Comparing the transfer learning performance of three well-known neural network architectures on the BreastPathQ dataset.

S.A.O. Bunk   
BMTTU/e  
0995198

*Abstract*— Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec iaculis justo purus, nec lobortis mauris pellentesque sed. Phasellus et lectus non sapien rhoncus convallis. Nam sit amet libero libero. Phasellus et risus convallis, imperdiet purus vel, venenatis ligula. Vestibulum eget quam ipsum. Aenean porta ac leo ut ornare. Maecenas scelerisque risus justo, ut pellentesque nisi luctus vel. Donec ullamcorper, odio ut dignissim finibus, dolor quam vestibulum tortor, a rhoncus tortor nisi at nunc. Phasellus vitae varius arcu, in condimentum massa. Maecenas faucibus enim sed velit tincidunt, a lacinia dolor mattis. Donec eget magna in arcu imperdiet blandit. Sed aliquam est ac placerat tempor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec iaculis justo purus, nec lobortis mauris pellentesque sed. Phasellus et lectus non sapien rhoncus convallis. Nam sit amet libero libero. Phasellus et risus convallis, imperdiet purus vel, venenatis ligula. Vestibulum eget quam ipsum. Aenean porta ac leo ut ornare. Maecenas scelerisque risus justo, ut pellentesque nisi luctus vel. Donec ullamcorper, odio ut dignissim finibus, dolor quam vestibulum tortor, a rhoncus tortor nisi at nunc.

# Introduction

In 2017, in the US alone, an estimated 252,710 new cases of invasive breast cancer and 63,410 new cases of in situ breast cancer were diagnosed amongst women [1]. Every woman undergoing treatment has some form of analysis performed on them to determine the appropriate treatment for their situation. These treatment options include techniques such as radiation therapy and chemotherapy [2][3]. Treatment may lead to a reduction in size or disappearance of the tumour. However, it is also possible that the tumour remains the same size, but the cellularity of the tumour reduces, meaning the tumour now consists of a lower number of larger cells.

Assessment of the cancer cellularity of a tumour is done to determine the effectiveness of previously applied treatment. The pathological examination of tissue removed during surgery allows for the determination of tumour cellularity. In current clinical practice pathologists manually determine the cellularity of tissue slides, which is a highly subjective and labour-intensive task. The variability in observers reduces the reliability and quality of the cellularity assessment [4].

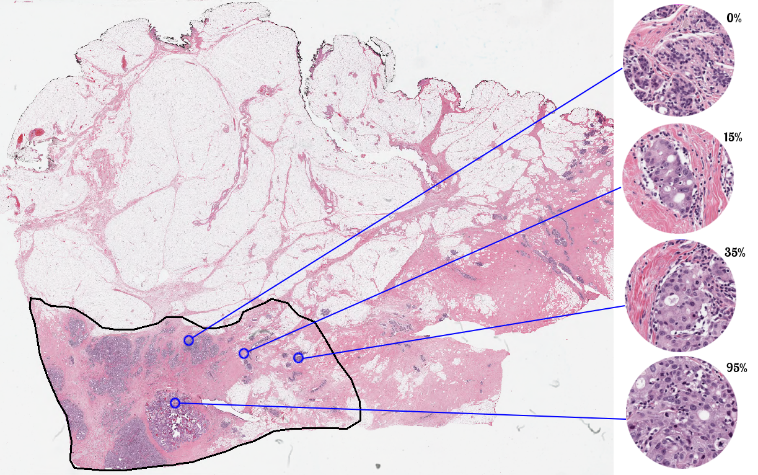


Figure 1: Haematoxylin and eosin stained slide used in the assessment of breast cancer cellularity. On the right the cellularity of specific points of the slide is estimated [4].

Automated assessment by image analysis can remove the intra- and inter-observer variability of cellularity determination and help increase the reproducibility of cellularity scores. In this study, we aim to assess cancer cellularity using deep learning techniques. Neural networks have achieved good results in histopathological image analysis tasks before and are highly useful tools for the computer-aided analysis of medical images [5][6][7].

There exist neural networks that are pre-trained and can easily be adapted to a different purpose. This process is called transfer learning and makes it so that even with a small dataset for the new task, the neural networks still achieve a good performance. A selection of networks we look at in this paper is InceptionV3 [8], VGG19 [9], Xception [10] and ResNet50 [11], all of which were pre-trained on the ImageNet dataset and are included with the Keras python library. The networks achieved top-5 accuracies on the ImageNet dataset of 0.937, 0.900, 0.945 and 0.921 respectively [12]. In this paper we re-train these networks to be used in the cancer cellularity determination task.

