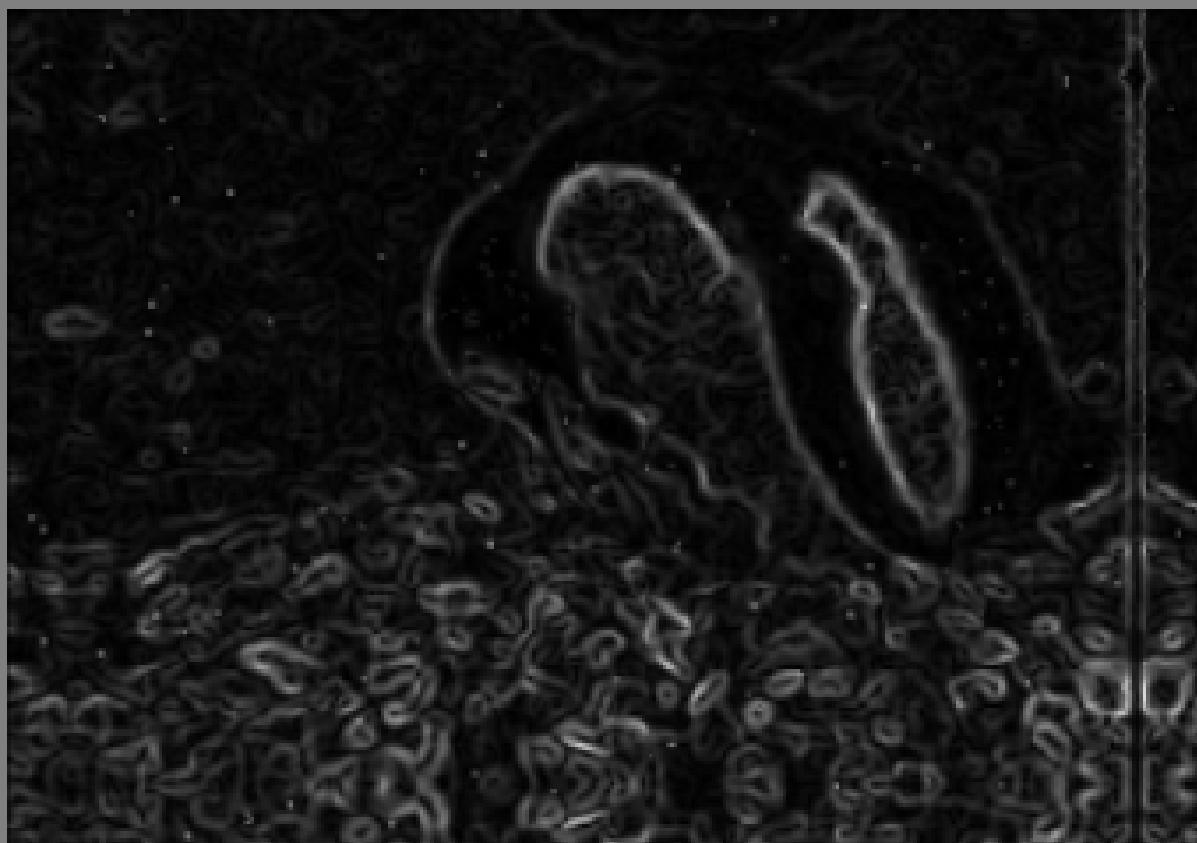


# Detection of Geometric Primitives in SAR Images



Mees J. Meester

Layout: typeset by the author using L<sup>A</sup>T<sub>E</sub>X.

Cover illustration: Sobel edge detection on image, indirectly taken from the Berkeley Segmentation Data Set and Benchmarks 500

# Detection of Geometric Primitives in SAR Images

In cooperation with OPT/NET B.V.

