frontiers | Frontiers in Medicine

ORIGINAL RESEARCH
published: 31 May 2022
doi: 10.3389/fmed.2022.894430

Building Efficient CNN Architectures for Histopathology Images Analysis: A Case-Study in Tumor-Infiltrating Lymphocytes Classification

*André L. S. Meirelles[1], Tahsin Kurc[2], Jun Kong[3], Renato Ferreira[4], Joel H. Saltz[2] and George Teodoro[1,4]**

[1] Department of Computer Science, Universidade de Brasília, Brasília, Brazil, [2] Biomedical Informatics Department, Stony Brook University, Stony Brook, NY, United States, [3] Department of Mathematics and Statistics and Computer Science, Georgia State University, Atlanta, GA, United States, [4] Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

OPEN ACCESS

Edited by:
Dachuan Zhang, First People's Hospital of Changzhou, China

Reviewed by:
Guotai Wang, University of Electronic Science and Technology of China, China
Arkadiusz Gertych, Cedars Sinai Medical Center, United States

***Correspondence:**
George Teodoro
george@dcc.ufmg.br

Specialty section:
This article was submitted to Pathology, a section of the journal Frontiers in Medicine

Received: *11 March 2022*
Accepted: *11 May 2022*
Published: *31 May 2022*

Citation:
Meirelles ALS, Kurc T, Kong J, Ferreira R, Saltz JH and Teodoro G (2022) Building Efficient CNN Architectures for Histopathology Images Analysis: A Case-Study in Tumor-Infiltrating Lymphocytes Classification. Front. Med. 9:894430. doi: 10.3389/fmed.2022.894430

Background: Deep learning methods have demonstrated remarkable performance in pathology image analysis, but they are computationally very demanding. The aim of our study is to reduce their computational cost to enable their use with large tissue image datasets.

Methods: We propose a method called Network Auto-Reduction (NAR) that simplifies a Convolutional Neural Network (CNN) by reducing the network to minimize the computational cost of doing a prediction. NAR performs a compound scaling in which the width, depth, and resolution dimensions of the network are reduced together to maintain a balance among them in the resulting simplified network. We compare our method with a state-of-the-art solution called ResRep. The evaluation is carried out with popular CNN architectures and a real-world application that identifies distributions of tumor-infiltrating lymphocytes in tissue images.

Results: The experimental results show that both ResRep and NAR are able to generate simplified, more efficient versions of ResNet50 V2. The simplified versions by ResRep and NAR require 1.32× and 3.26× fewer floating-point operations (FLOPs), respectively, than the original network without a loss in classification power as measured by the Area under the Curve (AUC) metric. When applied to a deeper and more computationally expensive network, Inception V4, NAR is able to generate a version that requires 4× lower than the original version with the same AUC performance.

Conclusions: NAR is able to achieve substantial reductions in the execution cost of two popular CNN architectures, while resulting in small or no loss in model accuracy. Such cost savings can significantly improve the use of deep learning methods in digital pathology. They can enable studies with larger tissue image datasets and facilitate the use of less expensive and more accessible graphics processing units (GPUs), thus reducing the computing costs of a study.

Keywords: digital pathology, deep learning, CNN simplification, tumor-infiltrating lymphocytes, efficient CNNs

1. INTRODUCTION

Pathology image analysis is quickly evolving thanks to advances in scanner technologies that now enable rapidly digitizing glass slides into high resolution whole slide images (WSIs). This has also been followed by several developments in computer aided diagnosis analysis tools and methods, which have improved the use of information computed from tissue characteristics in WSIs in disease classification, prediction of clinical outcomes, etc. (1–3). Deep learning methods have demonstrated significant improvements over traditional machine learning and other image analysis methods in a wide range of tissue image analysis tasks (4–10). Consequently, deep learning-based image analysis is rapidly becoming a mainstream approach in digital pathology.

The advances attained with the deep learning methods have also been accompanied by multiple challenges in order to make them more routinely used in pathology image analysis. For instance, these methods require a significant amount of annotated data to be used in training, which is particularly costly in digital pathology as it requires an expert pathologist to manually annotate large volumes of data (11, 12). Also, applications developed with deep learning should consider explainability to improve confidence in their use (13, 14).

We address another challenge with application of deep learning in digital pathology; the high computational cost of deep learning inference, which has adversely impacted the effective use of deep learning in many application domains (15). This problem is particularly more pronounced in digital pathology because WSIs are extremely high resolution images (in the range of 100K×100K pixels). A study analyzing thousands of WSIs would require substantial computing capacity. High computing requirements can significantly limit the use of deep learning in research and as a routine component of digital pathology workflows.

The demanding computational costs of deep learning models can be addressed by CNN simplification and acceleration techniques, such as: network pruning (16–18), sparsification (19, 20), quantization (21, 22), etc. Among network pruning solutions, there are those that concentrate on removing filters in the convolutional layers, which are referred to as channel or filter pruning (23–25). Other techniques act on a broader range of structures, removing full layers or even blocks of layers (26).

Network pruning solutions have been the focus of a number of publications, presenting good results in CNN speedup and also enabling lossless model compression (27). Filter pruning techniques and network pruning in general offer varying possibilities to select which filters from which layers should be excluded from the network or which structures to be removed. However, this is not performed in a balanced manner taking into consideration all model dimensions together, which may limit the performance and accuracy of the reduced network (28–30).

In this work, we present a novel approach that can generate more efficient Convolutional Neural Network (CNN) architectures to speed up the execution of model training and inference. Our approach, called Network Auto-Reduction (NAR), performs transformations in a given CNN architecture in order to reduce its width, depth, and resolution dimensions (also called components) to generate a novel architecture with the desired computational cost (in terms of number of FLOPs) and with minimal loss of accuracy. This simplification employs a compound scaling method with a set of fixed scaling coefficients. The goal is to maintain a balance among the components of the network—for instance, a larger input resolution would require more receptive fields and a larger number of channels to capture details of the input image as is theoretically shown in (28). NAR differs from most of the previous works that focus on reducing a single or a couple of the dimensions of the network (25, 27, 29, 31–33).

