Barbara Klaudel

The impact of filters on training a neural network to classify computed tomography images of kidney tumors

Aplikacja do wizualizacji działania nakładania **filtrów na obraz tomografii komputerowej (CT).**

For each filter, we tested how well it enhances a CT image with the following metrics.

- **Mean Squared Error (MSE)**

$$MSE(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

xi, yi - 2 finite discrete signals (2 images)

N - the number of signal samples, e.g. 130x130 = 16900

Benefits of MSE: low computational complexity (1 x multiplication and 2 x addition per sample), low memory usage (MSE can be computed independently for each sample), easy to interpret (MSE=0 -> x=y, MSE(x, y) = MSE(y, x) -> symmetry), good for optimization (symmetric convex differentiable functiun).

Drawbacks: it doesn't take into account the order of samples, assumes the quality of signal isn't influenced by the correlation between original and error signal, assumes all samples are equally important.

- **peak-signal-to-noise-ratio (PSNR)**

$$PSNR = 10log_{10}\frac{L^2}{MSE}$$

L - the range of acceptable pixel values, i.e. 256

PSNR is helpful when we compare images with different Ls, otherwise it is no more informative than MSE.

- **Structural Similarity Index Measure (SSIM)**

$$SSIM = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

µ- mean value, σ - variance/covariance

SSIM takes the value from rnge -1 to 1 (1 means images are identical)

Benefits: it takes into account that image signals are highly structured and there is a strong relationship among neighbors that store significant information about the structure and eelements of an image, it gives better results for unstructural disturbances than structural, it is symmetrical SSIM(x, y) = SSIM(y, x), it is somewhat alike to human eye assessment.

Drawbacks: sensitive to translation, scaling and rotation of images.

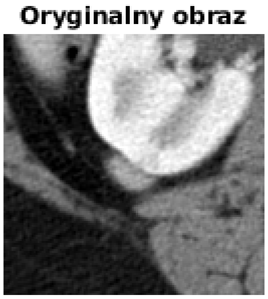
Based on: Mean Squared Error: Love It or Leave It? IEEE SIGNAL PROCESSING MAGAZINE [100] JANUARY 2009

1. We load a sample image. It is a fragment of CT image abdominal cavity presenting a kidney with benigh tumor - angiomyolipoma (AML).

```
pwd
```

```
ans =
'/home/basia/Documents/Filtry/filtry_barbara_klaudel'
```

```
I = imread('/home/basia/Documents/Filtry/filtry_barbara_klaudel/www.png');
imshow(I);
title('Original image');
```



Sobel Gradient Filter

```
sobelGradient = uint8(imgradient(I));
mse = immse(sobelGradient, I)
```

```
mse = 8.8994e+03
```

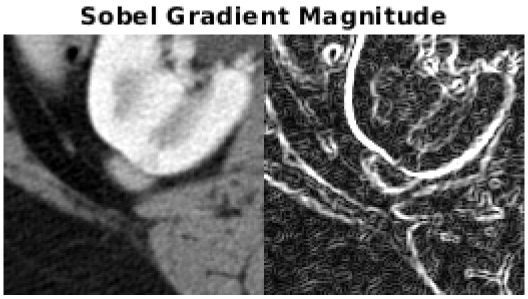
```
ssme = ssim(sobelGradient, I)
```

```
ssme = 0.0225
```

```
[peaksnr, snr] = psnr(sobelGradient, I);
peaksnr
```

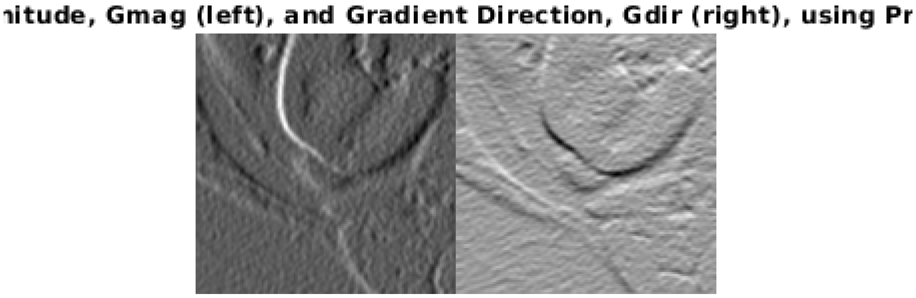
```
peaksnr = 8.6372
```

```
imshowpair(I, sobelGradient, 'montage');
title('Sobel Gradient Magnitude');
```

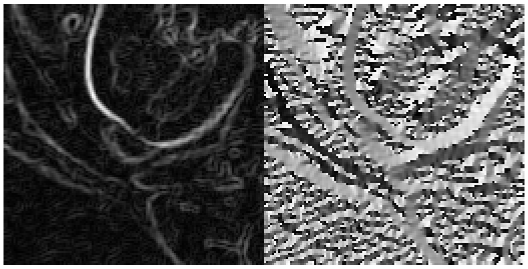


Gradient Magnitude and Direction

```
[Gx,Gy] = imgradientxy(I);
imshowpair(Gx, Gy, 'montage');
title('Gradient Magnitude, Gmag (left), and Gradient Direction, Gdir (right), using Pre
```

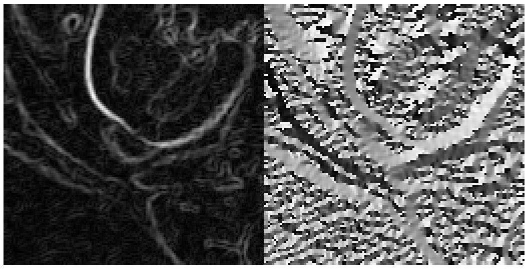

ıitude, Gmag (left), and Gradient Direction, Gdir (right), using Pr

```
[Gmag, Gdir] = imgradient(I,'prewitt');
imshowpair(Gmag,Gdir,'montage')
title('Gradient Magnitude (Left) and Gradient Direction (Right)')
```



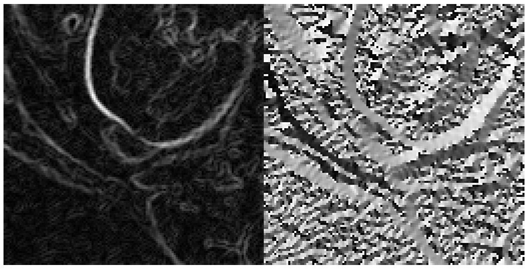
Gradient Magnitude (Left) and Gradient Direction (Right)

```
[Gmag, Gdir] = imgradient(I,'sobel');
imshowpair(Gmag,Gdir,'montage')
title('Gradient Magnitude (Left) and Gradient Direction (Right)')
```



Gradient Magnitude (Left) and Gradient Direction (Right)

```
[Gmag, Gdir] = imgradient(I,'central');
imshowpair(Gmag,Gdir,'montage')
title('Gradient Magnitude (Left) and Gradient Direction (Right)')
```



Gradient Magnitude (Left) and Gradient Direction (Right)