Mees J. Meester  
12217255

Bachelor thesis  
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*



University of Amsterdam  
Faculty of Science  
Science Park 904  
1098 XH Amsterdam

*Supervisor*  
MSc. Anil S. Baslamisli

Informatics Institute  
Faculty of Science  
University of Amsterdam  
Science Park 904  
1098 XH Amsterdam

June 25, 2021

# Contents

<b>1 Abstract</b>	<b>3</b>
<b>2 Introduction</b>	<b>4</b>
2.1 Geometric Primitives . . . . .	5
2.2 Synthetic Aperture Radar (SAR) . . . . .	6
2.2.1 Edge Detection for SAR Images . . . . .	7
<b>3 Related Work</b>	<b>9</b>
3.1 Optical and SAR Remote Sensing . . . . .	9
3.2 Noise . . . . .	11
3.2.1 Gaussian Smoothing . . . . .	13
3.2.2 Block-Matching and 3D Filtering (BM3D) . . . . .	13
3.2.3 Bilateral Filtering . . . . .	13
3.2.4 Anisotropic Diffusion . . . . .	14
3.2.5 Non-Local Means Denoising (NLMD) . . . . .	14
3.2.6 SAR Specific Denoising . . . . .	14
3.3 Geometric Primitives . . . . .	14
3.3.1 Edge and Boundary Detection . . . . .	15
3.3.2 Additional Features and Primitives . . . . .	19
<b>4 Approach</b>	<b>20</b>
4.1 Dataset . . . . .	21
4.2 Denoising . . . . .	21
4.3 Edge Detection . . . . .	22
4.3.1 Canny, LoG and GR . . . . .	22
4.3.2 Wavelet Transformation and Gabor Filtering . . . . .	23
4.3.3 K-means Clustering . . . . .	23
4.4 Fusion Methods . . . . .	24
4.5 Implementation Details . . . . .	24
4.5.1 BSDS500-speckled . . . . .	24
4.5.2 Denoising . . . . .	24

4.5.3	Edge Detection . . . . .	25
4.5.4	Fusion . . . . .	25
<b>5</b>	<b>Evaluation</b>	<b>27</b>
5.1	Denoising . . . . .	27
5.2	Edge Detection . . . . .	28
5.2.1	Low Level Methods . . . . .	29
5.2.2	LoG . . . . .	29
5.2.3	Canny . . . . .	29
5.2.4	K-means . . . . .	30
5.2.5	Gabor Filters . . . . .	31
5.2.6	GR . . . . .	31
5.2.7	Fusion . . . . .	32
5.3	BSDS500-speckled Benchmark . . . . .	32
5.4	Real SAR image . . . . .	33
5.4.1	Odessa, United States . . . . .	33
<b>6</b>	<b>Conclusion and Future Work</b>	<b>36</b>
6.1	Challenges . . . . .	37
<b>References</b>		<b>38</b>
<b>Appendices</b>		<b>48</b>
<b>A</b>	<b>Training - Qualitative Evaluation</b>	<b>49</b>
<b>B</b>	<b>Training - ROC curves</b>	<b>61</b>
<b>C</b>	<b>Training - Quantitative Evaluation</b>	<b>67</b>

# Chapter 1

## Abstract

This paper aims to evaluate the performance of edge detectors on remote-sensing synthetic-aperture radar (SAR) satellite images. The number of ground truth annotations for SAR images is scares. Therefore we evaluate on a large dataset of simulated SAR images, BSDS500-speckled. We evaluate how it relates to a real SAR image for evaluation edge detectors. With BSDS500-speckled, we evaluate a wide variate of edge detectors and tune their parameters for increased performance. Combinations of edge detectors made to further improve performance. We bundle the findings and BSDS500-speckled into a quantitative benchmark, which is missing in the field. Accordingly, this thesis aims to answer the following research questions: 1) How can we provide a comprehensive and fair experimental evaluation for SAR images on edge detection? 2) How do the traditional edge detection methods perform on SAR images? 3) What is the effect of fusion methods on the performance?

