Histopathologic Cancer Detection C3

Konrad Gerlach
konrad.gerlach@student.hpi.uni-potsdam.de

Frederic Sadrieh
frederic.sadrieh@student.hpi.uni-potsdam.de

Jacob Schäfer
jacob.schaefer@student.hpi.uni-potsdam.de

Alexander Sohn
alexander.sohn@student.hpi.uni-potsdam.de

Till Wenke
till.wenke@student.hpi.uni-potsdam.de

Abstract

Cancer is one of the deadliest diseases in the world. Especially when unnoticed, it can spread rapidly through the body. Therefore, detecting cancer quickly and reliably is of utmost importance. Histopathologic scans are scans of the human body at a microscopic level. Analyzing these scans, can help to diagnose certain diseases like cancer. It requires a trained professional to analyze these images and find patterns that could be an indicator for cancer. Deep Neural Networks can automate the prediction process and spot cancer cells more precisely. In this report, we implement a Convolutional Neural Network (CNN) able to achieve 95% prediction accuracy and an AUC-score of 0.9402, a sensitivity of 0.9427, a precision of 0.948, a specificity of 0.9647 and an F1-score of 0.9453 on unseen histopathologic scans. Furthermore, we realize feature visualization and saliency maps to allow, the user to see, on what basis the model predicts. With this, we can deduct learnings about our model and the dataset. For example, we can see what our CNN thinks a cancerous cell is, or we can prove that the model really learns to spot cancer and doesn't use other information from the dataset not intended for the classification (e.g., all cancerous scans are done with one microscope and the others with another). With this, we can detect problems within the dataset.

1 Introduction

Histopathologic scans are an important tool in medicine. While analyzing the human tissue on a microscopic level, a pathologist can detect different diseases like cancer. These tests are often the core of cancer detection [5]. With evermore scans and a shortage of experienced pathologists, automated or semi-automated detection methods are needed [2]. With this, the expert can focus on the difficult cases, while the detection for most cases is done for him.

Many advances in computer science, in recent years, have paved the way to solve these problems. With deep neural networks, cancer detection from histopathologic scans is possible and when trained on enough data, they can outperform a trained expert's judgment [2]. It has been shown that Convolutional Neural Networks, perform the best for image classification, because they can detect patterns in adjacent pixels[14].

Therefore, this project aims at building such a deep neural network, based on pathology images from the 'Histopathologic Cancer Detection' competition on Kaggle [8]. The goal is a Convolutional Neural Network, that can predict if a histopathologic scan shows signs of cancer. In this dataset, only the center of the scan is important for the decision. The minimum accuracy goal we set for our model is to achieve an accuracy of over 60%, because a random guess would have such an accuracy.

Furthermore, we aim to show the user how the model decides. Therefore, we implement different types of visualization as a form of explainable AI (XAI) [4]. This is important because it allows for a second opinion of a pathologist. Only the most important parts can be shown to him and he can skip the time-consuming task of finding these. In addition, for a patient, a cancer diagnosis is a life-changing event. For him to see why the diagnosis was made helps to raise acceptance.

2 Approach

2.1 Data Set

Our dataset is taken from the Kaggle Competition 'Histopathologic cancer detection' [8]. This dataset builds up on the PCam dataset [19]. The original PCam dataset contains duplicate images due to its probabilistic sampling, however, the version presented on Kaggle does not contain duplicates. The dataset consists of 220025 train images and 57458 test images of size 96x96 pixels. Special about the dataset, is that the labels only apply to the 32x32 pixel patch around the center. So a picture is classified as cancer, if and only if there is at least one pixel of cancer tissue in this 32x32 pixel region. The dataset is designed this way, to enable fully convolutional architectures without the need of padding. Of the training dataset 130908 images (59.5%) are labelled as non-cancer and 89117 (40.5%) are labelled as cancer. The images are uncompressed RGB images in the tiff format. For our training, we used a portion of the training dataset as a test dataset. We sampled $\frac{1}{3}$ randomly and used it as a test dataset. We made sure, to never mix test and training dataset during training and evaluating, by seeding the random generator.

2.2 Data Preparation

We used both a RGB and grayscale version of the dataset. For the transformation from RGB to grayscale, we used the built-in transformations of the torchvision library. The torchvision library relies on the python image library (pil) for this transformation.

