Bachelor's project

Image-based quality evaluation of otoscopy images

Weekly Report 9

Freja Rindel Peulicke, s185393

October 23, 2021

Literature

What has been done this week

1 Metric performance

1.1 Variance of Laplacian

This metric has been implemented for both jpg and png filetypes.

Outlier An outlier in the sharp images have been discovered. As can be seen in the image below, a few images are classified as much sharper than the rest of the images with a score of around 6 compared to the rest of the scores around 1.

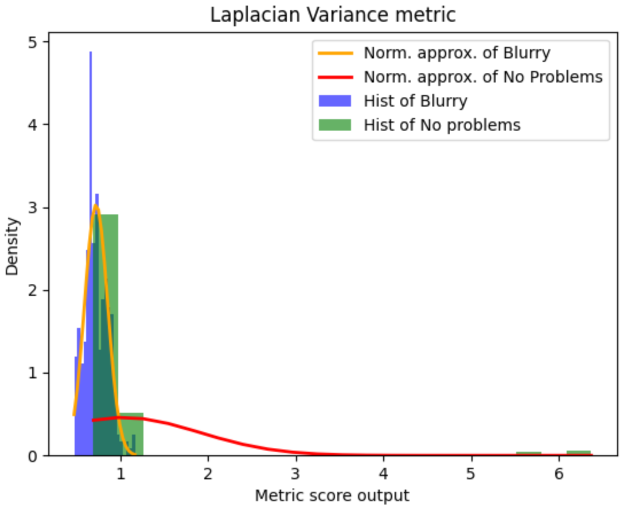
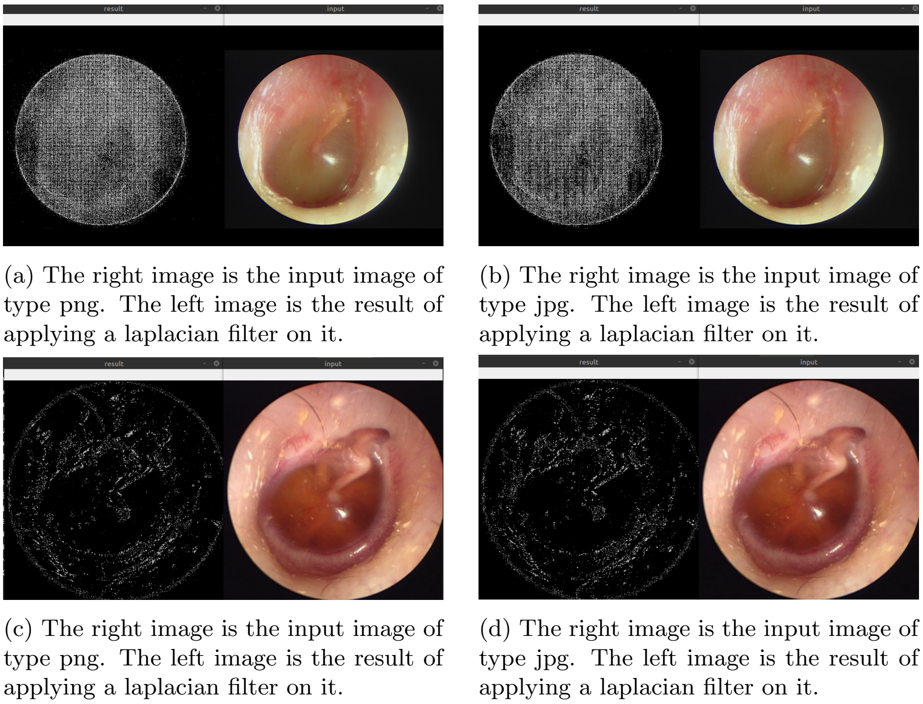


Figure 1: update diagram* A density plot on the training data of filetype jpg. The true blurry images are visualized in the blue histogram and the orange pdf-curve, and the sharp images are vizualised by the green histogram and the red pdf-curve. No gaussian blurred images have been included.

Displaying the image after applying the laplacian filter in the algorithm produces the output displayed in figure 2.



(a) The right image is the input image of type png. The left image is the result of applying a laplacian filter on it.

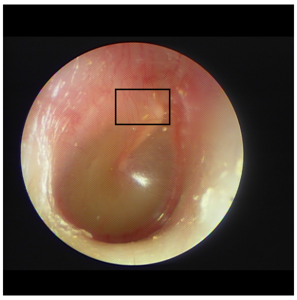
(b) The right image is the input image of type jpg. The left image is the result of applying a laplacian filter on it.

(c) The right image is the input image of type png. The left image is the result of applying a laplacian filter on it.

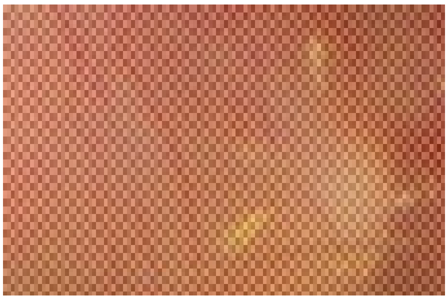
(d) The right image is the input image of type jpg. The left image is the result of applying a laplacian filter on it.

