Image Quality Assessment For Amateur Photographer

Lorenz Teply

Bachelor’s thesis

Image Quality Assessment For Amateur Photographer

Lorenz Teply

September 28, 2020

Institute for Data Processing
Technische Universität München

Lorenz Teply. *Image Quality Assessment For Amateur Photographer.* Bachelor's thesis, Technische Universität München, Munich, Germany, 2020.

Supervised by Prof. Dr.-Ing. K. Diepold and Stefan Röhrl; submitted on September 28, 2020 to the Department of Electrical Engineering and Information Technology of the Technische Universität München.

Institute for Data Processing, Technische Universität München, 80290 München, Germany, http://www.ldv.ei.tum.de/.

Abstract

The times in which film material and the development of analogue photos cost huge amounts of money and time are past since the digitization of photography. Meanwhile, many smartphones enable pictures in high to very high resolution and due to the fact that the costs for single lens reflex and system cameras are not too high anymore, more and more people are able to take a picture quickly in any situation and then store it on the device, the external hard drive or in the cloud.
Some of these memories are published on social media, but most are stored somewhere for a long time or forever. Some smartphones give a month's review with pictures or sort the pictures by meta data such as coordinates, so the memories are a little bit more sorted. Unfortunately, these free programs do not take over the sorting of blurred photos reliable, or at least such software is not easily available open-source.
But this does not eliminate the need to sort out your own photos from your vacation, weekend trip or family celebration in order to pick the good conveniently and keep them for yourself or make them available to your family, friends or other people afterwards.
For such or similar applications, this bachelor thesis aims to develop algorithms that can quickly and reliably sort out at least the blurred and too brightly or too darkly exposed pictures on a larger scale.
The exposure has been measured by the overall brightness and size of very bright or dark areas and combined to a filter. For the detection of blurriness GRAn and LAPn filter have been used and compared. The final algorithms combine the filter for exposure and each one of the blurriness filter. The performance of the algorithms with the two GRA filter has been equally. All in all the algorithm with the LAP filter had the best result.

Contents

1 Introduction

Pictures are important for the humanity all around the globe since thousands of years. Already 30.000 years ago scenes from daily life have been painted from the memories onto the walls of caves [17], and later on pottery or immortalized in mosaics. Coming closer to now there are also paintings about nature and portraits for supporting and showing the memories of travelers, families and in normal people. As it is an important part of the social life pioneers in science started to improve and automize the process of bringing the short living pictures from the eyesight down to paper for storing and sharing it. In the early 18th century it started with chemical processes [2] and since the 1990s the digital capturing of the surrounding became affordable for the broad society [3].
It is imaginable, that paintings, mosaics or comparable captures of memories have been expensive and also dependent on the skill and talent of the artist, at least if it had to be realistic
With the analog photography it already became easier and probably cheaper. Nevertheless, each photo needed extra film material with limited space for pictures, it took time for the final processing and took space for storing. So, each picture was more valuable and taking this in account more care was given to each "shot". All this mostly resulted in a smaller number of pictures and those had ideally a better quality, respecting the state of the technical possibilities.
Today over 70 Percent of the population in western Europe owns a smartphone [7]. Most smartphones have adequate up until excellent digital cameras integrated. Concerning the costs and effort of taking a picture, this changed the availability and number of pictures in general and for each subject drastically. Even though the cameras are becoming better and pictures out of focus or with a too dark or bright exposure are becoming less, people tend to shoot each picture several times to make sure having the best image.
Over months and years this can lead to an aggregation of a confusing stock of digital memories. It can be assumed that there will be also many images which are not worth to keep and in general probably too many to watch them all. In case these photos should be used as memories they need to be sorted and only the best ones should be kept. As there can be pictures which are out of focus, are too dark or too bright.

This thesis is discussing this issue and an approach for assessing images concerning their exposure and focus for filtering them automatically.
At the beginning there is a short section about the technique of digital cameras and the main reasons for images with lower quality. After knowing the issues the approaches for processing and interpreting the images mathematically will be explained. Before the development of the algorithm for sorting the images and finding the thresholds, the used datasets will be explained as well as their creation. Eventually the performance is evaluated and compared to other tools.

2 Background: Digital Images

To understand what digital images are and how they are created, there is a short explanation of the main contributing factors when taking a picture and different approaches of handling the information about the color.

2.1 Creation of Images

Many photos are now taken with small photo sensors integrated in smart phones. There, the adjustment possibilities of aperture, ISO, focal length and shutter speed known from system or SLR cameras are limited. It is tried to compensate this either by several camera modules or on software basis. In order to understand the basic principle, the functionality is explained further on a camera with lens.
The knowledge about the influencing factors by the variation of these exposure and focus parameters shall enable a better understanding of the results in the photos. Furthermore, evaluation vectors could be derived for exposure strength and focus.

Focal Length The term focal length used in cameras can also be described as the angle of aperture of the field of view. The change of the focal length leads to the effect of different opening angles on the photo, which enlarge or reduce the image area and in the opposite direction reduce or enlarge the subject. The changing of the focal length is often called zooming [18].

Focus The focus adjusts the optimal distance of the lens from the photo sensor in relation to the distance of the subject to the lens. This optimal distance creates a sharp subject for the area in the image that is the appropriate distance from the lens [18].

Depth of Focus Depth of field is the range of depth in a photo that appears sharp to our eyes. With a shallow depth of field, only a small area in the image plane is sharp. For example, the face in a portrait. In a landscape photo, on the other hand, the depth of field is usually the whole area and almost everything appears sharp to our eye. The depth of field is determined by the focal length, where the focus is on the image and the light sensitivity of the lens [18].

Shutter The shutter speed indicates the duration of the exposure of the film or sensor. The longer the exposure time, the more light rays can be captured. This results in a brighter image [18].

Aperture The aperture is a mostly adjustable hole in the lens. This allows to control the amount of light entering the lens. It can be compared to the pupil in the eye [18].

ISO The ISO value indicates the light sensitivity of the evaluation of the incident light rays. In analog photography, an entire film had the same light sensitivity. In digital photography, the sensitivity of the sensors and their evaluation can be adjusted. In simplified terms, it considers the number of incident light rays during the exposure differently weighted [18].

Sensor The image sensor is a light-sensitive module consisting of many semiconductors behind the lens or the lens of the camera and registers the incident light beams. This can then be read out and digitized [18].

At least several of those parameters need to be taken in account to get a good shot. Nevertheless, it is possible that at least one of these parameters is not perfectly set and it leads to a bad result concerning the quality. No matter whether the settings have been chosen manually or automatically.
To which quality issues this can lead will be explained in chapter 3.2.

2.2 Color spaces

Color spaces are different approaches to represent the color values of pixels in images. They can be optimized for different applications like color printers, compression, human eye, medical images.

RGB In the RGB color space, each pixel is assigned values for red, green and blue. They are 256 steps resulting in 8 bits each, which covers a value range from 0 (dark) to 255 (bright). These values are added together for the common color. Most common displays work with this color space [11].

HSV Alternatively, there is the HSV color space. This is based on the English terms Hue (color), Saturation and Value (brightness value). Here the colors are represented in a cylinder or cone. The height is the brightness, the angle is the color and the radius is the saturation. One motivation for this representation is the approximation to the perception of colors by the human eye [16].

Gray scale Compared to the pixels in RGB gray scale images have only one value. It is the variation of grey with 0 (black) to 255 (white). The conversion itself is simplified the average brightness of the color channels. In detail it more complex and for example discussed in the publication *Color Image to Gray scale Image Conversion* [1]. Here used conversations are based on the OpenCV library in Python [8].

These is a selection out of many color spaces to understand different methods. However, this paper will focus on RGB due to its simple structure and sufficient flexibility for the here discussed assessment.

3 Image Quality

Image quality can be separated into two fields. One is the content itself and the other is the quality of the way it is captured.
The content itself is often subjective and hardly measurable with physical or mathematical approaches. Therefore, this paper is concentrating on the other pillar of a good photo.
The sharpness and the exposure of the image will be evaluated and attempted to decide if it is valuable to keep or not for further evaluation of the content.

3.1 Sharpness

Before an evaluation can begin, a definition of sharpness is required. As already mentioned for the HSV color space, the human perception of an image differs from the physical representation. In human visual perception, certain things are perceived more strongly and others less strongly. These differences shall be explained here.

Physical Sharpness Physically or technically, an image is only sharp in one image plane. This means that only those objects which are at a certain distance from the camera are sharp. This distance is determined by the focus of the lens or objective and has the focal point exactly on the photo sensor. The focal point of objects closer or further away from the camera is not exactly on the height of the sensor and therefore out of focus.
If an image is reduced to an edge, this is reflected in the hardness of the transition from one color to another or from light to dark

Impression of Sharpness The impression of sharpness describes the sharpness perception of the human sense of sight. In Figure 3.1 you can see that the second column from the left also appears relatively sharp, although the transition extends over more than two pixels. In more complex images many edges are not as clear as the leftmost one and the image is still perceived as sharp. This means that the human eye's perception of sharpness is less accurate. It is also based on technical sharpness, but there are more factors that influence a good impression of sharpness.

Figure 3.1: Here are five columns with transitions of different thickness at the edges. From left to right the sharpness decreases noticeably.

Contrast Image contrast, the difference between light and dark areas of the image, has a strong effect on the sharpness impression. Images with increased contrast, especially at the edges, appear sharper. This effect is used in unsharp masking for improving the sharpness of images [15].

Resolution A problem that no longer occurs so frequently is a too low resolution when displaying or capturing the images. The resolution describes the number of pixels in an image or on a display. With a low-resolution screen, the image itself is not automatically blurred, but can only be displayed in a less than optimal way. If the image has a low resolution, details can no longer be displayed accurately, because areas of perceptible size are represented by one pixel and therefore only one color is available. This loss of detail is perceived as blurring.
A resolution of the image corresponding to the display size is not a guarantee for a sharp image, but a condition.

Level of Detail Pictures which show details clearly visible are perceived as sharper. Such details can be e.g. trees and their leaves or waves in water in landscape photos. If such details are expected but not seen, the image may not be perceived as sharp enough.

3.2 Quality Lowering Factors

3.2.1 Blurriness

Blurriness is the result, when the there is no focus. On pictures this can be all over the image or just not of the motive. For example in a portrait the face is sharp, but the background is blurry.

Out Of Focus Blur An image may be out of focus if the focus or focal point on the sensor has been adjusted for a distance that is not relevant to the image. This causes parts or the whole image to appear blurred [5].

Motion Blur The motion blur can be recognized by a blurred subject. This is caused, for example, by too long exposure time and a shaky hand or by subjects moving too fast [5].

3.2.2 Exposure

Exposure is basically resulting in the brightness of the image. If the camera settings for shutter, aperture and ISO are not suitable adjusted for the light of the scene it can either result to over or under exposed images.

Overexposure Overexposure can be recognized by larger very bright to white areas in images. This causes structures and thus information to be lost. This can be caused by a too high ISO value or too long exposure.

Underexposure The underexposure can be recognized by very dark or black areas in images. It is the opposite of overexposure.

Further not considered Factors There are also partly already mentioned factors like resolution which can affect the quality. These will not be considered further.

The goal of this work is to recognize images that are sharp and blurred for the human eye. Therefore, the result is focused on the impression of sharpness and is regarded as the sharpness to be evaluated. The determination and classification of contrast in the sense of evaluating the sharpness of an image requires a higher effort and will therefore not be discussed further in this paper. The recognition of technical sharpness is already known and there are proven procedures for this. To what extent these are suitable for the recognition of images that are sharp and no longer sharp for the human eye will be examined more closely here.

4 Analysis of Images

After the main problems for lower quality in images are identified, the next step is to detect these. The two factors blurriness and exposure will be discussed separately. In this chapter will give general information about image processing with three types of operators and their results. Their mathematical background is not considered more deeply.

4.1 Approaches for Image Quality Assessment in Focus

In the research and application area of computer vision there are a variety of approaches to evaluate the focus of a digital image. In a publication investigating the performance of different algorithms for image depth detection, six different ways of calculating the focus are distinguished [10]:

- "*Gradient-based operators* (GRAn). This family groups focus measure operators based on the gradient or first derivative of the image. These algorithms follow the assumption that focused images present more sharp edges than blurred ones. Thus, the gradient is used to measure the degree of focus."
- "*Laplacian-based operators* (LAPn). Similarly to the previous family, the goal of these operators is to measure the amount of edges present in images, although through the second derivative or Laplacian."
- "*Wavelet-based operators* (WAVn). The focus measure operators within this family take advantage of the capability of the coefficients of the discrete wavelet transform to describe the frequency and spatial content of images. Therefore, these coefficients can be utilized to measure the focus level."
- "*Statistics-based operators* (STAn). The focus measure operators within this family take advantage of several image statistics as texture descriptors in order to compute the focus level."
- "*DCT-based operators* (DCTn). Similarly to the wavelet-based operators, this family takes advantage of the discrete cosine transform (DCT) coefficients in order to compute the focus level of an image from its frequency content. None of the operators within this family have previously been used in SFF applications to our knowledge."