2.3 Neural Network Architectures

We chose a convolutional architecture for our network. Convolutional neural networks are common for image classification tasks and powerful. We looked at three different well known CNN architectures. Namely `VGG` [16], `LeNet` [11] and `AlexNet` [9] and developed our own inspired by them. We visualized the layers of our network in 1 with their different operations. We start with a fully convolutional part, that consists of 4 Layers. After that is a fully connected part with 3 Layers. We called our architecture `Big Konrad`, because of the large number of parameters in comparison to the other architecuters. Later we refer to our net by the name colour net if it was trained on the RGB data and `grayscale net` if it was trained on the grayscale data. To prevent overfitting, we used a technique called Dropout. During the learning phase, some neurons are temporarily, randomly dropped out of the netork while the remaining neurons are overweighted to retain the neuron's expected values [17]. We used different dropout rates in the fully convolutional and the fully connected part. To improve the convergence of our model, we used the technique of batch normalization proposed by Ioffe and Szegedy [7]. As an activation function, we used `LeakyReLU` throughout the network. As an optimizer, we used `Adam`. We also tested `sdg` and `rmsprop`. To determine optimal hyperparameters for our model, we conducted different sweeps to test out the parameters. Our sweeps resulted in the following hyperparameters:

1. Dropout convolutional Layer 0
2. Dropout fully connected 0.5
3. Weight decay 0

Our objective function is the binary cross entropy.

Layer	Operations

1	Dropout(p=0)
	Conv2d(i=3,o=128, kernel_size=7, padding=’same’)
	BatchNorm
	LeakyReLU
	MaxPooling(kernel_size=2, stride=2, padding=0)

2	Dropout(p=0)
	Conv2d(i=128, o=256, kernel_size=5, padding=’same’)
	BatchNorm
	LeakyReLU
	MaxPooling(kernel_size=2, stride=2, padding=0)

3	Dropout(p=0)
	Conv2d(i=256, o=512, kernel_size=3, padding=’same’)
	BatchNorm
	LeakyReLU
	MaxPooling(kernel_size=2, stride=2, padding=0)

4	Dropout(p=0)
	Conv2d(i=512, o=64, kernel_size=1, padding=’same’)
	BatchNorm
	LeakyReLU
	MaxPooling(kernel_size=2, stride=2, padding=0)

5	Flatten Dropout(p=0.5) Linear(in=2304, on=200) BatchNorm LeakyRelu

6	Dropout(p=0.5)
	Linear(in=200, on=400)
	BatchNorm LeakyReLU

7	Dropout(p=0.5)
	Linear(in=400, on=1)

Table 1: The layers and their operations

2.4 Feature Visualization

In order to understand the decision making process of our neural nets, we decided to implement some of the approaches to feature visualization described by Olah, Mordvintsev, and Schubert[13]. These approaches focus on finding example inputs which maximally or minimally activate neurons. This can help in deducing a neuron’s function within the neural net.

2.4.1 Dataset examples

A simple approach to understanding neuron activations is to find examples within the dataset which either maximize or minimize the activation of the neuron in question. We iterate over the dataset and calculate the forward pass for each example. For activation minimization the loss function is the activation of the neuron before any non-linearities are applied and for maximization this loss function is negated.

Figure 1 and 2 show images which activate the colour net’s class logit minimally/maximally respectively and thus display the most and least cancerous images according to the net.

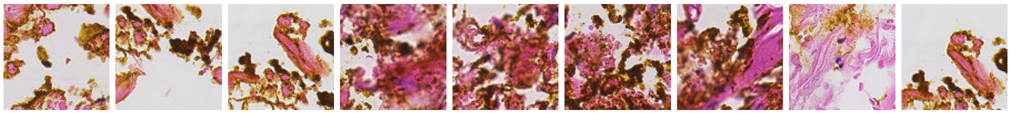


Figure 1: The least cancerous images according to the coloured net (all are non-cancerous)

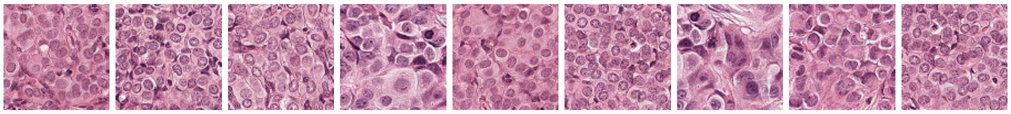


Figure 2: The most cancerous images according to the colour net (all are cancerous)