We experimentally evaluate our approach in a real-world application that classifies tumor-infiltrating lymphocytes (TILs) in WSIs (34, 35) (presented in Section 2.1). TILs are a type of white blood cells in the immune system, whose patterns found in the tissue images have been shown to have consistent correlations with patient overall survival in multiple cancer types (36–40). In our evaluation, we use ResNet50 V2 and Inception V4 as full, baseline networks and simplify them with NAR. We compare NAR to a state-of-the-art method, called ResRep (27). ResRep is designed to carry out lossless channel pruning (filter pruning) to slim down a CNN through a reduction in the width or number of output channels of convolutional layers. The experimental evaluation shows that NAR can generate CNNs with demands up to 4× lower than the original CNN, while delivering the same classification quality (AUC). The simplified networks generated by NAR are more efficient, with smaller requirements for the same AUC values when compared with the networks generated by ResRep.

The rest of this document is organized as follows: Section 2 presents the motivating TIL classification application, the NAR strategy proposed here and summarizes the ResRep approach. Section 3 shows the performance evaluation in detail and Section 4 discusses the main finds and promising directions for future work.

2. MATERIALS AND METHODS

2.1. Tumor-Infiltrating Lymphocytes (TIL) Classification Using Deep Learning

This work is motivated by analyses carried with deep learning models of WSIs to identify and classify spatial patterns of TILs (34, 35). There is increasing evidence that TIL patterns in cancer tissue correlate with clinical outcomes; for example, high densities of TILs indicate favorable outcomes, such as longer survivals for patients (37). Quantitative analyses of TIL patterns can provide valuable information about interactions between cancer and immune system and novel bio-markers for prediction of prognosis and treatment response.

WSIs allow a researcher to carry out quantitative investigations of the tumor microenvironment at the subcellular level. This has motivated the development of image analysis methods to extract and characterize quantitative imaging features from WSIs (1–3, 41, 42).

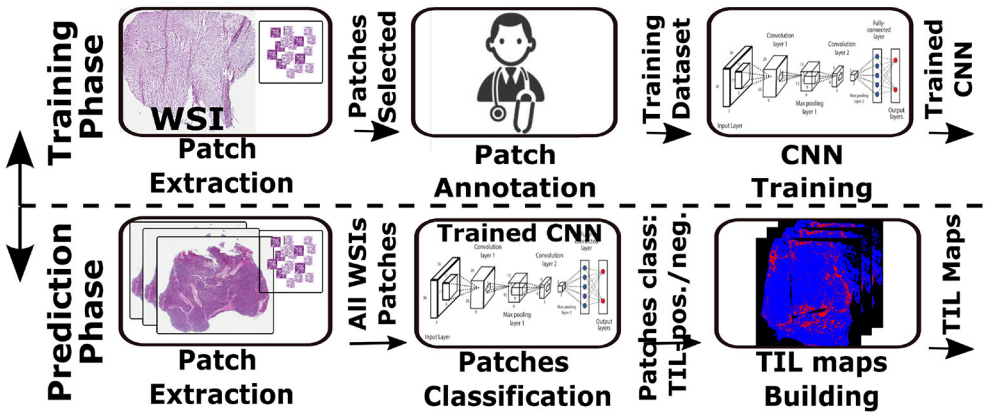

FIGURE 1 | Use-case TIL analysis workflow. CNN is trained to identify TIL rich tissue based on patches annotated by expert pathologist (top). The CNN model is then used to classify input WSI in a patch basis. The result is a TIL map presenting TIL rich regions in the input tissue.

Deep learning methods based on Convolutional Neural Networks (CNNs) have emerged as an effective approach for image analysis in several domains. CNNs have been employed for a variety of tissue image analysis tasks, including object identification, segmentation, and recognition of spatial patterns (34, 43–49).

Figure 1 shows a TIL analysis pipeline, based on the work done in (34), that predicts distributions of TILs in images of hematoxylin and eosin (H&E) stained tissue specimens. In this pipeline, an input image is partitioned into small patches—the size of a patch is 50 × 50 square microns in our application. A CNN classification model classifies the patches into TIL-positive and TIL-negative classes (a binary classification operation). As is shown in the figure, the pipeline is composed of a training phase and a prediction phase. In the training phase (shown in the top), the CNN learns to classify input image patches. In this process, patches are extracted from multiple WSIs, pathologists review and annotate them, and the CNN classification model is trained. The selection of patches and model training is repeated until the desired accuracy level is reached. The prediction phase (bottom part of the image) applies the trained model to input patches from unseen WSIs to compute TIL maps that identify tissue regions with TILs—TIL-positive patches are shown as Red dots on a Blue background, which represents tissue.

While CNNs have been applied successfully for TIL analysis (34, 35), scaling the analysis to thousands of WSIs is challenging, because of the CNNs high computational cost. This poses a major limitation to a broader adoption of CNN-based methods in the digital pathology domain. We propose a method that intelligently simplifies a CNN to reduce its computational cost while minimizing loss of model accuracy. The proposed method is discussed in the next section.

2.2. Network Auto-Reduction (NAR)

We propose Network Auto-Reduction (NAR) to simplify CNNs and reduce their execution cost in the inference (prediction) phase. Several approaches have been proposed for CNN simplification. Most of the prior approaches aim to reduce one of the dimensions of the CNN: depth, width or input resolution (27). Some studies proposed removing specific CNN filters (25, 29, 31–33), or introducing weight sparsity (18) or applying a combination of both (26, 27). In most of those cases, the CNN is re-trained multiple times while the reduction operations are iteratively applied. This is computationally expensive and may not even be feasible in applications that employ large training datasets.

NAR simplifies a CNN by modifying the depth, width, and input resolution of the model together. The goal is to maintain a balance between network building blocks in order for the simplified CNN to attain good accuracy, as demonstrated in previous work (28, 50, 51). The compound simplification process is illustrated in **Figure 2**.