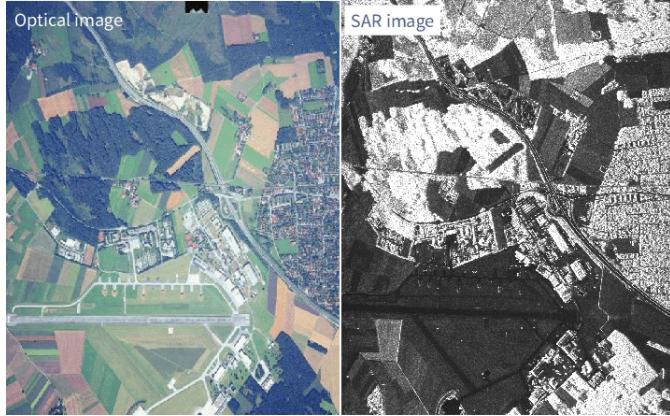
# Chapter 2

## Introduction

Remote sensing satellite images capture the earth's surface. Therefore, they enable surveillance, analysis of infrastructure agriculture, and natural disaster management (Sharma, Kumar, Desai, & Gujrati, 2008). Optical and synthetic-aperture radar (SAR) equipment obtain the two primary forms of satellite data, as illustrated in Figure 2.1. Both forms measure different wavelengths, which results in different advantages and disadvantages.

Optical remote sensing measures photometric quantities such that the wavelengths are within the spectrum of visible light and weighted to mimic the human vision. On the other hand, SAR technology utilizes longer wavelengths (Herndon, Meyer, Flores, Cherrington, & Kucera, 2020). The main advantage of the (longer) SAR bands is that these microwave signals can penetrate the weather, making it possible to capture areas under clouds. On the other hand, the optical signals are limited by the cloud coverage due to their small wavelengths. Hence, it is not feasible to monitor a region covered by (heavy) clouds by optical imaging. Moreover, optical imaging depends on the (external) light from the sun for illumination. Thus, regions should be properly illuminated to achieve a decent optical image. When regions do not receive sufficient illumination due to atmospheric and photometric effects, (near) real-time surveillance is not possible. On the other hand, SAR itself actively sends and retrieves signals to the earth's surface. These signals bounce differently from the surfaces depending on the wavelengths and material properties; some are absorbed by the surface and others may bounce multiple times. Thus, the surfaces are illuminated by the differences in how the waves are reflected back. Therefore, SAR is able to analyse the earth's surface in areas any time of the day, even when it is dark, making it ideal for surveillance.

Figure 2.1: Optical and SAR images. Credits: Ley et al. (2018)



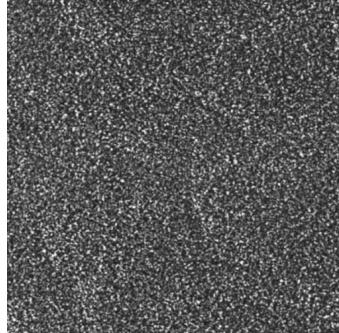
## 2.1 Geometric Primitives

To manually realize surveillance and natural disaster management is infeasible because human resources are expensive, and algorithms are much faster in detecting patterns than humans. For example, for the case of flood detection in Houston, Texas region after Hurricane Harvey in 2017, a team of analysts spent almost a week to generate a map covering an area of  $\sim 1$  kilometre<sup>2</sup>, whereas OPT/NET’s AI engine took  $\sim 15$  minutes to cover an area of  $\sim 68$  kilometres<sup>2</sup> with extremely high accuracy (Rao, 2021).

It is possible to automate the monitoring tasks by detecting distinct geometric features like lines, edges and blobs. Automated recognition of the essential features of images is achieved by artificial intelligence (AI), specifically its specialisation in computer vision. In this thesis, the emphasis is on the edge detection task for SAR images using computer vision techniques. Since edges form the most fundamental features of the geometric primitives, they can later be utilized to identify other shapes such as lines, corners and junctions. Moreover, these features are often used as salient image regions to be used for pre-segmentation for object detection and recognition in remote sensing image processing. For example, roads can be detected by identifying lines (Chen, Zhu, Xie, Hu, & Zeng, 2018), or blob-like structures might give clues on silos at an oil refinery.

