. Xu, *et al.*, "Slimmable neural networks," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net (2019).

20 Y. Bhalgat, Y. Zhang, J. M. Lin, *et al.*, "Structured convolutions for efficient neural network design," *Advances in Neural Information Processing Systems* **33**, 5553–5564 (2020).

21 Q. Qiu, X. Cheng, G. Sapiro, *et al.*, "Dcfnet: Deep neural network with decomposed convolutional filters," in *International Conference on Machine Learning*, 4198–4207, PMLR (2018).

22 Y. Li, S. Gu, L. V. Gool, *et al.*, "Learning filter basis for convolutional neural network compression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5623–5632 (2019).

23 W. Kang and D. Kim, "Deeply shared filter bases for parameter-efficient convolutional neural networks," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, *et al.*, Eds., **34**, 7397–7408, Curran Associates, Inc. (2021).

24 Z. Yang, Y. Wang, C. Liu, *et al.*, "Legonet: Efficient convolutional neural networks with lego filters," in *International Conference on Machine Learning*, 7005–7014, PMLR (2019).

25 P. Savarese and M. Maire, "Learning implicitly recurrent cnns through parameter sharing," in *International Conference on Learning Representations*, (2019).

26 Y. Yang, J. Yu, N. Jojic, *et al.*, "Fsnet: Compression of deep convolutional neural networks by filter summary," in *International Conference on Learning Representations*, (2020).

27 Y. Wang, C. Xu, C. Xu, *et al.*, "Learning versatile filters for efficient convolutional neural networks," *Advances in Neural Information Processing Systems* **31** (2018).

28 K. Han, Y. Wang, C. Xu, *et al.*, "Learning versatile convolution filters for efficient visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(11), 7731–7746 (2021).

29 K. Han, Y. Wang, Q. Tian, *et al.*, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1580–1589 (2020).

30 Y. Cheng, F. X. Yu, R. S. Feris, *et al.*, "An exploration of parameter redundancy in deep networks with circulant projections," in *Proceedings of the IEEE international conference on computer vision*, 2857–2865 (2015).

31 C. Deng, S. Liao, Y. Xie, *et al.*, "Permdnn: efficient compressed dnn architecture with permuted diagonal matrices," in *Proceedings of the 51st Annual IEEE/ACM International Symposium on Microarchitecture*, 189–202 (2018).

32 A. Araujo, B. Negrevergne, Y. Chevaleyre, *et al.*, "Training compact deep learning models

for video classification using circulant matrices," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 271–286 (2018).

33 C. Liu, W. Ding, X. Xia, *et al.*, "Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnns with circulant back propagation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2691–2699 (2019).

34 Y. Wang, C. Xu, C. Xu, *et al.*, "Beyond filters: Compact feature map for portable deep model," in *International Conference on Machine Learning*, 3703–3711, PMLR (2017).

35 J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feed-forward acceleration," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds. (2015).

36 Y. Wang, P. Spincemaille, Z. Liu, *et al.*, "Clinical quantitative susceptibility mapping (qsm): Biometal imaging and its emerging roles in patient care," *Journal of magnetic resonance imaging* **46**(4), 951–971 (2017).

37 Y. Wang and T. Liu, "Quantitative susceptibility mapping (qsm): decoding mri data for a tissue magnetic biomarker," *Magnetic resonance in medicine* **73**(1), 82–101 (2015).

38 C. Liu, W. Li, K. A. Tong, *et al.*, "Susceptibility-weighted imaging and quantitative susceptibility mapping in the brain," *Journal of magnetic resonance imaging* **42**(1), 23–41 (2015).

39 T. Liu, P. Spincemaille, L. De Rochefort, *et al.*, "Calculation of susceptibility through multiple orientation sampling (cosmos): a method for conditioning the inverse problem from measured magnetic field map to susceptibility source image in mri," *Magnetic Resonance*

in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine **61**(1), 196–204 (2009).

40 K. G. B. Rasmussen, M. Kristensen, R. G. Blendal, *et al.*, "Deepqsm-using deep learning to solve the dipole inversion for mri susceptibility mapping," *BioRxiv* , 278036 (2018).

41 J. Yoon, E. Gong, I. Chatnuntawech, *et al.*, "Quantitative susceptibility mapping using deep neural network: Qsmnet," *Neuroimage* **179**, 199–206 (2018).

42 W. Jung, J. Yoon, S. Ji, *et al.*, "Exploring linearity of deep neural network trained qsm: Qsmnet+," *Neuroimage* **211**, 116619 (2020).