Our method is inspired by the approach proposed by Tan et al. (52) to scale up simple CNNs. Their method was designed to increase the size of a simple CNN in order to improve its prediction performance. Here, on the other hand, we address the problem of simplifying a CNN that is already known to perform well in the target domain, but has a high computation cost. Tan et al. (52) formulated the problem of scaling-up a CNN as an optimization problem defined in Equation (1), given a target memory consumption (TM) and (TF):

$$\begin{aligned} &max_{d,w,r} \;\; Accuracy(M(d,w,r)) \\ &\quad s.t \;\; M(d,w,r) = \bigodot_{i=1,\ldots,s} \hat{\mathcal{F}}_i^{d.\hat{L}_i} \left(X_{\langle r.\hat{H}^i, r.\hat{W}^i, w.\hat{C}^i \rangle} \right) \\ &Memory(M) \leq TM; \\ &\quad (M) \leq TF, \end{aligned} \tag{1}$$

Here, $\bigodot_{i=1,\ldots,s}$ is the composition of the layers of a given CNN M. Each layer i can be viewed as the application of function $\hat{F}_i$ on its input tensor X_i, with dimensions $\hat{H}^i, \hat{W}^i, \hat{C}^i$ (height, width, channels). The layers can be repeated in a sequence of $\hat{L}_i$ occurrences. The transformation process changes all three components of a network simultaneously, *depth* (the number of layers $\hat{L}_i$), *width* (the number of channels $\hat{C}_i$) and *resolution* (the

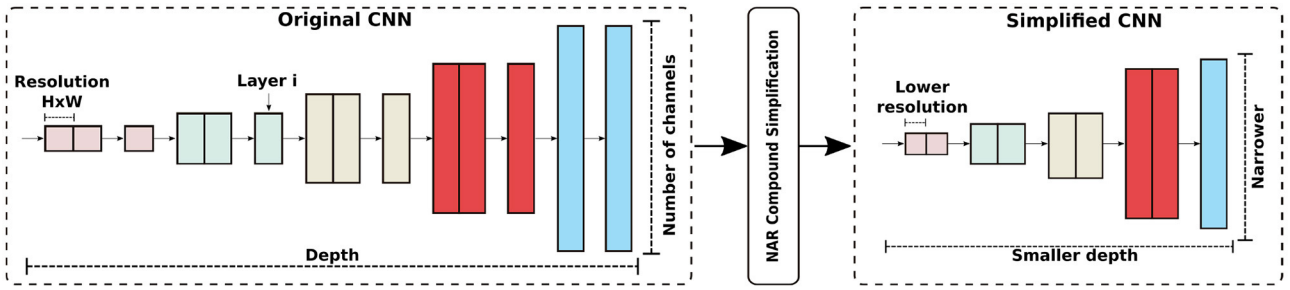

FIGURE 2 | NAR compound CNN simplification modifies depth, width, and input resolution in order to have a balance among CNN components.

height $\hat{H}_i$ and width $\hat{W}_i$ of tensor X_i) in a balanced way. The scaling coefficients d, w, r used by Tan et al., enabled creating a bigger network M with $d.\hat{L}^i$ occurrences of layer i and input size $r.\hat{H}^i, r.\hat{W}^i, w.\hat{C}^i$, except for layer $i = 0$, in which the input dimensions are the same as the input image dimensions and channels. For given values of d, w, r, the cost of the scaled-up CNN is increased proportionally to $d.w^2.r^2$.

According to Tan et al., it is critical to balance the scaling coefficients in order to obtain the best accuracy/efficiency relation for a given resource constraint. To that end, a uniform compound scaling strategy is used to distribute the cost increase among these parameters through a ϕ coefficient, such that $d = \alpha^{\phi}, w = \beta^{\phi}$, and $r = \gamma^{\phi}$ with a restriction that $\alpha.\beta^2.\gamma^2 \approx 2$. The values of α, β, and γ that produce the best accuracy are determined by a model grid search (52).

In NAR, we apply a reduction factor to each CNN component such that $d = \alpha^{-\phi}, w = \beta^{-\phi}, and\ r = \gamma^{-\phi}$ with the same restriction valid for α, β, and γ. This results in a theoretical reduction of $\frac{1}{2^{\phi}}$ for every value of ϕ. Therefore, NAR generates reduced versions of any block based CNN.

2.3. ResRep

ResRep (27) is a state-of-the-art CNN pruning strategy that uses structural re-parameterization to reduce a network's width. It implements a two step solution, referred to as *remembering* and *forgetting* steps inspired by neurobiology research. In the remembering step, the network is trained with the addition of *compactor* layers attached to the original convolutional layers. The goal is to identify filters that contribute little to the learning process. The compactors are 1×1 convolutional layers that apply gradient penalties, making some channels' gradients approach zero. The forgetting step is executed after the remembering step and reconstructs the original model based on the compactor trained network, but without some channels.

A key feature of ResRep is the mechanism by which channels are selected to be removed from the original network. The selection process uses a "gradient resetting" scheme, applied to the compactors' gradients only. A group Lasso penalty is used in conjunction with the training objective function to produce a channel-wide sparsity. The gradient resetting operation is formulated in Equation (1).

$$L_{total}(X, Y, \Theta) = L_{perf}(X, Y, \Theta) + \lambda P(K) \quad (2)$$

$$G(\mathbf{F}) = \frac{\partial L_{total}(X, Y, \Theta)}{\partial \mathbf{F}} \leftarrow \frac{\partial L_{perf}(X, Y, \Theta)}{\partial \mathbf{F}} * m + \lambda \frac{\mathbf{F}}{||\mathbf{F}||_E}. \quad (3)$$

Here, L_{total} is the objective function applied to input X with label Y, given current network weights Θ. The λ is a penalty strength factor and $P(K)$ is the Lasso penalty added to the regular cost function L_{perf}. The gradients for each filter ($\mathbf{F}$) of the convolutional layer may be zeroed with a binary mask m. The final gradient $G(\mathbf{F})$ is compared to a threshold value (ϵ). If it is below the threshold, the filter is removed. It is expected that $G(\mathbf{F})$ will be close to zero for filters for which the binary mask m is 0, since only the penalties are considered.

3. RESULTS

The network cost reduction techniques were evaluated with the TIL classification application described in Section 2.1 and two popular CNN architectures, ResNet50 V2 (53) and Inception V4 (54)—the two CNNs had been successfully employed for whole slide image analysis in a previous work (35). The CNNs were trained with 4,300 image patches extracted from a set of 56 WSIs from 10 tumor tissue types, including breast, prostate and pancreatic cancer, in The Cancer Genome Atlas (TCGA) repository (55). Fifteen thousand patches extracted from another set of 5 WSIs comprised the test dataset. The full list of the WSIs is given in **Supplementary Tables 3, 4**, which also includes the percentage of TIL positive patches in each WSI. The images were downloaded in their native Aperio SVS file format. SVS files have a hierarchical representation that stores multiple resolutions of the same image. We used the highest resolution available for each WSI. If an image is obtained at 40x or 20x magnifications, the physical dimensions of a pixel are 0.25×0.25 μm or 0.5×0.5 μm, respectively. We employed the OpenSlide library (http://openslide.org/formats/aperio/) to read the images and extract patches. The images along with their TIL classification (Map) are publicly available (https://cancerimagingarchive.net/datascope/TCGA_TilMap/).