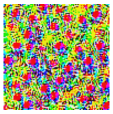
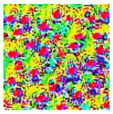
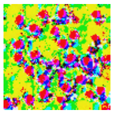
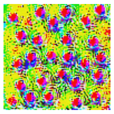
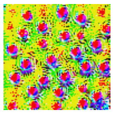
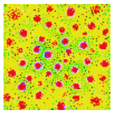
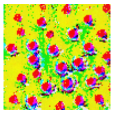
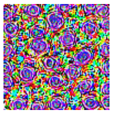
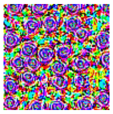
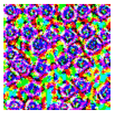
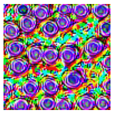
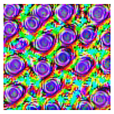
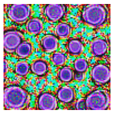
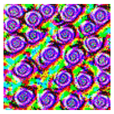
Minimization							
Maximization							
Blur standard deviation.	-	0.5	1	-	-	-	0.75
max. Rotation	-	-	-	5°	15°	180°	15°
max. Translation Fraction	-	-	-	5%	10%	20%	10%
Scaling Factor Range	-	-	-	0.9-1.1	0.8-1.2	0.5-1.8	0.8-1.2

Table 2: Different regularization configurations for feature visualization in colour Big-K model

2.4.2 Visualization by optimization

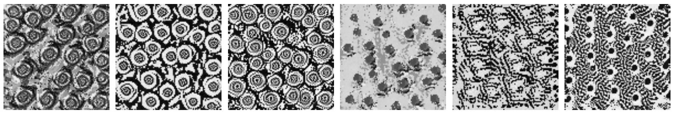
Visualization by optimization generates sample images which maximize or minimize the activation of neurons. We initially generate a random input image. This image is then fed through the forward pass of our network and the loss is calculated. The loss function is again the activation or negated activation of the neuron in question. Instead of adjusting the network's weights to minimize the loss functions - as one would during training - we calculate the derivative of the loss with respect to the input image and adjust the image to minimize the loss function. This process is then repeated until we arrive at a sufficiently optimal sample image.

It is important to mention, that this approach does not yield sample images following the same distribution as the images from our dataset are sampled from. Olah, Mordvintsev, and Schubert specifically mentions the issue of high frequency noise patterns being present within these generated samples and that they seem to be closely related to adversarial examples. Adversarial examples use non-random, imperceptible perturbations of an input image to cause the neural network to misclassify the image [18]. As adversarial examples obviously undermine the purpose of our net we are not interested in visualizing them, as they do not help in understanding the net's actual purpose. Olah, Mordvintsev, and Schubert propose multiple different techniques to mitigate these patterns via regularization.

One of the techniques is to enforce the transformation robustness of the generated sample image. This is achieved by applying a random affine transformation, i.e. randomly rotating, scaling and translating the sample input, before calculating the forward pass.

Olah, Mordvintsev, and Schubert. also outline a more direct way of addressing the high frequency patterns through frequency penalization. This can be achieved implicitly by blurring the input image before the above mentioned transformations are applied.

Table 2 shows different regularization settings applied to maximizing and minimizing the class logit (output layer) of our colour trained Big-K model. The images are initially randomly generated using a random multivariate uniform distribution and then optimized using an Adam optimizer with learning rate 0.01 for 1500 steps. Gaussian Blur is applied with a kernel size of 5 and the image is translated in both x and y directions.