Smooth Gradient

```
sigma = 2;
smoothImage = imgaussfilt(I,sigma);
smoothGradient = imgradient(smoothImage,'CentralDifference');
mse = immse(smoothImage, I)
```

```
mse = 136.3212
```

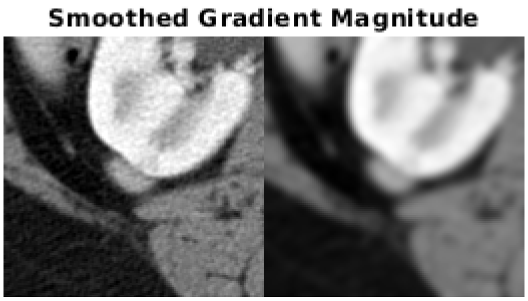
```
ssme = ssim(smoothImage, I)
```

```
ssme = 0.7157
```

```
[peaksnr, snr] = psnr(smoothImage, I);
peaksnr
```

```
peaksnr = 26.7852
```

```
imshowpair(I, smoothImage, 'montage');
title('Smoothed Gradient Magnitude');
```



Anisotropic Diffusion (Perona & Malik)

Anisotropic diffusion with default parameters

- Often used with CT images
- It sharpens areas, which are borders of an image. It smoothes the insides of areas.
- High diffusion value is desired in homogenous regions. In inhomogenous regions (the regions with high gradient values) low diffusion is desired.
- A space-variant diffusion factor is used to differentiate insides and borders of regions.

$$I = div(c(x,y,t)\nabla I) = c(x,y,z)\Delta I + \nabla c \cdot \nabla I$$

```
ad = imdiffusefilt(I);
```

```
mse = immse(ad, I)
```

```
mse = 45.4299
```

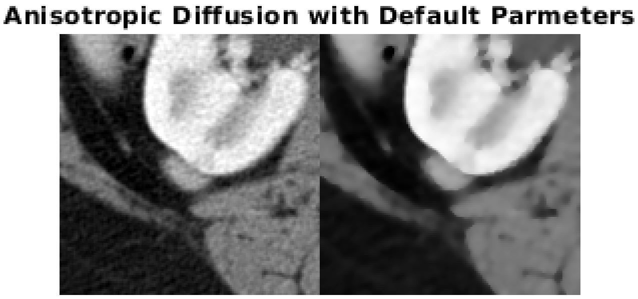
```
ssme = ssim(ad, I)
```

```
ssme = 0.8202
```

```
[peaksnr, snr] = psnr(ad, I);
peaksnr
```

```
peaksnr = 31.5574
```

```
imshowpair(I, ad, 'montage');
title('Anisotropic Diffusion with Default Parmeters')
```



Anisotropic diffusion with estimated parameters

- We can choose from exponential and quadratic methods. Exponential method is suggested if there is strong contrast on the borders, while quadratic method, when we have a lot of homogenous regions.
- We choose optimal number of iteration and gradient treshold value with imdiffuseest matlab function. Gradient treshold value designates the starting value from which we assume that a region is a border.

```
[gradThresh,numIter] = imdiffuseest(I,'ConductionMethod','quadratic');
ade = imdiffusefilt(I,'ConductionMethod','quadratic', ...
    'GradientThreshold',gradThresh,'NumberOfIterations',numIter);
mse = immse(ade, I)
```

```
mse = 41.8651
```

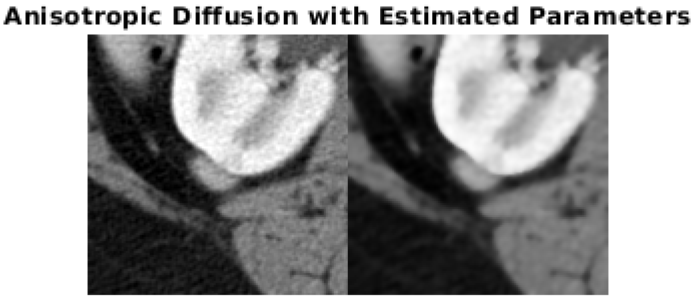
```
ssme = ssim(ade, I)
```

```
ssme = 0.8545
```

```
[peaksnr, snr] = psnr(ade, I);
peaksnr
```

```
peaksnr = 31.9123
```

```
imshowpair(I, ade, 'montage');
title('Anisotropic Diffusion with Estimated Parameters')
```



Sharpen Image

- it uses a method of unsharp masking, which subtracts the blurred image ffrom the original image. The procedure goes as follows:

1. Detect edges and create a mask.
2. Use the mask to enhance contrast around the edges.

To simplify: the filter sort of "overdoes" function values in place of light and dark edges.

Check a nice visualization at: https://www.cambridgeincolour.com/tutorials/unsharp-mask.htm

```
sharpen = imsharpen(I);
mse = immse(sharpen, I)
```

```
mse = 21.7911
```

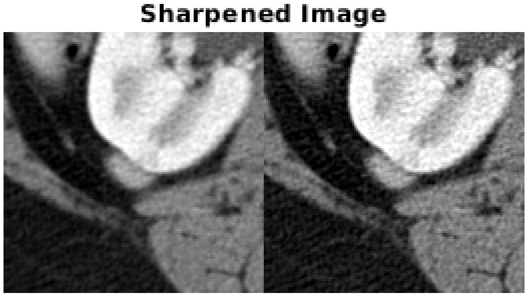
```
ssme = ssim(sharpen, I)
```

```
ssme = 0.9611
```

```
[peaksnr, snr] = psnr(sobelGradient, I);
peaksnr
```

```
peaksnr = 8.6372
```

```
imshowpair(I,sharpen,'montage')
title('Sharpened Image');
```


Sharpened Image

Wiener Filter

- most commonly used for deconvolution and noise reduction
- often used to enhance the soft tissue areas, e.g. a cyst consisting of connective tissue
- it uses value of local mean and variance independently for each pixel.

$$\mu = \frac{1}{NM} \sum_{n_1,n_2 \in n} a(n_1,n_2) \quad \sigma^2 = \frac{1}{NM} \sum_{n_1,n_2 \in n} a^2(n_1,n_2) - \mu$$

η - local neighborhood of a pixel with a value N by M

Based on mean and varince, Wiener filter is created, separately for each pixel.

$$b(n_1,n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(a(n1,n2) - \mu)$$

v^2 - noise variance, if uknown it is estimated based on mean value of all local variances

```
K = wiener2(I,[5 5]);
mse = immse(K, I)
```

```
mse = 66.6433
```

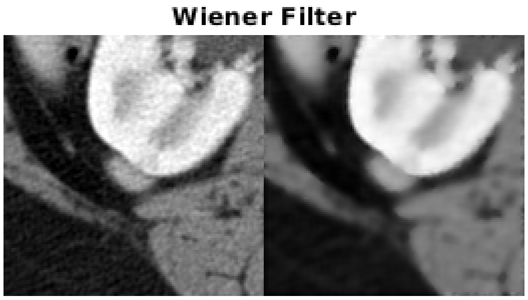
```
ssme = ssim(K, I)
```

```
ssme = 0.7659
```

```
[peaksnr, snr] = psnr(K, I);
peaksnr
```

```
peaksnr = 29.8932
```

```
imshowpair(I, K, 'montage');
title('Wiener Filter');
```