TABLE 1 | Number of parameters and layers organization in original ResNet50 V2 and NAR simplified networks.

CNN	ORIGINAL	NAR $\phi = 1$	NAR $\phi = 2$	NAR $\phi = 3$
ResNet50 V2 (53)				
# Params	23,568,898	14,583,140	11,274,413	8,514,988
Conv 1	7×7, 64, stride 2			
Stage 1	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 58 \\ 3 \times 3, 58 \\ 1 \times 1, 232 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 53 \\ 3 \times 3, 53 \\ 1 \times 1, 212 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 48 \\ 3 \times 3, 48 \\ 1 \times 1, 192 \end{bmatrix} \times 2$
Stage 2	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 106 \\ 3 \times 3, 106 \\ 1 \times 1, 424 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 116 \\ 3 \times 3, 116 \\ 1 \times 1, 464 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 96 \\ 3 \times 3, 96 \\ 1 \times 1, 384 \end{bmatrix} \times 2$
Stage 3	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 233 \\ 3 \times 3, 233 \\ 1 \times 1, 932 \end{bmatrix} \times 5$	$\begin{bmatrix} 1 \times 1, 212 \\ 3 \times 3, 212 \\ 1 \times 1, 848 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 192 \\ 3 \times 3, 192 \\ 1 \times 1, 768 \end{bmatrix} \times 3$
Stage 4	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 465 \\ 3 \times 3, 465 \\ 1 \times 1, 1860 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 423 \\ 3 \times 3, 423 \\ 1 \times 1, 1692 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 385 \\ 3 \times 3, 385 \\ 1 \times 1, 1540 \end{bmatrix} \times 2$

The parameter count considers a binary classification problem.

In all of the original and simplified CNN configurations, an input image patch covers a tissue area of $50 \times 50 \mu$m, which was resized to the expected input image size of each CNN. The number of patches that a CNN has to process to analyze a WSI is the same as the other CNNs, regardless of the input size required by each CNN.

The deep learning models were trained and tested on a machine running Linux, equipped with 2 Intel Xeon Gold 6248 "Cascade Lake" CPUs (with 20 cores each), 512 GB of DDR4 RAM, and an NVIDIA Tesla V100 GPU with 32 GB of dedicated memory. In all of the experiments, the models were trained from scratch for a varying number of epochs (50 for NAR and 180 for ResRep, which requires a larger number of epochs to simplify the CNN) using Adam optimization algorithm, a learning rate of 0.0005, and weight decay of 0.0005. StepLR was used as learning rate scheduler for ResRep, with step size of 5 epochs and gamma as 0.5 (learning rate reduction factor). In addition to NAR and ResRep, we have also evaluated a reduction strategy in which only the input image is reduced. This strategy is called input reduction (IR). With IR, we evaluated the impact of the compound reduction implemented by NAR against input data reduction only. The IR strategy results in smaller feature maps in memory but does not require changes to the CNN architecture, which remains exactly the same as the original.

The classification performances of the models trained with the simplified CNNs generated by ResRep and NAR were evaluated using the Area Under the ROC Curve (AUC) metric, the values of which were computed as the mean of values from 3 runs. The values of $\alpha = 1.2, \beta = 1.1$ and $\gamma = 1.15$ used here that lead to the best performance were determined using a grid search (52). The execution cost of each model was measured in terms of the number of Giga- (G) required to process a given input patch covering an area of $50 \times 50 \mu$m. The total count considers both convolutional and dense layers, given, respectively, by the relations $F_{conv} = 2 * Number\ of\ channels * Kernel\ shape * Output\ shape$ and $F_{dense} = 2 * Input\ size * Output\ size$. The NAR codes were developed using Keras and Tensorflow, while ResRep was implemented with PyTorch.

3.1. Simplification of ResNet50 V2 by NAR, ResRep, and IR

This set of experiments compare NAR, ResRep, and the Input Reduction (IR) approaches in simplifying the ResNet50 V2. The value of ϕ in NAR was varied between 1 and 3. Values greater than 3 generated simplified architectures that were purely sequential models that did not resemble the original model at all. Moreover, $\phi = 3$ resulted in significant drop in classification performance.

The simplified CNNs generated by different configurations of NAR and ResRep are summarized in **Tables 1**, **2**, respectively. As is shown in **Table 1**, NAR reduces multiple components of the network; this is illustrated by different number of blocks in each stage and different filter quantity in each convolutional layer. ResRep, on the other hand, primarily prunes the filters in the last stages of the network. In **Table 2**, *P* marks positions where filters have been pruned. The filters in stage 2 are not pruned until $\epsilon = 0.90$ and no filters are pruned in stage 1.

Table 3 shows the computational requirements and classification performances of the models generated from the simplified networks. NAR with $\phi = 2$ generated a network with 70% reduction in computational requirements compared to the original network. Additionally, the AUC value obtained by the simplified model is the same as that achieved by the original model. ResRep also was able to generate simplified networks with no loss of AUC performance. However, as is shown in the table, these networks had higher computational requirements than the networks generated by NAR. Further, the IR strategy achieved competitive results as compared to ResRep, although it is a relatively simple approach. NAR has attained an overall better performance (smaller G) than IR for the same AUC. Further, it is

TABLE 2 | Number of parameters and layers for the ResRep reduced networks (binary classification).