Figure 4.1: Lenna in color

- "*Miscellaneous operators* (MISn). This family groups operators that do not belong to any of the previous five groups."

The result of the authors was, that operators based on Laplace performed best on average. Also the majority of projects in GitHub seem to use algorithms based on the Laplacian. To get a comparison for the application in this paper and because they are related to Laplacian operators also GRAn will be considered.

4.2 Operations on Images

Chosen operators from GRAn and LAPn will be discussed here. Both are operations on the pixel values. For simplification only gray scale images will be used here.
The operations are on functions with two dimensions. The mathematical interpretation of an image is shown for better understanding in original Lenna in 4.1 and her 3D plot of in gray scale in figure4.2. Images are seen as matrices with each value representing the pixel's intensity.
The pixel values are discrete so for also the operations are discrete approximations. This is reached with a convolution over the image with a filter frame. These filter frames are sometimes also called kernels. Their size can differ with different effects for example at the edges of images. Here only 3×3 kernels will be discussed.

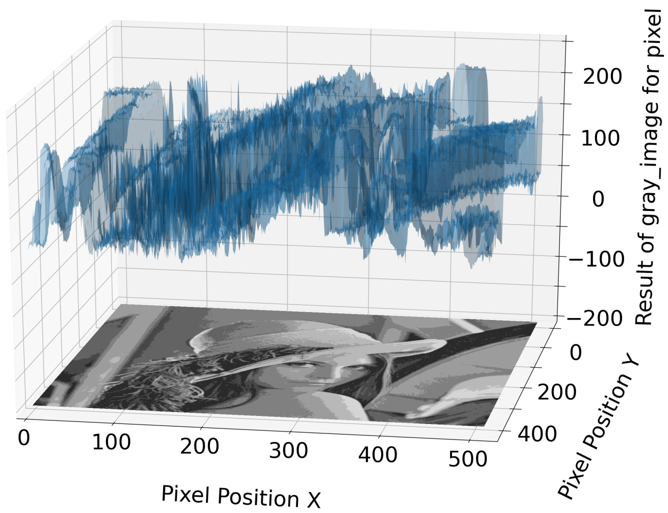

Figure 4.2: 3D plot of Lenna (Fig. 4.1) in grey scale. On the surface Lenna can be seen in gray scale. It is the 2D projection of the plot.

4.2.1 Gradient-based operators

As previously mentioned gradient based operators are the first derivative or extensions of these. Here will be discussed the frequent applied Sobel kernel and Scharr kernel.

Sobel The Sobel operator has kernels for the x and for the y derivative. The 3D kernels are reached by the the multiplication of the x- and y-derivative vector and the 1D Gaussian filter [6][9]. In the Sobel Kernel

$$Sobel_X = \underbrace{\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}}_{1DGaussianfilter} * \underbrace{\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}}_{XDerivative} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.1}$$

the Gaussian filter smooths the operation and due to this noise has less impact. The result after the $Sobel_X$ kernel was applied is plotted in figure 4.3.

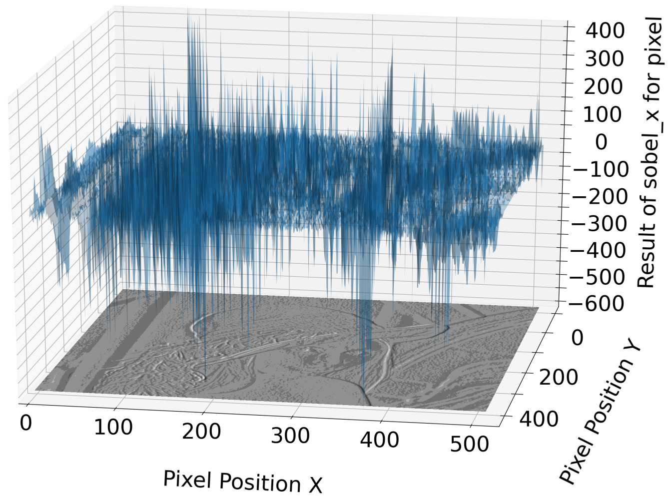

Figure 4.3: 3D plot of Lenna (Fig. 4.1) in gray scale after the $Sobel_X$ filter was applied. On the surface Lenna can be seen in gray scale. It is the 2D projection of the plot.

Scharr The $Scharr$ kernel is working similar to the $Sobel$ kernel. The improvement is, that

$$\underbrace{\frac{1}{32} * \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix}}_{Scharr_X} \quad \text{and} \quad \underbrace{\frac{1}{32} * \begin{bmatrix} 3 & 10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & -3 \end{bmatrix}}_{Scharr_Y} \tag{4.2}$$

are maximally optimized concerning the rotational symmetry of the Gradient [13]. The result after the $Scharr_Y$ kernel was applied is plotted in figure Ap.F1.

4.2.2 Laplacian-based operators

The discrete Laplacian operator

$$[13] \quad Laplacian = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{Laplacefilter} \tag{4.3}$$

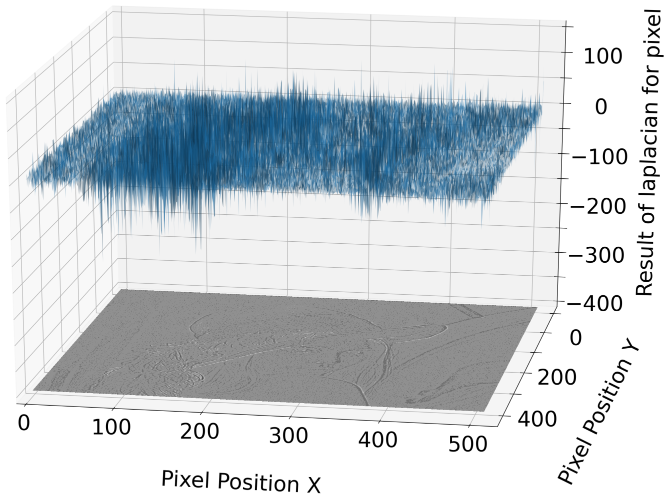

Figure 4.4: 3D plot of Lenna (Fig. 4.1) in gray scale after the $Laplacian$ filter was applied. On the surface Lenna can be seen in gray scale. It is the 2D projection of the plot.

in the second dimension is approximating the second derivative of the image pixel values. The application of the Laplacian operator on Lenna can be seen in figure 4.4.

4.3 Blurriness/Focus

After the three operators have been explained the results need to be interpreted and adopted for detecting blurriness.
In chapter 3.1 and figure 3.1 was explained that the impression of Sharpness is highly influenced by the edges and their texture.

Compared to the plots from Lenna, figure 4.6 shows a better understandable result of the $Sobel_X$ operator applied to an image. The original image with the columns is figure 3.1. In figure 4.6 it is visible, that the sobel operator is "reacting" in the edges from figure 4.5 with (local) maxima when it is going from dark to bright and (local) minimum when it is changing from bright to dark.
Another observation is the different intensity of the amplitudes. The "sharper" the edge is, as in 3.1 explained, the higher is the amplitude.
The observations of the plots for $Sobel_Y$ (fig. Ap.F2), $Scharr_X$ (fig. Ap.F3), $Scharr_Y$ (fig. Ap.F4) and $Laplacian$ (fig. 4.7) are different. Whereby the plot

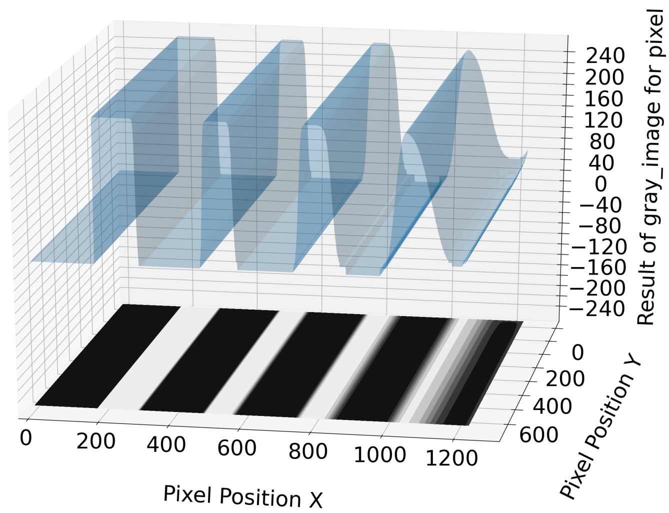

Figure 4.5: 3D of (Fig. 4.1).On the surface the edges can be seen.

of $Scharr_X$ is only distinguishing in the height of the amplitude, the plot for the $Laplacian$ has not only a different amplitude, but also two zero crossings at each edge. Nevertheless, these differences are only resulting in a different height of the result in the later explained variance in equation 4.4.
As the image 3.1 only has edges in one direction, the other does not have edges. Therefore, the plots for $Sobel_Y$ and $Scharr_Y$ are more or less zero. The plot Ap.F1, the result of $Scharr_Y$, shows the edges. But compared to figure 4.3, the result of $Scharr_X$, different edges of the images lead to deflection. Consequently, it is necessary to consider both dimensions when using GRAn operators for analyzing the whole image.
In the most cases one sharp edge does not make the whole image sharp. Therefore, the edges together with their quality of transition need to be count.
The variance

$$VAR\_SOB_X = \sum_x^X \sum_y^Y \left[|S_x(x,y)| - \overline{S} \right]^2 \tag{4.4}$$

takes both together in account. The equation subtracts the mean

$$\overline{S_x} = \frac{1}{XY} \sum_x^X \sum_y^Y |S_x(x,y)| \tag{4.5}$$

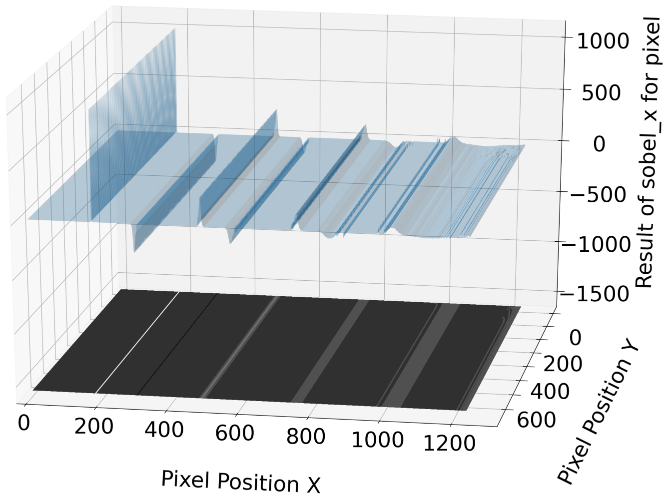

Figure 4.6: 3D of (Fig. 3.1) after the $Sobel_X$ filter was applied. On the surface the edges can be seen.

from

$$|S(x, y)| \tag{4.6}$$

as the absolute value of the result after the $Sobel$ kernel was convoluted over the image. By the definition of the variance each summand is squared.

Due to all values are summed up, many high values in the first derivative, which was already detected as a good indicator for a sharp edge, result in a high end result. Therefore, a high variance very likely means a sharp image.

For the $Sobel_Y$, $Scharr_X$, $Scharr_Y$ and $Laplacian$ the calculations are the same, therefore they will not be further explained and defined as VAR_SOB_Y, VAR_SCHARR_X, VAR_SCHARR_Y and VAR_LAP

4.4 Exposure

After Blurriness, Exposure is the second quality characteristic to filter for in the later algorithm. For that two ways to measure this will be explained.

Average Exposure Over- or Underexposed images mostly have a high amount of either bright or dark colors in the image. Consequently this is leading to a higher or lower average of the brightness of the image. This average can be calculated

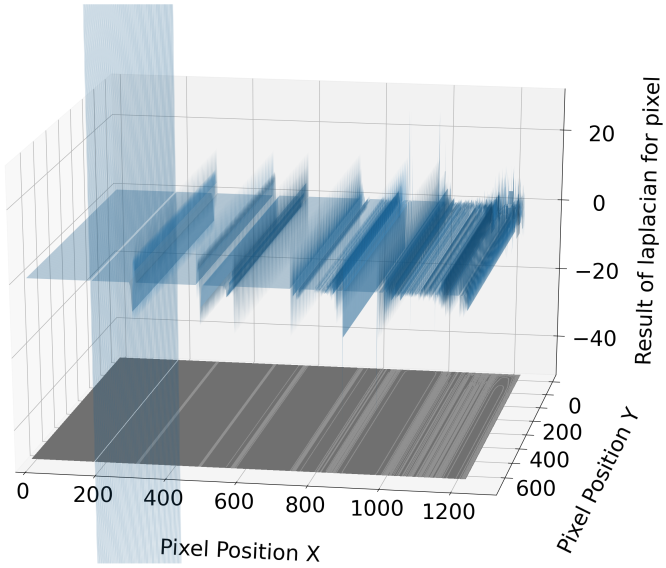

Figure 4.7: 3D of (Fig. 3.1) after the Laplacian filter was applied. On the surface the edges can be seen.

similar to the mean of the Sobel (eq. 4.5). Instead of the matrix after the $Sobel_X$ was applied, the one from the gray scale image is used.

