Identification the presence of metastases from histopathology images

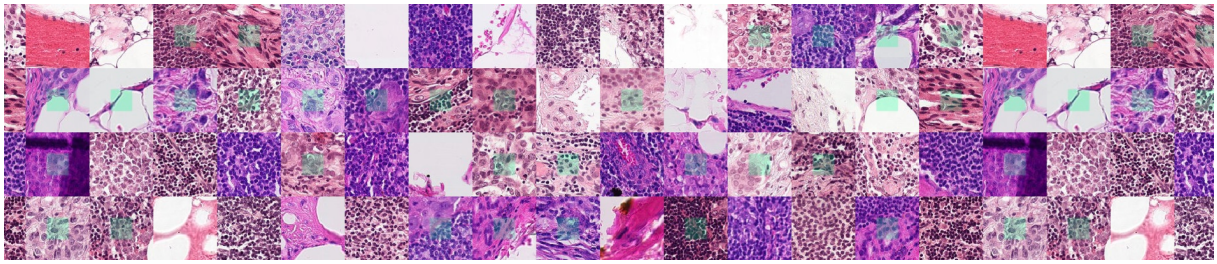
Maciej Draguła

5 February 2020

1 Introduction

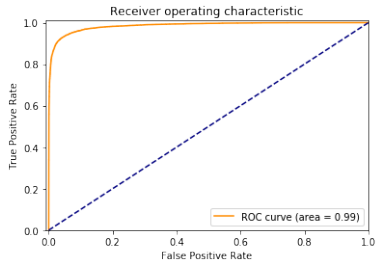
What is the problem?

The goal is to identify the presence of metastases from 96 x 96px digital histopathology images so this is a binary image classification problem. One of possible challanges is that the metastases can be as small as single cells in a large area of tissue. The problem is taken from Kaggle.



What is exactly on the images?

The histopathological images are microscope images of lymph nodes that were stained with hematoxylin and eosin (H&E). H&E allows to detect various objects like nuclei (blue), cytoplasm and extracellular parts (both red and pink). Hematoxylin binds to negatively charged substances like nucleic acids and pink eois to positively charged substances like amino acids side chains which makes visible difference on the image. According to American Cancer Society lymph nodes are the first place a breast cancer is likely to spread.



How to find the best solution?

In binary classification one of the good metric is area under the ROC curve. The ROC curve is a plot of True positive rate against False positive rate at various thresholds. The area under the

curve is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one so the bigger the area under the curve the better classifier we have. You can see an example of almost perfect classifier below.

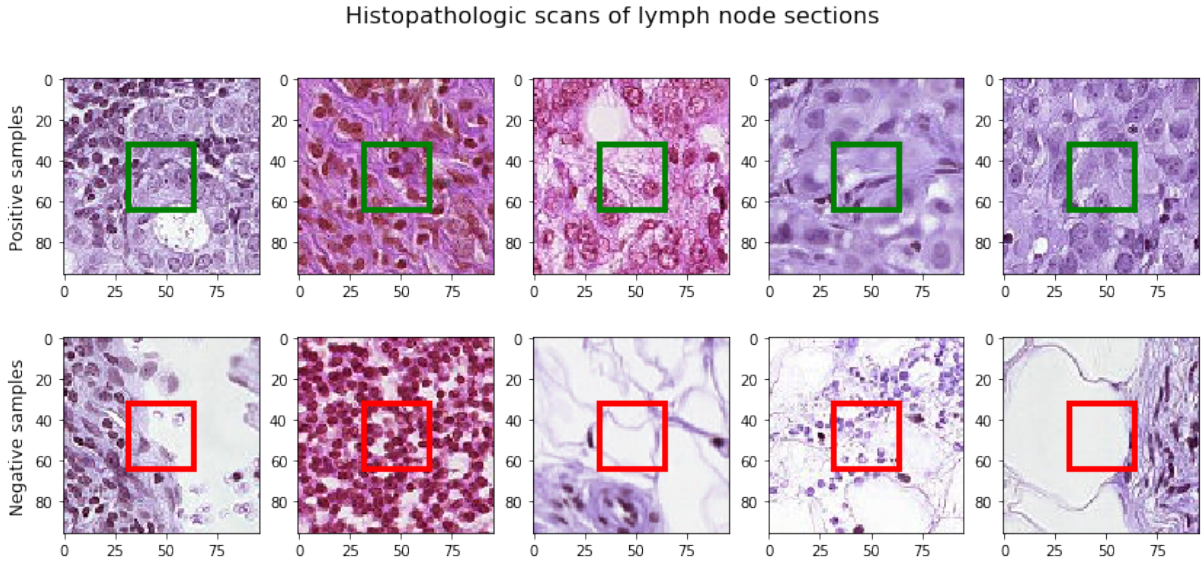
2 Description of data

What kind of data do we have?