An edge manifests itself by an abrupt change in pixel intensity values, often identified by a significant shift in first or second derivative. Mostly, a convolutional filter (kernel) approximates the gradients or second derivatives of an image. Applying the kernel yields edge responses and a threshold decides which changes are considered as true edges. The most common edge filters are the Sobel, Roberts Cross, and Prewitt operators (Duda & Hart, 1973; Roberts, 1965; Prewitt, 1970). These detectors are at the bottom of the hierarchy, considered as low level and

Figure 2.2: Speckle noise on a homogeneous patch. Credits: Simard et al. (1998)



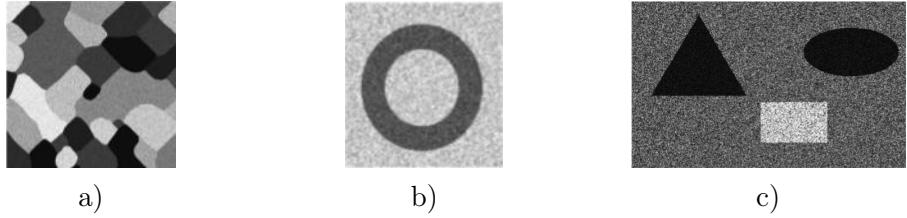
straightforward without any parameters to tune. More advanced ones such as the Laplacian of Gaussian or Canny, include a smoothing operation to prevent detecting noise as false edges (Marr & Hildreth, 1980; Canny, 1986). Obviously, each edge detector has both advantages and disadvantages. For example, the smoothing process may help eliminate the false edges, but it may also prevent the ability to detect micro edges. In that sense, decision fusion methods can be utilized to combine the (outputs of the) multiple detectors and can average out the different detectors' errors. On the other hand, recent state-of-the-art edge detectors use data-driven supervised convolutional neural networks (CNN) to *learn* specialized kernels (Shen, Wang, Wang, Bai, & Zhang, 2015; Bertasius, Shi, & Torresani, 2015; Xie & Tu, 2015; Y. Liu et al., 2019).

## 2.2 Synthetic Aperture Radar (SAR)

More than 40 years ago NASA sent the first civilian SAR-satellite into space (Moreira et al., 2013). Since then, SAR technology has been broadly used for capturing the earth's surface. Space-borne SAR systems TerraSAR-X, COSMO SkyMed, HRWS, Radarsat-2 and Capella-2 reach a spatial resolution of 1 metre or higher (Braun & Hochschild, 2017; Nava, Catani, & Monserrat, 2021). The Sentinel-1 SAR satellite covers areas up till 6400 kilometres<sup>2</sup>. Thanks to their high resolution, wide-area coverage, and weather and illumination invariant qualities SAR images are widely appreciated.

Nonetheless, SAR images also bring challenges. Unlike optical images, they are highly corrupted with speckle noise, see Figure 2.2. The noise characteristics make them exceedingly challenging to process. In that sense, basic edge detection models tend to produce unsatisfactory results for radar images (Touzi, Lopes, & Bousquet, 1988). The main problem is that the speckle noise patterns may be identified as false edges. Furthermore, it is not possible to entirely denoise the

Figure 2.3: Examples of simulated images. a) Q.-R. Wei and Feng (2015); Xiang et al. (2017b); Schou et al. (2003) b) Zhan et al. (2013) c) Xiang et al. (2017b)



speckled images (Singh & Shree, 2016). Thus, the task is an active area of research. Mostly, the problem is approached by edge-preserving denoising algorithms (Salehi, Vahidi, Abdeljawad, Khan, & Rad, 2020; D'Hondt, Guillaso, & Hellwich, 2013; G. Liu, Yang, Xia, & Liao, 2013; Chierchia, El Gheche, Scarpa, & Verdoliva, 2017). CNNs also demonstrate promising results for both denoising and also for edge detecting (Cozzolino, Verdoliva, Scarpa, & Poggi, 2020; Lattari et al., 2019; Lu et al., 2021; C. Liu, Tupin, & Gousseau, 2020; El-Sayed & Sennari, 2014).