Percentage dark and bright areas Images can not be only too bright or dark over the whole image. It is also possible, that there are just areas too bright or dark. These areas can be detected by counting the number of pixels with a certain value above or underneath of thresholds. After investigating images the result was, that dark and white areas are not only the extreme 0 and 255. It turned out, that pixels darker than 15 and brighter than 240 are a good definition for to dark or to bright.

5 Datasets

The algorithm for assessing images has to be developed and verified on datasets. This chapter is explaining the how the qualities of the images are rated and how the Test- and Verification Datasets have been created.

5.1 Rating of images

To understand which threshold is needed for the extracted parameters and to evaluate the performance afterwards, already rated images need to be used. Both sets have different requirements to the development of the rating, which will be discussed later.
The algorithm should detect the level of focus and the level of exposure. To get a better overview of the sets the images have been categorized by the evaluates in either *Animal*, *Landscape*, *Macro*, *Night*, *Portrait* or *Other*. Furthermore, the kind of exposure was classified to *OK*, *Over*- or *Underexposed* and the kind of blurriness to *OK*, *Motion* or *Out Of Focus*. The rating for the quality of focus and exposure was collected separately and orientating on the *Mean Opinion* Score.
Relying on an article from the Springer Jounal *Mulitmedia Systems* [14] the

> "Mean opinion score (MOS) has become a very popular indicator of perceived media quality."

Its advantage is a easily understandable rating. They can differ by their discrete levels. Here the often used *5-point MOS scale* with the discrete levels *excellent*, *good*, *fair*, *poor* and *bad* has been used. Once for the quality of focus and once for the exposure.

5.2 Webapp for creating Datasets

To create the datasets it is necessary to have an application to walk through the images and safe the rating. As there was no such tool found, which meets the above mentioned requirements, it had to be developed.
The figure 5.1 shows the structure of the webapp and database on a high level. The webapp itself is written with *Python*, *HTML* and *CSS*. The *Flask* library was used for the server and the database is *SQLite*.

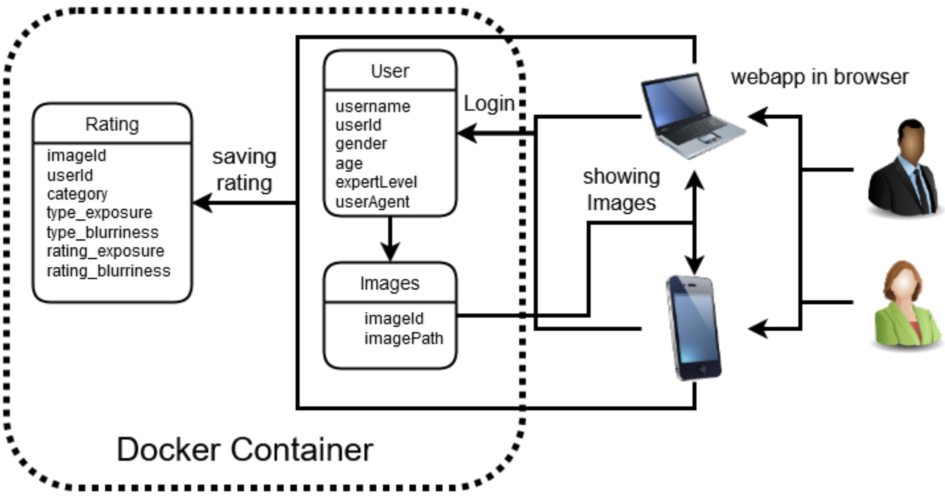

Figure 5.1: High Level flowchart of the webapp.

The whole application is deployed in a *Docker Compose* environment with an additional container for a *nginx* reverse proxy. It was hosted on a server of the *Chair for Data Processing (LDV)* of *Technical University of Munich* and temporary public reachable under `http://demo.ldv.ei.tum.de/`.

5.3 Test Dataset

The Test Dataset consists of 362 images. These are a mixture of casual images and special images were specially made for this dataset. For example Macro shots with different focus or the same motive of a house and garden with different focus and different exposure.
For a better understanding of the consistence there are number of images per labeling or sorting. The category other contains mostly images for with the same motive with different parameters to find the threshold more granular.
The final rating of these images can be seen in the tables 5.1, 5.2, 5.3 and 5.4. They have been asseset by one person and are only used for finding the threshold. It is not used for verification.

	Animal	Landscape	Macro	Night	Other	Portrait
# images	12	77	35	5	227	6

Table 5.1: Test Dataset/Number by category

	OK	Motion	Out Of Focus
# images	226	39	97

Table 5.2: Test Dataset/Number by type of blurriness

	OK	Over Exposed	Under Exposed
# images	212	80	70

Table 5.3: Test Dataset/Number by type of exposure

	Excellent	Good	Fair	Poor	Bad
# images focus	109	70	50	48	85
# images exposure	57	127	59	36	83

Table 5.4: Test Dataset/Rating for level of focus and exposure

Assessment of the Test Dataset Generally, the Test Dataset has a reasonable size with 362 images. Furthermore, it contains images with small changes in focus and exposure from the same motive. So a granular determination of thresholds should work good. Nevertheless, the dataset also has disadvantages. Even though there are many images, three categories only have a small number. In total it will be sufficient to have a realistic cut through of images which are collected for memories by amateur photographers. But as it is only rated by one person, the algorithm will very likely orientate on subjective preferences.

5.4 Verification Dataset

Compared to the Testset the Verificationset only contains 40 images. The overview can be seen in tables 5.6, 5.7, 5.8 and 5.9. It was rated by 16 people from the age between 18 and 51. Out of these only 11 have been rating all 40 images and used to the create the final verification set. More information can be found in table 5.5.

	gender	expert Level	platform
# persons	(5/6)	(1/0/7/3/0)	(5/6)

Table 5.5: Verification Dataset/conditions of voters. Gender [male/female]; Expert Level [excellent/good/fair/poor/bad]; Platform [monitor/mobile]

The answers for each image have been differing from person to person. To get a final dataset with one rating per rating category of the image the average

$$average\_rating = \left\lfloor \frac{\sum_{rating=1}^{x} r_{rating}}{x} + 0.5 \right\rfloor \tag{5.1}$$

of each rating r for the exposure and blurriness have been calculated. Where x is the number of ratings. The discrete levels have been mapped to 1 for excellent to 5 for bad.
For *category*, *type of blurriness* and *type of exposure* the most chosen option has been taken.
This dataset will only be used to evaluate the performance of the algorithm.

	Animal	Landscape	Macro	Night	Other	Portrait
# images	3	17	2	7	6	1

Table 5.6: Verification Dataset/Number by category

	OK	Motion	Out Of Focus
# images	32	2	6

Table 5.7: Verification Dataset/Number by type of blurriness

	OK	Over Exposed	Under Exposed
# images	28	6	6

Table 5.8: Verification Dataset/Number by type of exposure

	Excellent	Good	Fair	Poor	Bad
# images focus	10	21	2	6	1
# images exposure	9	17	9	5	0

Table 5.9: Verification Dataset/Rating for level of focus and exposure

Assessment of the Verification Dataset Even though the dataset is well distributed by gender of test persons and the platforms they have used, there are several issues. Unfortunately the skill level of the persons is mostly on a low level. This resulted in unclear ratings for several pictures. Due to the small size of the dataset and because the test persons rated a few images to another category than expected, the categories *Animal*, *Macro* and *Portrait* only have a very small number of pictures. In summary, this might still give a more or less realistic picture of peoples opinion about images and so it can be used to compare with the developed algorithm. Nevertheless, a dataset with more test persons would give a more stable result.

6 Sorting Algorithm

Eventually the explained theory about images and their analysis is going to be combined and successive applied to reach a filter. As already done in other chapters, the two possibly quality lowering factors exposure and blurriness are covered in different sections. In the end of the chapter they will be combined.

6.1 Approach

All these single filters are relying on a threshold, which will be determined. The challenge is, that there are not just excellent and bad images, but the five in chapter 5.1 explained levels. Therefore, binary classification testing and optimization can not be used here.
The approach here will be to firstly plot a boxplot to find ranges where a good threshold possibly can be. Afterwards within this range the percentual drop rate for discrete thresholds is plotted. The images which are dropped depending on the filter, either with a higher or lower value than the threshold. This can be equated with the percentage of images which would be dropped by such filter with given threshold. As there are five ratings, each rating is plotted for each discrete threshold in a different color.
In the plots the relation between possibly lost images with excellent rating and kept images with bad rating has the highest priority when finding the threshold. With second priority, but mostly more granular, the lost-keep relation between good and poor images is used to find the possibly optimum for the threshold. The viewpoint is to prefer keeping possibly excellent and good images over dropping many poor and bad images.
Such complexity, especially when the filters are combined, raises the idea to use Machine Learning Models for finding the perfect threshold. Presumable, this is leading to a better result, thought this adds a higher complexity and would exceed the frame of this paper. Hence, the determination of the threshold will be done by observation of plots.

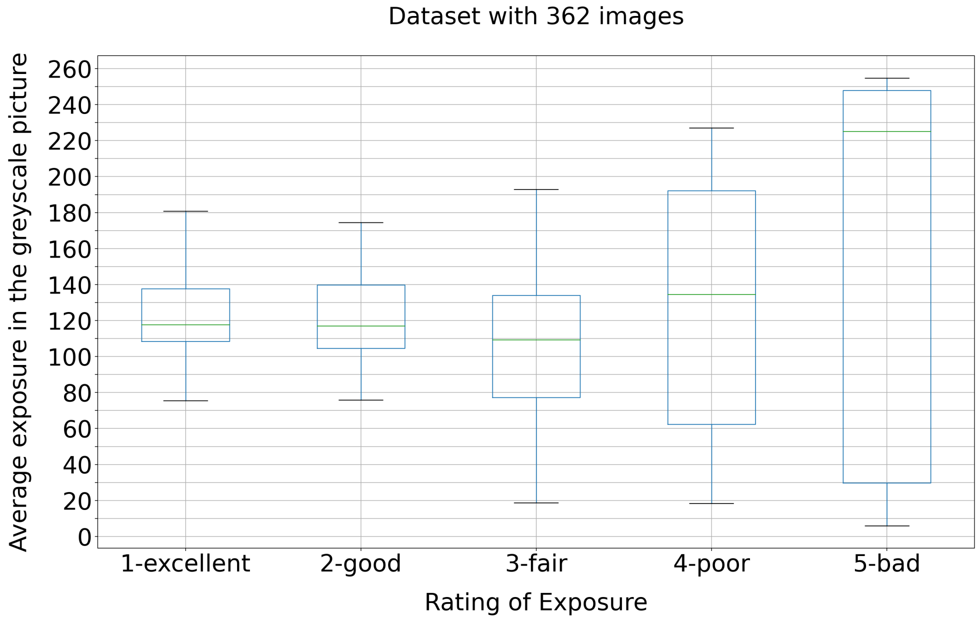

Figure 6.1: Boxplot over the average exposure from the images from Test Dataset (chapter 5.3).

6.2 Exposure

For the two values covered in chapter 4.4, the exposure thresholds will be searched separately and in the end combined to one filter for exposure. The ratings in the plots in this section are only referring to the exposure.

Average Exposure In chapter 4.4 was explained, that brighter or darker, so over- or underexposed, images have a higher or smaller average exposure. This means well exposed images should have a average exposure in the middle between 0 and 255.

In the boxplot 6.1 can be seen, that excellent and good exposed images have a mean of their exposure in the middle between 0 and 255. Poor and bad exposed are more distributed. This boxplot confirms the expectations and validates the dataset concerning the relation of rating and average exposure dataset.