Assessing the effectiveness of any method on a certain task requires an evaluation metric. To compare automatically assessed cancer cellularity scores to manually obtained ones, a fair metric needs to be chosen. This study ventures to compare the three different networks with a multitude of metrics. We utilize existing metrics like Spearman correlation, Kendall’s tau [18], mean-squared error and prediction probability [13]. It is expected that networks which performed better on the ImageNet dataset will show the highest scores [14]. The results showed that Xception had the best scores for Kendall’s Tau, prediction probability, and Spearman correlation, but the worst score for mean squared error. VGG19 performed the worst according to the three metrics Xception scored the best in, and the best in the metric Xception scored the worst in. InceptionV3 scored between both other networks in every metric. ResNet50 did not produce proper results.

The dataset the networks will be applied on is the BreastPathQ dataset. This dataset is available in the SPIE-AAPM-NCI BreastPathQ: Cancer Cellularity Challenge [4].

# Methods

The BreastPathQ training set contains 2579 patches extracted from 96 whole slide images. The whole slide images were made from breast tissue obtained from 64 patients. The training set has one tumour cellularity score assigned per patch [4]. The training set was split up into three parts based on the patient ID to create our own training, validation and test datasets. 45 patients had their associated patches turned into 1489 train patches, 8 patients and their 429 patches became the validation dataset, and 10 patients and their 476 patches became the test dataset. This split is also shown in Table 1.

Table 1: The augmentation techniques and their range values.

|  |  |  |
| --- | --- | --- |
| Training Set | Validation Set | Test Set |
| 45 patients | 8 patients | 10 patients |
| 1489 patches | 429 patches | 476 patches |

## VGG19

Developed at the University of Oxford, this network was an attempt at refining the implementation of convolutional layers in the ImageNet classification task by Krizhevsky et al. [18]. The depth of the network was steadily increased by adding more convolutional layers onto the original architecture. VGG19 has a simple linear layout, using stacked convolutional layers [9]. The layout of these layers can be seen in Figure 2. VGG19 has a topological depth of 26 layers. These 26 layers consist of 143,667,240 parameters [12]. Compared to the architecture VGG19 is based on, it provided improved classification performance with a top-5 test error of 6.8% compared to the original top-5 test error of 16.4 on the ImageNet dataset [9].

Figure 2: The layers of the VGG network. The orange colour indicates a block of consecutive convolutional layers, with the amount indicated at the beginning. The grey blocks are dense layers, and the blue blocks are simple functional layers.

## InceptionV3

This network was developed at Google, evolving from GoogLeNet (InceptionV1) [15] into InceptionV2 [16] and finally into InceptionV3 used in this paper [8]. The InceptionV3 network has a fundamental building block known as an Inception module. An example of an Inception module as found in InceptionV3 can be seen in Figure 3. In the InceptionV3 network such modules are stacked behind one another to form the core of the InceptionV3 network. This is different from the VGG19 network, which was a simple stack of convolutional layers, with no parallel neurons. An Inception module is a convolutional feature extractor, but they have been empirically shown to perform better than a convolutional layer whilst having fewer parameters. The InceptionV3 network as available in Keras has 23,851,784 parameters, and a topological depth of 159 [12].