(a) grayscale conversion of color net class logit maximization (b) grayscale image to maximize class logit of colour net (c) grayscale net class logit maximization (d) grayscale conversion of color net class logit minimization (e) grayscale image to minimize class logit of colour net (f) grayscale net class logit minimization

Figure 3: Grayscale maximizations and minimizations of class logits for grayscale net and colour net

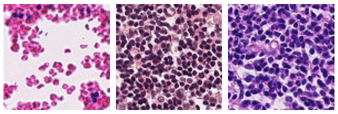
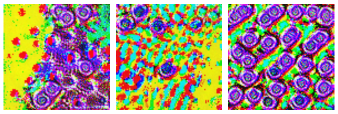


Figure 4: Dataset images without cancer containing the typical small round cells

The combination of the random affine transformation with blurring seems to produce the best results and reduce noise the most. If not otherwise specified, the rightmost settings will be used for all further feature visualizations throughout this report. A blur with standard deviation of 0.75 was chosen as 0.5 did not reduce noise sufficiently and 1 was too high in that it reduced the level of detail by producing large segments containing only a single colour. The latter can be explained by these segments being blurred together and thus being more smoothly blended during training.

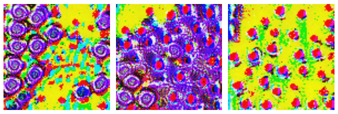
Figure 3 shows the class logit maximization (Figure 3c) and minimization (Figure 3f) for our grayscale Big-K model compared to the optimized images for our colour Big-K model converted to grayscale (Figure 3a, 3d). We also optimized grayscale images to maximize (Figure 3b) /minimize (Figure 3e) the class logit of the colour net shown in the same figure.
All three maximizations contain the same distinct rosette pattern. The grayscale version for the colour net (3a) contains channel-like shapes between the rosette patterns, which do not appear in the more densely packed grayscale net's image (3c). However the grayscale optimized image for the colour net (3b) does not show these channels either. This seems to indicate that their colour is more important than their structure for the activation of the class logit. The rosette patterns seem similar to the larger cells contained in the cancerous images of our dataset. This is also indicated by the purple coloring of the rosettes observed in Table 2, which is similar to the cells in images containing cancer shown in Figure 4.
The minimizations differ more strongly. The main similarity lies in Figures 3e,3d and 3f containing many small dark spots, which are similar to the healthy cells contained in our dataset. Both Figure 3e and 3f also contain very high frequency ridge patterns.

2.4.3 Combined analysis



(a) neuron 1 (b) neuron 2 (c) neuron 3

Figure 5: first linear layer neuron minimization



(a) neuron 1 (b) neuron 2 (c) neuron 3

Figure 6: first linear layer neuron maximization

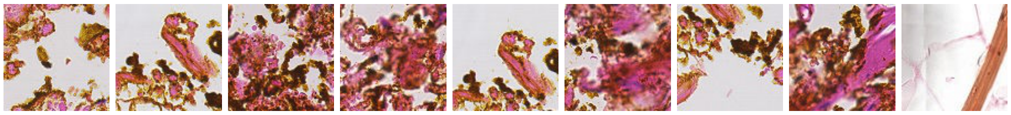


Figure 7: data examples which maximize the activation of neuron 3 in the first linear layer (all are non-cancerous)

We are now going to analyze the front layers of the colour net using both visualization by optimization and dataset examples. Figure 5 and 6 show images which maximally/minimally activate the first three neurons of the first linear layer of our model. We can observe, that these images are compositions of the same basic patterns shown in Table 2 just arranged differently. It is thus questionable how much added value the linear layers provide when even the first linear layer is capable of detecting high-level patterns. As the positions of the cancerous part (purple) and the non-cancerous parts of the image (yellow-green) are swapped by maximization and minimization it seems that the neurons just detect different cancer locations (e.g. Is cancer located at the edges of the image or at its center? for neuron 1). Additionally Figure 7 and 8 seem to indicate, that neuron 3 is already capable of functioning as a cancer detector all of its own.

2.5 Saliency Maps

In order to see what parts of actual histopathologic images make the model decide that it contains cancer and to finally find out how a cancer cell looks like, we used saliency maps. They are another neural network interpretation technique as presented by Simonyan, Vedaldi, and Zisserman [15]. Through this tool we aim to determine which parts of an image are important for the model's decision and we can see if it pays attention to the right parts i.e. cancerous cells.

2.5.1 Setup of Saliency Maps

To determine how strongly one value/ pixel in the input image influences the models decision, we fed dataset input images into the model and backpropagated to get the gradient of the output w.r.t. all input values as described by Simonyan, Vedaldi, and Zisserman. Intuitively a value with gradient of large magnitude indicates that this value has to be changed just slightly to affect the class score. Meaning the model pays special attention to this value when making its decision [15]. Therefore one can conclude that these should be the potentially cancerous cells/ pixels when given a cancerous image.