Overexposed For the over exposed images the range in the plot 6.2 is between 125, the middle, and 225. The further the value for the threshold is away from 125 the less images with excellent and good rating are dropped. Because the drop rate of bad images is constant between 60% and 65% until 175, it is possible to

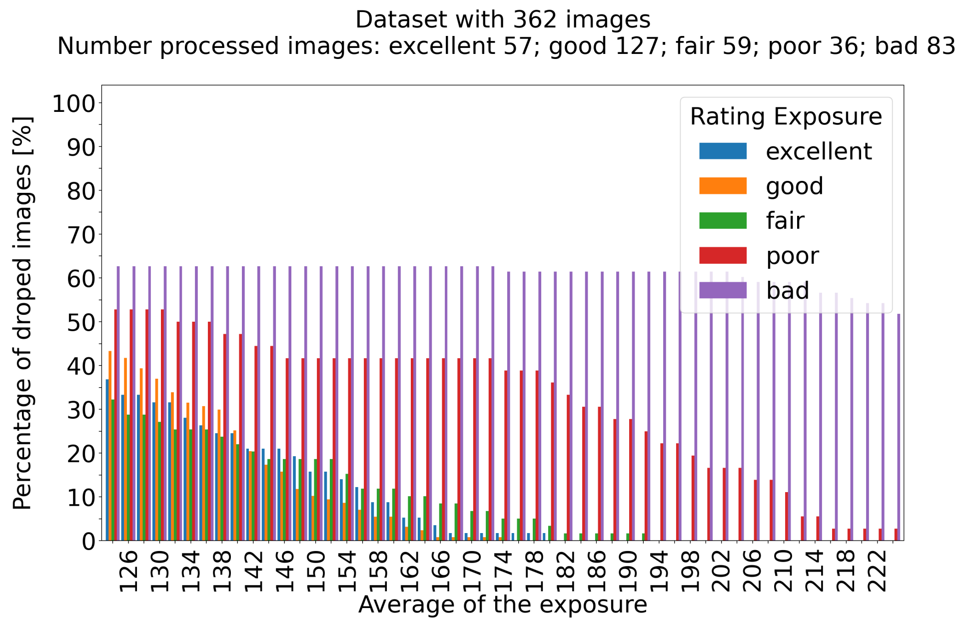

Figure 6.2: Plot of the drop rate for images with a higher value for the average exposure of the image than the threshold. This plot is for over exposed images

concentrate on the excellent and good images. From 169 on the drop rate for these is under 5%. Finally when taking the poor rated images in account a threshold of 170 seems promising, as the rate is still over 40%.

Underexposed For the under exposed images the range in the plot 6.3 is between 25 and 125. The closer the values for the threshold are coming to 25, the less images are dropped. Compared to the plot for the over exposed images 6.2 the drop rates are lower for most of the categories. The rates for bad and poor images stay around 35% until the a threshold of 43, meanwhile possibly lost images with excellent and good rating are becoming significantly less. At a threshold of 75 neither a single excellent nor a single good images is dropped. Therefore this is a very promising threshold.

Percentage dark and bright areas The boxplot 6.4 shows distribution by rating for the in chapter 4.4 explained value for the percentage dark and bright areas of the images.
As expected better rated images have a lower percentage and worse rated a higher percentage. The range between 0 until 50, the bottom of bad, seems to be the most promising.

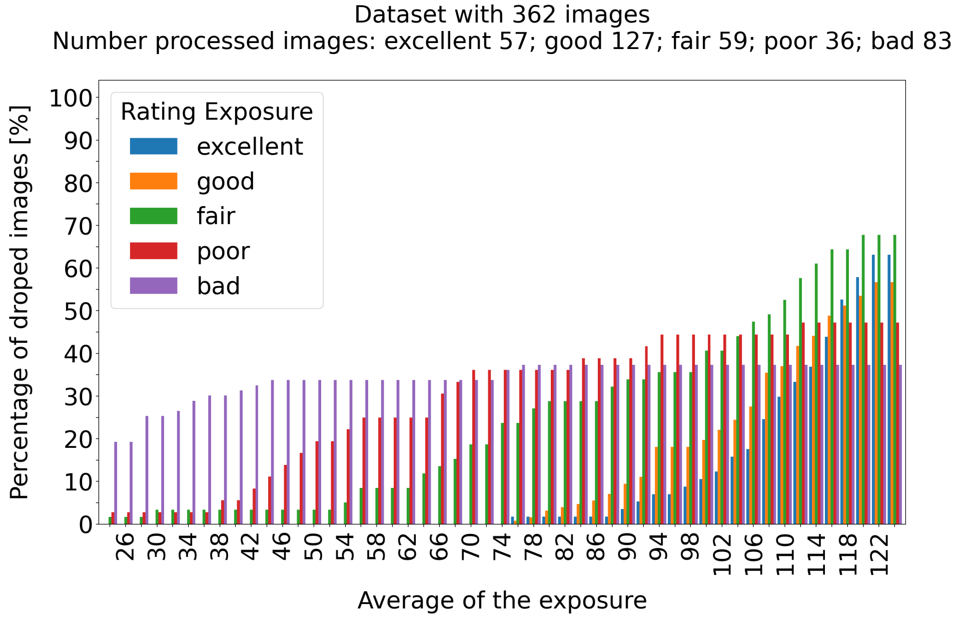

Figure 6.3: Plot of the drop rate for images with a lower value for the average exposure of the image than the threshold. This plot is for under exposed images.

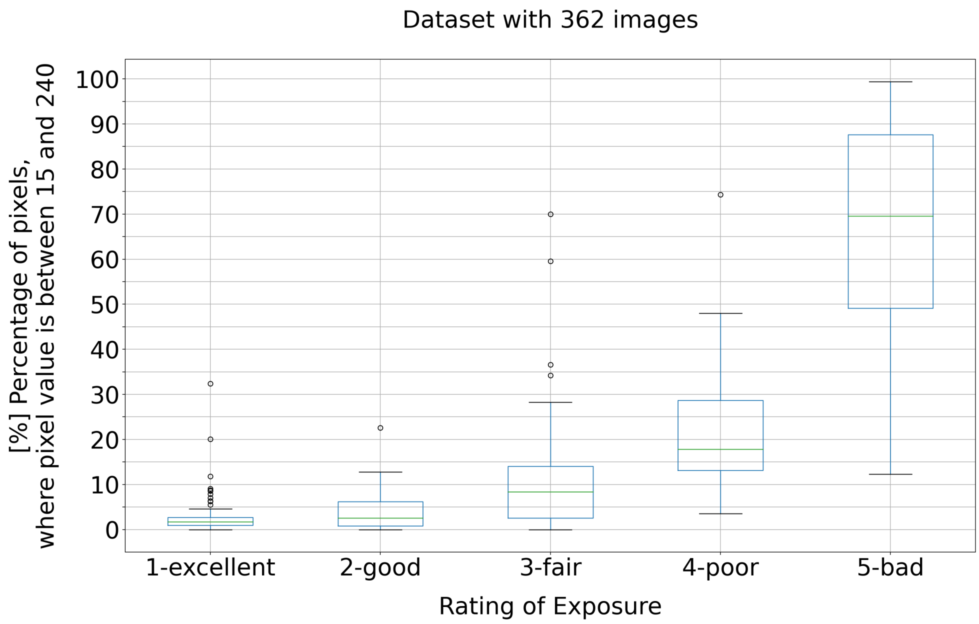

Figure 6.4: Boxplot over the percentage of dark and bright areas from the images from Test Dataset (chapter 5.3).

Figure 6.5: Plot of the drop rate for images with a higher percentage of dark and bright areas of the image than the threshold.

This range is plotted in figure 6.5. From a threshold of 12 on the drop rate for excellent images is at around 5% while it is still at 100% for bad images. Furthermore, taking fair and poor rated images in account, a threshold of 12 seems promising. Nearly no good images are dropped while over 80% of the poor images are detected.

Combination With exception of several images, like captured at night, an image which is too bright or dark in general is mostly not well exposed. It is the same with images with bigger bright or dark areas. But it is not always given, that both can be found at the same time. The filter for average exposure is focusing on the detection of the first mentioned and the filter for the percentage of dark and bright areas is more sensitive to dark or bright areas.
Therefore, a combination of both should improve the result. As a change of the thresholds for the average exposure are not changing the keep-drop relation reasonable, the thresholds will be kept and only images with average exposure 75 and 170 be kept. The thresholds for finding the percentage of dark and bright areas was defined in chapter 4.4. Only the threshold for the value percentage will stay variable like in figure 6.5. The resulting plot is 6.6 over a dataset with remaining 231 images.

Compared to plot 6.5, in plot 6.6 big steps for poor images can be seen. This

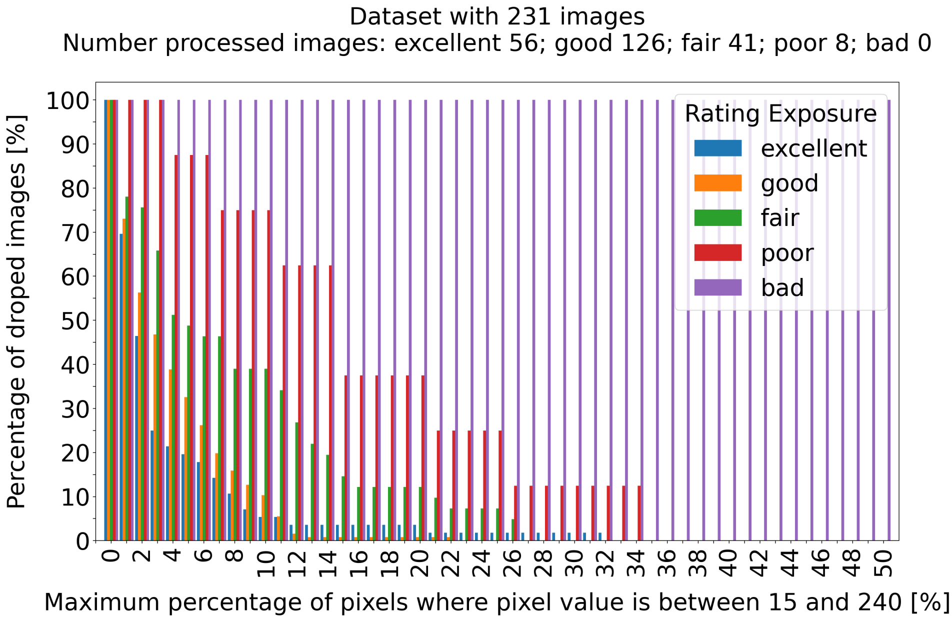

Figure 6.6: Plot of the drop rate for images with a higher percentage of dark and bright areas of the image than the threshold. The dataset contains only images with an average exposure between 75 and 170.

is caused by the small number of 8 poor images. Nevertheless the before chosen threshold of 12% is still promising and detects 5 more poor exposed images, than just the filter for average exposure.

6.3 Blurriness

For the three in chapter 4.3 covered values to recognize the sharpness thresholds will be searched separately. The ratings in the plots in this section are only referring to the blurriness.
For $Sobel$, $Scharr$ and $Laplacian$ a higher variance indicates more and sharper edges. Therefore, in all three filters the thresholds are minimum values the variance of an image has to have to be detected as not blurry.

Sobel The Boxplot 6.7 shows the distribution by rating for the VAR_SOB_X application on the images from the test dataset. The values between 0 and 1000 seem to be interesting, because the range of excellent and good rated images is nearly reaching zero. Furthermore, can be seen, that there are several outliers with poor and bad rating. With this filter they will not be detected.
When comparing the boxplot 6.8 for the VAR_SOB_Y with the one for the

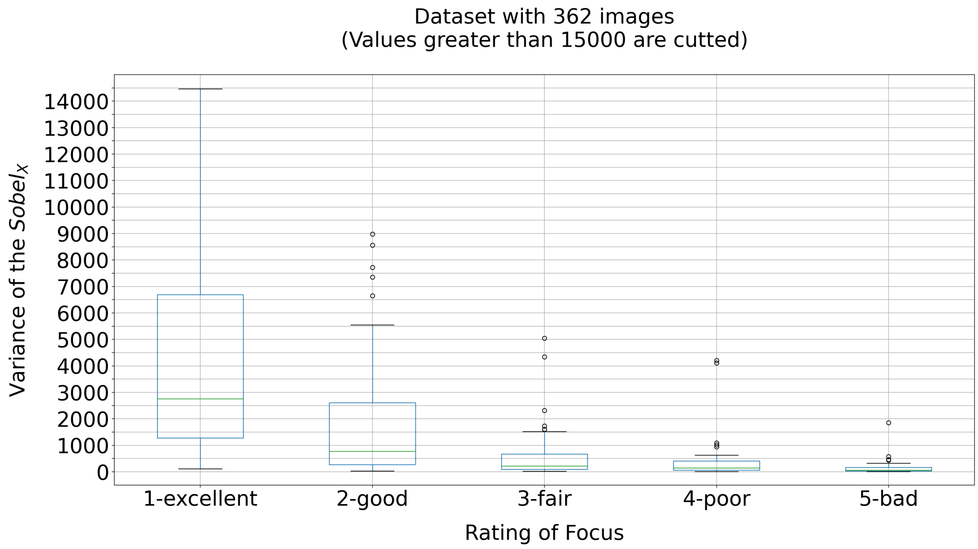

Figure 6.7: Boxplot of the VAR_SOB_X from the images from Test Dataset (chapter 5.3). Data points with values above 15000 have been cut, to see the range with lower values better. Original Boxplot can be found in appendix Ap.F5.

VAR_SOB_X (6.7) a similarity can be seen. Mainly only the outliers are different. The same can be observed in the plots with the drop rates for VAR_SOB_X (Ap.F7) and VAR_SOB_Y (Ap.F8). This raises the assumption to link the two thresholds and find one for both together. Therefore VAR_SOB_{XY} will be used further.