CNN	$\epsilon = 0.82$	$\epsilon = 0.84$	$\epsilon = 0.86$	$\epsilon = 0.88$	$\epsilon = 0.90$	$\epsilon = 0.92$	$\epsilon = 0.94$
ResNet50 V2 (53)							
# Params	12,527,836	9,421,008	8,663,740	9,225,475	7,931,287	4,882,052	4,696,612
Conv 1	1×1, 64	1 × 1, 64	1 × 1, 64	7 × 7, 64, stride 2	1 × 1, 64	1 × 1, 64	1 × 1, 64
Stage 1	3 × 3, 64 × 3 1 × 1, 256 1 × 1, 128	3 × 3, 64 × 3 1 × 1, 256 1 × 1, 128	3 × 3, 64 × 3 1 × 1, 256 1 × 1, 128	3 × 3, 64 × 3 1 × 1, 256 1 × 1, 128	3 × 3, 64 × 3 1 × 1, 256 1 × 1, 128	3 × 3, 64 × 3 1 × 1, 256 1 × 1, *P*	3 × 3, 64 × 3 1 × 1, 256 1 × 1, *P*
Stage 2	3 × 3, 128 × 4 1 × 1, 512 1 × 1, *P*	3 × 3, 128 × 4 1 × 1, 512 1 × 1, *P*	3 × 3, 128 × 4 1 × 1, 512 1 × 1, *P*	3 × 3, 128 × 4 1 × 1, 512 1 × 1, *P*	3 × 3, *P* × 4 1 × 1, 512 1 × 1, *P*	3 × 3, *P* × 4 1 × 1, 512 1 × 1, *P*	3 × 3, *P* × 4 1 × 1, 512 1 × 1, *P*
Stage 3	3 × 3, *P* × 6 1 × 1, 1024 1 × 1, *P*	3 × 3, *P* × 6 1 × 1, 1024 1 × 1, *P*	3 × 3, *P* × 6 1 × 1, 1024 1 × 1, *P*	3 × 3, *P* × 6 1 × 1, 1024 1 × 1, *P*	3 × 3, *P* × 6 1 × 1, 1024 1 × 1, *P*	3 × 3, *P* × 6 1 × 1, 1024 1 × 1, *P*	3 × 3, *P* × 6 1 × 1, 1024 1 × 1, *P*
Stage 4	3 × 3, *P* × 3 1 × 1, 2048	3 × 3, *P* × 3 1 × 1, 2048	3 × 3, *P* × 3 1 × 1, 2048	3 × 3, *P* × 3 1 × 1, 2048	3 × 3, *P* × 3 1 × 1, 2048	3 × 3, *P* × 3 1 × 1, 2048	3 × 3, *P* × 3 1 × 1, 2048

In each reduction level, P indicates the block position where channels were pruned.

TABLE 3 | AUC, Giga- (G) correspondent to model input size, number of parameter layers, and total of model layers of ResNet50 V2 and simplified networks by ResRep, IR, and NAR.

CNN	AUC	G	Input size	Param. layers	# of layers
ResNet50 V2 (53)	0.86	9.65	240 × 240	50	225
ResNet ResRep $\epsilon = 0.82$	0.87	8.34			
ResNet ResRep $\epsilon = 0.84$	0.82	7.91			
ResNet ResRep $\epsilon = 0.86$	0.84	7.63			
ResNet ResRep $\epsilon = 0.88$	0.81	7.68	240 × 240	50	225
ResNet ResRep $\epsilon = 0.90$	**0.86**	**7.27**			
ResNet ResRep $\epsilon = 0.92$	0.69	6.10			
ResNet ResRep $\epsilon = 0.94$	0.73	6.09			
ResNet50 V2 IR 1	0.88	7.90	209 × 209		
ResNet50 V2 IR 2	0.88	5.83	181 × 181		
ResNet50 V2 IR 3	**0.86**	**4.24**	157 × 157	50	225
ResNet50 V2 IR 4	0.84	3.49	137 × 137		
ResNet50 V2 IR 5	0.81	2.56	119 × 119		
ResNet50 V2 IR 6	0.79	2.03	104 × 104		
ResNet NAR $\phi = 1$	0.84	5.15	209×209	42	170
ResNet NAR $\phi = 2$	**0.86**	**2.96**	181×181	36	160
ResNet NAR $\phi = 3$	0.80	1.53	157× 157	30	134

Bold values are those with good quality/performance trade offs.

noticeable that when the input image is reduced bellow a certain size (e.g., 119×119), the AUC of IR is significantly impacted.

An interesting configuration of ResRep occurred when ϵ was set to 0.90. The computational requirements of the simplified network was 75.0% of that of the original network, and the simplified network attained an equivalent AUC level. However, when a higher simplification value was used, there was a significant drop in AUC. For the same AUC values (e.g., 0.86), NAR generated CNNs with smaller computational requirements.

3.2. NAR and IR Performance for the Inception V4 CNN

This set of experiments measures the performance of NAR and IR with Inception V4 (54). The Inception is a deeper network than ResNet50 V2 and has a higher computational cost, thus it is another interesting case for evaluating our approach. We unfortunately have not been able to use ResRep to simplify the Inception. This CNN has a more complex architecture with multiple shortcuts and the ResRep code/documentation available does not implement Inception neither it provides clear directions on how to apply the method to other complex architectures (27).

The results of the NAR simplified networks as the ϕ parameter is varied are shown in **Table 4**. First, it is noticeable that the original Inception showed a better classification performance as compared to ResNet (0.92 vs. 0.87). As compared to the IR strategy, NAR has again attained better performance for the same AUC level. Once again, for the best AUC score of each strategy and 0.87 AUC values, NAR requires, respectively, about 2.35× and 4.93× less FLOPs to compute an inference. These observations once again show the importance of a balanced compound network reduction as performed by NAR.

TABLE 4 | AUC, Giga-FLOPs (GFLOPs) correspondent to input sizes, number of parameter layers, and total layers of Inception V4 and simplified networks produced by NAR.

CNN	AUC	G	Input size	Param. layers	# of layers
Inception V4 (54)	0.92	15.48	240 × 240	245	861
Inception IR 1	**0.91**	**9.80**	209 × 209		
Inception IR 2	0.89	6.64	181 × 181		
Inception IR 3	0.88	4.79	158 × 158	245	861
Inception IR 4	0.87	2.76	137 × 137		
Inception IR 5	0.86	1.92	119 × 119		
Inception IR 6	0.77	1.14	104 × 104		
Inception NAR $\phi = 1$	**0.91**	**8.14**	209 × 209	206	723
Inception NAR $\phi = 2$	**0.92**	**4.17**	181 × 181	179	627
Inception NAR $\phi = 3$	0.90	2.21	158 × 158	145	507
Inception NAR $\phi = 4$	0.88	1.02	137 × 137	123	429
Inception NAR $\phi = 5$	0.87	0.56	119 × 119	101	351
Inception NAR $\phi = 6$	0.84	0.28	104 × 104	91	315

Bold values are those with good quality/performance trade offs.