About 220k training images and 57k evaluation images. We are not going to use evaluation images because the contest is over and we just want to test performance of models. The dataset is a subset of PCam dataset.
A positive labels means that that there is a least one pixel of tumour tissue in the center region (32 x 32px) of the image. Because tumour tissue in the outer region does not influence the label, it might be a good idea to crop the images to center region (this means that negatively labelled image could contain metastases in the outer region). We decided to focus on 48 x 48px center region however other crops were not investigated.
It is easy to see that the training set is not balanced. The ratio is close to 60/40 meaning that there are more negative images.

Understanding data

Lymph node metastases features are described in very complicated way but to simplify: irregular nuclear shapes, sizes or staining shades can indicate metastases. The features might be very subtle thus the problem of classifying metastases is not an easy task for a specialist and extremely hard for an untrained eye like mine. See the example images below with highlighted center areas.

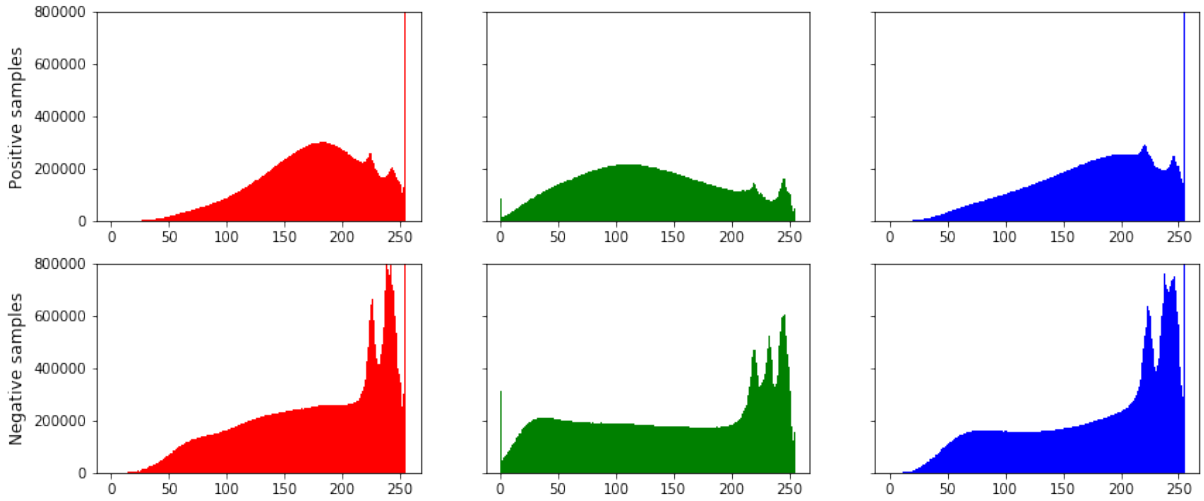


3 Feature engineering

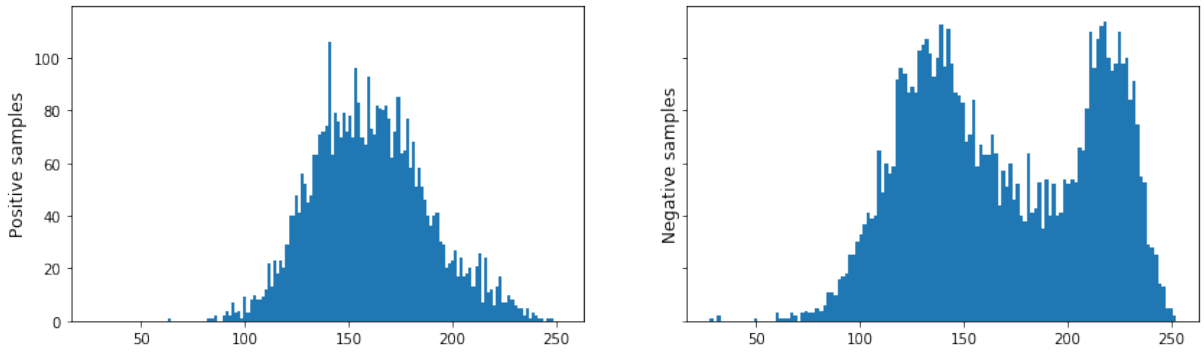
Color channels

As mentioned before it is hard to describe useful features but let's do some feature engineering on each channel: red, green and blue. First let's take a look how looks the distribution of colors

in each class.