Plot 6.9 combines both thresholds and shows the drop rate of pictures which have a VAR_SOB_{XY} smaller than the thresholds. Compared to the figures Ap.F7 and Ap.F8 the drop rate of excellent images became lower. The other rates did not change significantly. Unfortunately the drop rate of good images keeps high. To still have a practical drop rate of bad images, it could be a good trade off to choose 260. The drop rate for good images is at 20% while still 85% of bad and over 50% of poor images are dropped.

Scharr For the VAR_SCHARR_{XY} the procedure is the same as for VAR_SOB_{XY} in chapter 6.3. The plots can be found in appendix. The box-plots are Ap.F10 for X and Ap.F12 for Y and the plots for getting the threshold are Ap.F13 for X, Ap.F14 for Y and Ap.F15 for the plot with linked thresholds.
The here determined threshold for VAR_SCHARR_{XY} is 4200.

Laplacian Compared to $Sobel$ and $Scharr$ there is only one $Laplacian$ kernel. So there is only one threshold to determine. The range for the plot 6.10 was observed in the boxplot Ap.F16. Compared to the plot 6.9 for finding the

Figure 6.8: Boxplot of the VAR_SOB_Y from the images from Test Dataset (chapter 5.3). Data points with values above 15000 have been cut, to see the area with lower values better. Original Boxplot can be found in appendix Ap.F6.

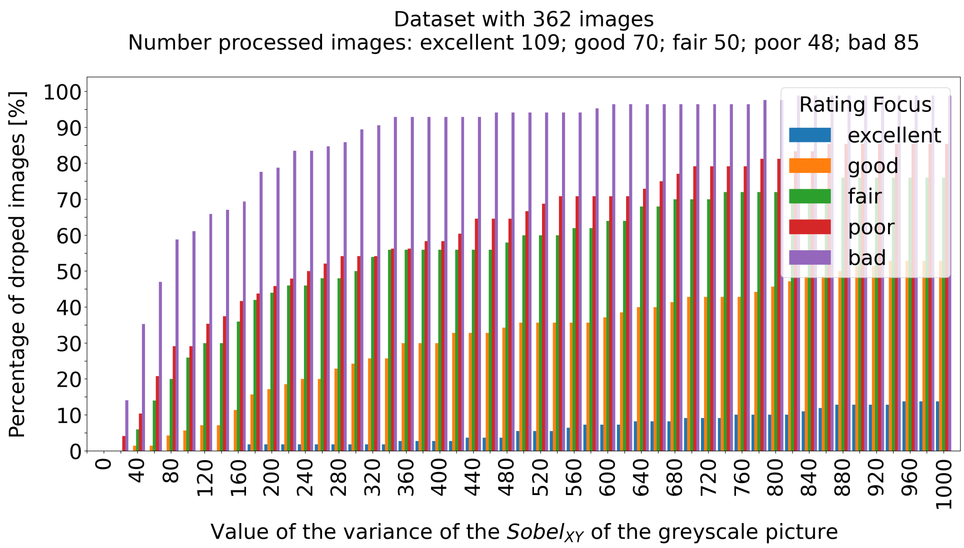

Figure 6.9: Plot of the drop rate for images with a lower VAR_SOB_{XY} of the image than the thresholds.

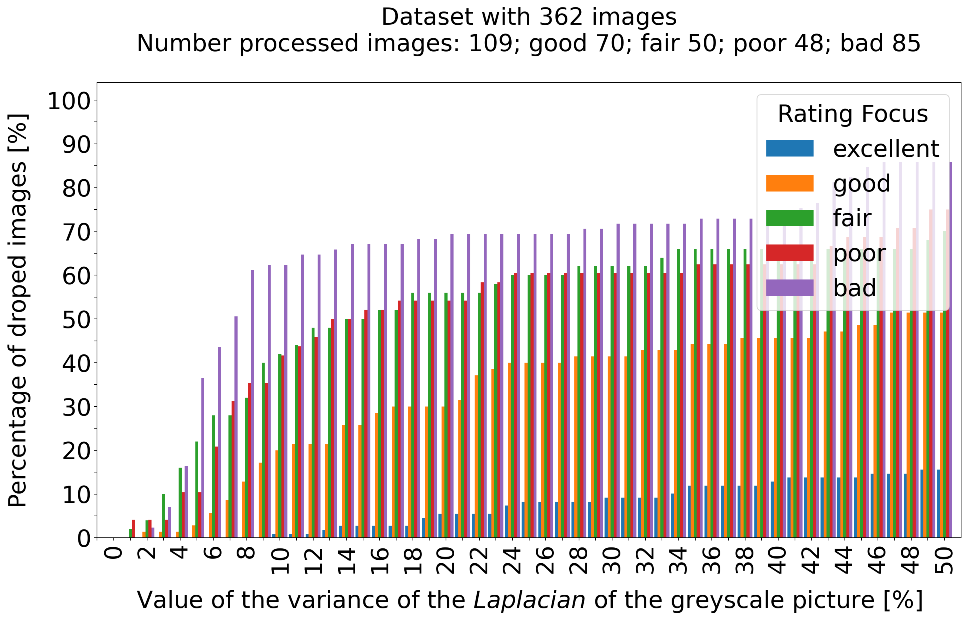

Figure 6.10: Plot of the droprate for images with a lower VAR_LAP of the image than the threshold.

VAR_SOB_{XY} threshold, the drop rate of bad images is lower while the rate for excellent images is staying over 5% until 23. Furthermore, the drop rate of good images is comparable high. Therefore it is necessary to choose a threshold below 10, to keep the drop rate of good image below 20% and the one for excellent below 5%.

6.4 Combination

Conclusively, after thresholds for the exposure and focus filter are found, a filter combining both needs to be set up.
The filter for exposure is the combination of average exposure and percentage of dark and bright areas. Only images with average exposure between 75 and 170 and with a percentage of dark and bright areas lower than 0.12 are kept.
There are three filter for blurry images. Each of them is combined with the exposure filter. Thus there are three different filter in the end.
The filter with the $Laplacian$ drops all images with a VAR_LAP below 10. The $Sobel$ filter images with a VAR_SOB_{XY} below 260 and the $Scharr$ filter below 4200 for VAR_SCHARR_{XY}.

7 Performance

When the algorithm and thresholds are found, it is necessary to understand how the results are performing and to compare them with other similar tools. For a better comparison the decision for images to keep and drop is brought to a binary classification. With this definition at first the performance on the Test Dataset and Verification Dataset is measured, before it is compared to two other tools.

7.1 Binary Classification for Performance Evaluation

These filter are not separating between the five categories 5.1. They are deciding by binary classification if the image should be kept or dropped. Therefore, also the images from the datasets need to be transformed into binary classification.
Images are counted as

$$Image_{GOOD} = (e_e \wedge (e_f \vee g_f)) \vee (g_e \wedge (e_f \vee g_f)) \tag{7.1}$$

which means, that images with the ratings excellent exposure (e_e), good exposure (g_e), excellent focus (e_f) or good focus (g_f) are counted as worth to keep.
The complementary are

$$Image_{BAD} = \neg Image_{GOOD} \tag{7.2}$$

bad images.

Coefficients for Performance Evaluation As there is a binary classification existing now, it is possible to use well known coefficients. Here will be used $Recall$, $Precision$, $BalancedAccuracy$ and the F_2Score.
The $BalancedAccuracy$ is used, because in the Test Dataset are twice more bad images than good ones.
Already during determining the threshold, it was prioritized to keep all good and excellent images and the number of images mistakenly kept had second priority. This means the $Recall$, which is the percentage of detected good images from all good images, is more important. Therefore the F_2Score is used.

7.2 Performance on Test Dataset

The tables Ap.T1 to Ap.T3 show the performance of the filter algorithms on the Test Dataset with the above mentioned thresholds in confusion matrices.

	Scharr	Sobel	Sobel wC	Lapl.	Lapl. wC
$Precision$	0.7236	0.7236	0.6242	0.5732	0.5749
$Recall$	0.8396	0.8396	0.8774	0.8868	0.9057
$BalancedAccuracy$	0.8534	0.8534	0.8293	0.8067	0.8142
F_2score	0.7854	0.7854	0.8116	0.818	0.8308

Table 7.1: Performance parameters for all three algorithmon the Test Dataset. $Sobel$ and $Laplacian$ also with Containtering (wC)

Comparing the tables Ap.T1 and Ap.T2 no differences can be seen. This allows the conlcusion, that for this application, the $Sobel$ and $Scharr$ filter have the same performance. Therefore only the $Sobel$ filter will be discussed, even though further results and interpretations will also count for the $Scharr$ filter.

Issues In the table Ap.T1 with the performance of the $Sobel$ filter can be seen, that 13 images which should be kept, are dropped. After inspecting these images, one issue can be, that one dimension of $Sobel$ is over the threshold and the other under. But because both are counted equal, these images are dropped. There are images which have mostly edges in one direction. An extreme example is figure 3.1 with the results for the two dimensions in figures 4.6 and 4.6. A possibility to improve this can be adding another loop before deciding. Here it is defined as $Containering$[1] and means, that if one value of the two dimensions is within the threshold, but the other is not, this dimension is checked again, but with a bit lower threshold. Here it is set to 90% of the original threshold.

A similar issue is popping up, where the exposure is good, but he focus not so good, but still fine. And the same the other way around. For this the same approach as above, called $Containering$[1], can be used. Thus the thresholds for exposure are set adjusted to 60 and 185 for average exposure and 0.2 for the percentage of dark and bright areas, when the focus is within the thresholds. And the focus thresholds are reduced to 9 for the VAR_LAP and to 234 for the VAR_SOB_{XY}, when the exposure is good.
The results with $Containering$ can be seen in table 7.1. While the performance of the $Sobel$ filter lost $Precision$ and $Accuracy$ and only improved in the $Recall$, the $Laplacian$ improved in all classifications.

[1]Containering: The name for this method is inspired by the term containering for collecting still good food out of trash containers from supermarkets. This is done for reducing senseless food waste. .

The attempt with $Containering$ was partly successfully, hence it will be also used with the Verification Dataset.

7.3 Performance on Verification Datasets

The tables Ap.T6 to Ap.T9 show the performance of the filter algorithms on the Verification Dataset with the thresholds determined with the Test Dataset in confusion matrices.
Here are also the results for the algorithm without $Containering$ to see the affect, and if it really makes a difference.

	Sobel	Sobel wC	Lapl.	Lapl. wC
$Precision$	0.5714	0.5926	0.5652	0.5862
$Recall$	0.5714	0.7619	0.619	0.8095
$BalancedAccuracy$	0.5489	0.5915	0.5464	0.589
F_2score	0.5682	0.7092	0.6052	0.7417

Table 7.2: Performance parameters for all three algorithm on the Verification Dataset. $Sobel$ and $Laplacian$ also with Containtering (wC)

Assessment Performance on Verification Dataset The performance on the Verification Dataset in table 7.2 could not reach the performance on the Test Dataset in table 7.1. But this is expectable, as the thresholds have been determined on the Test Dataset.
However the performance, especially with $Containering$ is promising. With $Laplacian$ and $Containering$ only 4 images are mistakenly dropped, while 7 out of 19 images are detected as bad images. Investigating the mistakenly dropped images it can be seen, that only images with the category *night* and *macro* are dropped. Also from the mistakenly kept images 3 are from *night*.

Comparison Sobel and Laplacian

In table 7.2 the performance of the algorithms using $Sobel$ or $Laplacian$ can be compared. Even thought it looks here, like $Sobel$ has a higher $Precision$ while $Laplacian$ reaches a higher $Recall$, it is inaccurate to use this to compare both operators. The different results here are very likely depending on the different and not directly comparable thresholds. Furthermore, these algorithms here are also combined with the exposure filter, which can add unpredictable bias. Therefore the comparison here could only be made between $Sobel$ and $Laplacian$ linked to the here chosen thresholds. A proper comparison could only be done by having comparable thresholds and preferable without the exposure filter. For example if both have

the same $Precision$ the $Recall$ could be compared. A visual comparison can also be made between fig. 6.9 and fig. 6.10.
With the here used thresholds the algorithm with the $Laplacian$ filter and $Containering$ seems the most promising.

7.4 Performance compared to other Tools

To understand the performance of the developed algorithm better, there will be a comparison with two tools found in the internet. The first one is a basic version of the here developed and the second one is an Application for iOS devices.
Unfortunately both tools are only focusing on the detection of blurry images.

Pyimagesearch

On the webpage and blog *pyimagesearch.com* is an article about *"Blur detection with OpenCV"*[12]. There is a description about detecting blurry images with the python library OpenCV. It works just with the VAR_LAP and suggests a threshold of 100.

All 40 from the Verification Dataset have been processed with the threshold of 100 for the VAR_LAP. The result can be seen in table 7.3. This results in a $Precision$ of 0.5556, a $Recall$ of 0.7143, $BalancedAccuracy$ of 0.5414 and a F_2Score of 0.6757.

