CS680 Project Report

A Study of ResNets and DenseNets in Breast Cancer Detection

Shengye Chen
School of Computer Science
University of Waterloo
j57chen@uwaterloo.ca

Abstract

Deep learning models have been increasingly used in the medical imaging field in recent years. These models can be trained to analyze medical images such as MRIs to identify abnormalities or features of interest. This project is an application of two deep learning models – ResNets and DenseNets, to identify breast cancer using MRI images. I use a public dataset on Kaggle to train 3 ResNets and 3 DenseNets with different number of layers. ResNet-152V2 and DenseNet-169 achieve the highest validation accuracy of 99.65%, but DenseNet-169 has better parameter efficiency than ResNet-152V2.

1 Introduction

Breast cancer is the second most common cancer diagnosed in Canada, especially among women aged over 50. 1 in 8 women will be diagnosed with breast cancer in their lifetime [1]. The importance of early detection of breast cancer cannot be overstated. [2] has shown that women who are diagnosed with early stage breast cancer have a much better chance of surviving and living a longer, healthier life. Early diagnosis and treatment also reduce the costs associated with treating the more advanced stages of breast cancer. Early detection and diagnosis of breast cancer can also reduce the risk of recurrence and reduce the need for more extensive treatment. For these reasons, early detection of breast cancer is a critical component of a comprehensive healthcare program.

Analysis of medical imaging is a very crucial task to diagnose any diseases at the earliest, but manual screening for breast cancer is time-consuming and can be subjective. Deep convolutional neural networks (CNN) offer a promising solution to these challenges, by providing a fast and objective way to analyze medical images. Deep learning models can be trained on large datasets of medical images, where the correct labels (i.e., the presence or absence of cancer) are provided. This allows the models to learn the patterns in the images that are indicative of cancer. Once trained, the algorithms can then be used to automatically analyze new images and make predictions about the presence of cancer. In addition to being fast and objective, deep learning models can also be highly accurate. They can be trained to achieve high levels of sensitivity and specificity. This can help reduce the number of false positives and false negatives, and improve the overall accuracy of breast cancer screening.

Based on the CDC guideline [3], four types of medical imaging can be used to diagnose breast cancer – breast ultrasound (sonograms), diagnostic mammogram (X-ray), histopathological images, and breast magnetic resonance imaging (MRI). In this study, MRI images are chosen to train deep CNNs due to three reasons: *(a)* MRI has a higher sensitivity for detecting breast cancer than X-ray and sonograms [4]; *(b)* unlike X-ray, MRI do not use any dangerous radiation; and *(c)* MRI does not require taking tissue samples from patients compared to histopathological images. Based on these reasons, MRI images are chosen in this study as they are the most suitable way to quickly screen breast cancer patients at an early stage.

In this study, the structure of ResNets and DenseNets is analyzed and their effectiveness in classifying breast cancer MRI images is evaluated. I find that DenseNets have a higher accuracy rate compared to ResNets, despite having fewer layers and parameters. The most successful model for the dataset in this study is DenseNet-169, which has a validation accuracy of 99.65%. While ResNet-101V2 also produces the same accuracy, it requires nearly five times as many parameters as DenseNet-169. My code is publicly available at https://github.com/jacky1c/BreastCancerDetectionProject.

2 Related Works

The traditional computer vision methods were taken into consideration as a part of this study. However, researches have shown that this approach has the disadvantage of being sensitive to various conditions and factors, such as lighting, object orientation, and noise, which could significantly impact the quality of the extracted features and ultimately the performance of the model. As a result, solutions that rely on hand-crafted features might not be as effective when deployed in real-world conditions as they appeared to be based on controlled data. However, it has been argued that deep learning-based methods are often more effective at generalizing to new data compared to traditional computer vision methods, especially when there is a large volume of data available and techniques such as transfer learning and data augmentation are used.

The applications of CNNs in medical imaging field have been a part of CNN research since their initial discovery. CNNs are particularly well-suited to image classification and segmentation tasks, and have been used to detect and classify various types of abnormalities in medical images, such as tumors, lesions, and abnormalities in organs.