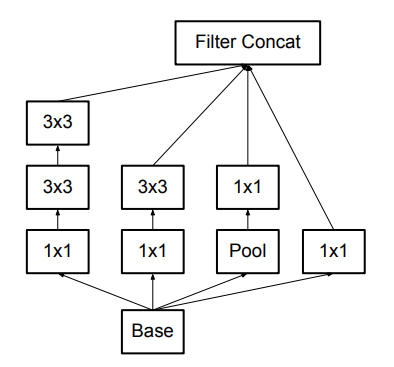
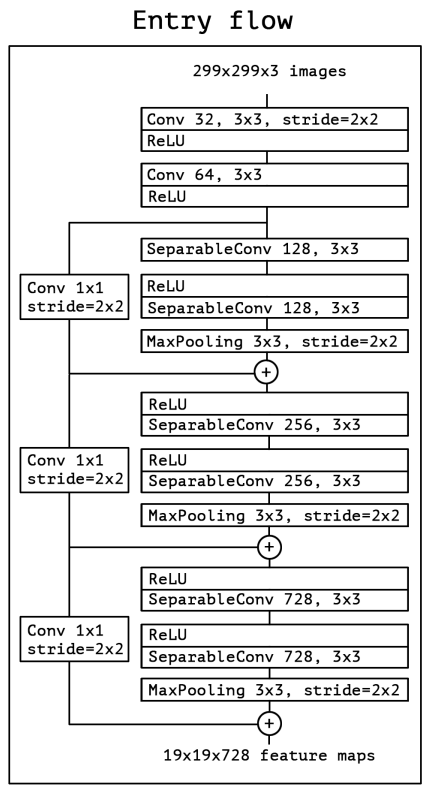
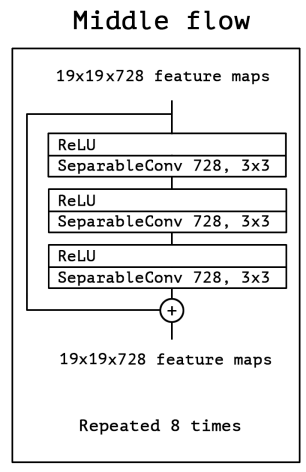


Figure 3: An Inception module as found in the InceptionV3 architecture. The InceptionV3 network consists of a stack of such modules. From [8].

## Xception

Like InceptionV3, Xception was also developed at Google, but by a different person. It builds on InceptionV3, by replacing its Inception modules with depthwise separable convolutions (a depthwise convolution followed by a pointwise convolution) [10]. This is done because the author feels that an Inception module with a maximally large amount of parallel convolutional layers can be seen as a depthwise separable convolution. Xception slightly outperforms InceptionV3 on the ImageNet dataset, and greatly outperforms InceptionV3 on a different, much larger dataset. This different and larger image classification dataset contains 350 million images belonging to one of 17,000 classes. The Xception architecture can be seen in Figure 4. As included in Keras, Xception has a topological depth of 126 layers, consisting of 22,910,480 parameters [12]. This similarity in parameter count of Xception and InceptionV3 means that the gain in performance of Xception is not down to increased parameters, but due to a better architecture.





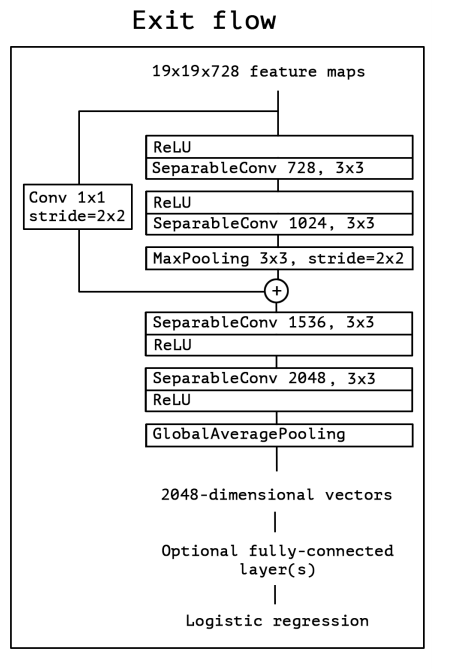


Figure 4: The building blocks of the Xception architecture. All Convolutional and Separable Convolutional layers are followed by batch normalization. Adapted from [10].

## ResNet50

Developed at Microsoft Research, ResNet50 gains its name from residual learning, where a shortcut connection skips a couple of layers. In the case of ResNet50, these shortcuts perform identity mapping, with their values simply being added to the output of the skipped layers [11]. This can be seen in Figure 5. ResNet50 is made from several such blocks stacked upon each other. The exact composition of ResNet50 can be seen in Figure 6. As implemented in Keras, ResNet50 has a topological depth of 50, with 25,636,712 parameters [12]. Even having so much fewer parameters than VGG19, ResNet50 still manages to outperform VGG19 on the ImageNet classification task, achieving a top-5 error of 5.25%.

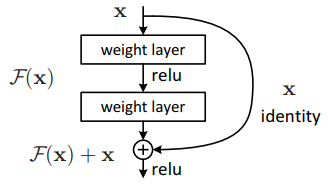


Figure 5: A small part of a residual learning network showcasing the identity mapping. From [11].

Figure 6: The ResNet50 architecture. Each of the convolutional blocks has a shortcut around them.

