Supplementary Information

Supplementary Notes

1. *AlexNet*

This CNN based architecture has eight layers of learnable parameters. The model comprises five convolutional and max pooling layers and three fully connected layers and a final output layer. Each of the layers in the convolutional blocks and fully connected blocks use ReLu (Rectified Linear Unit) activation functions. This architecture was first used to classify the ImageNet dataset.

2. *GoogLeNet*

This architecture is the first to use inception blocks. Inception blocks use convolutions that aim to decrease the number of learnable parameters and in turn makes the architecture go deeper. In an inception block there is a common input through which convolutions of filter sizes 1x1, 3x3, 5x5 and a 3x3 max pooling is performed and the output of each of these modules are stacked together to a single output. Supplementary Figure 1 shows the architecture of an inception block. GoogLeNet has twenty-two layers in total, three convolutional layers followed by nine inception blocks (these inception blocks are occasionally followed by Max pooling layers) and a final block that consists of average pooling, dropout, and linear layers.

3. *Inception V3*

This is one of the state-of-the-art architectures which is 42 layers deep, its architecture is like GoogLeNet with few improvisations and advancements. The few characteristics of Inception V3 are: (i) Factorizing larger convolutions into smaller ones which reduces the number of parameters and reduces the cost of computation. e.g., a 5x5 convolutional layer is factorized into two 3x3 convolutional layers. (ii) Spatial factorization into asymmetric convolutions: e.g., replacing a 3x3 convolutional layer with a 1x3 and a 3x1 convolutional layers. This reduces the number of parameters and decreases the cost of computation. (iii) Auxiliary classifiers: usage of these classifiers improves the convergence of deep networks, hence improving the accuracy. (iv) Efficient Grid size reduction of feature maps.

4. *ResNet*

This is a 34 layered residual network, the core idea of ResNet is to address the vanishing gradient problem faced by many deep-neural networks by pioneering 'identity shortcut connection' which skips one or more layers. Supplementary Figure 2 shows are typical residual block. This CNN was designed typically for mobile and embedded vision applications. This architecture uses depthwise separable convolutions that helps in reducing the latency of the application. A depthwise separable convolution is a depthwise convolution succeeded by a pointwise convolution. A depthwise convolution is a channel-wise kxk spatial convolution, whereas a pointwise convolution is simply a 1x1 convolution which is used to change the dimension.

5. *MobileNet*

This CNN was designed typically for mobile and embedded vision applications. This architecture uses depthwise separable convolutions that helps in reducing the latency of the application. A depthwise separable convolution is a depthwise convolution succeeded by a pointwise convolution. A depthwise convolution is a channel-wise kxk spatial convolution, whereas a pointwise convolution is simply a 1x1 convolution which is used to change the dimension.

6. *Xception*

Xception stands for “extreme inception.” This architecture uses features of both Inception v3 and MobileNet architectures. It uses a modified depthwise separable convolution i.e., it has a pointwise convolution and then a depthwise convolution (opposite of MobileNet), this feature is inspired by inception v3 where a 1x1 convolution precedes a spatial convolution. Xception does not use any intermediate activation function like ReLu. Also, convolutions are not performed across all the channels, therefore the connections are fewer, and the model is less deep. It uses 'skip connections' like the ResNet.