The use of deep CNN is adopted by Aljuaid et al. [5] to classify benign and malignant lesions in the Breast Cancer Histopathological imaging Database (BreakHis) where benign and malignant images have four sub-classes individually. For binary classification of benign or malignant cancer, they achieve accuracy rates of 99.7%, 97.66%, and 96.94% using ResNet, InceptionV3Net, and ShuffleNet, respectively. Average accuracies for multi-class classification are 97.81%, 96.07%, and 95.79% for ResNet, Inception-V3Net, and ShuffleNet, respectively.

MRI images are used in [6] where researchers propose a Back Propagation Boosting Recurrent Widening Model (BPBRW) with a Hybrid Krill Herd African Buffalo Optimization (HKH-ABO) method to diagnose breast cancer at an earlier stage. Furthermore, Wienmed filter is established for pre-processing the MRI noisy image content. They demonstrate their model have a 99.6% accuracy rate.

In [7], Yadav et al. find that transfer learning on VGG16 and InceptionV3 models is a more useful classification method on a small dataset compared to a capsule network training from scratch. In transfer learning, retraining specific features on a new target dataset is essential to improve performance. Additionally, it is important to ensure that the complexity of the network is appropriate for the size of the dataset.

3 Deep Learning Models and Techniques

3.1 ResNets

ResNet, short for Residual Network, is a type of deep learning architecture that was introduced in 2015 by He et al. [8]. It was developed to address the vanishing gradient problem, which can make it difficult for the network to learn from the data, when neural networks gets deeper.

Residual blocks. The key idea behind ResNet is the use of residual blocks [8], which allow the network to bypass one or more layers of the network and directly access the output of a preceding layer. In general, the residual block is defined as

$$x_{l+1} = f(F(x_l, W_l) \oplus h(x_l)) \tag{1}$$

, where x_l and x_{l+1} are the input and output of the l^{th} layer.

Function $F(\cdot)$ is the residual function to be learned from the data. It contains operations of batch normalization [9], ReLU [10], and convolution.

Function $h(\cdot)$ is the shortcut connection, which is an identity mapping, i.e. $h(x) = x$. The shortcut connection helps to prevent the vanishing gradient problem, as the backpropagation can flow directly through the shortcut connections, allowing the network to learn more effectively. When the output dimension is bigger than the input dimension, the identity mapping can be chosen in two ways: either applying zero padding or using 1×1 convolutions.

There are two variations of residual blocks with regards to the choice of function $f(\cdot)$ in Eqn.(1). In the original version [8], function $f(\cdot)$ is the ReLU [10] function. In the second version [11], function $f(\cdot)$ is chosen as identity mapping and the residual block becomes $x_{l+1} = F(x_l, W_l) \oplus x_l$. The comparison of two residual block designs is shown as in Fig. 1.