The choice was made to not include the top of the pre-trained networks InceptionV3, VGG19, Xception and ResNet50, as the top’s main function is to classify the input into the 1000 categories of the ImageNet dataset [8]. Instead, three custom layers were added to replace the top layers. These layers were constructed such that they would yield a single output, which would be the input image’s cellularity. These added custom layers are shown in Figure 7.

Figure 7: Layers added on top of the pre-trained networks.

A variety of data augmentation techniques were applied to the image patches. During training, these techniques were applied randomly to the image patches within a certain range as shown in Table 1.

Table 2: The augmentation techniques and their range values.

|  |  |
| --- | --- |
| Augmentation Technique | Value range (+/-) |
| Rotation | π |
| X translation | \*image width |
| Y Translation | \*image height |
| Rescaling | \*image size |
| X flipping | Yes |
| Y flipping | Yes |
| Per-pixel rescaling |  |
| Channel Shifting | 15 |
| Shearing | \*image size |
| Zooming | \*image size |

InceptionV3, VGG19 and Xception were trained for 100 epochs using the Adam optimizer with a learning rate of 0.001, with a mean squared logarithmic error loss. All but the three added custom layers had their weights frozen during the initial training of the added custom layers. The training was optimized to achieve the lowest possible loss on the validation set. Epoch were the validation loss was lower than the previous lowest validation loss had their weights saved.

After completing the initial training of the added custom layers, the InceptionV3 and Xception networks also had some of their convolutional layers re-trained. For the InceptionV3 network the first 41 layers and for the Xception the first 66 layers remained frozen, the rest of the layers was re-trained using the Adam optimizer with a learning rate of 0.0001 and the same loss as before. Again, the networks were optimized on the mean squared error of the validation set.

ResNet50 had all of it layers re-trained using the Adam optimizer with a learning rate of 0.0001 and the same mean squared logarithmic loss. ResNet50 was also optimized based on the mean squared error of the validation set.

Each network had four different instances of itself trained. For each of these four instances, predictions on the test set were produced. The predicted cellularity values on the test set were saved so that statistical analysis could be performed on the predictions of the test set. The predictions made on the test set were compared to the ground truth of the test set using four different statistical metrics, namely Kendall’s Tau, Prediction Probability, Mean Square Error and Spearman Correlation.

Kendall’s Tau is a measure for the rank relationship between two different variables and their observations. A value of 1 would mean a perfect relationship and 0 no relationship.

Prediction Probability is an adaptation of scipy’s implementation of Kendall’s Tau by the BreastPathQ challenge team [13] based on [19]. Prediction Probability shows the highest scores for networks that correctly rank the predictions, in other words; if the ground truth cellularity value of a patch is higher than that of another patch, the network that correctly predicts a higher cellularity value will score higher than those that don’t. This metric, along with Kendall’s Tau, is important if the goal of the network is to rank cellularity scores, and exact values aren’t overly important.

Mean Squared Error simply calculates how far off the predictions are from their ground truth, with a higher value meaning the predictions are further from the truth. This metric is important in cases where an exact cellularity value is required for the intended application of the network.

Spearman Correlation is also a measure for the rank relationship of two variables and is thus very similar in purpose to Kendall’s Tau.

# Results

Of each statistical metric obtained for each network, the mean was determined between the four trained instances of each network. These means can be seen in Table 2. It can be seen that Xception performed the best in three metrics, and VGG19 performed the worst the same three metrics. InceptionV3 was the middle performer in each metric. The performance in the mean squared error was the worst for Xception and the best for VGG19. A complete swap in ranking compared to their performance in the other three metrics.

Looking at the standard deviations in Table 4 does show something interesting as well. The standard deviation of VGG19’s scores is significantly lower than the standard deviation of InceptionV3’s and Xception’s scores. The standard deviation for the InceptionV3 and Xception are more similar, but Xception has the highest standard deviations.

Table 3: Mean metric values for each network and their four instances. Cells coloured green score the highest in their given metric and cells coloured red score the lowest. The uncoloured cells fall in between, aside from those belonging to ResNet50.