	Condition Positive	Condition Negative
Prediction Positive	15	12
Prediction Negative	6	7

Table 7.3: Confusion Matrix of the result from "pyimagesearch.com" on the Verification Dataset

Gemini Photos: Gallery Cleaner

Gemini Photos: Gallery Cleaner[4] is an application for iOS devices from the company *MacPaw Inc.*
The app advertises with "SELECTS THE NOT-SO-GOOD ONES" and further "blurred ones". This future is part of the free version.

All 40 from the Verification Dataset have been processed by the app. The result can be seen in table 7.4. This results in a $Precision$ of 0.5384, a $Recall$ of 1, $BalancedAccuracy$ of 0.5263 and a F_2Score of 0.8536.

	Condition Positive	Condition Negative
Prediction Positive	21	18
Prediction Negative	0	1

Table 7.4: Confusion Matrix of the result from the Gemini App on the Verification Dataset

Assessment Performance other Tools The results of *pyimagesearch.com* are comparable with the results of $Sobel$ and $Laplacian$. Even though it is only meant to detect the blurry images, it has good results.
Except of $Sobel$ without $Containering$, in total the in this paper developed thresholds have better results. At first sight this is different for the app *Gimini*. The results of the classification coefficients are promising. Looking into table 7.4 it can be seen, that *Gemini* did not loose a single good image, though the number of detected bad images is 1 out of 19. In sum this tool did perform well on the first glance, but actually did not do anything.

8 Conclusion

Before reaching the summary and giving an outlook the development of the algorithm will be explained in a nutshell.

Approach For understanding the characteristics of images which are affecting the quality an overview was given. The procedure of creating a digital image helped to find possible settings of the motive and camera, which can result in bad images.
Furthermore, the explanation of the factors in the digital images, which are lowering the quality, allowed to get a deeper understanding about the issues in images. This helped to find tendencies to search for and analyse.
These observations have been interpreted mathematically and the images processed with several operators and calculations. This resulted in values for detecting over- and underexposure, very dark and bright areas in images, and the number and their character of edges, which are a indicator for the focus.

Datasets To determine and evaluate the algorithm it is necessary to have two datasets. To get those datasets a webapp was developed, which allowed to label and rate images in the webbrowser. There was one dataset with 362 images for developing the algorithm and finding thresholds. This was labeled and rated by a single person. The dataset for verifying the developed threshold is the result of 11 test persons labeling and rating the images.

Algorithms The thresholds for the different decisions for exposure and focus respectively blurriness have been determined by observation of the drop rates for each rating level in a range of thresholds. This helped to make granular decisions considering the requirement of higher importance to keep good images, while the detection of bad images had second priority. To improve the interference between exposure and blurriness in images, there was developed a mechanism named $Containering$. If either exposure or focus is good, the other category is again evaluated with a less stringent threshold.

Performance This improved the performance, albeit not as much as expected. The performance was evaluated with the verification dataset and also compared with two other tools. It outperformed iOS tool Gemini Photos and was slightly better than just the Laplacianfilter with the threshold from Pyimagesearch.

A deeper investigation of the results on the verification dataset revealed, that the here developed algorithm has problems with images of the categories *night* and *macro*.

Summary The developed algorithm with its thresholds can support reducing numbers of images and detects around 40% of the bad images of all considered categories. Excluding images from the categories *night* and *macro* will improve this result and also reduces the mistakenly lost good images.
With a detection rate of 40% from the bad images, the algorithm will not safe time on small stocks of photos. It becomes more relevant when a huge number of images needs to be processed. As the motivation was to support creating a set of memories out of a bigger stock of images, it is meeting the purpose with on a useful level, especially when the weaknesses are considered.
Even though the algorithm can automate detecting bad images concerning exposure and focus reasonable, the survey for creating the verification dataset showed, that for several images nearly all opinions are consistent. But for several images the distribution of the ratings showed, that there is a significant level of subjectiveness in the rating. Therefore the thresholds determined in this thesis is not overall objective and probably fits the best to the person who created the Test Dataset.

Outlook In this thesis the approach and algorithm with its thresholds was explained and developed. The results are snippets of codes for testing and evaluating the algorithm and the thresholds, but is not comfortable to use productive on folders with pictures. Hence, the next step would be to use one of the in this thesis developed algorithms, and build a piece of software around, which is convenient to use productive and maybe even add new features.

Bibliography

1. S. Chandran. In Color Image to Grayscale Image Conversion, pp. 196 – 199, apr 2010. URL https://www.researchgate.net/publication/224130500_Color_Image_to_Grayscale_Image_Conversion.

2. Duden Learnattack GmbH. Fotografie und ihre geschichtliche Entwicklung. 2010. URL https://www.lernhelfer.de/schuelerlexikon/kunst/artikel/fotografie-und-ihre-geschichtliche-entwicklung. Last Accessed: 2020-09-06.

3. D. van Hall. Die Geschichte der Digitalkamera und der digitalen Bildaufzeichnung. 2015. URL https://www.digitalkameramuseum.de/de/geschichte. Last Accessed: 2020-09-06.

4. M. Inc. *Gemini Photos: Gallery Cleaner*, oct 2020. URL https://apps.apple.com/app/gemini-photos-gallery-cleaner/id1277110040?ref=producthunt&pt=118808&mt=8&ign-mpt=uo%3D4.

5. Jain, Sindhu and Goswami, Sudhir. A comparative study of various image restoration techniques with different types of blur. 11 2015.

6. D.J. Kroon. In Numerical optimization of kernel based image derivatives, dec 2009. URL http://www.k-zone.nl/Kroon_DerivativePaper.pdf.

7. C. Liu. Worldwide Internet and Mobile Users. In *EMartketer's updated estimates for 2015*, p. 11, aug 2015. URL https://insights.ap.org/uploads/images/eMarketer_Estimates_2015.pdf.

8. N.N. *OpenCV Docu - Color conversions*, N/A. URL https://docs.opencv.org/master/de/d25/imgproc_color_conversions.html#color_convert_rgb_gray. Last Accessed: 2020-09-06.

9. J.L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia. In Diatom autofocusing in brightfield microscopy: a comparative study, 3, pp. 314–317 vol.3, 2000. URL https://ieeexplore.ieee.org/abstract/document/903548.

10. S. Pertuz, D. Puig, and M.Á. García. Analysis of focus measure operators for shape-from-focus. In *Pattern Recognition*, 46, pp. 1415–1432, 2013. URL

https://www.semanticscholar.org/paper/Analysis-of-focus-measure-operators-for-Pertuz-Puig/8c675bf5b542b98bf81dcf70bd869ab52ab8aae9.

11. S. Romaniello. *Photoshop CS3 Channels and Masks Bible*. Bible (Wiley). Wiley, 2007. URL https://books.google.de/books?id=hD2j1J8HvD0C.

12. A. Rosebrock. In Blur detection with OpenCV, 2015. URL https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/. Last Accessed: 2020-08-15.

13. H. Scharr. *Optimale Operatoren in der Digitalen Bildverarbeitung*. Ph.D. thesis, University of Heidelberg, apr 2000. URL http://archiv.ub.uni-heidelberg.de/volltextserver/962/1/Diss.pdf.

14. R. Streijl, S. Winkler, and D. Hands. Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives. In *Multimedia Systems, Springer*, 2014. URL https://link.springer.com/article/10.1007/s00530-014-0446-1.

15. R. Strickland. *Image-Processing Techniques for Tumor Detection*. CRC Press, 2002. URL https://books.google.de/books?id=HiGMZDT1IXYC.

16. S. Sural, Gang Qian, and S. Pramanik. In Segmentation and histogram generation using the HSV color space for image retrieval, 2, 2002.

17. Valladas, H., Clottes, J., Geneste, J. Evolution of prehistoric cave art. In *Nature*, apr 2001. URL https://www.nature.com/articles/35097160?free=2.

18. S. Ventura. *DSLR Photography for Beginners: Take better photos, make better memories.* Salvatore Ventura, 2019. URL https://books.google.de/books?id=5BqzDwAAQBAJ.

Appendix

Figures

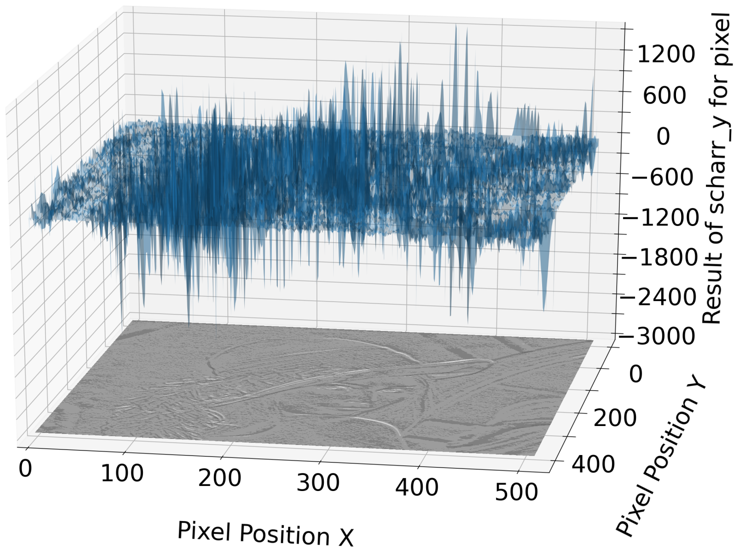

Figure Ap.F1: 3D plot of Lenna (Fig. 4.1) in greyscale after the $Scharr_Y$ filter was applied. On the surface Lenna can be seen in grayscale. It is the 2D projection of the plot.

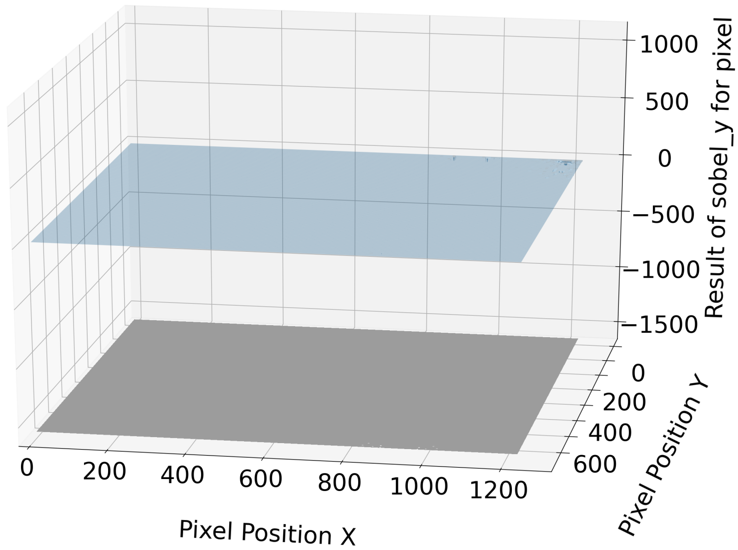

Figure Ap.F2: 3D of (Fig. 3.1) after the Sobel Y filter was applied. On the surface the edges can be seen.

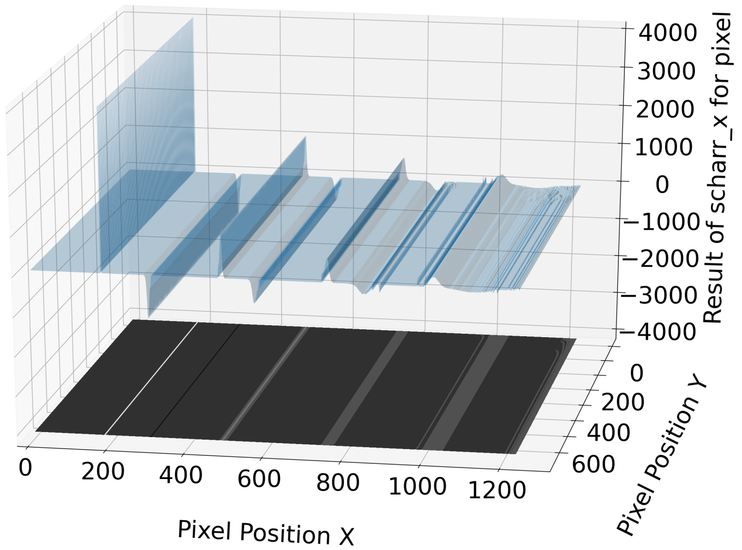

Figure Ap.F3: 3D of (Fig. 3.1) after the $Scharr_X$ filter was applied. On the surface the edges can be seen.