For visualization of the gradients we use colourmaps colouring different gradient values in different colours. We used `inferno` (black-purple-yellow) [1] As a perceptually uniform sequential colourmap enabling the viewer to perceive even smallest differences [12].

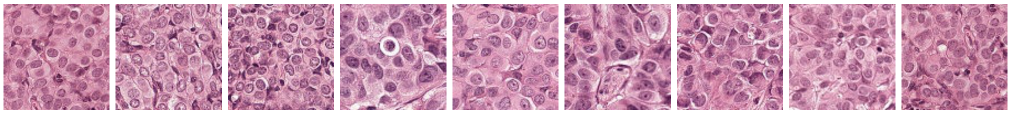


Figure 8: data examples which minimize the activation of neuron 3 in the first linear layer (all are cancerous)

Figure 9: The inferno color map for small to big values [1]

One way to look at the images is channel-wise in Figures 10 and 11. Although the red-channel is the one that seems most important with the most yellowish region, all of them draw a quite similar picture. Consequently it is valid to take the largest absolute values across all channels for one pixel to point out which pixel in the image influenced the models decision the most as suggested by Simonyan, Vedaldi, and Zisserman [15]. One can see that this also leads to a more smoothly coloured saliency map.

Not only from the samples in Figures 10 and 11, we can assume that for correctly predicted cancerous images multiple regions in the image are important for its decision. Whereas for correctly predicted non-cancerous images there is often only one larger or even no region of high interest for the model.

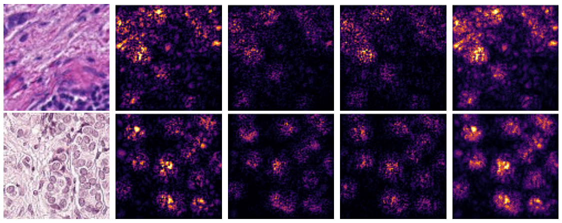


Figure 10: From left to right: cancerous image with class logits above 0.999 and corresponding saliency maps for its red, green, blue channel and for all channels together

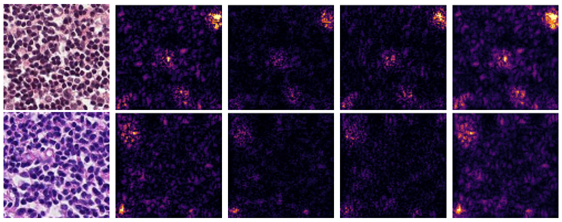


Figure 11: From left to right: non-cancerous image with class logits below 0.001 and corresponding saliency maps for its red, green, blue channel and for all channels together

2.5.2 Learnings About Our Models

According to Figure 12 our model definitively learned certainty on which parts of an image are relevant for a classification. In the far worse colour model (83% test accuracy) the same regions are of interest, but they are narrowed down in the better model. In addition we can also see that our grayscale model does not pay attention to the same regions as Big-K. However it is remarkable that we can see almost same sized larger circles of interest but again it does not focus on particular spots as much as the colour model. Knowing this, 93% test accuracy for this model is a quite good result but it might also relate to that we reached a plateau while training.

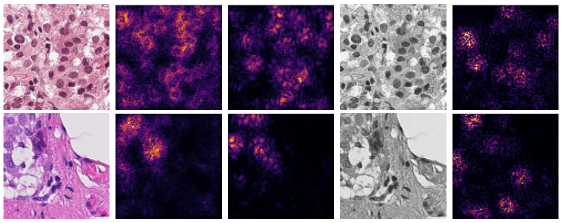


Figure 12: From left to right: a cancerous input image with a saliency map from a bad and our best model and the same image in grayscale and its saliency map using our gray scale model

Figure 13 shows us that the Big-K model does not always concentrate on the center of the image. However the images are just labeled based on cancer pixels in the 32x32 center of the image. It can be concluded that it did not learn the correct attributes of such cancerous images. It rather also uses knowledge about the larger surrounding of cancer cells to make its classification which was only introduced for convenience reasons [8].