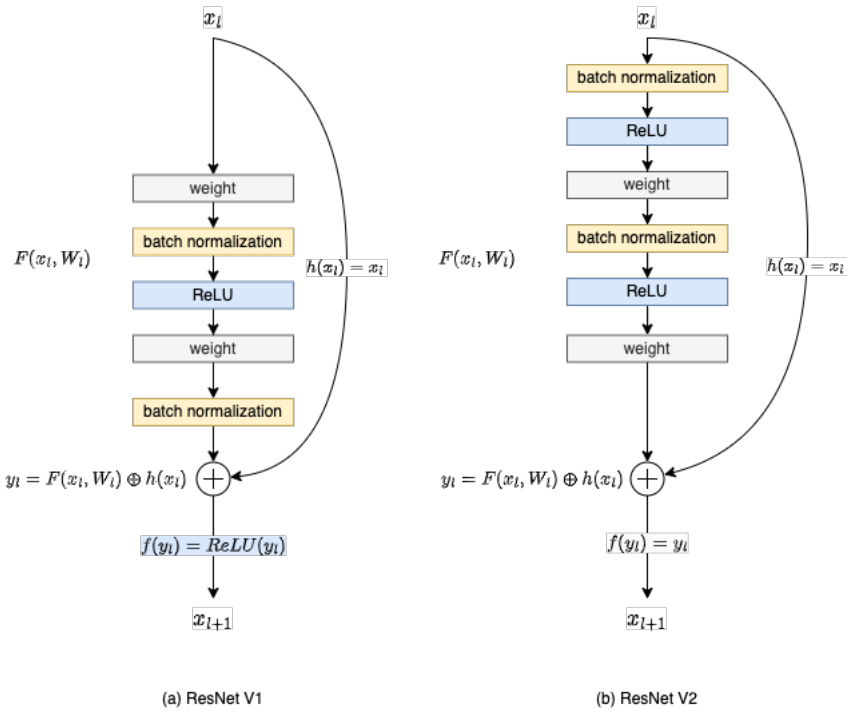

Figure 1: A comparison of *(a)* residual blocks in ResNet V1 [8] and *(b)* residual blocks in ResNet V2 [11].

ResNet architecture. The output of a residual block is directly connected to the input of the next residual block to form a residual network. One of the benefits of ResNet is that it can be easily trained using standard stochastic gradient descent (SGD) algorithms, without the need for specialized training techniques like those used for very deep networks. This makes it easier to use and more widely applicable than other deep learning architectures.

Each layer in ResNets has its own weights and thus the number of parameters of ResNets is large. However, [12] shows that many layers contribute very little and in fact can be randomly dropped during training. In other words, the extra number of weights do not add the same improvement to the network when depth increases.

In summary, ResNet is a powerful and effective deep learning architecture that uses shortcut connections to overcome the vanishing gradient problem. It has been successfully applied to a number of tasks, and its ease of use makes it a popular choice for researchers and practitioners working in the field of deep learning.

3.2 DenseNets

DenseNet was introduced in 2016 by Huang et al. [13] as another deep learning architecture to alleviate the vanishing gradients problem. It is based on the concept of dense connectivity, where

each layer in the network is connected to every other layer in a feed-forward fashion. This allows the network to learn more effectively and can lead to more accurate predictions.

DenseNet and ResNet share a similar concept – namely, that both of them utilize shortcut connections in order to address training accuracy degradation problem. The innovation of DenseNets is the shortcut connections are inserted from any layer to all of its subsequent layers to further improve the backpropagation flow. This change increases the number of shortcut connections in a network from L (in ResNets) to $\frac{L(L+1)}{2}$ where L is the number of layers. Compared to ResNets, where each layer receives a state from a preceding layer and passes its knowledge down to the next layer, each layer in DenseNets receives a collective knowledge from all the earlier layers.

Dense blocks. Instead of combining features by element-wise addition, DenseNet concatenates features before they are passed into the next layer. In general, a dense block is formally defined as

$$x_{l+1} = F([x_0, x_1, \ldots, x_l]) \tag{2}$$

, where $[x_0, x_1, \ldots, x_l]$ is a single tensor consisting of the feature maps produced in layers $0, 1, \ldots, l$. Similar to [11], function $F(\cdot)$ in DenseNets contains operations of batch normalization [9], ReLU [10], and a 3×3 convolution, i.e., BN-ReLU-Conv(3×3).

The number of feature maps produced by function $F(\cdot)$ is referred as the growth rate k of the network. If the number of channels in the initial input layer, i.e., the 0^{th} layer, and the growth rate of each function $F(\cdot)$ at the l^{th} layer are the same, then the number of input feature maps of the l^{th} layer is $k \times l$. The bottleneck layers are introduced to reduce the number of input feature maps in order to improve computational efficiency. In [13], a bottleneck layer, which is essentially a different choice of the composite function $F(\cdot)$, is defined as BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3).

DenseNet architecture. Feature map sizes remain the same within dense blocks so that they can be concatenated easily, but feature map sizes can be different between two adjacent dense blocks in order to achieve down-sampling. Therefore, transition layers are inserted in between dense blocks to reduce the size of feature maps. In [13], transition layers contain operations of a 1×1 convolution followed by a 2×2 average pooling. At the end of the last dense block, a global average pooling is performed and then a softmax classifier is attached to produce the final output. A comparison of residual blocks and dense block architecture is shown in Fig. 2.

Compression. The number of feature maps can be further reduced at transition layers by compression. If the dense block before a transition layer has m feature maps, the transition layer with compression rate of θ generates $\lfloor \theta m \rfloor$ output feature maps. This is implemented by setting the number of output features as θ generates $\lfloor \theta m \rfloor$ in Conv2d function.