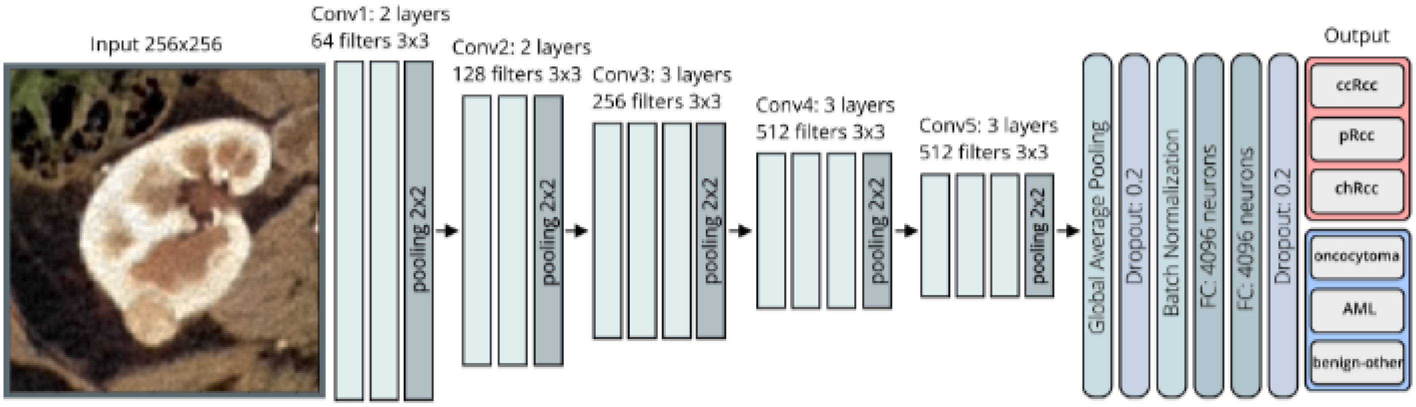
Transfer Learning

We checked if applying a filter improves the accuracy of kidney tumor classification. We checked sharpen, Wiener and anisotropic diffusion with esitmated parameters.

We used the dataset of CT images of abdominal cavity collected for TITAN project (https://titan.radiato.ai/). Images come from St. Adalbert's hospital in Gdańsk and from publicly available dataset KITS19. The images present kidney tumors of 6 types: 3 malignant (clear cell renal cell carcinoma (ccRCC), chromophobe renal cell carcinoma (chRCC), papillary renal cell carcinoma (pRCC)) and 3 benign (angiomyolipoma (AML), oncocytoma, benign-other). The dataset consists of middle slices of arterial phase. Multiple slices from the same study where used insteasd of data augmentation.

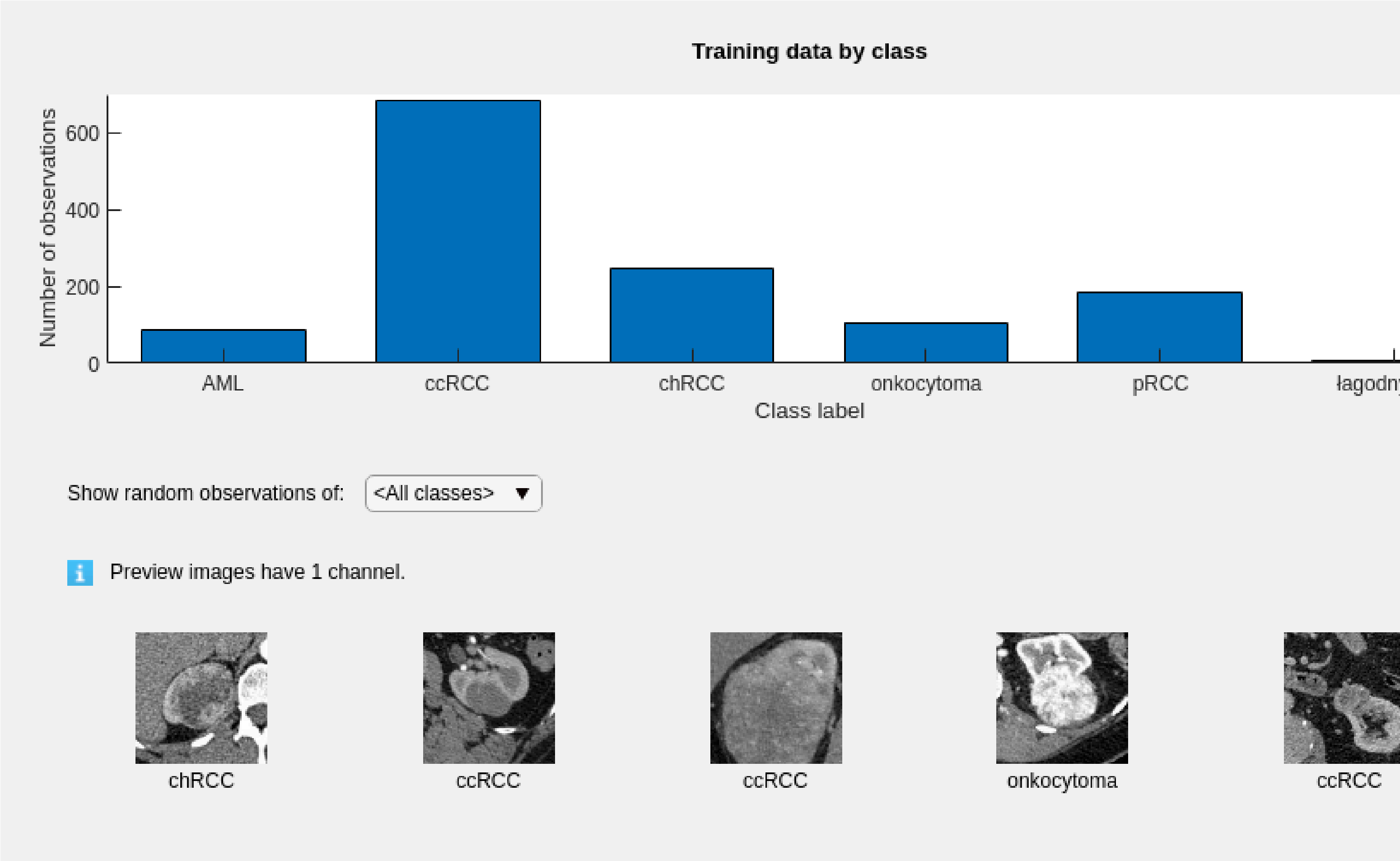
The images present a kidney with a tumor. The image size is 130x130 pixels. After cropping, the images were converted to png from DICOM wiyh abdominal cavity window.

We used VGG-16 model pretrained on ImageNet. We used 80-20 validation split.