43 Y. Gao, X. Zhu, B. A. Moffat, *et al.*, "xqsm: quantitative susceptibility mapping with octave convolutional and noise-regularized neural networks," *NMR in Biomedicine* **34**(3), e4461 (2021).

44 Z. Li, S. Ying, J. Wang, *et al.*, "Reconstruction of quantitative susceptibility mapping from total field maps with local field maps guided uu-net," *IEEE Journal of Biomedical and Health Informatics* **27**(4), 2047–2058 (2023).

45 W. Jung, S. Bollmann, and J. Lee, "Overview of quantitative susceptibility mapping using deep learning: Current status, challenges and opportunities," *NMR in Biomedicine* , e4292 (2020).

46 C. Langkammer, F. Schweser, K. Shmueli, *et al.*, "Quantitative susceptibility mapping: report from the 2016 reconstruction challenge," *Magnetic resonance in medicine* **79**(3), 1661–1673 (2018).

47 D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International*

Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds. (2015).

48 Y. Fang, H. Zhang, J. Xie, *et al.*, "Sensitivity of chest ct for covid-19: Comparison to rt-pcr," *Radiology* , 200432 (2020). PMID: 32073353.

49 N. Awasthi, A. Dayal, L. R. Cenkeramaddi, *et al.*, "Mini-covidnet: Efficient lightweight deep neural network for ultrasound based point-of-care detection of covid-19," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* **68**(6), 2023–2037 (2021).

50 S. Roy, W. Menapace, S. Oei, *et al.*, "Deep learning for classification and localization of covid-19 markers in point-of-care lung ultrasound," *IEEE transactions on medical imaging* **39**(8), 2676–2687 (2020).

51 J. Wang, Y. Bao, Y. Wen, *et al.*, "Prior-attention residual learning for more discriminative covid-19 screening in ct images," *IEEE Transactions on Medical Imaging* **39**(8), 2572–2583 (2020).

52 C. Li, L. Dong, Q. Dou, *et al.*, "Self-ensembling co-training framework for semi-supervised covid-19 ct segmentation," *IEEE Journal of Biomedical and Health Informatics* **25**(11), 4140–4151 (2021).

53 J. P. Kanne, "Chest ct findings in 2019 novel coronavirus (2019-ncov) infections from wuhan, china: Key points for the radiologist," (2020). PMID: 32017662.

54 M. Chung, A. Bernheim, X. Mei, *et al.*, "Ct imaging features of 2019 novel coronavirus (2019-ncov)," *Radiology* **295**(1), 202–207 (2020).

55 N. Paluru, A. Dayal, H. B. Jenssen, *et al.*, "Anam-net: Anamorphic depth embedding-based

lightweight cnn for segmentation of anomalies in covid-19 chest ct images," *IEEE Transactions on Neural Networks and Learning Systems* **32**(3), 932–946 (2021).

56 D.-P. Fan, T. Zhou, G.-P. Ji, *et al.*, "Inf-net: Automatic covid-19 lung infection segmentation from ct images," *IEEE Transactions on Medical Imaging* **39**(8), 2626–2637 (2020).

57 G. Wang, X. Liu, C. Li, *et al.*, "A noise-robust framework for automatic segmentation of covid-19 pneumonia lesions from ct images," *IEEE Transactions on Medical Imaging* **39**(8), 2653–2663 (2020).

58 M. Jun, G. Cheng, W. Yixin, *et al.*, "COVID-19 CT Lung and Infection Segmentation Dataset," (2020).

59 D. S. Kermany, M. Goldbaum, W. Cai, *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell* **172**(5), 1122–1131 (2018).

60 Y. Rong, D. Xiang, W. Zhu, *et al.*, "Surrogate-assisted retinal oct image classification based on convolutional neural networks," *IEEE Journal of Biomedical and Health Informatics* **23**(1), 253–263 (2019).

61 L. Fang, C. Wang, S. Li, *et al.*, "Attention to lesion: Lesion-aware convolutional neural network for retinal optical coherence tomography image classification," *IEEE transactions on medical imaging* **38**(8), 1959–1970 (2019).

62 N. Paluru, H. Ravishankar, S. Hegde, *et al.*, "Self distillation for improving the generalizability of retinal disease diagnosis using optical coherence tomography images," *IEEE Journal of Selected Topics in Quantum Electronics* **29**(4: Biophotonics), 1–12 (2023).