As the total number of layers and the growth rate increase, Huang et al. [13] observe that generally DenseNet have a better accuracy rate without compression or bottleneck layers. This makes intuitive sense because DenseNet can have more comprehensive knowledge about the features. Compression and bottleneck layers can be added if the number of feature maps is a constraint. They also appear to be an effective way to make DenseNets less prone to overfitting [13].

DenseNet is a promising CNN architecture that can improve the flow of information through the network and reduce the number of parameters. Its effectiveness has been demonstrated on a number of tasks. In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2017, a DenseNet model achieved a top-1 error rate of 22.16% [13], which was a significant improvement over ResNet of 24.33% [8].

3.3 Transfer Learning

Transfer learning is a machine learning technique that allows a model trained on one task to be applied to a different but related task. It is based on the idea that the knowledge learned by a model on one task can be a useful starting point for other tasks, even if the input data and output predictions are different.

Transfer learning has become increasingly popular in recent years, as it can significantly reduce the amount of data and computation required to train a model for a new task. This is because the pre-trained model can be used as a starting point, and only the final layers of the network need to be

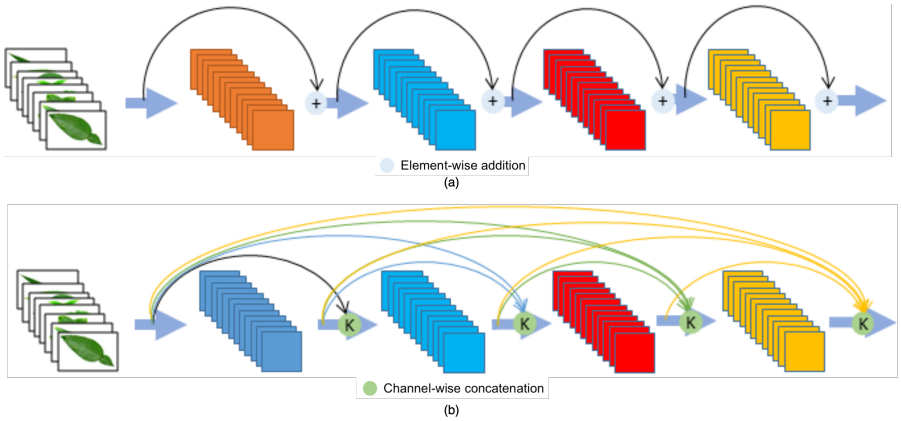

Figure 2: A comparison of *(a)* Four residual blocks being connected in a ResNet and *(b)* one dense block in a DenseNet [14].

re-trained on the new data. In research work published by Thrun et al. [15], 129,450 clinical skin cancer images were classified using pre-trained models and they achieve excellent results. In this study, I use the learned weights on the ImageNet [16] to initialize the training weights for models to learn to classify breast cancer MRI images.

4 Experiments

4.1 Dataset

I use a public dataset on Kaggle [17] which contains 540×250 pixels grey-scale MRI images where regions of interest are not annotated. There are a total of 1400 images in the training set with the same number of images for healthy and sick patients. The data provider claims the 80 images in the test set are split into equal number of healthy and sick samples, but all of them should actually be sick images. This mistake initially caused me confusions because the test accuracy was always around 50% even though the validation accuracy was very high. In fact, all of the test image names start with an "S" (for sick), which is another evidence to prove all test images are sick samples. Therefore, the training and test sets in this study are not exactly the same as how the images are provided. Instead, I split 74 (5%) images as the test set from all the images, and then split 281 (20%) images as the validation set from the remaining images. As a result, 1125 images are used to train CNN models.

Data augmentation. I adopt a standard data augmentation scheme (zooming, rotating, and flipping) that is widely used on medical images. Training and validation steps are implemented with augmented images to enhance model robustness. False positive and false negative rates are calculated using both of the original and the augmented test images. Data augmentation is denoted with a "+" mark at the end of the dataset name in Table 1.