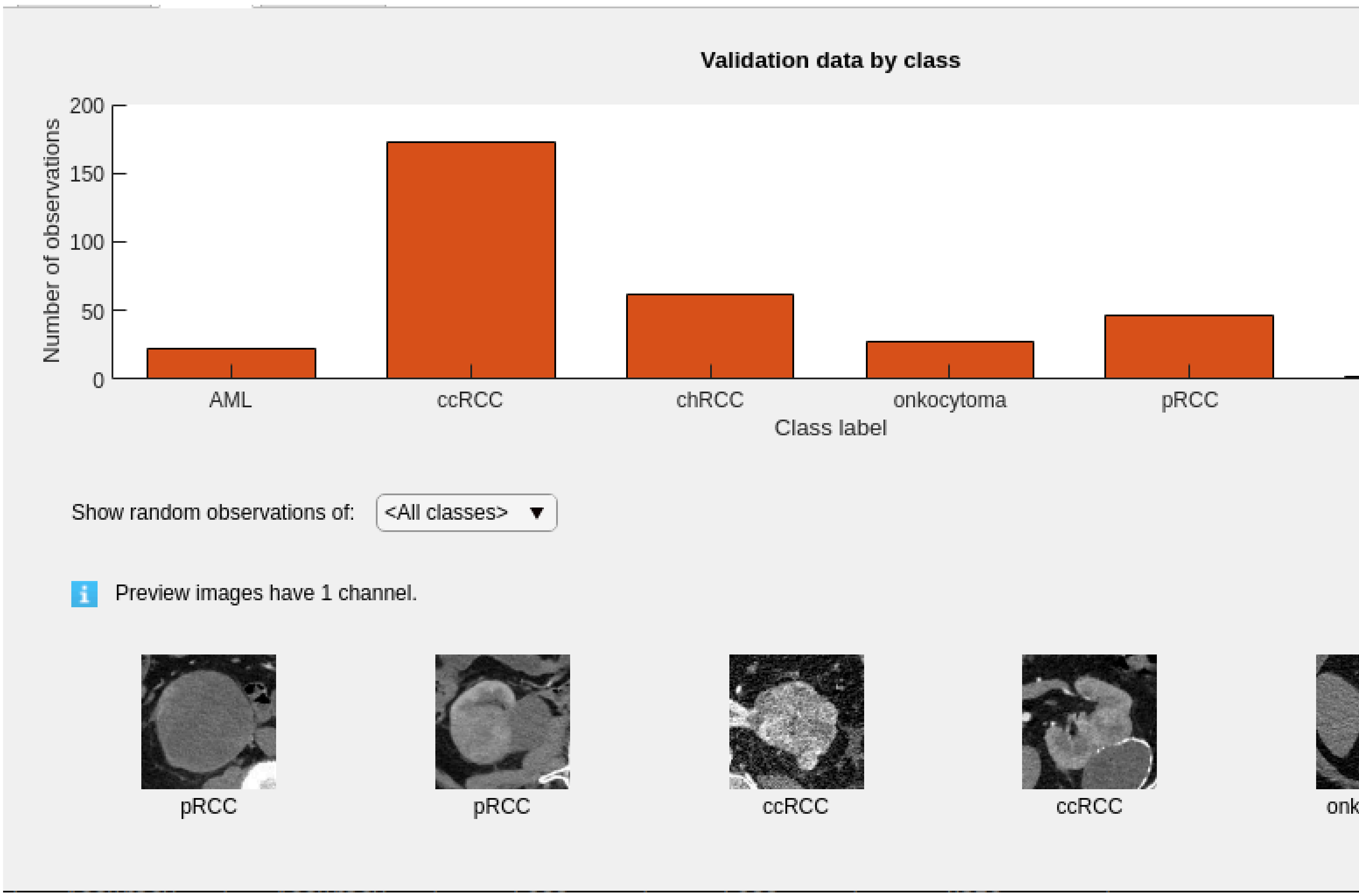
Architecture visualization - source: Aleksander Obuchowski, unpublished research

1. **Basic dataset - no filters.**



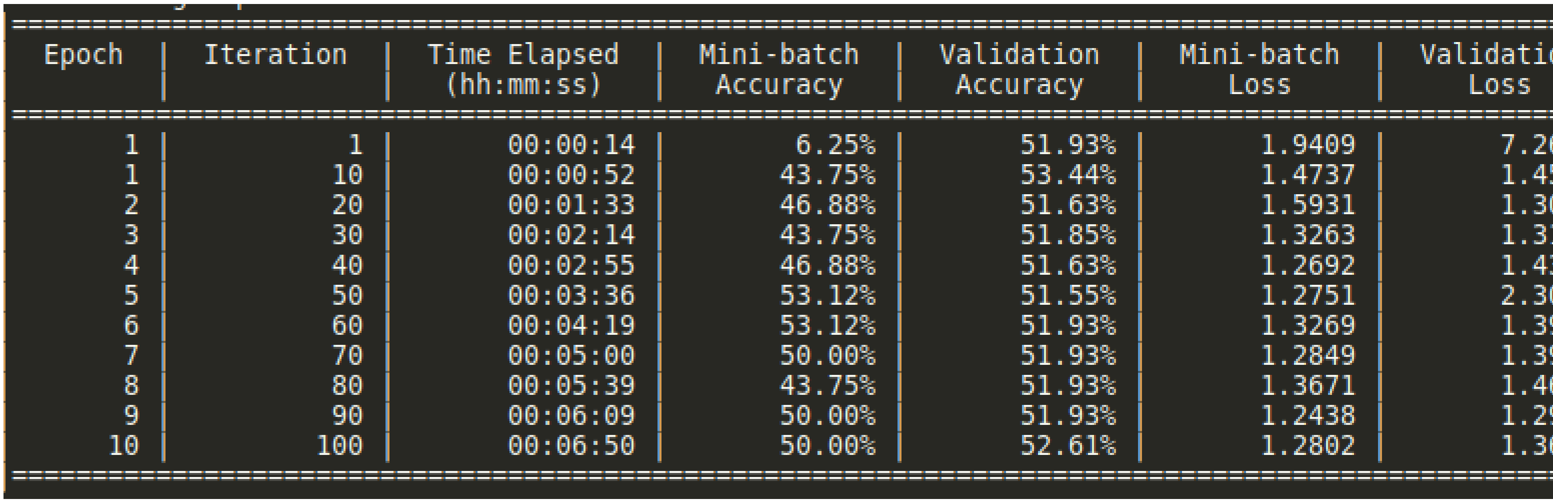
Show random observations of: <All classes> ▼

Preview images have 1 channel.