7. *DenseNet*

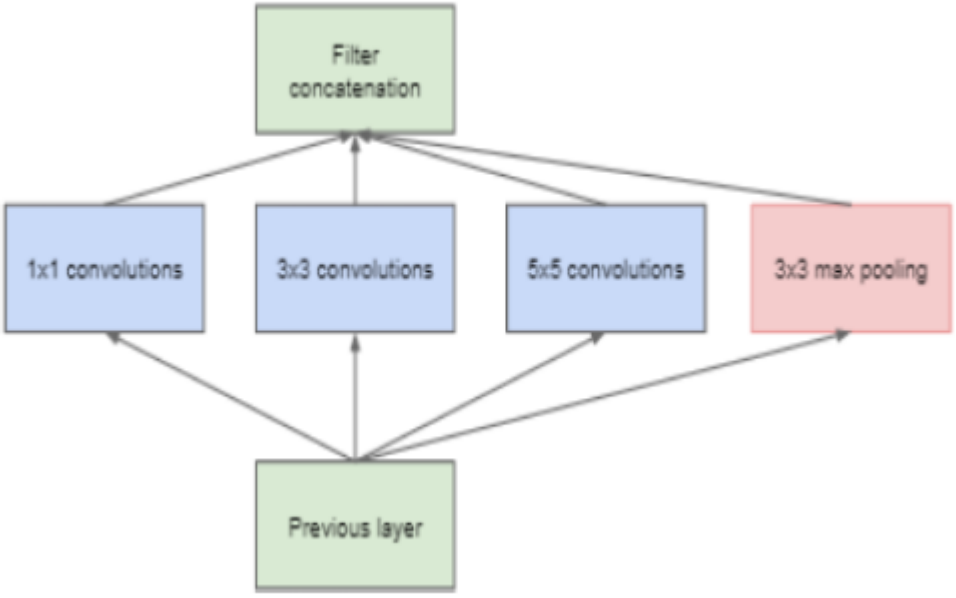
These are densely connected convolutional networks with a very narrow set of layers, the salient feature of this architecture is it re-uses its own features thus exploiting the full potential of its own network which reduces the number

of learnable parameters by removing the redundancy to re-learn its feature maps. Dense Nets can handle the vanishing gradient problem as the loss function of each layer provides the gradient of the layers.

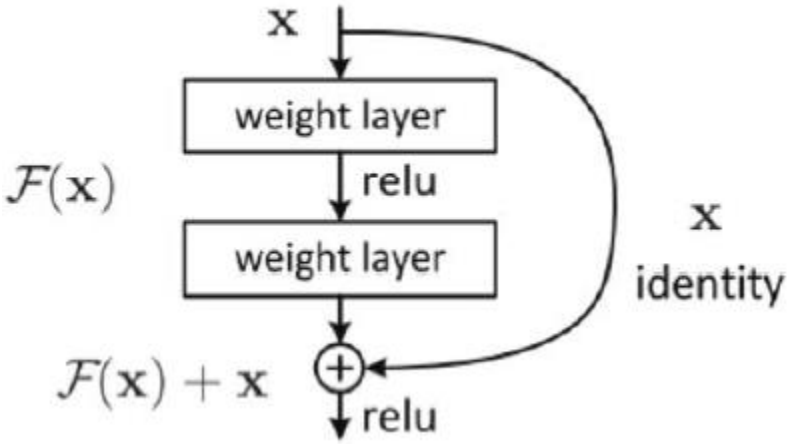
8. *ResNeXt*

This CNN architecture has features of VGG, ResNet and Inception. It uses repetitive blocks (like VGG), skip connections (as in ResNet) and auxiliary classifiers (like Inception v3). ResNeXt replaces the standard residual blocks with one that leverages a "split-transform-merge" strategy.