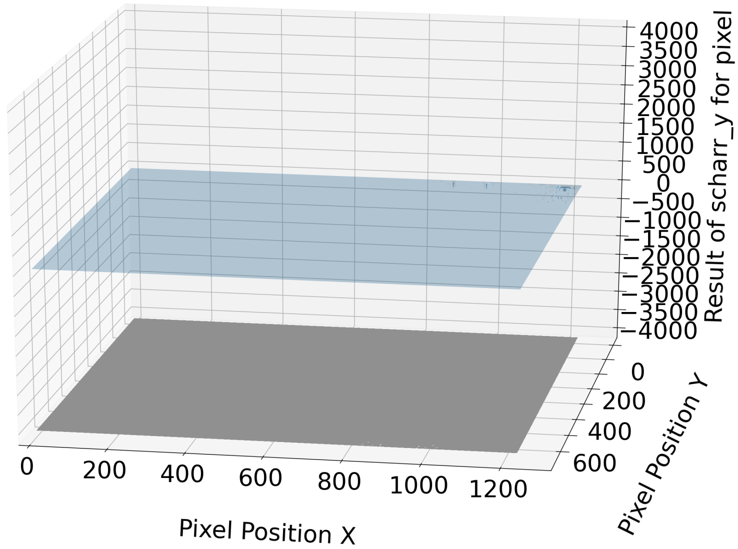

Figure Ap.F4: 3D of (Fig. 3.1) after the $Scharr_Y$ filter was applied. On the surface the edges can be seen.

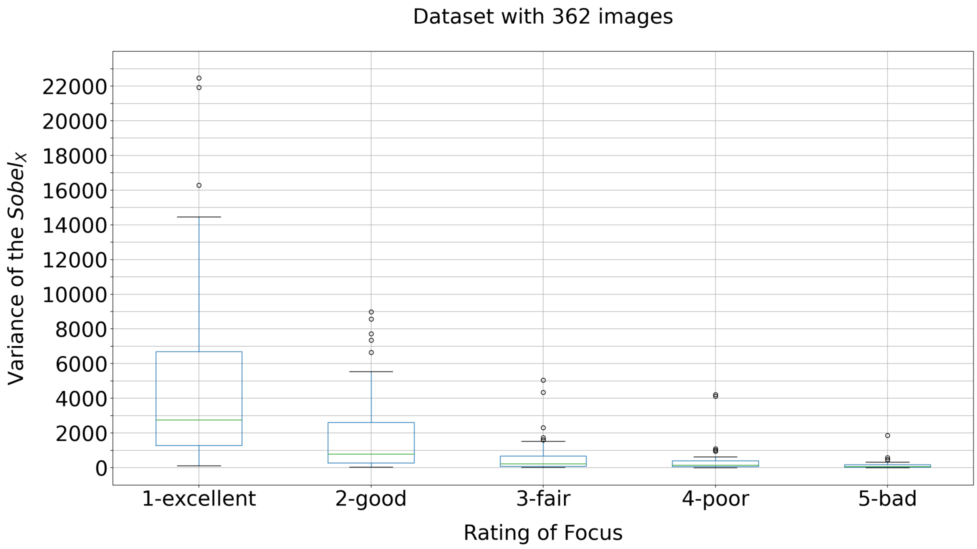

Figure Ap.F5: Boxplot of the variance of the $Sobel_X$ from the images from Test Dataset (chapter 5.3).

Figure Ap.F6: Boxplot of the variance of the $Sobel_Y$ from the images from Test Dataset (chapter 5.3).

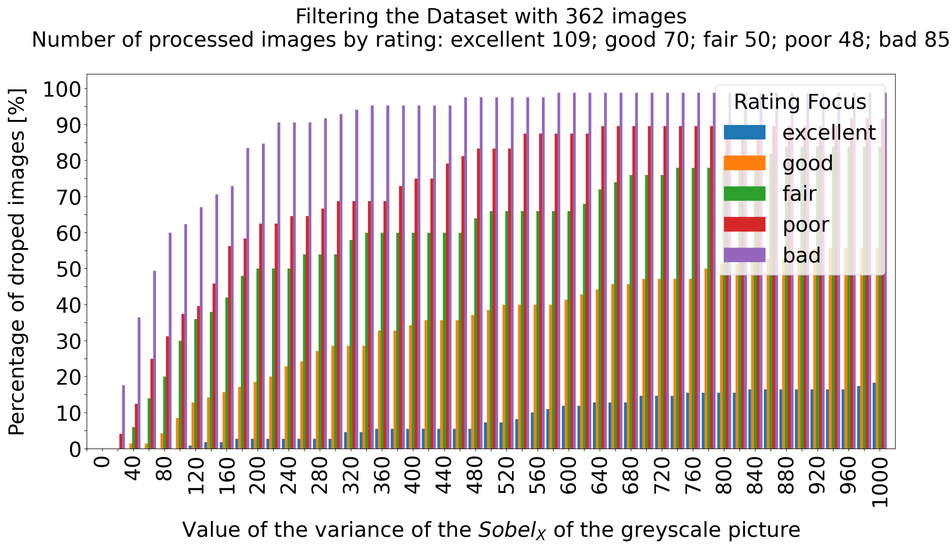

Figure Ap.F7: Plot of the droprate for images with a lower variance of $Sobel_X$ of the image than the threshold.

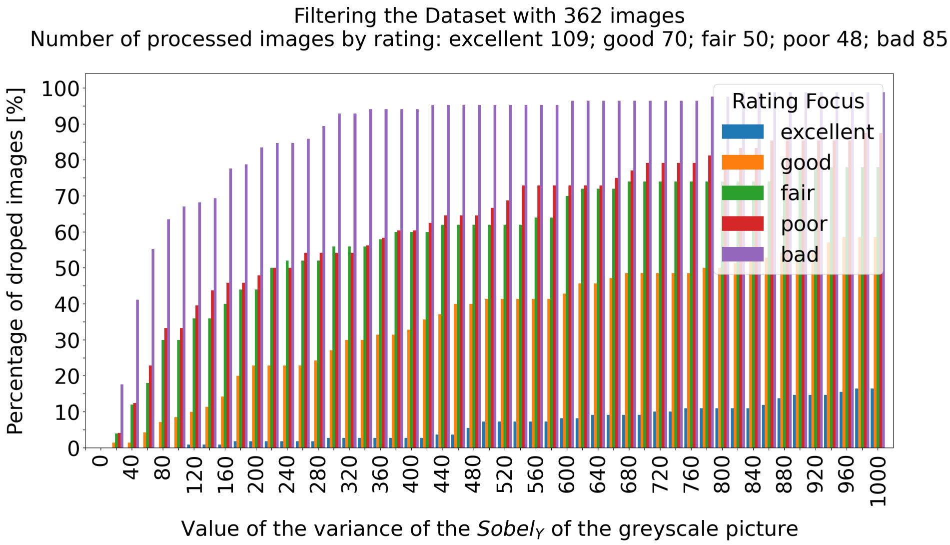

Figure Ap.F8: Plot of the droprate for images with a lower variance of $Sobel_Y$ of the image than the threshold.

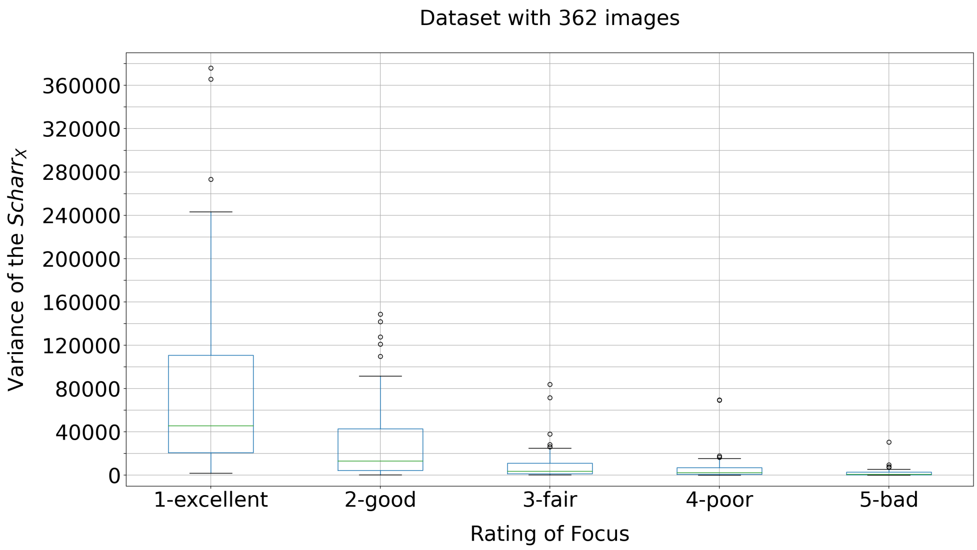

Figure Ap.F9: Boxplot of the variance of the $Scharr_X$ from the images from Test Dataset (chapter 5.3).

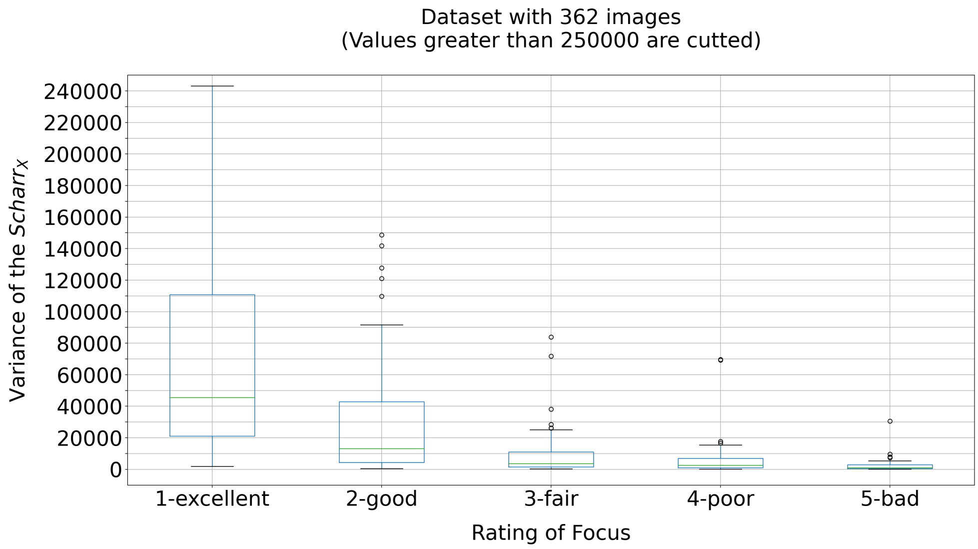

Figure Ap.F10: Boxplot of the variance of the $Scharr_X$ from the images from Test Dataset (chapter 5.3). Datapoints with values above 250000 have been cutted, to see the area with lower values better. Original Boxplot can be found in appendix Ap.F9.

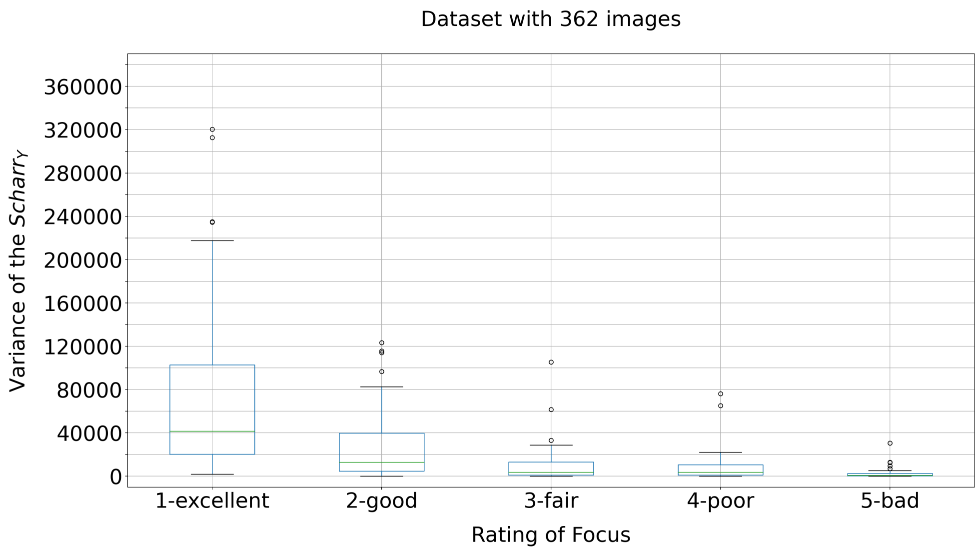

Figure Ap.F11: Boxplot of the variance of the $Scharr_Y$ from the images from Test Dataset (chapter 5.3).