### 2.2.1 Edge Detection for SAR Images

A number of SAR specific edge detectors have been proposed to deal with the speckle noise characteristics. Examples are a ratio-based edge detector (RBED), a multiscale edge detector based on Gabor filters, and a constant false alarm rate (CFAR) edge detector (Q.-R. Wei & Feng, 2015; Xiang, Wang, Wan, & You, 2017a; Schou, Skriver, Nielsen, & Conradsen, 2003). Although these edge detectors can achieve robust and effective results in their own evaluations, the comparisons tend to include only a couple of (simulated) SAR images. For instance, Xiang et al. (2017a) use a simple synthetic image corrupted with the speckle noise and a real TerraSAR-X image for the evaluations. The same synthetic image is also utilized by Q.-R. Wei and Feng (2015) together with two real TerraSAR-X images. Likewise, the same synthetic image is used by Jing, Jin, and Xiang (2020) together with two real SAR images. Moreover, Zhan, You, and Fuqing (2013) utilize another simple synthetic image corrupted with the speckle noise and two real SAR images. Xiang, Wang, Wan, and You (2017b) simulate a single synthetic image with three different options and also use three real images; two images from the GF-3 satellite and one from TerraSAR-X. Similarly, Luo, An, Wang, and Huang (2020) use a single simulated synthetic image and two real Mini-SAR images. Finally, Q.-R. Wei and Feng (2018) evaluate two simulated synthetic images and a single real TerraSAR-X SAR image. Some of the synthetic images are shown in Figure 2.3.

Synthetically generated images provide ground-truth edge maps so that the authors calibrate the parameters of their algorithms using quantitative evaluations.

Then, the parameters achieving the highest performance are selected and applied to real SAR images to provide qualitative evaluations. Nonetheless, the quantitative evaluations and parameters selection are based only on a couple simple images, as provided in Figure 2.3. Therefore, the generalized performance of the proposed methods are not truly reflected as real world patterns are much more complex and diverse. The main reason behind is the lack of large-scale datasets with ground-truth edge annotations. That also emerges another problem that a quantitative benchmark is missing in the field. Thus, at the moment, it is not possible to fully evaluate a (new or existing) method. Moreover, there is simply no fair way to compare results to other research since they either do not use the same evaluation images or a small number of comparison methods. In that sense, Bachofer, Quénéhervé, Zwiener, Maerker, and Hochschild (2016) provide the only comparison of different combinations of speckle reduction techniques and edge detection methods. However, they evaluate four fundamental methods on only four images with multilooking. Therefore, a comprehensive evaluation is also missing.

In this thesis, we aim to close the aforementioned gaps. Recently, C. Liu et al. (2020) have simulated a large-scale SAR dataset, BSDS500-speckled, exploiting an optical dataset for edge detection in natural images to train their CNN for edge detection in SAR images. The dataset is generated by multiplying the grayscale intensity optical images with a 1-look speckle noise following a Nakagami distribution. Including different augmentations, the dataset includes 28800 training images and 200 test images. Therefore, instead of using a couple of simple synthetic images, we propose to use the training set of the BSDS500-speckled for parameter tuning. The detectors with the best performing parameters on the training set are eventually evaluated on the test set to form the benchmark. With this benchmark, we provide the most extensive experimental evaluation for SAR images on edge detection and thereby addressing the lack of large-scale experimental surveys in the remote sensing field. The survey also includes the performance evaluations of a number of denoising algorithms. To that end, we evaluate the following edge detectors: Canny, Farid, Gabor, Gradient by Ratio, a K-means based clustering edge detector, Laplacian of Gaussian, Prewitt, Roberts Cross, Scharr, Sobel and a 2D Wavelet discrete transformation. Additionally, we explore a bundle of decision fusion methods which aims to combine the outputs of different algorithms. We hope that our work will serve a baseline for future algorithms. Additionally, we describe how these image properties link to the detection of geometric primitives. Accordingly, this thesis aims to answer the following research questions: 1) How can we provide a comprehensive and fair experimental evaluation for SAR images on edge detection? 2) How do the traditional edge detection methods perform on SAR images? 3) What is the effect of fusion methods on the performance?