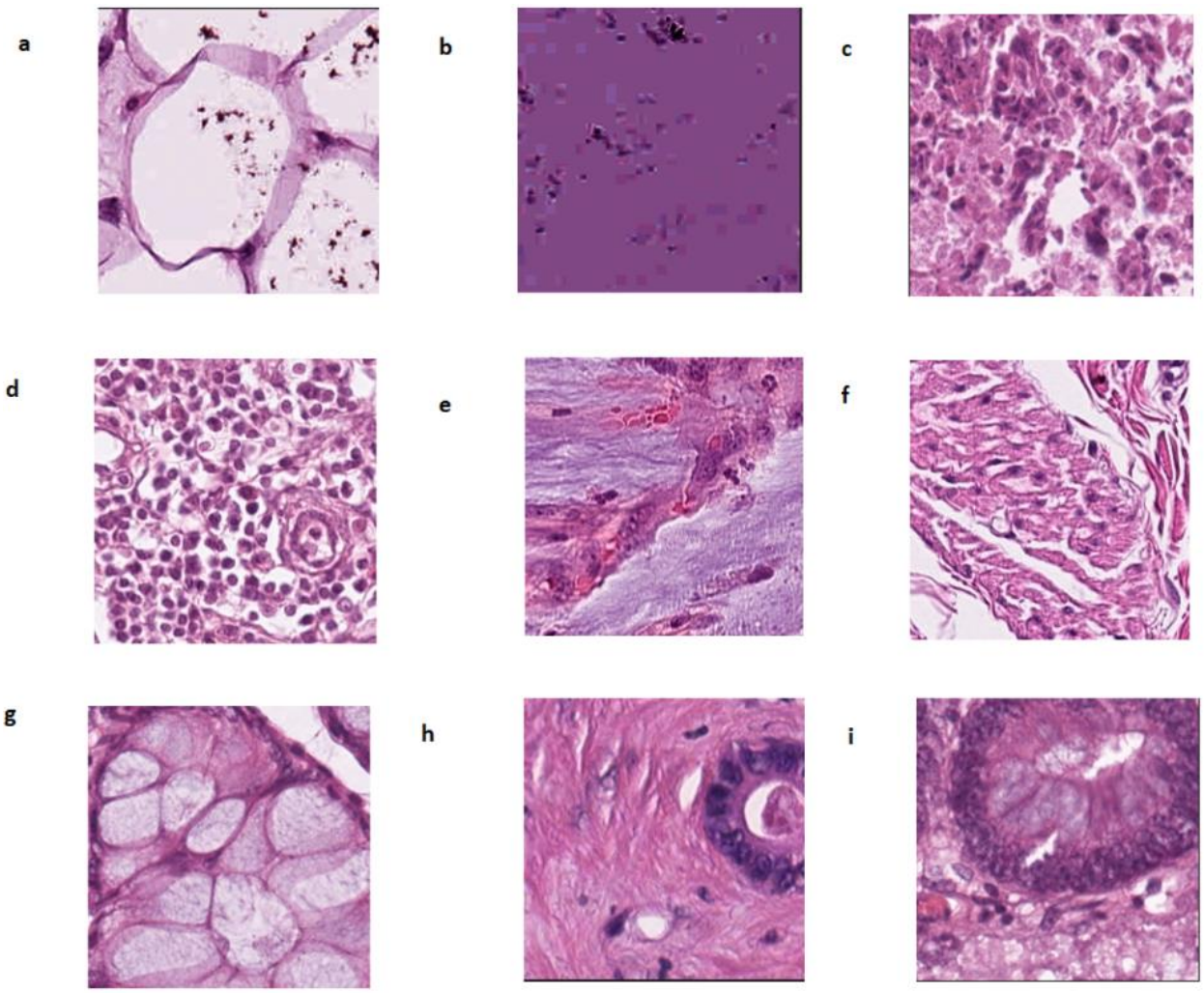
Supplementary Figures



Supplementary Figure 1: Architecture of an Inception Block



Supplementary Figure 2: A Residual Block



Supplementary Figure 3: (a) ADI, (b) BACK, (c) DEB, (d) LYM, (e) MUC, (f) MUS, (g) NORM, (h) STR, (i) TUM.

Supplementary Table

Architecture	Cancer detection				Tissue classification
	Accuracy	Precision	Recall	F1 score	Accuracy
AlexNet	95.39%	94.16%	96.78%	95.45%	91.80%
GoogLeNet	99.12%	99.11%	99.25%	99.13%	98.86%
ResNet	98.78%	98.67%	98.89%	98.78%	97.04%
Inception V3	97.24%	97.43%	97.02%	97.23%	96.64%
MobileNet	98.85%	98.19%	99.53%	98.86%	96.22%
Xception	99.14%	98.60%	99.70%	99.14%	97.16%
ResNeXt	97.92%	97.13%	98.75%	97.93%	93.63%
DenseNet	98.92%	98.17%	99.70%	98.93%	96.37%

Supplementary Table 1: Performance of Deep Learning Models

Architecture	ADI	BACK	DEB	LYM	MUC	MUS	NORM	STR	TUM
Alexnet	99.10%	99.30%	92.90%	98.90%	93.30%	84%	86.40%	87.10%	85.10%
GoogleNet	99.78%	100%	98.89%	100%	99.11%	99.33%	98.67%	96.44%	97.56%
ResNet	99.78%	99.33%	97.56%	99.78%	97.78%	95.78%	94.22%	92.89%	95.33%
Inception V3	98.67%	97.11%	95.78%	100%	98.44%	93.33%	97.56%	94.89%	92.22%
Xception	98.67%	99.78%	97.56%	99.78%	98.22%	96.67%	96.22%	95.33%	92.22%
DenseNet	99.78%	99.11%	94.89%	99.11%	97.78%	86.89%	97.78%	94.44%	97.51%

Supplementary Table 2: Tissue Classification Accuracies