Figure 2: The above images displays the laplacian output of two different true sharp images of filetype png and jpg. The lower images represent a normal output of the laplacian filter, while the upper images produces outliers.

Looking closely at the original image (both jpg and png), one can see a filter-like pattern, which produces the very white output.



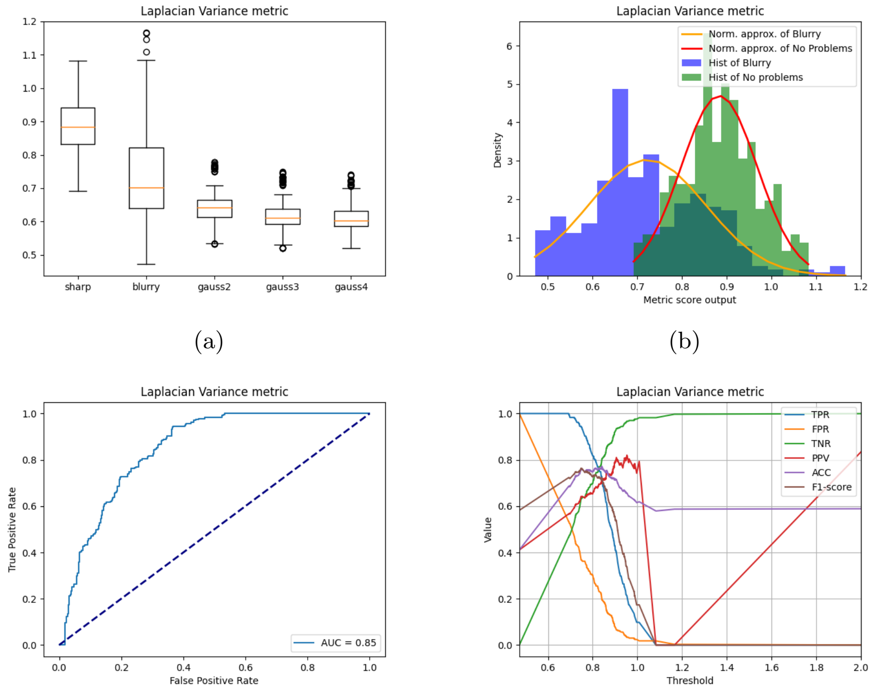
(a)



(b)

This image and the synthetic sharp data produced from it will be disregarded from the following analysis.

JPG

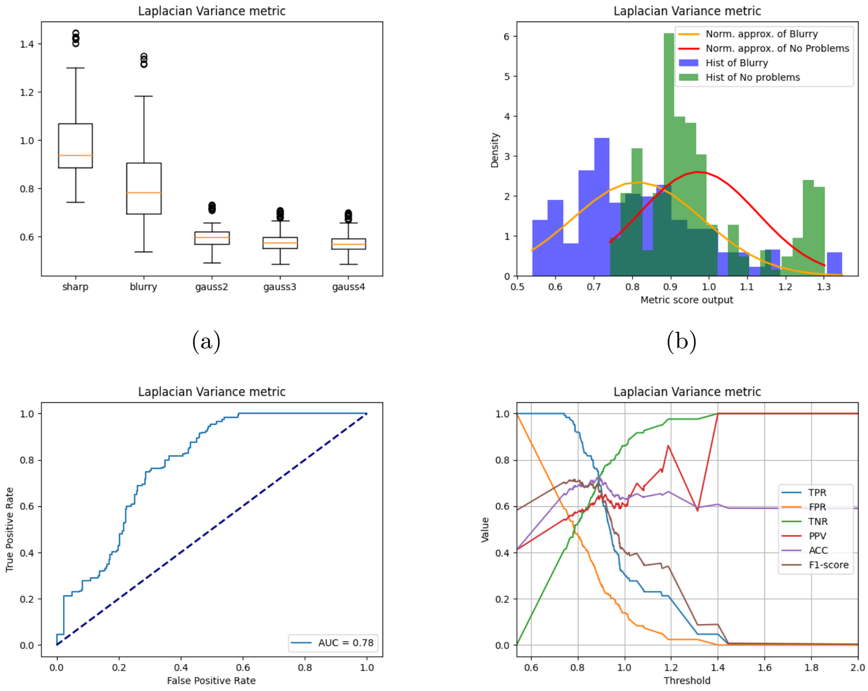


(a) (b)

(c) (d)

Figure 4

PNG



(a)

(b)

(c)

(d)

Figure 5

1.2 ”Training” the metrics

The goal is to avoid false positives (**FP**: classified as sharp when blurry), as these can distort the results from the data later on, while still achieving some positive outputs.