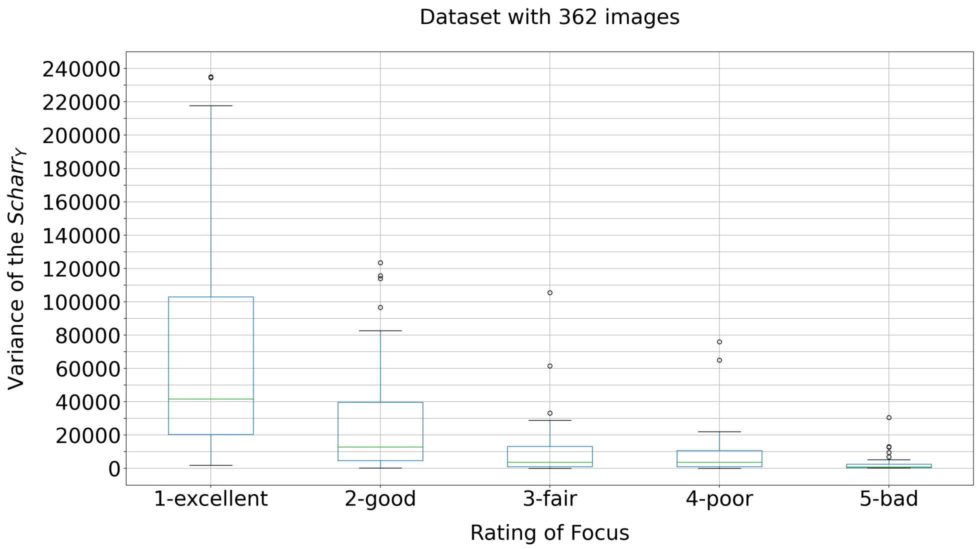

Figure Ap.F12: Boxplot of the variance of the $Scharr_Y$ from the images from Test Dataset (chapter 5.3). Datapoints with values above 250000 have been cutted, to see the are with lower values better Original Boxplot can be found in appendix Ap.F11.

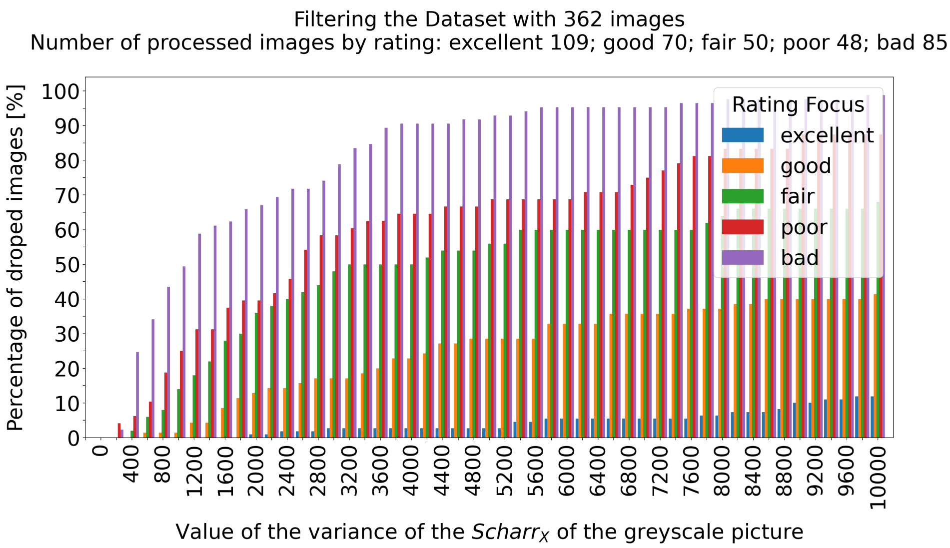

Figure Ap.F13: Plot of the droprate for images with a lower variance of $Scharr_X$ of the image than the threshold.

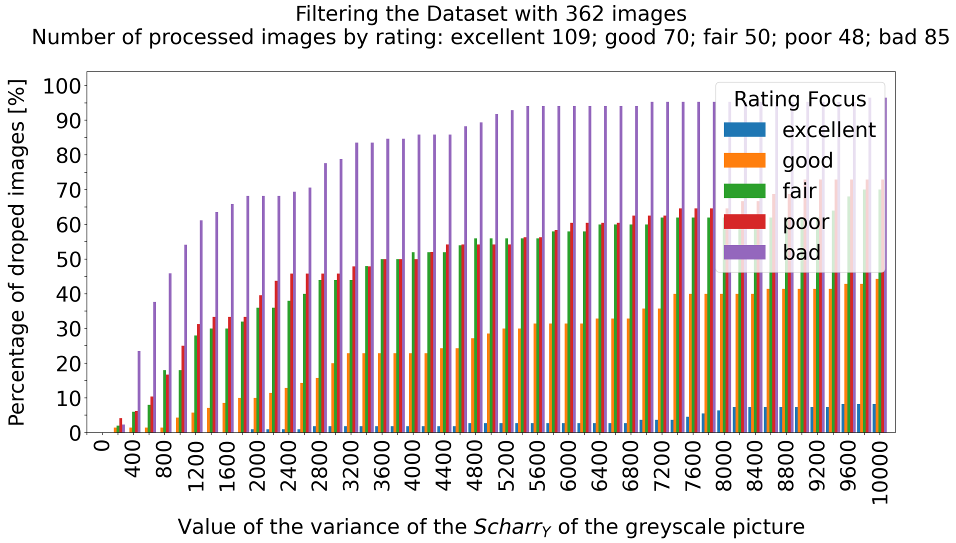

Figure Ap.F14: Plot of the droprate for images with a lower variance of $Scharr_Y$ of the image than the threshold.

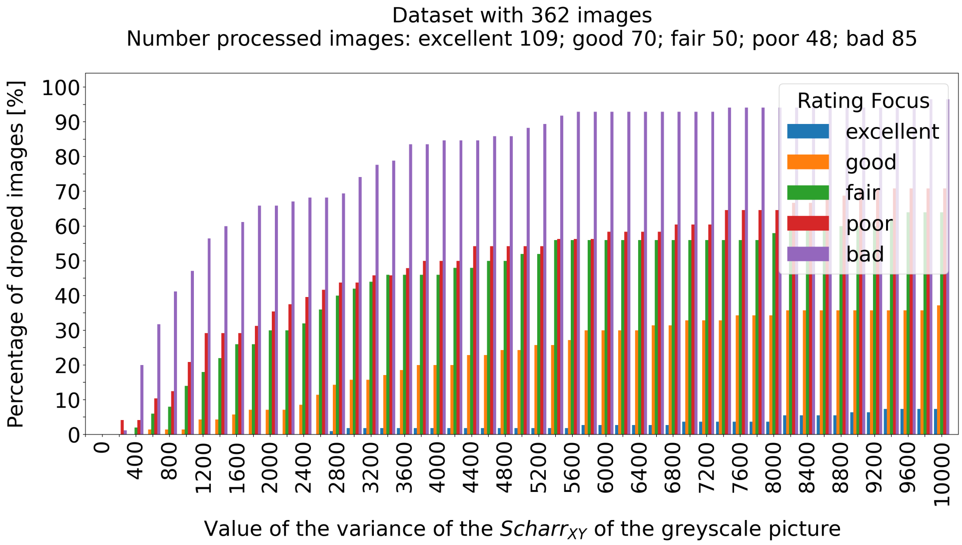

Figure Ap.F15: Plot of the droprate for images with a lower variance of $Scharr_{XY}$ of the image than the thresholds.

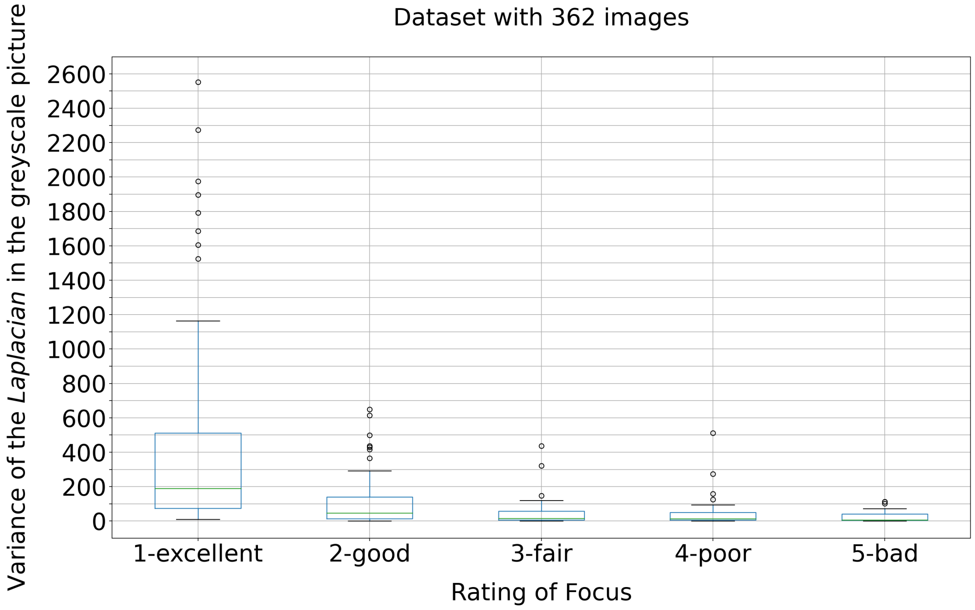

Figure Ap.F16: Boxplot of the variance of the $Laplacian$ from the images from Test Dataset (chapter 5.3).

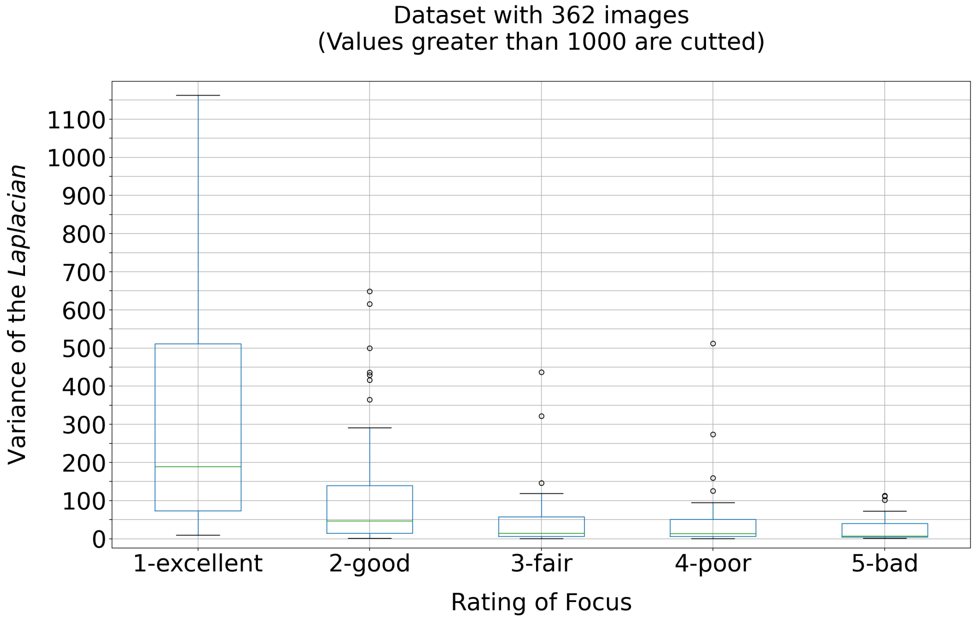

Figure Ap.F17: Boxplot of the variance of the $Laplacian$ from the images from Test Dataset (chapter 5.3). Datapoints with values above 1000 have been cutted, to see the are with lower values better. Original Boxplot can be found in appendix Ap.F16.

Tables

	Condition Positive	Condition Negative
Prediction Positive	89	34
Prediction Negative	17	222

Table Ap.T1: Confusion Matrix of the combined Filter with $Sobel$ on the Test Dataset

	Condition Positive	Condition Negative
Prediction Positive	89	34
Prediction Negative	17	222

Table Ap.T2: Confusion Matrix of the combined Filter with $Scharr$ on the Test Dataset

	Condition Positive	Condition Negative
Prediction Positive	94	70
Prediction Negative	12	186

Table Ap.T3: Confusion Matrix of the combined Filter with $Laplacian$ on the Test Dataset

	Condition Positive	Condition Negative
Prediction Positive	93	56
Prediction Negative	13	200

Table Ap.T4: Confusion Matrix of the combined Filter with $Sobel$ on the Test Dataset with containering enabled.

	Condition Positive	Condition Negative
Prediction Positive	96	71
Prediction Negative	10	185

Table Ap.T5: Confusion Matrix of the combined Filter with $Laplacian$ on the Test Dataset with containering enabled.

	Condition Positive	Condition Negative
Prediction Positive	12	9
Prediction Negative	9	10

Table Ap.T6: Confusion Matrix of the combined Filter with $Sobel$ on the Verification Dataset

	Condition Positive	Condition Negative
Prediction Positive	13	10
Prediction Negative	8	9

Table Ap.T7: Confusion Matrix of the combined Filter with $Laplacian$ on the Verification Dataset

	Condition Positive	Condition Negative
Prediction Positive	16	11
Prediction Negative	5	8

Table Ap.T8: Confusion Matrix of the combined Filter with $Sobel$ on the Verification Dataset with containering enabled.

	Condition Positive	Condition Negative
Prediction Positive	17	12
Prediction Negative	4	7

Table Ap.T9: Confusion Matrix of the combined Filter with $Laplacian$ on the Verification Dataset with containering enabled.