Further, the NAR simplified version had a far better trade-off in terms of the GFLOPs required to attain a certain AUC when compared to ResNet. For instance, NAR $\phi = 5$ reached an AUC of 0.87 with only 0.56 GFLOPs. A comparable performance level required at least 8.34 GFLOPs and 2.96 GFLOPs with the simplified ResNet networks, respectively, with ResRep and NAR. The results also show that the simplified Inception V4 can sustain the same AUC level as the original network with a computational cost reduction of about 4× (NAR $\phi = 2$).

4. DISCUSSION

Overall, the experimental evaluation shows that it is possible to simplify a classification CNN to reduce its computational requirements in the inference phase, while maintaining model performance comparable to the original CNN. The ResNet50 models generated by ResRep with $\epsilon = 0.90$ and by NAR with $\phi = 2$ practically achieved the same AUC scores as the models from the original ResNet50 V2 network and were computationally 1.32× and 3.26× cheaper, respectively. Our method, NAR, produced more efficient networks than ResRep. We attribute this improvement to the fact that NAR employs an approach that simplifies the multiple components of a network in a more balanced manner. The analysis of the shape structure of the simplified CNNs with both methods (shown in **Tables 1**, **2**) highlights the main differences among their simplification strategies. ResRep mainly modified the latest layers of the CNNs, while NAR carried out a more homogeneous simplification over all of the network stages. Previous work (28, 50, 51) demonstrated that such a balance among the CNN components is important to maximize classification quality. The better compromises of NAR vs. IR strategy also demonstrate in practice that the compound reduction performed by the first is important into maximizing AUC while reducing the FLOPs demand.

This observation aligns well with the goal of our NAR method, which is to modify the width, depth, and input resolution components of a network together and in a simple way. Additionally, NAR is easier to use, requiring few alterations to an original network, without the need to change the training dynamics. ResRep, on the other hand, is harder to use as it requires changing the network with extra layers and also includes new CNN training penalties etc. This is even harder with deeper CNNs that are becoming more popular.

In the experiments with Inception V4, which is a deeper network than ResNet50, we observed that the original Inception V4 has achieved overall better AUC than the original ResNet50, but it was about 1.6× more expensive. The simplified version generated by NAR with $\phi = 5$ achieved an AUC value of 0.87, which is comparable to the original ResNet50 network, and was faster than the simplified ResNet with the same AUC value; the simplified Inception V4 model required 0.56 GFLOPs while the simplified ResNet50 model required 2.96 GFLOPs (about 5.3× more expensive). Our experimental evaluation suggests that during the development of a deep learning network, it may be better to focus on the classification performance of the network and worry less about its computational requirements and further apply a network simplification step after the network architecture has been fine-tuned fo r classification performance.

In our work we used classification of TILs in whole slide images as the driving application use case. We expect that our method can be generalized to other classification problems in digital pathology. Characterization of TIL patterns in whole slide images is an important use case. Multiple studies have shown that there is a correlation between the density and spatial organization of TILs and clinical outcomes (37, 38, 56, 57). Characterizations of TIL patterns can lead to better understanding of cancer mechanisms and improve cancer staging (58). There is an increasing number of computational pathology approaches to generate such characterizations (34, 59, 60).

Applications of deep learning methods for TIL analysis on a large number of whole slide images is desirable, as they can result in a better understanding of TIL patterns. It is important to employ effective and efficient deep learning methods in order to facilitate such applications. We have shown that our approach can reduce computational requirements by roughly of 4× without impacting overall classification quality for two real-world CNN networks. This is a significant improvement in execution cost and can enable a broader use of these techniques in digital pathology. We also believe this paper opens multiple interesting directions for future work. First, as briefly discussed, it would be important to evaluate a larger number of CNN architectures to analyze how simplification methods would affect their AUC and count. This could answer the question regarding whether the developer should worry or not about the FLOPs required or network complexity during the development, or if this could be resolved by simplification methods in all cases. Second, we also want to expand this analysis with additional pathology image analysis applications,

including not only additional classification applications but also segmentation tasks, for instance.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

AM implemented the code, performed the experiments, and organized the dataset. AM, TK, and GT performed the experimental analysis. AM and GT wrote the first draft of the manuscript. AM, TK, JK, RF, JS, and GT wrote the final manuscript. All authors contributed to conception and design of the study. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported in part by 1UG3CA225021 from the NCI, R01LM011119-01 and R01LM009239 from the NLM, CNPq, Capes/Brazil Grants PROCAD-183794, FAPEMIG, PROCAD/UFMG, K25CA181503, and U01CA242936 from National Institute of Health and generous donations from Bob Beals and Betsy Barton. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation Grant Number ACI-1548562. Specifically, it used the Bridges system, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmed.2022.894430/full#supplementary-material