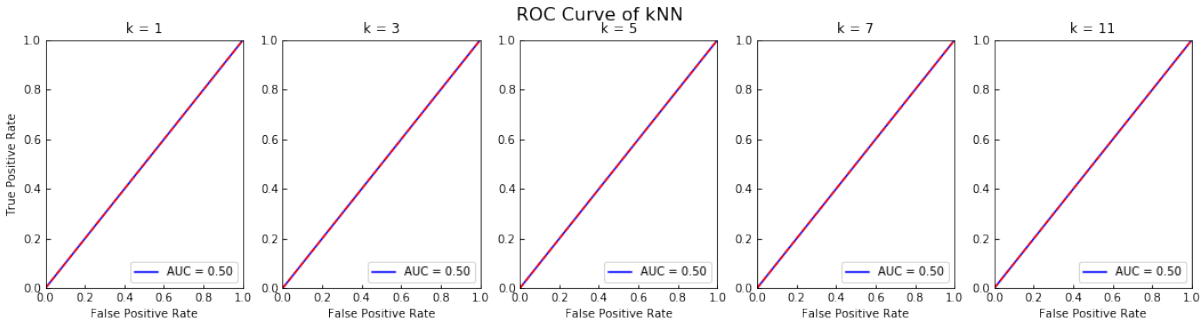
Now we can clearly spot the difference between classes. Positive samples usually have darker colors (remember: 0 is black, 255). The peaks of distributions from first row are on the left-hand side of peaks from second row. Interesting fact is that many white spots appear in both classes. You can notice a very high value for 255 in blue channel. It might be an artefact which we may need to deal with later. To make sure that negative samples seems to consist of brighter colors than negative samples let's take a look on mean image brightness.



Here we can also see that distribution of mean brightness for positive samples tends to be unimodal with the peak around 160. Whereas for negative samples, it seems to behave like bimodal distribution with peaks in 140 and 220.

4 k-Nearest Neighbours

At very beginning let's check very naive algorithm - kNN. Because the data set is large and because of RAM limitation we will use on 20k first images and split them in ratio 80:20. See blow the ROC curves for each for each k to see the quality of classifiers. The result is not unexpected. The space is huge so kNN cannot fit correctly and works like random classifier.



5 Convolutional Neural Net

Model description

The model for this problem will be convolutional neural net using ReLU as an activation function with batch normalization, max pooling and dropouts. Below you can find an architecture of the net.

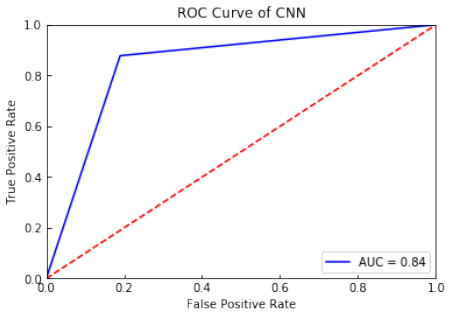
- Convolution layer (32, 64, 128 filters, 3 x 3 kernel)
- Batch normalization layer
- Activation layer (ReLU)
- Convolution layer (32, 64, 128 filters, 3 x 3 kernel)
- Batch normalization layer
- Activation layer (ReLU)
- Max pooling layer (2 x 2)
- Dropout layer

(This block is repeated three times with size of filters given in curly brackets). At the end an output will be flatten and full connected to 256 neurons and finally converted to values from 0 to 1 using sigmoid.

- Flatten layer
- Fully conected layer (256)
- Batch normalization layer
- Activation layer (ReLU)
- Dropout layer
- Fully conected layer (1) with sigmoid

Training and validation

The model was compiled with Adam optimizer and binary cross-entropy as loss function. This time we use full dataset and again we split the data using 80% for training and 20% for validation.
We decided to set batch size on 50 which means that the network will process 50 images at ones. We also added reshuffling the data between the epochs which may improve the outcome. Finally we get 86% accuracy on training set and 83.8% on test set. You can see the ROC curve below. It is not sure why it has such shape with one sharp point.



6 Neural Architecture Search

What is NAS?

Neural Architecture Search (NAS) is the process of automating architecture engineering i.e. finding the design of our machine learning model. Where we need to provide a NAS system with a dataset and a task (classification, regression, etc), and it will give us the architecture. And this architecture will perform best among all other architecture for that given task when trained by the dataset provided. NAS can be seen as a subfield of AutoML and has a significant overlap with hyperparameter optimization. To understand NAS we need to look deeply into what it is doing. It finds an architecture from all possible architectures by following a search strategy that will maximize the performance. The following figure summarizes the NAS algorithm.

Search space defines what neural architecture a NAS approach might discover in principle. The search strategy and performance estimation can be solved with RL. Reinforcement learning is the problem faced by an agent that must learn behaviour through trial and error interactions with a dynamic environment to maximize some reward. In the case of NAS, the agent produces the model architecture. Then the model is trained on the dataset and the performance of the model on the validation data is taken as a reward.

Model description

We decided to use built-in NASNetMobile. The output from the net is passed through GlobalAveragePooling to fully connected layer with sigmoid used as an activation function.

Training and validation

Again we will use Adam as optimizer and binary cross entropy as a loss function. The data is is again split between training and validation in ratio 80:20.
At the end we have accuracy on training set 93.2% and 88.9% accuracy.

7 Conclusions and further work

Needless to say, that classic algorithm like kNN is not enough efficient to process this multidimensional data set. The result that NAS beat earlier proposed architecture is expected because the goal of NAS is to find the best possible architecture.
There are few areas in which the work could be done but because of time limits we skipped them:

- better data augmentation (now we use only rotations and mirror images)
- one could compare results with other classic pretrained net like ResNet
- dataset reduction: maybe it is a good idea to remove very bright samples because the appear in both classes
- cross validation