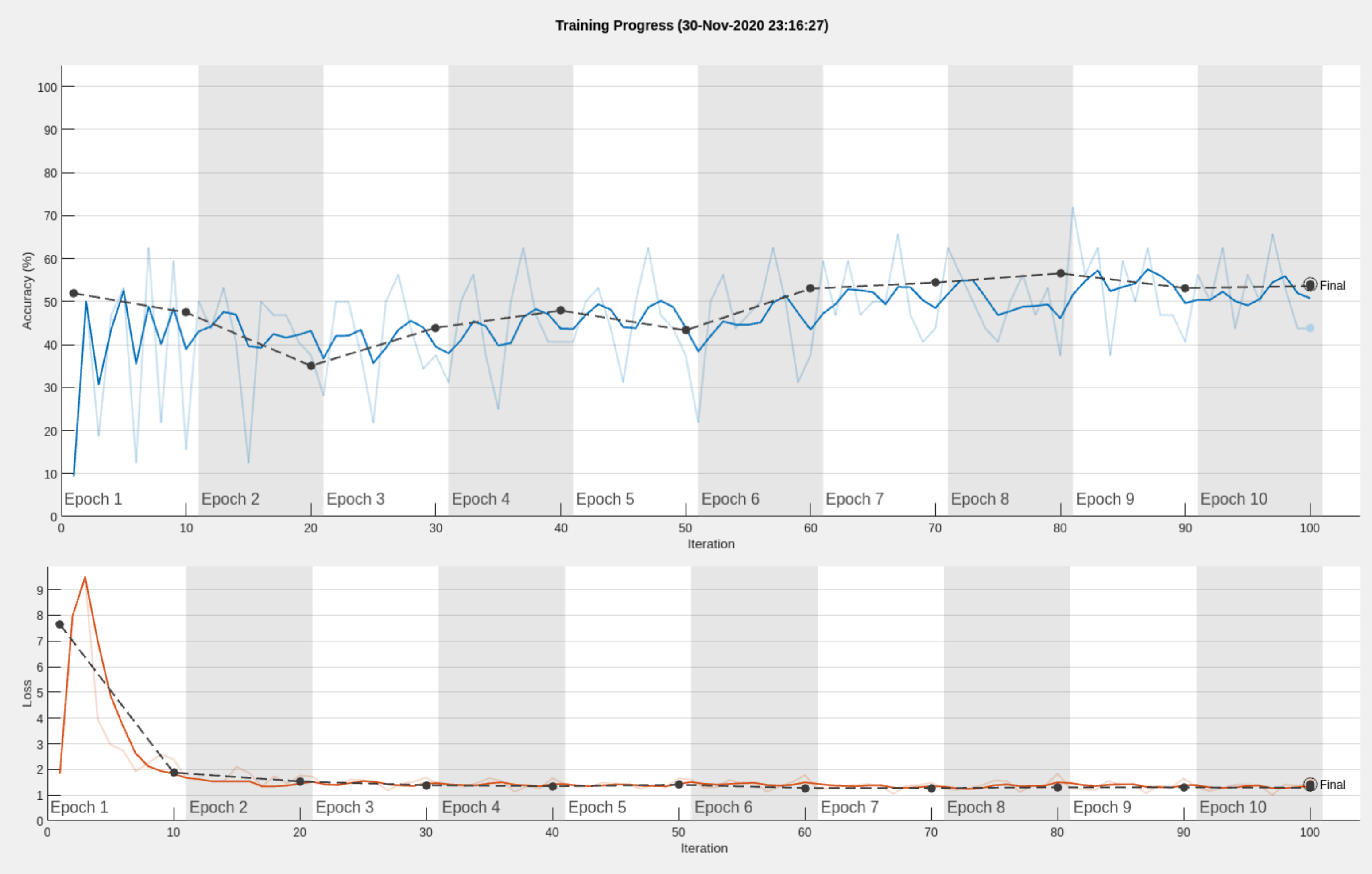


Show random observations of: <All classes> ▼

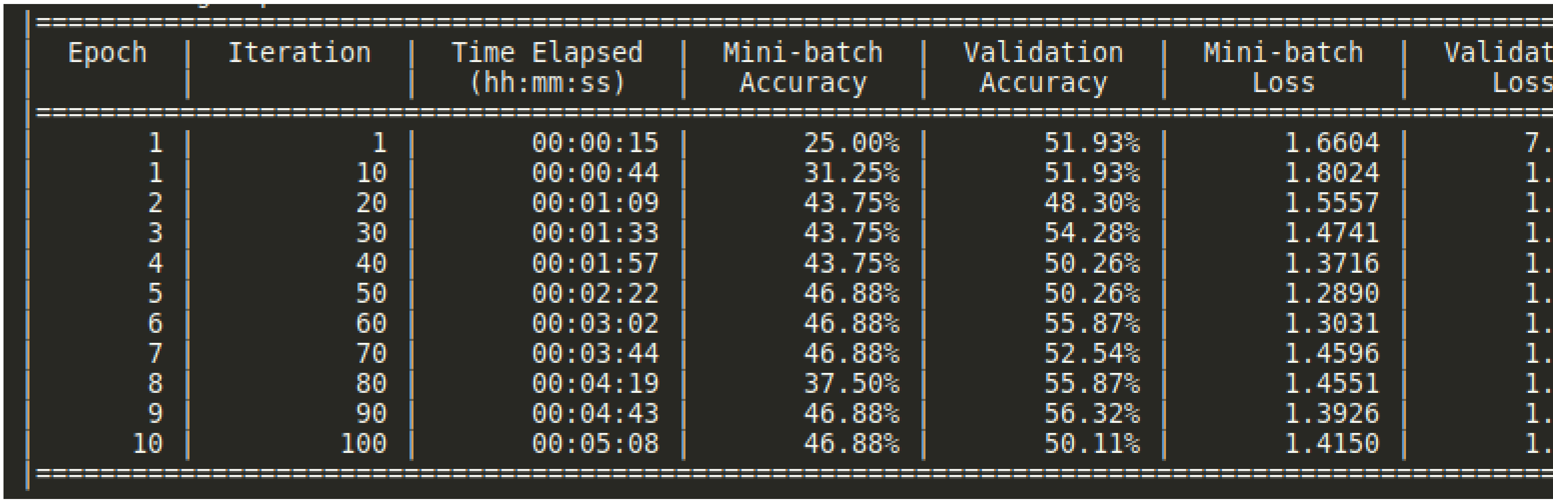
Preview images have 1 channel.