REFERENCES

1. Gurcan MN, Boucheron LE, Can A, Madabhushi A, Rajpoot NM, Yener B. Histopathological image analysis: a review. *IEEE Rev Biomed Eng*. (2009) 2:147–71. doi: 10.1109/RBME.2009.2034865
2. Li C, Xue D, Hu Z, Chen H, Yao Y, Zhang Y, et al. A survey for breast histopathology image analysis using classical and deep neural networks. In: *International Conference on Information Technologies in Biomedicine*. Springer (2019). p. 222–33. doi: 10.1007/978-3-030-23762-2_20
3. Madabhushi A, Lee G. *Image Analysis and Machine Learning in Digital Pathology: Challenges and Opportunities*. Stockholm: Elsevier (2016). doi: 10.1016/j.media.2016.06.037
4. Barker J, Hoogi A, Depeursinge A, Rubin DL. Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles. *Med Image Anal*. (2016) 30:60–71. doi: 10.1016/j.media.2015.12.002
5. Spanhol FA, Oliveira LS, Petitjean C. Breast cancer histopathological image classification using convolutional neural networks. *2016 International Joint Conference on Neural Networks (IJCNN)*. Vancouver, BC (2016). doi: 10.1109/IJCNN.2016.7727519
6. Xu Y, Xia Z, Ai Y, Zhang F, Lai M, Chang EIC. Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation. In: IEEE, editor. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, QLD: IEEE (2015). doi: 10.1109/ICASSP.2015.7178109
7. Dimitriou N, Arandjelovic O, Caie PD. Deep learning for whole slide image analysis: an overview. *Front Med*. (2019) 6:264. doi: 10.3389/fmed.2019.00264
8. Wang S, Yang DM, Rong R, Zhan X, Xiao G. Pathology image analysis using segmentation deep learning algorithms. *Am J Pathol*. (2019) 189:1686–98. doi: 10.1016/j.ajpath.2019.05.007
9. Binder T, Tantaoui EM, Pati P, Catena R, Set-Aghayan A, Gabrani M. Multi-organ gland segmentation using deep learning. *Front Med*. (2019) 6:173. doi: 10.3389/fmed.2019.00173
10. Serag A, Ion-Margineanu A, Qureshi H, McMillan R, Saint Martin MJ, Diamond J, et al. Translational AI and deep learning in diagnostic pathology. *Front Med*. (2019) 6:185. doi: 10.3389/fmed.2019.00185
11. Grote A, Schaadt NS, Forestier G, Wemmert C, Feuerhake F. Crowdsourcing of histological image labeling and object delineation by medical students. *IEEE Trans Med Imaging*. (2018) 38:1284–94. doi: 10.1109/TMI.2018.2883237
12. Ørting S, Doyle A, van Hilten MHA, Inel O, Madan CR, Mavridis P, et al. A survey of crowdsourcing in medical image analysis. *arXiv preprint arXiv:190209159*. (2019). doi: 10.15346/hc.v7i1.1
13. Singh A, Sengupta S, Lakshminarayanan V. Explainable deep learning models in medical image analysis. *J Imaging*. (2020) 6:52. doi: 10.3390/jimaging6060052
14. Amann J, Vetter D, Blomberg SN, Christensen HC, Coffee M, Gerke S, et al. To explain or not to explain? Artificial intelligence explainability in clinical decision support systems. *PLoS Digit Health*. (2022) 1:e0000016. doi: 10.1371/journal.pdig.0000016
15. Thompson NC, Greenewald KH, Lee K, Manso GF. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*. (2020). doi: 10.48550/arXiv.2007.05558
16. Han S, Mao H, Dally WJ. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:151000149*. (2015).
17. Lin S, Ji R, Li Y, Wu Y, Huang F, Zhang B. Accelerating convolutional networks via global & dynamic filter pruning. In: *IJCAI*. Shenyang (2018). p. 8. doi: 10.24963/ijcai.2018/336
18. Shao M, Dai J, Kuang J, Meng D. A dynamic CNN pruning method based on matrix similarity. *Signal Image Video Process*. (2021) 15:381–9. doi: 10.1007/s11760-020-01760-x
19. Ding X, Zhou X, Guo Y, Han J, Liu J, et al. Global sparse momentum SGD for pruning very deep neural networks. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc, Fox EB, editors. *Advances in Neural Information Processing Systems 32*. Vancouver, BC: Curran Associates, Inc (2019). p. 6382–94.
20. Han S, Pool J, Tran J, Dally W. Learning both weights and connections for efficient neural network. In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R, editors. *Advances in Neural Information Processing Systems 28*. Montreal, QC: Curran Associates, Inc (2015). p. 1–9
21. Ba J, Caruana R. Do deep nets really need to be deep? In: Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ, editors. *Advances in Neural Information Processing Systems 27*. Montreal, QC: Curran Associates, Inc (2014). p. 1–9.