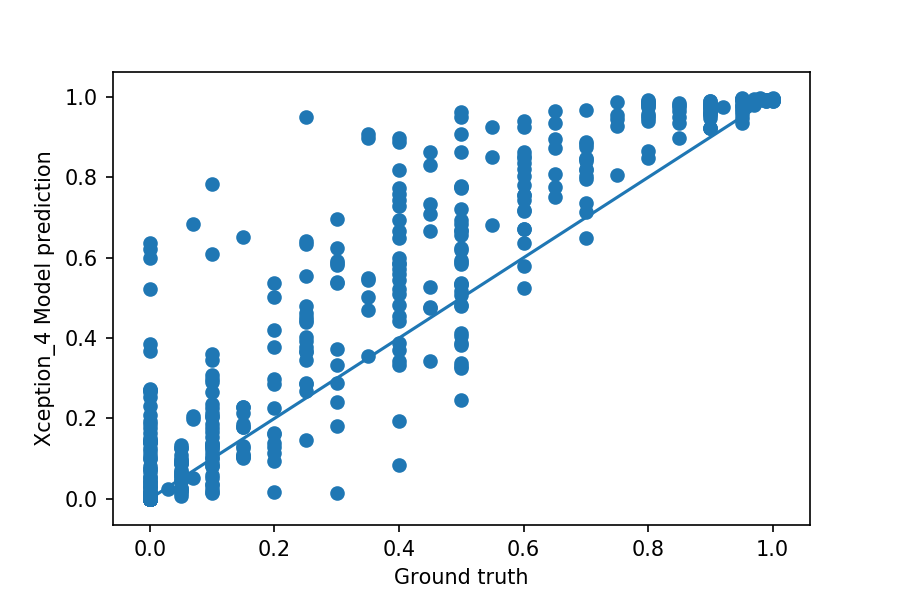
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Kendall's Tau | Prediction Probability | Mean Squared Error | Spearman Correlation |
| InceptionV3 | 0.70794 | 0.87822 | 0.02627 | 0.86496 |
| VGG19 | 0.66692 | 0.85630 | 0.02570 | 0.83327 |
| Xception | 0.73808 | 0.89433 | 0.03028 | 0.88256 |
| ResNet50 | 0.07075 | 0.53158 | 0.15687 | 0.09227 |

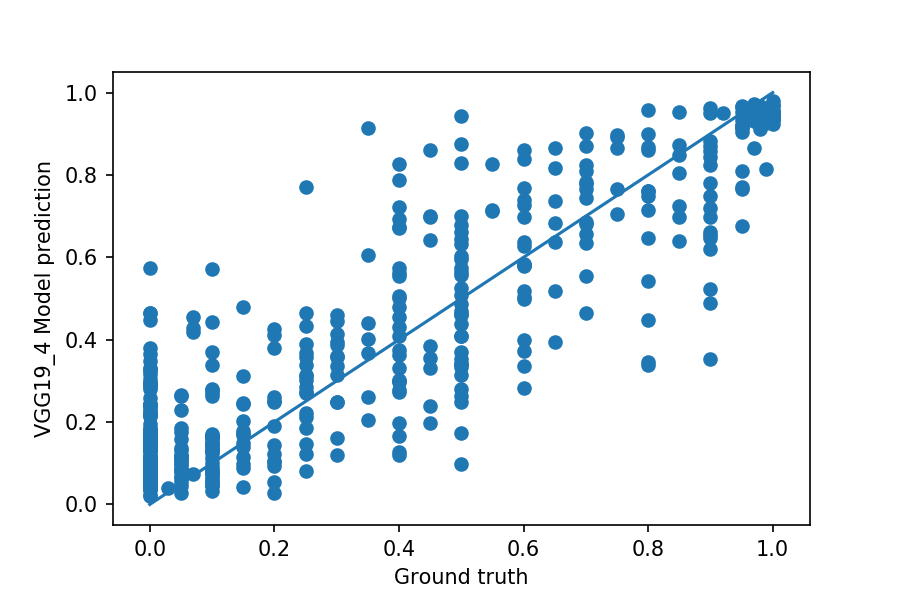
Table 4: The standard deviation of each metric for each network.