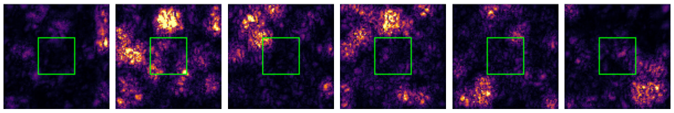


Figure 13: Examples of saliency maps across all channels for cancerous images - inner 32x32 pixels are surrounded by a green square

2.5.3 Learnings About Cancer

In order to see which structures in the histopathologic images were of interest for the model, we surrounded them in red in Figure 14. An area was marked when it lies in the 5 pixel surrounding of one of the 0.01% highest gradient pixels. Thereby cancer cells should be highlighted. And we finally know what Big-K thinks is a cancer cell.

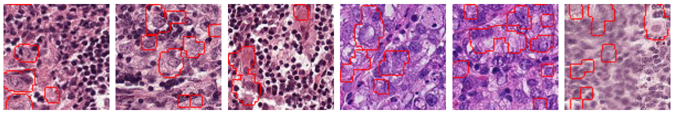


Figure 14: Regions the model found most interesting surrounded in red for cancerous images

Under the assumption that circular structures are some sort of cells, we can take away that it points on irregular structures such as larger cells among a majority of smaller ones, areas that not seem to include any cell or seem to be blurred. However not all similar looking regions seem to be of interest for our model even in one image. We cannot determine why one region is marked whereas another one is not. Thus we cannot generalize what specific structures go along with a cancerous image.

By applying the techniques of this chapter to artificial inputs from the previous one in Figure 15 we can confirm that the model classifies as cancerous when there are big purple circles and small pink cells are a sign for no cancer. One could assume that these are perfect cancer and healthy cells, respectively. As seen in Figure 15 Big-K identifies these structures very accurately.

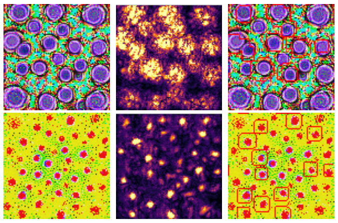


Figure 15: Regions of interest from highly (top) and non-cancerous (bottom) artificial inputs from Feature Visualization via saliency maps

3 Evaluation

The goal of our model is to help identify cancerous and non cancerous images. Therefore, we should evaluate the models capability to distinguish between those two classes of images. A common metric to do so is calculating the Area under the ROC(receiver operating characteristic) Curve. When calculating a ROC curve, the idea is to measure the true positives rate (TPR) and the false positives rate (FPR) at different classification thresholds. To get one point of the curve, you would first set the threshold to i.e. 30%, so that all images for which the model predicts a probability of more than 30% are classified as cancerous. Then, the TPR and FPR is calculated for this specific threshold.

In submitting predictions for unlabeled test data to kaggle [8], we received a AUC-Score of 0.9393. Thus, our model is fairly capable of distinguishing between the classes. However, a lot of the submissions for the initial kaggle competition perform even better.

One should note, that we didn't use this metric to optimize our model. Instead, we focused on accuracy. This is known to be a pitfall when training classification models. However, as described above, the train data set is more or less balanced and a high accuracy is, while not being optimal, a useful metric to evaluate and optimize the models performance.

With the goal of optimizing accuracy, we performed sweeps to fix important hyper parameters.

In doing so, we were able to train our model (which we described throughout the report) to 95% test accuracy.

Further, when training the model for an extended time, we observed that the train accuracy reaches more than $99,97\%$. In the meantime, the test accuracy does not improve significantly above 95%. Although the model is still able to generalize, it has learned the provided train data by heart. This is an indicator for the model being too large for the performed task. A model with less parameters is probably able to achieve a comparable test accuracy.

We can conclude that our model is indeed able to distinguish between cancerous and non cancerous images. However, the models performance and it's size can be improved.

4 Conclusion

During the project we succeeded in developing a well performing model. Further, we developed visualizations that help to understand how the model forms it's decision.

A next step could be to conduct pathologists in order to verify our and the models assumptions about what cancerous and non-cancerous parts of the images are - indicating how well we could trust the highlighting in Learnings About Cancer. With this, saliency maps may even have a practical use case, in a real world context. Having the predicted label and the important areas outlined could help a doctor in deciding whether the model is correct more quickly. A next step, to further develop this feature, would be to talk to doctors that could make use of it.