# Chapter 3

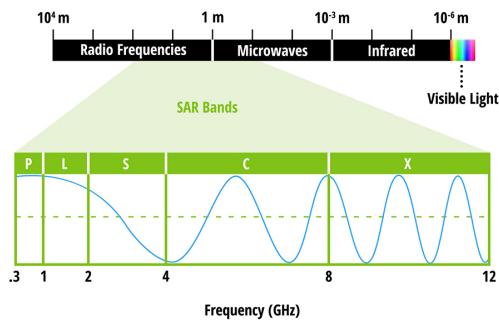
## Related Work

### 3.1 Optical and SAR Remote Sensing

In 1978, NASA sent the first civilian SAR satellite into space (Moreira et al., 2013). SAR is based on Real-Aperature Radar (RAR). With RAR, the resolution of images significantly decreases with the increasing distance from the captured surface due to the longer microwave wavelengths. For most applications, the resolution of the radar images becomes insufficient. The size of the antennas on RAR equipment has to be expanded to hundreds of meters to enable a sufficient resolution, which is not practical in space. On the other hand, SAR synthesises a substantial antenna. From a physical antenna of 10 metres in the direction of flight, SAR can synthesise one of 10 kilometres (*Overview*, 2020). Due to the simulated long antenna, space-borne SAR can reach a very high spatial resolution. For instance, the new Capella-2 reaches a spatial resolution of 50 centimetres per pixel (Nava et al., 2021). Space-borne SAR systems TerraSAR-X, COSMO SkyMed, HRWS, Radarsat-2 reach a spatial resolution of 1 metre or higher as well (Braun & Hochschild, 2017; Nava et al., 2021). These high-resolution SAR images are commonly acquired with the *spotlight* mode. This mode gets accurate data by capturing a sequence of images of the same terrain within seconds. Another mode is the *stripmap* mode, which captures large areas. In contrast to the spotlight mode, the sensors are not pointing at one specific location. This mode requires more computation and has a smaller spatial resolution than the spotlight in general (Singh, Diwakar, Shankar, Shree, & Kumar, 2021). However, the advantage is being able to analyse significant terrains; the Sentinel-1 satellite stripmap mode covers areas up to 80 by 80 kilometres (Braun & Hochschild, 2017).

Remote sensing satellite images enable surveillance, analysis of infrastructure, agriculture, and natural disaster management (Sharma et al., 2008). For example, with remote sensing surveillance, illegal fishing can be detected (Rao, 2020). Space-

Figure 3.1: Radar frequency bands. Credits: Herndon et al. (2020)



borne remote sensing is a cheap alternative for the detection of oil spills (Jha, Levy, & Gao, 2008). Remote sensing in precision agriculture has developed rapidly in the last years (Wójtowicz, Wójtowicz, Piekarczyk, et al., 2016). Moreover, detecting infestations in weeds can be realized (Thorp & Tian, 2004). Bajwa, Rupe, and Mason (2017) use remote sensing images to identify deceases in soybeans, and one of the models can recognise 58% of deceased plants and 97% of healthy plants. In addition, SAR is the preferred space-borne equipment for mapping floods (Martinis & Rieke, 2015). Combining optical and radar remote sensing images in dark areas allow a relatively quick and reliable response to floods (Hussain & Shan, 2010). After disasters occur, satellite images also help in the recovery, for example, with the damage claim process (Simonovic et al., 2002, pp. 96-97).