22. Banner R, Nahshan Y, Soudry D. Post training 4-bit quantization of convolutional networks for rapid-deployment. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, editors. *Advances in Neural Information Processing Systems 32*. Vancouver, BC: Curran Associates, Inc (2019). p. 1–9.
23. Xu S, Huang A, Chen L, Zhang B. Convolutional neural network pruning: a survey. In: *2020 39th Chinese Control Conference (CCC)*. Long Beach, CA: IEEE (2020). p. 7458–63. doi: 10.23919/CCC50068.2020.9189610
24. He Y, Liu P, Wang Z, Hu Z, Yang Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Montreal, QC (2019). p. 4340–9. doi: 10.1109/CVPR.2019.00447
25. Luo JH, Zhang H, Zhou HY, Xie CW, Wu J, Lin W. Thinet: pruning CNN filters for a thinner net. *IEEE Trans Pattern Anal Mach Intell*. (2018) 41:2525–38. doi: 10.1109/TPAMI.2018.2858232
26. Lin S, Ji R, Yan C, Zhang B, Cao L, Ye Q, et al. Towards optimal structured cnn pruning via generative adversarial learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2019). p. 2790–9. doi: 10.1109/CVPR.2019.00290
27. Ding X, Hao T, Tan J, Liu J, Han J, Guo Y, et al. ResRep: lossless CNN pruning via decoupling remembering and forgetting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Berlin (2021). p. 4510–20. doi: 10.1109/ICCV48922.2021.00447
28. Lu Z, Pu H, Wang F, Hu Z, Wang L. The expressive power of neural networks: a view from the width. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17*. Red Hook, NY: Curran Associates Inc. (2017). p. 6232–40.
29. Zou J, Rui T, Zhou Y, Yang C, Zhang S. Convolutional neural network simplification via feature map pruning. *Comput Electric Eng*. (2018) 70:950–8. doi: 10.1016/j.compeleceng.2018.01.036
30. Hajabdollahi M, Esfandiarpoor R, Najarian K, Karimi N, Samavi S, Soroushmehr SR. Hierarchical pruning for simplification of convolutional neural networks in diabetic retinopathy classification. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Long Beach, CA: IEEE (2019). p. 970–3. doi: 10.1109/EMBC.2019.8857769
31. Cheng Y, Wang D, Zhou P, Zhang T. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:171009282*. (2017). doi: 10.48550/arXiv.1710.09282
32. Ding X, Ding G, Guo Y, Han J, Yan C. Approximated oracle filter pruning for destructive CNN width optimization. In: *International Conference on Machine Learning*. PMLR (2019). p. 1607–16.
33. Osaku D, Gomes J, Falcão AX. Convolutional neural network simplification with progressive retraining. *arXiv preprint arXiv:210104699*. (2021). doi: 10.1016/j.patrec.2021.06.032
34. Saltz J, Gupta R, Hou L, Kurc T, Singh P, Nguyen V, et al. Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images. *Cell Rep*. (2018) 23:181. doi: 10.1016/j.celrep.2018.03.086
35. Le H, Gupta RR, Hou L, Abousamra S, Fassler D, Kurc TM, et al. Utilizing automated breast cancer detection to identify spatial distributions of tumor infiltrating lymphocytes in invasive breast cancer. *Am J Pathol*. (2020) 190:1491–504. doi: 10.1016/j.ajpath.2020.03.012
36. Oble DA, Loewe R, Yu P, Mihm MC. Focus on TILs: prognostic significance of tumor infiltrating lymphocytes in human melanoma. *Cancer Immunity Arch*. (2009) 9:3.
37. Angell H, Galon J. From the immune contexture to the Immunoscore: the role of prognostic and predictive immune markers in cancer. *Curr Opin Immunol*. (2013) 25:261–7. doi: 10.1016/j.coi.2013.03.004
38. Mlecnik B, Bindea G, Pagés F, Galon J. Tumor immunosurveillance in human cancers. *Cancer Metastasis Rev*. (2011) 30:5–12. doi: 10.1007/s10555-011-9270-7
39. Teng MW, Ngiow SF, Ribas A, Smyth MJ. Classifying cancers based on T-cell infiltration and PD-L1. *Cancer Res*. (2015) 75:2139–45. doi: 10.1158/0008-5472.CAN-15-0255
40. Rakaee M, Kilvaer TK, Dalen SM, Richardsen E, Paulsen EE, Hald SM, et al. Evaluation of tumor-infiltrating lymphocytes using routine H&E slides predicts patient survival in resected non-small cell lung cancer. *Hum Pathol*. (2018) 79:188–98. doi: 10.1016/j.humpath.2018.05.017
41. Wang S, Yang DM, Rong R, Zhan X, Fujimoto J, Liu H, et al. Artificial intelligence in lung cancer pathology image analysis. *Cancers*. (2019) 11:1673. doi: 10.3390/cancers11111673
42. Komura D, Ishikawa S. Machine learning methods for histopathological image analysis. *Comput Struct Biotechnol J*. (2018) 16:34–42. doi: 10.1016/j.csbj.2018.01.001
43. Linder N, Taylor JC, Colling R, Pell R, Alveyn E, Joseph J, et al. Deep learning for detecting tumour-infiltrating lymphocytes in testicular germ cell tumours. *J Clin Pathol*. (2019) 72:157–64. doi: 10.1136/jclinpath-2018-205328
44. Goyal M, Knackstedt T, Yan S, Hassanpour S. Artificial intelligence-based image classification methods for diagnosis of skin cancer: challenges and opportunities. *Comput Biol Med*. (2020) 127:104065. doi: 10.1016/j.compbiomed.2020.104065
45. Li J, Li W, Sisk A, Ye H, Wallace WD, Speier W, et al. A multi-resolution model for histopathology image classification and localization with multiple instance learning. *Comput Biol Med*. (2021) 131:104253. doi: 10.1016/j.compbiomed.2021.104253
46. George K, Faziludeen S, Sankaran P, Joseph K P. Breast cancer detection from biopsy images using nucleus guided transfer learning and belief based fusion. *Comput Biol Med*. (2020) 124:103954. doi: 10.1016/j.compbiomed.2020.103954
47. Klauschen F, Müller KR, Binder A, Bockmayr M, Hägele M, Seegerer P, et al. Scoring of tumor-infiltrating lymphocytes: From visual estimation to machine learning. In: *Seminars in Cancer Biology*. vol. 52. Elsevier (2018). p. 151–7. doi: 10.1016/j.semcancer.2018.07.001
48. Garcia E, Hermoza R, Castanon CB, Cano L, Castillo M, Castanneda C. Automatic lymphocyte detection on gastric cancer IHC images using deep learning. In: *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE (2017). p. 200–4. doi: 10.1109/CBMS.2017.94
49. Roy K, Banik D, Bhattacharjee D, Nasipuri M. Patch-based system for classification of breast histology images using deep learning. *Comput Med Imaging Graph*. (2019) 71:90–3. doi: 10.1016/j.compmedimag.2018.11.003
50. Zagoruyko S, Komodakis N. Wide residual networks. In: Richard C, Wilson ERH, Smith WAP, editors. *Proceedings of the British Machine Vision Conference (BMVC)*. York: BMVA Press (2016). p. 87.1–87.12. doi: 10.5244/C.30.87
51. Raghu M, Poole B, Kleinberg J, Ganguli S, Dickstein JS. On the expressive power of deep neural networks. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70. ICML'17*. (2017). p. 2847–54.
52. Tan M, Le QV. EfficientNet: rethinking model scaling for convolutional neural networks. In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Long Beach, CA(2019).
53. He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. In: *European Conference on Computer Vision*. Amsterdam: Springer. (2016). p. 630–45. doi: 10.1007/978-3-319-46493-0_38
54. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: *Thirty-First AAAI Conference on Artificial Intelligence*. San Francisco, CA(2017).
55. National Human Genome Research Institute. *The Cancer Genome Atlas*. (2017). Available online at: https://cancergenome.nih.gov/
56. Zitvogel L, Tesniere A, Kroemer G. Cancer despite immunosurveillance: immunoselection and immunosubversion. *Nat Rev Immunol*. (2006) 6:715–27. doi: 10.1038/nri1936
57. Fridman WH, Pages F, Sautés-Fridman C, Galon J. The immune contexture in human tumours: impact on clinical outcome. *Nat Rev Cancer*. (2012) 12:298–306. doi: 10.1038/nrc3245
58. Broussard EK, Disis ML. TNM staging in colorectal cancer: T is for T cell and M is for memory. *J Clin Oncol*. (2011) 29:601–3 doi: 10.1200/JCO.2010.32.9078

59. Rutledge WC, Kong J, Gao J, Gutman DA, Cooper LA, Appin C, et al. Tumor-infiltrating lymphocytes in glioblastoma are associated with specific genomic alterations and related to transcriptional class. *Clin Cancer Res*. (2013) 19:4951–60. doi: 10.1158/1078-0432.CCR-13-0551
60. Lazar AJ, McLellan MD, Bailey MH, Miller CA, Appelbaum EL, Cordes MG, et al. Comprehensive and integrated genomic characterization of adult soft tissue sarcomas. *Cell*. (2017) 171:950–65.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.