Further, we identified several aspects in which we can improve our model. For instance, we want to optimize hyper parameters using different metrics, such as AUC.

We also want to investigate the topic of clean-It. Deep Learning models are known for requiring a lot of computing resources and thus electrical energy. Decreasing factors such as model size or training time can significantly reduce the required energy consumption.

5 CO2 Emission Related to Experiments

Experiments were conducted using Google Cloud Platform and private resources. We assume that we would have used Google Cloud Platform only, in their most carbon emitting region asia-south1[10] (as we had no influence on the region), which has a carbon efficiency of 0.92 kgCO$_2$eq/kWh. A cumulative of 240 hours of computation was performed on hardware of type Tesla K80 (TDP of 300W)[3].

Total emissions are estimated to be 66.24 kgCO$_2$eq of which 100 percents were directly offset by the cloud provider. We take into account that this can only be an approximation [6].

Estimations were conducted using the MachineLearning Impact calculator presented in [10].

References

[1] *Choosing colormaps in matplotlib.* URL: https://matplotlib.org/stable/tutorials/colors/colormaps.html#mycarta-banding.

[2] Miao Cui and David Y. Zhang. “Artificial intelligence and computational pathology”. In: *Laboratory Investigation* 101.4 (Apr. 2021), pp. 412–422. ISSN: 1530-0307. DOI: 10.1038/s41374-020-00514-0. URL: https://doi.org/10.1038/s41374-020-00514-0.

[3] David. *Alternative to colab pro: Comparing Google’s Jupyter notebooks to gradient notebooks (updated!)* July 2022. URL: https://blog.paperspace.com/alternative-to-google-colab-pro/.

[4] David Gunning and David Aha. “DARPA’s Explainable Artificial Intelligence (XAI) Program”. In: *AI Magazine* 40.2 (June 2019), pp. 44–58. DOI: 10.1609/aimag.v40i2.2850. URL: https://ojs.aaai.org/index.php/aimagazine/article/view/2850.

[5] Lei He et al. “Histology image analysis for carcinoma detection and grading”. en. In: *Comput Methods Programs Biomed* 107.3 (Mar. 2012), pp. 538–556.

[6] Peter Henderson et al. “Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning”. In: *CoRR* abs/2002.05651 (2020). arXiv: 2002.05651. URL: https://arxiv.org/abs/2002.05651.

[7] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 448–456. URL: https://proceedings.mlr.press/v37/ioffe15.html.

[8] Kaggle. *Histopathologic Cancer Deteciton*. Kaggle, 2018.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[10] Alexandre Lacoste et al. “Quantifying the Carbon Emissions of Machine Learning”. In: *arXiv preprint arXiv:1910.09700* (2019).

[11] Yann LeCun et al. “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE*. Vol. 86. 11. 1998, pp. 2278–2324. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7665.

[12] Matteomycarta et al. *The rainbow is dead...long live the rainbow! – the rainbow is dead...long live the rainbow! – perceptual palettes, part 4 – cie lab heated body*. Dec. 2014. URL: https://mycartablog.com/2012/10/14/the-rainbow-is-deadlong-live-the-rainbow-part-4-cie-lab-heated-body/.

[13] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. “Feature Visualization”. In: *Distill* (2017). https://distill.pub/2017/feature-visualization. DOI: 10.23915/distill.00007.

[14] Neha Sharma, Vibhor Jain, and Anju Mishra. “An Analysis Of Convolutional Neural Networks For Image Classification”. In: *Procedia Computer Science* 132 (2018). International Conference on Computational Intelligence and Data Science, pp. 377–384. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.198. URL: https://www.sciencedirect.com/science/article/pii/S1877050918309335.

[15] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2013. DOI: 10.48550/ARXIV.1312.6034. URL: https://arxiv.org/abs/1312.6034.

[16] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556 (2014). URL: http://arxiv.org/abs/1409.1556.

[17] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[18] Christian Szegedy et al. “Intriguing properties of neural networks”. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6199.

[19] Bastiaan S Veeling et al. “Rotation Equivariant CNNs for Digital Pathology”. In: (June 2018). arXiv: 1806.03962 [cs.CV].