Optical remote sensing is passive remote sensing which relies on solar illumination. Optical remote sensing measures photometric quantities such that the wavelengths are within the spectrum of human visible light (between 400 nanometres and 800 nanometres). The human eye responds to green light stronger than others, so that green has a greater luminous flux when the same intensity in green appears. Recorded wavelengths in optical sensors are weighted to mimic these properties. On the other hand, SAR equipment is a form of active remote sensing; it sends wavelengths and records the time it takes to reflect to its sensors. SAR technology uses a wavelength of 3.8 to 2.4-centimetres for high-resolution images (Herndon et al., 2020). Figure 3.1 shows the difference in wavelengths.

The main advantage of the long SAR bands is that these microwave signals can penetrate the weather, making it possible to capture areas under clouds. Similarly, longer SAR wavelengths can penetrate leaves, making it possible to show inundations in areas covered by trees as well (Martinis & Rieke, 2015). On the other hand, cloud and leaf coverage limits optical signals due to their small wavelengths. Hence, it is not feasible to monitor a region covered by (heavy) clouds by optical

Figure 3.2: Monitoring a region covered by heavy clouds. Credits: Insight (2017)

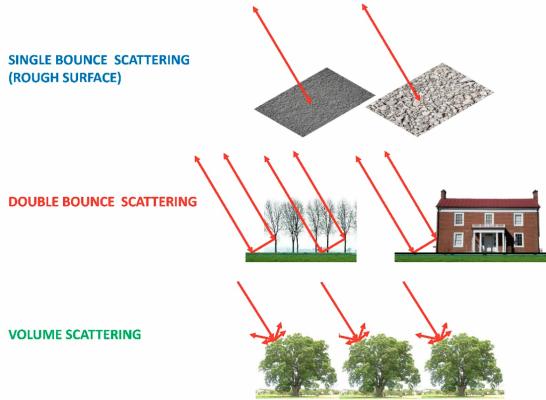


imaging, see Figure 3.2. Moreover, optical imaging depends on the (external) light from the sun for illumination. Thus, regions should be illuminated appropriately to achieve a decent optical image. When regions do not receive sufficient illumination due to atmospheric and photometric effects, (near) real-time surveillance is not possible. On the other hand, SAR itself actively sends and retrieves signals to the earth's surface. These signals bounce differently from the surfaces depending on the wavelengths and material properties. For example, calm water and other smooth surfaces mirror the signals in a different direction, which prevents them from reaching back to the SAR sensors (Martinis & Rieke, 2015). Thus, (calm) water appears black or dark on SAR images, which makes water easy to detect. When SAR sends signals to a rough area, signals scatter in different directions and only partly bounce back towards the sensors. In urban areas, often it is assumed that a double bounce occurs; signals reflect on the earth perpendicular and bounce on an building parallel to their original direction back to the sensors (Delgado Blasco, Fitrzyk, Patruno, Ruiz-Armenteros, & Marconcini, 2020). The main scatter mechanisms; single bounce (rough surface), double bounce and volume scattering and are shown in Figure 3.3.

## 3.2 Noise

Noise is a random variation of pixel intensities arising from the acquisition process of the digital images. The random noise is known by various types, of which speckle, quantisation, impulse and Poisson noise are the most frequently discussed (Goyal, Dogra, Agrawal, Sohi, & Sharma, 2020). Additive White Gaussian noise has the normal distribution probability density function and it is the most common one (Barbu, 2013; Boncelet, 2009). Salt-and-pepper impulse noise