Choosing the threshold:
Let the user choose an acceptable TNR (specificity), e.g. 98%.
After this, find the corresponding threshold and choose the metric with the highest accuracy, as it will provide a possibility that some images will be ”accepted”, that is, classified as sharp.

We could also calculate summed distance, d, of some rates... TPR (sensitivity), F1-score, precision... to the top and bottom border (1 and 0) according to which border is desired for that particular rate. Then choose the metric with smallest d.

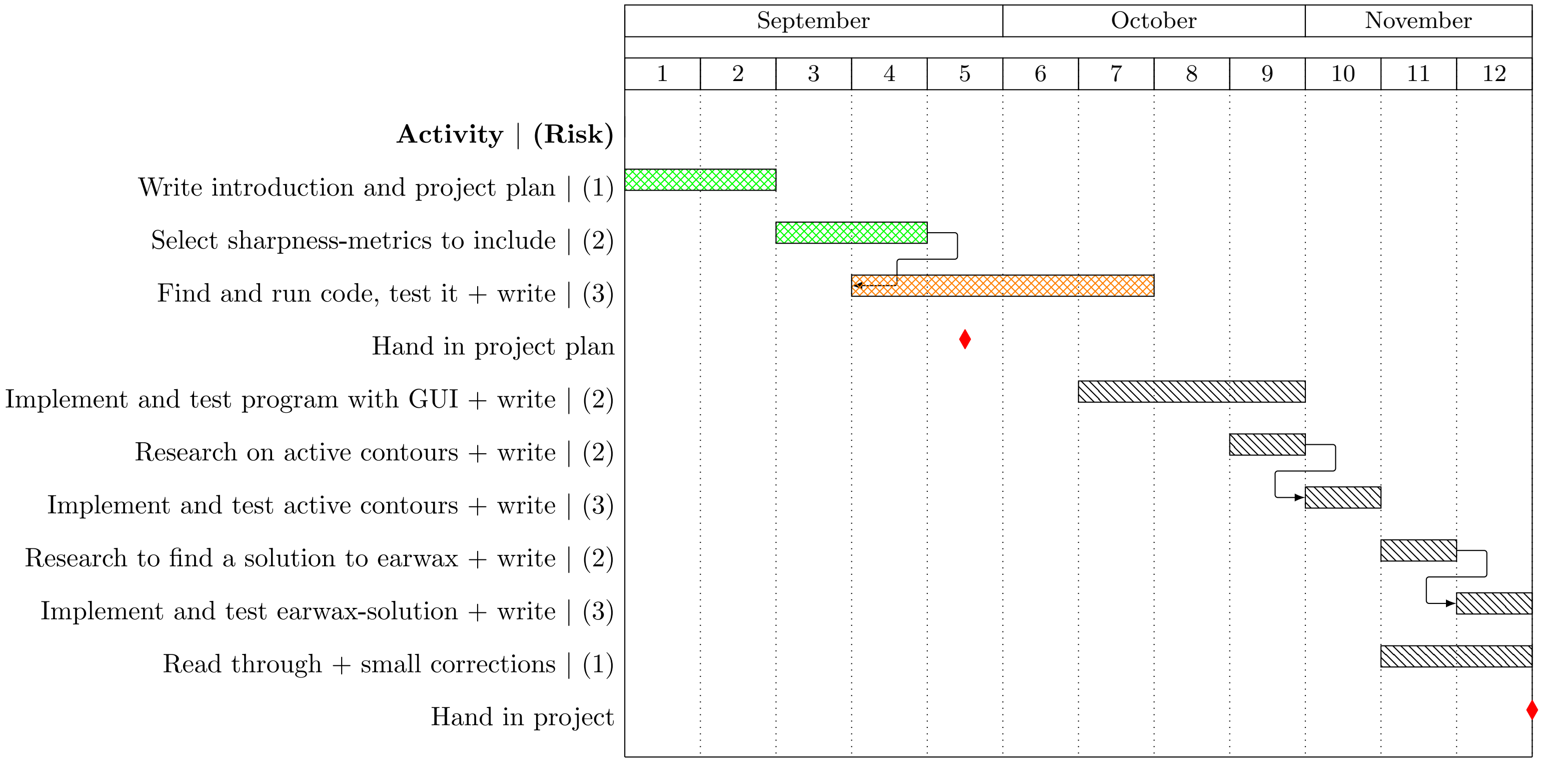
- maximize TNR (how many are blurry when blurry)
- maximize TPR (we would still like some possibility that images are classified ”sharp”)
- maximize PPV (part declared sharp when sharp)

- high f1-score for general score + do not produce FP (to not filter away too many negatives)

The following graphs displays output on the training data set excluding the Gaussian blurred images.

Project status according to the study plan

Overall Project Plan



Plan for the next weeks

1. Decide parameters to choose metric on (e.g. let user decide FNR and then minimize/maximize some parameter and choose the metric, that performs best on those criteria)
2. Implement choosing metric
3. Test the metrics on a test data set (the images not used for training) and report results
4. Start designing and implementing the final program

References