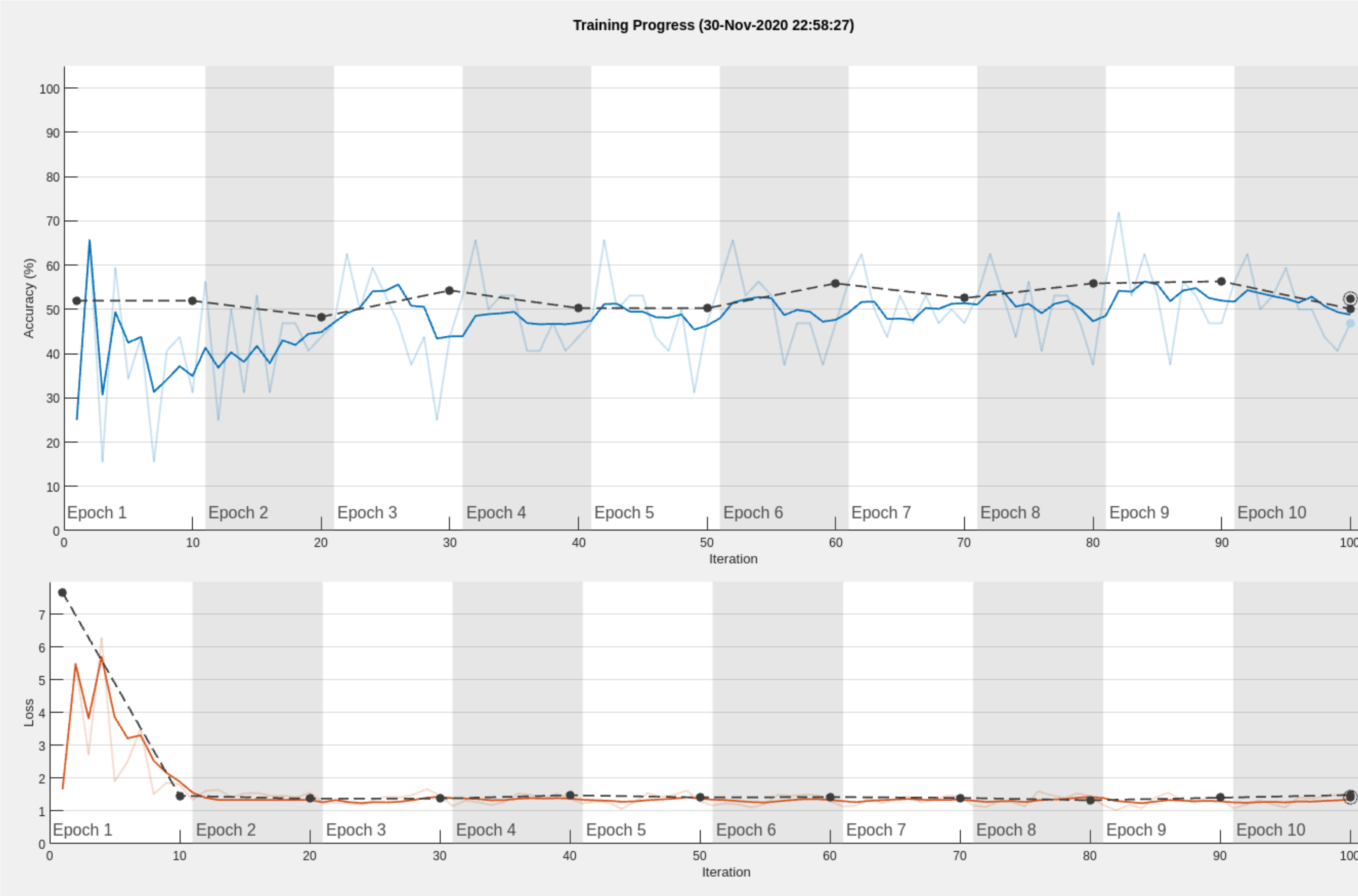
```
================================================================================================================
 Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Validation  |  Mini-batch  |  Validati
        |             |   (hh:mm:ss)   |   Accuracy   |   Accuracy   |     Loss     |     Loss
================================================================================================================
      1 |           1 |       00:00:14 |        6.25% |       51.93% |       1.9409 |       7.20
      1 |          10 |       00:00:52 |       43.75% |       53.44% |       1.4737 |       1.45
      2 |          20 |       00:01:33 |       46.88% |       51.63% |       1.5931 |       1.30
      3 |          30 |       00:02:14 |       43.75% |       51.85% |       1.3263 |       1.3
      4 |          40 |       00:02:55 |       46.88% |       51.63% |       1.2692 |       1.43
      5 |          50 |       00:03:36 |       53.12% |       51.55% |       1.2751 |       2.30
      6 |          60 |       00:04:19 |       53.12% |       51.93% |       1.3269 |       1.39
      7 |          70 |       00:05:00 |       50.00% |       51.93% |       1.2849 |       1.39
      8 |          80 |       00:05:39 |       43.75% |       51.93% |       1.3671 |       1.40
      9 |          90 |       00:06:09 |       50.00% |       51.93% |       1.2438 |       1.29
     10 |         100 |       00:06:50 |       50.00% |       52.61% |       1.2802 |       1.30
================================================================================================================
```



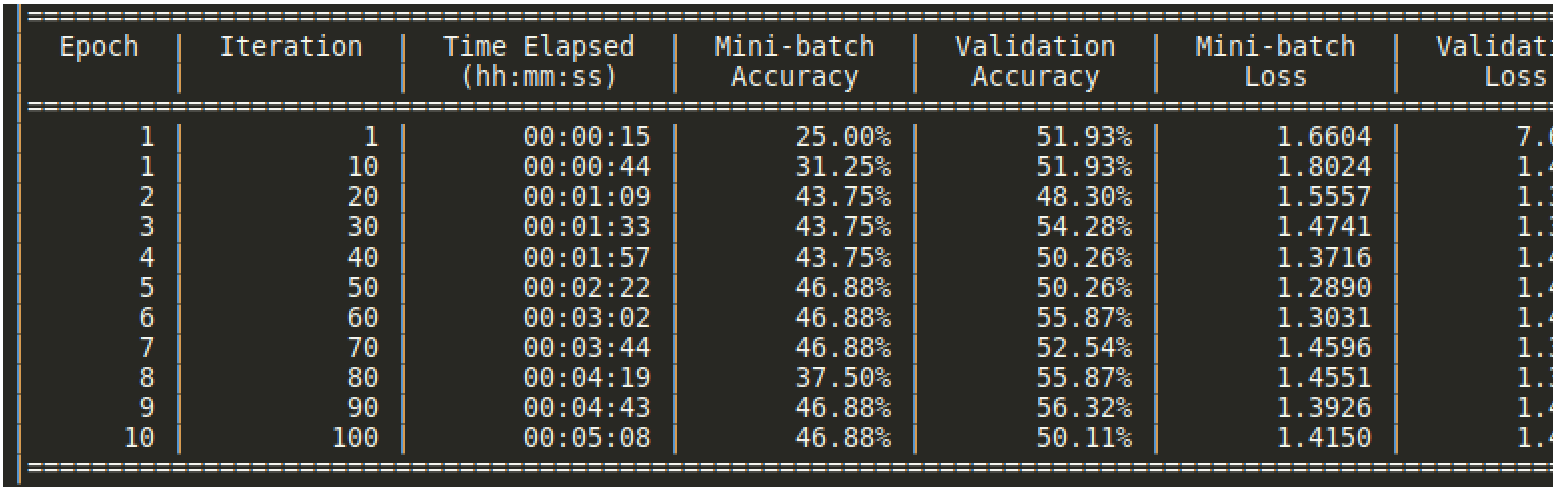
2. Dyfuzja anizotropowa z oszacowanymi parametrami