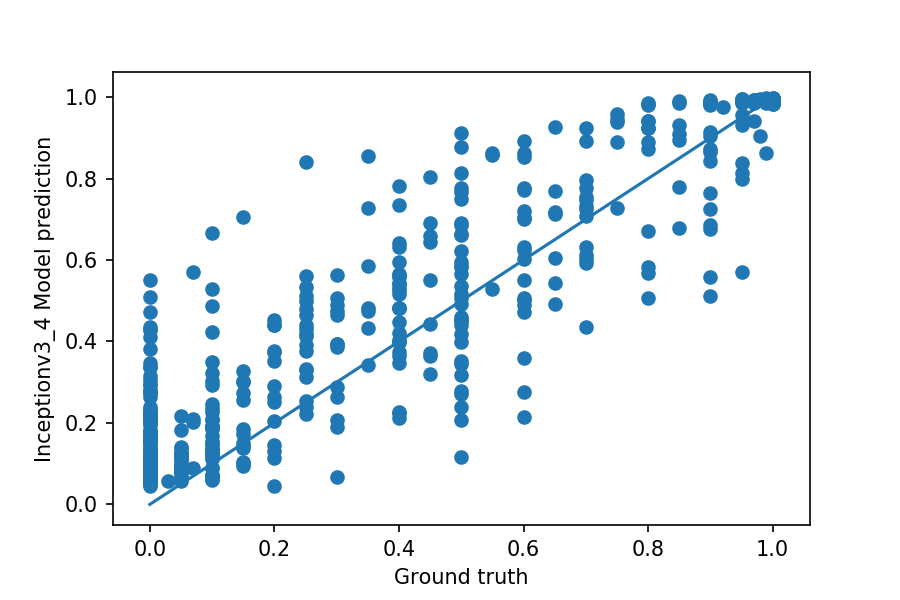
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Kendall's Tau | Prediction Probability | Mean Squared Error | Spearman Correlation |
| InceptionV3 | ±0.03922 | ±0.02096 | ±0.00711 | ±0.02950 |
| VGG19 | ±0.00755 | ±0.00403 | ±0.00188 | ±0.00544 |
| Xception | ±0.07630 | ±0.04076 | ±0.00852 | ±0.06006 |
| ResNet50 | ±0.08454 | ±0.03651 | ±0.0342 | ±0.10282 |

In Figure 8, predicted cellularity scores on the test have been plotted against the ground truth for the Xception, VGG19, InceptionV3 and ResNet50 networks. It can be seen that the Xception network often predicts higher values than the ground truth, whereas VGG19 predicts both higher and lower values than the ground truth with a similar frequency.

ResNet50 scored terribly in each metric, because of its terribly inaccurate predictions, as can also be seen in Figure 8 D. The fourth instance of the network never predicted a cellularity value higher than 0.4.







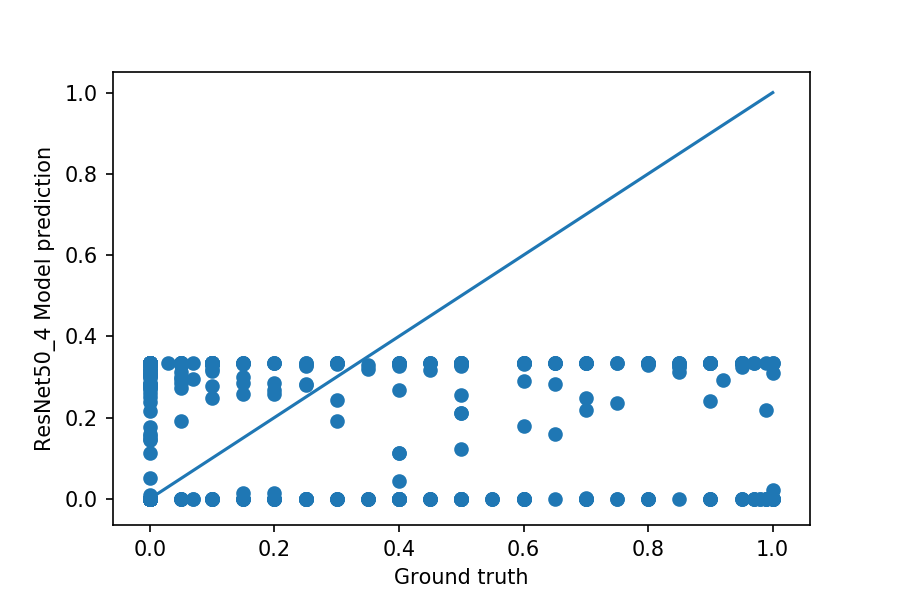


Figure 8: A: The prediction of the Xception 4 model plotted against the ground truth of the test dataset.   
B: The prediction of the VGG19 4 model plotted against the ground truth of the test dataset.  
C: The prediction of the InceptionV3 4 model plotted against the ground truth of the test dataset.   
D: The prediction of the ResNe50 4 model plotted against the ground truth of the test dataset.  
The blue line is where the dots should fall if the predictions were completely accurate.