63 F. Li, H. Chen, Z. Liu, *et al.*, "Deep learning-based automated detection of retinal diseases

using optical coherence tomography images," *Biomedical Optics Express* **10**(12), 6204–6226 (2019).

64 A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 8026–8037 (2019).

65 J. Zhang, Z. Liu, S. Zhang, *et al.*, "Fidelity imposed network edit (fine) for solving ill-posed image reconstruction," *Neuroimage* **211**, 116579 (2020).

66 K.-W. Lai, M. Aggarwal, P. van Zijl, *et al.*, "Learned proximal networks for quantitative susceptibility mapping," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 125–135, Springer (2020).

67 O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, *LNCS* **9351**, 234–241, Springer (2015). (available on arXiv:1505.04597 [cs.CV]).

68 F. N. Iandola, S. Han, M. W. Moskewicz, *et al.*, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360* (2016).

69 R. R. Selvaraju, M. Cogswell, A. Das, *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 618–626 (2017).

Naveen Paluru received his Ph.D. in Medical Image Analysis from the Department of Computational and Data Sciences, Indian Institute of Science, Bangalore. He was awarded the Best Ph.D. Thesis Medal for the year 2024, and his doctoral research was funded by the Prime

Minister's Research Fellowship. He is currently a Senior AI Scientist at the Science and Technology Organization, GE HealthCare, India. His research interests lie at the intersection of artificial intelligence, signal processing, and medical imaging.

 Mehak Arora holds an MS in Electrical and Computer Engineering from Georgia Institute of Technology. She is currently pursuing a PhD in Computer Engineering at Duke University, where she is building prognostic risk estimations models from multimodal, longitudinal clinical data. Her research interests lie at the intersection of machine learning and clinical informatics.

 Phaneendra Yalavarthy holds a Ph.D. from Dartmouth College and leads TANUH, the AI-Centre of Excellence in Healthcare and is also a Professor of Computational and Data Sciences at IISc Bangalore. Awards include INAE Young Engineer (2013), and S. Ramachandran-National Bioscience Award (2020). His research interests incudes computational/deep learning in medical imaging, physiological signals, and non-destructive imaging. A senior IEEE/OSA/SPIE member and serves as an associate editor of IEEE Transactions on Medical Imaging.

List of Figures

4 Representative susceptibility reconstruction results (sagittal view) across orientation 1 of considered methods (column-wise) on a sample test volume. The high-frequency error component (HFEN(%) $\downarrow$) with reference to the gold standard COSMOS was provided for each slice. Overall quantitative metrics of considered methods across the test volumes are shown in Table 1.

5 Error maps of the QSM reconstructions shown in Fig. 4, computed with respect to the COSMOS reconstructions as reference. Error maps are shown for all the compared QSM methods. The comparison between the baseline QSMnet and the proposed Rolling QSMnet is highlighted, illustrating differences in spatial error distribution and overall agreement with COSMOS.

6 Representative segmentation results on a patient volume (test cases). The input slices (test cases) are shown in the first column. The respective annotations (ground truth) are presented in the second column. The predictions of the UNet and Rolling UNet are presented in the fourth and fifth columns, respectively. Abnormalities in the lung region are indicated in red, and the normal lung region is indicated in green. The average Dice Similarity scores are given below the corresponding slices. Overall quantitative metrics of considered methods across the test cases are shown in Table 2.

7 Feature representations of CT images from the initial layers of UNet driven architectures while segmenting COVID-19 anomalies. The input CT images are presented in the first column. The final predictions are presented in the fifth column. The features obtained with standard convolutions (UNet) and from rolling convolutions (Rolling UNet) are shown in the first and second rows, respectively. (a) clearly distinguishes the lung region from the background (through edge extraction), (b) most of the neurons fired for the abnormal region (shown in red color), and (c) activated neurons focused on the normal/healthy tissue (shown in green color) of the lung.

8 Example OCT images (first row) from UCSD dataset for each class of retinal disease given correspondingly on top of each image (column-wise), with abnormal region shown by a bounding box. The subsequent rows show the overlaid gradients on the input images, specifically the Grad-CAM visualizations. Overall quantitative metrics of considered methods across the test cases are shown in Table 3

9 One-vs-Rest (OVR) Receiver Operating Characteristic (ROC) curves for OCT image classification across the four categories - Class 0 (CNV), Class 1 (DME), Class 2 (Drusen), and Class 3 (Normal) corresponding to (a) ResNet18 and (b) Rolling ResNet18.

List of Tables