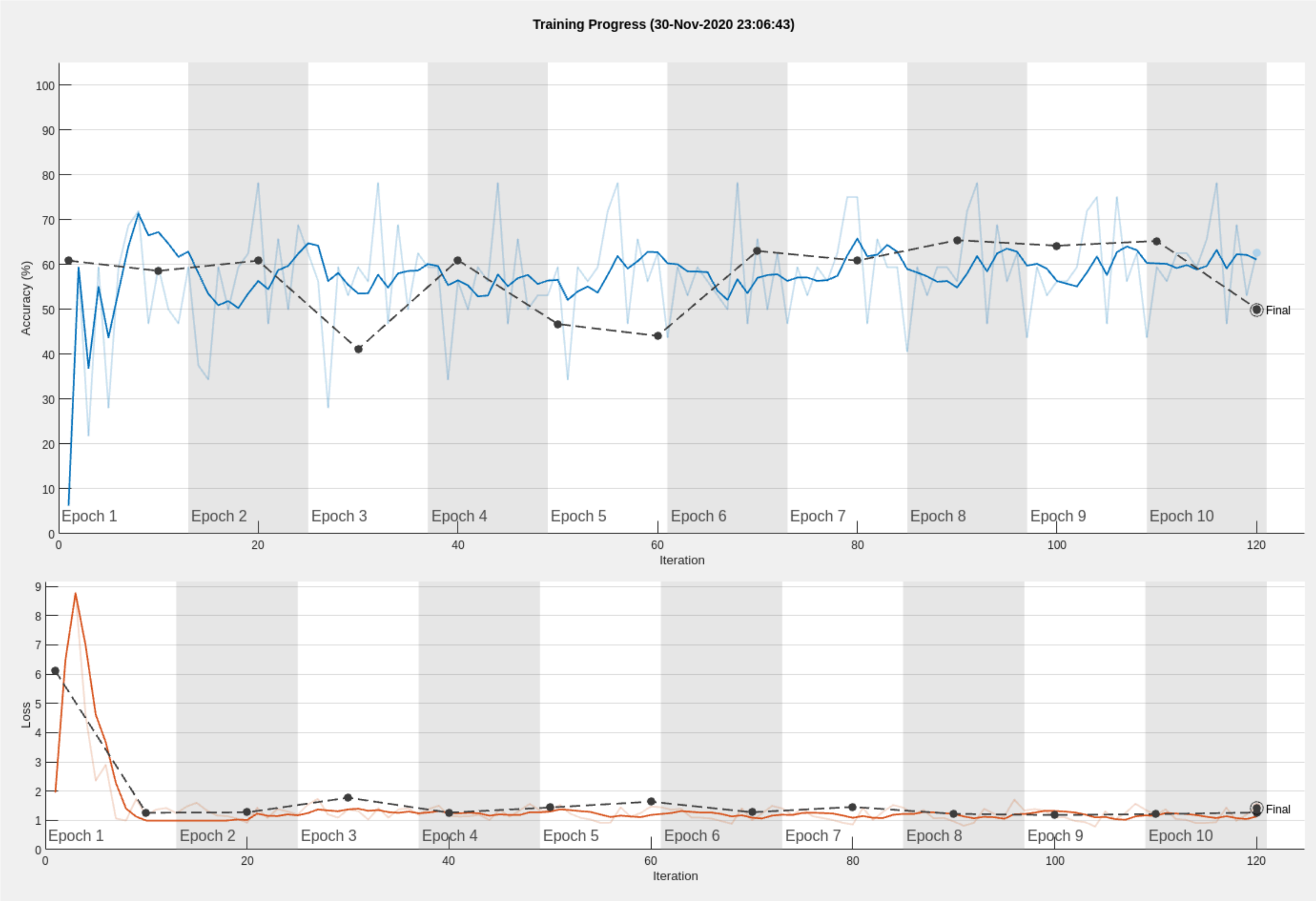
```
|========================================================================================================
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Validation  |  Mini-batch  |  Validat
|         |             |   (hh:mm:ss)   |   Accuracy   |   Accuracy   |     Loss     |     Loss
|========================================================================================================
|       1 |           1 |       00:00:15 |       25.00% |       51.93% |       1.6604 |          7.
|       1 |          10 |       00:00:44 |       31.25% |       51.93% |       1.8024 |          1.
|       2 |          20 |       00:01:09 |       43.75% |       48.30% |       1.5557 |          1.
|       3 |          30 |       00:01:33 |       43.75% |       54.28% |       1.4741 |          1.
|       4 |          40 |       00:01:57 |       43.75% |       50.26% |       1.3716 |          1.
|       5 |          50 |       00:02:22 |       46.88% |       50.26% |       1.2890 |          1.
|       6 |          60 |       00:03:02 |       46.88% |       55.87% |       1.3031 |          1.
|       7 |          70 |       00:03:44 |       46.88% |       52.54% |       1.4596 |          1.
|       8 |          80 |       00:04:19 |       37.50% |       55.87% |       1.4551 |          1.
|       9 |          90 |       00:04:43 |       46.88% |       56.32% |       1.3926 |          1.
|      10 |         100 |       00:05:08 |       46.88% |       50.11% |       1.4150 |          1.
|========================================================================================================
```



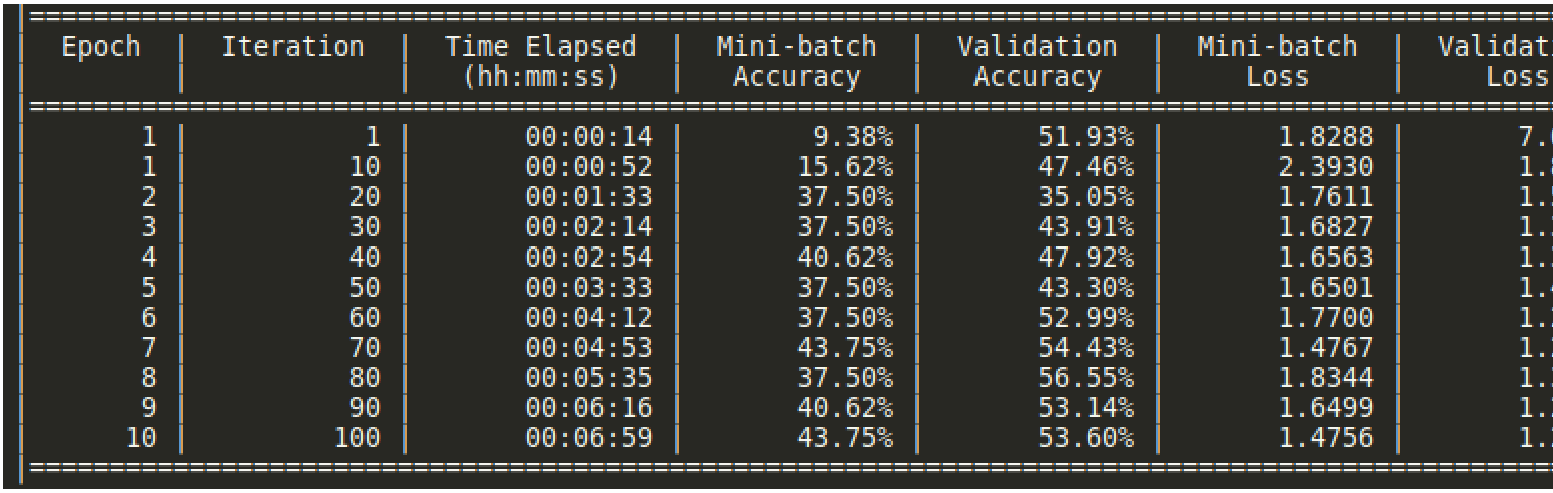
3. **Filtr** wyostrzający.