# Discussion and Conclusion

As expected, the networks with a better performance on the ImageNet dataset scored better in the majority of metrics. However, it is surprising that Xception, which performed best in most metrics, had the highest mean squared error. As the other metrics all look at statistical correlation and not direct accuracy, this must mean that Xception was better at correctly predicting the relative ranking of patch cellularities, but was worse at predicting their exact values. This behaviour was indeed seen in Figure 3 A and B, where Xception predicted fewer cellularities lower than the associated ground truth value than VGG19 did. The cellularity values predicted by Xception were however usually higher than their ground truth counterparts.

Xception’s high performance is likely due to its application of depthwise separable convolution. As Xception is based off of InceptionV3, with the replacement of the Inception modules with depthwise separable convolutional layers being the only major difference. The higher performance of Xception as compared to InceptionV3 is a major indication for this assessment to be true.

From these results it becomes clear that the choice of metric with which to rank the performance of neural networks should be tailored to the desired application. If the goal is to rank the cellularities of different patches, utilizing metrics such as the prediction probability to grade the performance makes the most sense. However, if exact cellularity scores are required, utilizing a metric which grades networks on how close they come to the ground truth value would be the preferred option.

The reasons for ResNet50’s terrible performance on this data isn’t entirely understood. Different training runs produced vastly different predictions, which could mean that the random initialization of parameters has something to do with its terrible performance. It could also be that ResNet50’s use of identity mapping causes trouble when applied to this dataset, as this is the major distinguishing factor between ResNet50 and the other networks.

For future research it could be interesting to not only grade the network’s performance using a particular metric, but also use that metric as the loss during the training of the network. This network could then be compared to a network trained with a different loss metric than the grading metric, to see how they each perform with regards to the metric used to actually grade the network performance. A requirement for this to be possible is that the loss metric is differentiable.

##### Acknowledgment

Thanks go out to S.C. Wetstein for providing guidance during the execution of the performed research and associated writing of the paper.

##### References

1. American Cancer Society. *Breast Cancer Facts & Figures 2017-2018*. Retrieved from <https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/breast-cancer-facts-and-figures/breast-cancer-facts-and-figures-2017-2018.pdf>
2. Early Breast Cancer Trialists' Collaborative Group. (2011). Effect of radiotherapy after breast-conserving surgery on 10-year recurrence and 15-year breast cancer death: meta-analysis of individual patient data for 10 801 women in 17 randomised trials. *The Lancet*, *378*(9804), 1707-1716.
3. Von Minckwitz, G., Untch, M., Blohmer, J. U., Costa, S. D., Eidtmann, H., Fasching, P. A., ... & Jackisch, C. (2012). Definition and impact of pathologic complete response on prognosis after neoadjuvant chemotherapy in various intrinsic breast cancer subtypes. J Clin Oncol, 30(15), 1796-1804.
4. Grand-Challenges. *SPIE-AAPM-NCI BreastPathQ: Cancer Cellularity Challenge.* Retrieved from <https://breastpathq.grand-challenge.org/>
5. Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.
6. Veta, M., Heng, Y. J., Stathonikos, N., Bejnordi, B. E., Beca, F., Wollmann, T., ... & Hedlund, M. (2019). Predicting breast tumor proliferation from whole-slide images: the TUPAC16 challenge. *Medical image analysis*, *54*, 111-121.
7. Wang, D., Khosla, A., Gargeya, R., Irshad, H., & Beck, A. H. (2016). Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*.
8. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).
9. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
10. Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).
11. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
12. Keras, Applications, retrieved from <https://keras.io/applications/>
13. SPIE-AAPM-NCI BreastPathQ: Cancer Cellularity Challenge 2019, retrieved from: <http://spiechallenges.cloudapp.net/competitions/14>
14. Kornblith, S., Shlens, J., & Le, Q. V. (2018). Do better imagenet models transfer better?. *arXiv preprint arXiv:1805.08974*.
15. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.
16. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of The 32nd International Conference on Machine Learning, pages 448–456, 2015.
17. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
18. Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, *30*(1/2), 81-93.
19. Smith, W. D., Dutton, R. C., & Smith, N. T. (1996). A measure of association for assessing prediction accuracy that is a generalization of non‐parametric ROC area. *Statistics in Medicine*, *15*(11), 1199-1215.