4.2 Training and Evaluation

I utilize the publicly available Keras implementation of ResNet and DenseNet models [18]. The weights produced by the models pre-trained on ImageNet [16] is used as initializers. Followed by the ResNets and DenseNets layers, the same output layers (i.e., a global average pooling layer, a dropout layer, a bath normalization layer, and a dense layer with softmax activation function) are added to produce the final outcomes. I adopt cross entropy as the loss function and Adam [19] as the optimizer. When no improvement is seen for 5 consecutive epochs, learning rate is reduced by a factor of 5. I select the model with the lowest validation error during training and the test errors are only evaluated once for each model setting.

Table 1: Accuracy rates of ResNets and DenseNets. ”+” denotes data augmentation.

Model	Depth	Params	Train+	Validation+	Test		Test+	
					FP	FN	FP	FN
ResNet-50V2	103	23.58M	99.38%	98.58%	5.41%	0%	2.70%	1.35%
ResNet-101V2	205	42.64M	99.29%	98.94%	1.35%	0%	1.35%	0%
ResNet-152V2	307	58.34M	**99.56%**	**99.65%**	1.35%	0%	1.35%	0%
DenseNet-121	242	7.04M	99.56%	99.29%	1.35%	0%	1.35%	0%
DenseNet-169	338	12.65M	99.64%	**99.65%**	1.35%	0%	1.35%	0%
DenseNet-201	402	18.33M	**99.91%**	98.94%	1.35%	0%	1.35%	0%

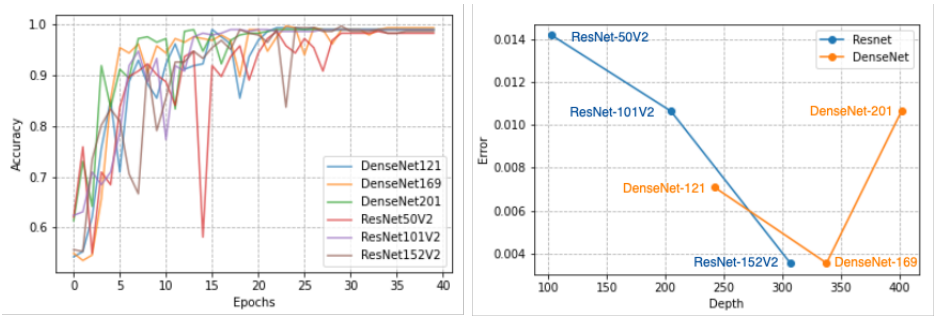

Figure 3: *Left:* Validation accuracy curves of ResNets and DenseNets. *Right:* Comparison of the effectiveness of network depth on validation error between ResNets and DenseNets.

4.3 Classification Results

Overall, each model in this study produces a satisfying result especially when areas of interest are not annotated and people who do not have a medical background cannot distinguish healthy images from sick images. I also observe that models pre-trained on ImageNet [16] have resulted in improved performance on the new task, compared to training models from scratch. Moreover, all of the models except ResNet-50V2 achieve false negative rate of 0%, which means none of the sick images are missed by our models. ResNets and DenseNets demonstrate to perform excellent jobs of image classification. ResNet-152V2 and DenseNet-169 achieve the best validation accuracy of 99.65%.

4.4 Analysis of Results

In general, as the network grows deeper the accuracy will increase, but the deepest model in this study does not achieve the smallest validation error as shown in the right graph in Fig. 3. DenseNet-201 achieves the highest training accuracy (99.91%) among all of the models, but it starts overfitting because the scale of the dataset does not match the complexity of the model. This phenomenon demonstrates that overfitting could not be entirely eliminated from DenseNets if the model is not chosen carefully, even though DenseNet is shown to have a regularizing effect[13], which reduces the risk of overfitting in ResNets [8] when training set size is small. Therefore, it is important to find a proper network complexity that matches the scale of the dataset.

One of the main advantages of DenseNet is that it can significantly reduce the number of parameters in the network because feature maps can be passed from any preceding layer directly. As shown in Table 1, compared to the ResNet-152V2, which has 307 layers and requires 58.34 million parameters, the 338-layer DenseNet-169 with only 12.65 million parameters achieves the same validation accuracy and a better training accuracy.

5 Conclusion

In this study, I have analyzed the architecture of ResNets and DenseNets. Both of them are created to alleviate the vanishing gradients problem when CNNs depth increases. In general, DenseNets achieve higher accuracy over ResNets with more layers and less number of parameters.