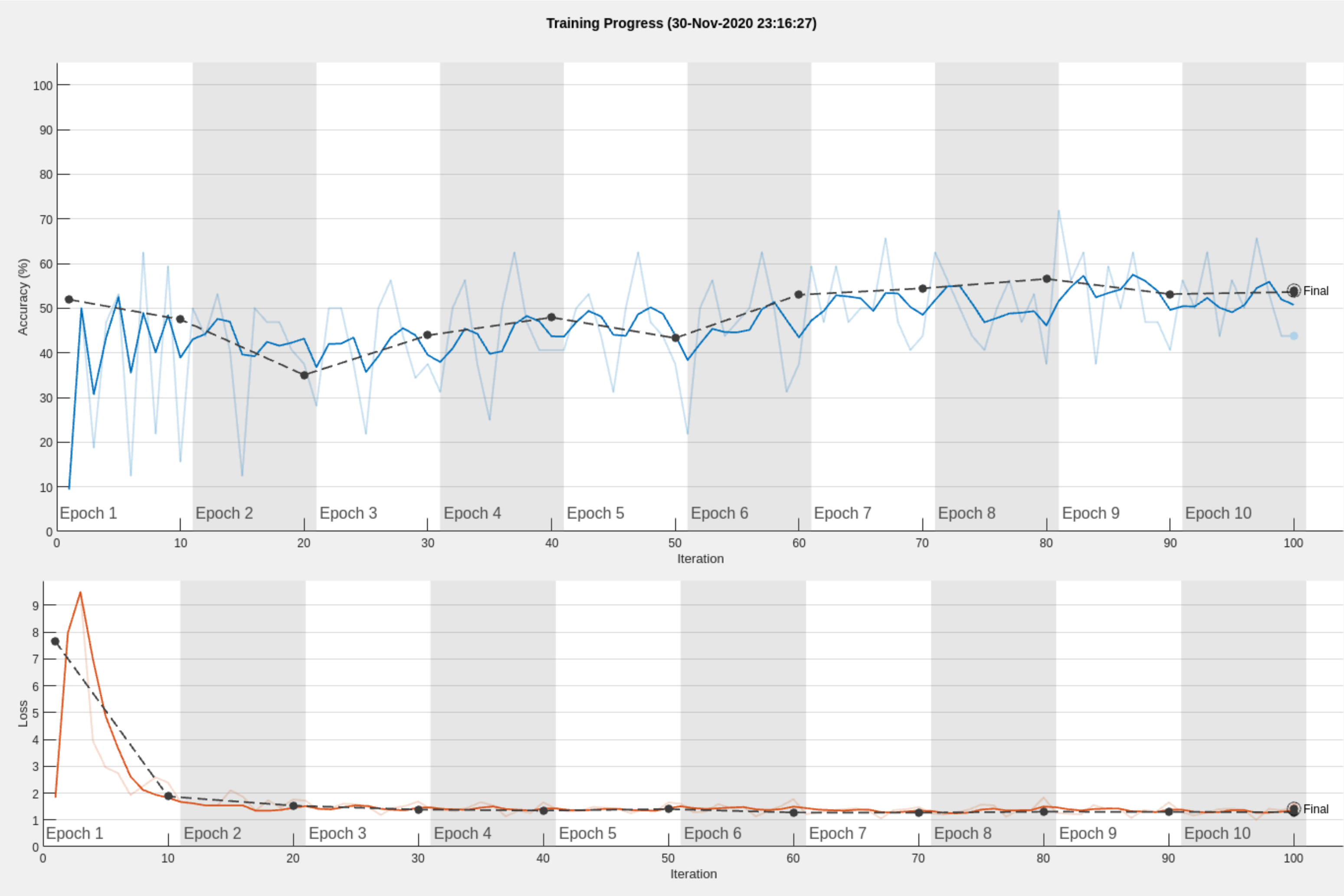
```
|=========================================================================================================================
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Validation  |  Mini-batch  |  Validati
|         |             |   (hh:mm:ss)   |   Accuracy   |   Accuracy   |     Loss     |     Loss
|=========================================================================================================================
|       1 |           1 |       00:00:15 |       25.00% |       51.93% |       1.6604 |       7.
|       1 |          10 |       00:00:44 |       31.25% |       51.93% |       1.8024 |       1.4
|       2 |          20 |       00:01:09 |       43.75% |       48.30% |       1.5557 |       1.3
|       3 |          30 |       00:01:33 |       43.75% |       54.28% |       1.4741 |       1.3
|       4 |          40 |       00:01:57 |       43.75% |       50.26% |       1.3716 |       1.4
|       5 |          50 |       00:02:22 |       46.88% |       50.26% |       1.2890 |       1.4
|       6 |          60 |       00:03:02 |       46.88% |       55.87% |       1.3031 |       1.4
|       7 |          70 |       00:03:44 |       46.88% |       52.54% |       1.4596 |       1.3
|       8 |          80 |       00:04:19 |       37.50% |       55.87% |       1.4551 |       1.3
|       9 |          90 |       00:04:43 |       46.88% |       56.32% |       1.3926 |       1.4
|      10 |         100 |       00:05:08 |       46.88% |       50.11% |       1.4150 |       1.4
|=========================================================================================================================
```



4. Filtr Wienera.



```
===========================================================================================================
| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validati
|       |           | (hh:mm:ss)   | Accuracy   | Accuracy   | Loss       | Loss
===========================================================================================================
|     1 |         1 |     00:00:14 |      9.38% |     51.93% |     1.8288 |     7.
|     1 |        10 |     00:00:52 |     15.62% |     47.46% |     2.3930 |     1.
|     2 |        20 |     00:01:33 |     37.50% |     35.05% |     1.7611 |     1.
|     3 |        30 |     00:02:14 |     37.50% |     43.91% |     1.6827 |     1.
|     4 |        40 |     00:02:54 |     40.62% |     47.92% |     1.6563 |     1.
|     5 |        50 |     00:03:33 |     37.50% |     43.30% |     1.6501 |     1.
|     6 |        60 |     00:04:12 |     37.50% |     52.99% |     1.7700 |     1.
|     7 |        70 |     00:04:53 |     43.75% |     54.43% |     1.4767 |     1.
|     8 |        80 |     00:05:35 |     37.50% |     56.55% |     1.8344 |     1.
|     9 |        90 |     00:06:16 |     40.62% |     53.14% |     1.6499 |     1.
|    10 |       100 |     00:06:59 |     43.75% |     53.60% |     1.4756 |     1.
===========================================================================================================
```



CONCLUSIONS

The sharpen filter turned out to give the best results. It improved accuracy on validation datased by a few percentage points.

Anisotropic diffusion improved the results only slightly.

Wiener filter gave results comparable to the unfiltered data.