ResNets and DenseNets are then used to classify breast cancer MRI images and their performance have been analyzed. The best model for the dataset in this study is DenseNet-169 which achieves a validation accuracy of 99.65%. Even though ResNet-101V2 also achieves the same validation accuracy, it uses nearly 5 times number of parameters than DenseNet-169. Another important observation is that generally deeper networks can produce more precise predictions, but finding a proper network complexity that matches the scale of the dataset should not be omitted.

References

[1] Darren R. Brenner, Hannah K. Weir, Alain A. Demers, Larry F. Ellison, Cheryl Louzado, Amanda Shaw, Donna Turner, Ryan R. Woods, and Leah M. Smith. "Projected estimates of cancer in Canada in 2020". *Canadian Medical Association Journal*, vol. 192, no. 9 (2020) (cit. on p. 1).

[2] Ophira Ginsburg et al. "Breast cancer early detection: A phased approach to implementation". en. *Cancer*, vol. 126 Suppl 10, no. Suppl 10 (May 2020), pp. 2379–2393 (cit. on p. 1).

[3] Centers for Disease Control and Prevention. *How Is Breast Cancer Diagnosed?* URL: https://www.cdc.gov/cancer/breast/basic_info/diagnosis.htm (visited on 10/25/2022) (cit. on p. 1).

[4] Jan Endrikat, Gilda Schmidt, Daniel Haverstock, Olaf Weber, Zuzana Jirakova Trnkova, and Jörg Barkhausen. "Sensitivity of Contrast-Enhanced Breast MRI vs X-ray Mammography Based on Cancer Histology, Tumor Grading, Receptor Status, and Molecular Subtype: A Supplemental Analysis of 2 Large Phase III Studies". en. *Breast Cancer (Auckl.)*, vol. 16 (Jan. 2022), p. 117822342210921 (cit. on p. 1).

[5] Hanan Aljuaid, Nazik Alturki, Najah Alsubaie, Lucia Cavallaro, and Antonio Liotta. "Computer-aided diagnosis for breast cancer classification using deep neural networks and transfer learning". *Computer Methods and Programs in Biomedicine*, vol. 223 (2022), p. 106951 (cit. on p. 2).

[6] Kranti Kumar Dewangan, Deepak Kumar Dewangan, Satya Prakash Sahu, and Rekhram Janghel. "Breast cancer diagnosis in an early stage using novel deep learning with hybrid optimization technique". *Multimedia Tools and Applications*, vol. 81, no. 10 (Apr. 2022), pp. 13935–13960 (cit. on p. 2).

[7] Samir S Yadav and Shivajirao M Jadhav. "Deep convolutional neural network based medical image classification for disease diagnosis". *Journal of Big Data*, vol. 6, no. 1 (Dec. 2019), p. 113 (cit. on p. 2).

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. 2015 (cit. on pp. 2–4, 6).

[9] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015 (cit. on pp. 2, 4).

[10] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, 2010, pp. 807–814 (cit. on pp. 2–4).

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Identity Mappings in Deep Residual Networks*. 2016 (cit. on pp. 3, 4).

[12] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. *Deep Networks with Stochastic Depth*. 2016 (cit. on p. 3).

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269 (cit. on pp. 3, 4, 6).

[14] Muammer Turkoglu, Muzaffer Aslan, Ali Arı3, Zeynep Mine Alçin, and Davut Hanbay. "A multi-division convolutional neural network-based plant identification system" (2021) (cit. on p. 5).

[15] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. "Dermatologist-level classification of skin cancer with deep neural networks". en. *Nature*, vol. 542, no. 7639 (Feb. 2017), pp. 115–118 (cit. on p. 5).

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255 (cit. on pp. 5, 6).

[17] Uzair Khan. *Breast Cancer Patients MRI's Dataset*. URL: https://www.kaggle.com/datasets/uzairkhan45/breast-cancer-patients-mris (visited on 10/25/2022) (cit. on p. 5).

[18] Keras Team. *Keras Applications*. URL: https://keras.io/api/applications/ (visited on 12/15/2022) (cit. on p. 5).

[19] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014 (cit. on p. 5).