Figure 3.3: Main backscatter mechanisms. Credits: Delgado Blasco et al. (2020)



may occur due to defective sensors (Chan, Ho, & Nikolova, 2005). Quantisation noise usually occurs in the acquisition phase when variables get converted to another level or from continuous to discrete (Boncelet, 2009). Speckle noise gets created because of random interference between the coherent returns due to the differences in the surface within pixels (Boncelet, 2009). To that end, denoising algorithms aim to decrease the amount of noise while preserving important structures. Noise can be modelled as additive or multiplicative. An additive noise image ( $y_{add}$ ) models noise as being added to an unknown high-quality image, resulting in a degraded noisy image (Gu & Timofte, 2019). With a multiplicative noise image ( $y_{mul}$ ), noise is modelled as being multiplied by the image. See Equation 3.1 and Equation 3.2 for clarification, where  $n$  denotes the pure noise,  $x$  is the clean image and  $y$  denotes the corrupted image.

$$y_{add} = x + n \quad (3.1)$$

$$\begin{aligned} y_{mul} &= x \cdot n \\ \log(y_{mul}) &= \log(x) + \log(n) \end{aligned} \quad (3.2)$$

Multiplicative speckle noise corrupts SAR images because of random interference of electromagnetic waves (Saevarsson, Sveinsson, & Benediktsson, 2004). There are various techniques to reduce the speckle. For instance, by combining statistically uncorrelated speckle patterns multilook images can be created. The disadvantage of this method is the decreased system resolution (Ouchi, 1985). In this paper, we consider the more challenging 1-look images. In addition, most denoising algorithms only model additive noise, making them less fitting for the speckle-noise. A way to still use the additive denoising algorithms is to take the

natural logarithm (log) of a noisy SAR image before applying the algorithm (Equation 3.2): the log of a multiplication of two instances is the sum of the log of those two instances. Some denoising algorithms are specifically designed to deal with multiplicative noise, making the use of the log and its product rule unnecessary.

The two procedures for denoising are spatial and transform domain filtering. Linear and non-linear filters divide the former into two categories. The non-linear filters do not assume a distribution of the random noise (Jain & Tyagi, 2016). Transform domain filtering first transforms the noisy images and attempts to denoise the transformed image. Preserving image features, including edges and corners, is a major challenge in reducing noise (Jain & Tyagi, 2016). The median, averaging and Gaussian and other common filters have the drawback of smoothing out some of the true edges (Barbu, 2013). On the other hand, low level denoising methods use only a convolution with hand-crafted filters. More advanced methods utilize deep learning (H. Li, 2014; K. Zhang, Zuo, Chen, Meng, & Zhang, 2017; Cozzolino et al., 2020; Lattari et al., 2019; Lu et al., 2021).

### 3.2.1 Gaussian Smoothing

With Gaussian smoothing, a Gaussian (the normal distribution) is convolved over an image. It is a local operation that averages neighbourhoods according to a Gaussian distribution with a given standard deviation. The advantage of this filter is that blobs are preserved, while with a strong mean filter, blobs can blend together. Gaussian smoothing has been widely used (B. Pan, 2013).

### 3.2.2 Block-Matching and 3D Filtering (BM3D)

Block-Matching and 3D Filtering (BM3D) is a block-matching algorithm proposed by Dabov, Foi, Katkovnik, and Egiazarian (2007). It takes a 2D block of an image and then finds similar blocks within the image. These similar blocks do not only have a similar average intensity but a comparable noise distribution. They are grouped into a 3D array. Then, the 3D arrays are processed with collaborative filtering. This grouping method reveals fine details while preserving important structures (Dabov et al., 2007).

### 3.2.3 Bilateral Filtering

Bilateral Filtering is another smoothing method that preserves edges (Tomasi & Manduchi, 1998). It is non-iterative and straightforward. It takes the average of surrounding pixels, which generally becomes problematic at edges since they get averaged out. The variation of intensities is taken into account to prevent this.

### 3.2.4 Anisotropic Diffusion

Isotropic smoothing takes the average of neighbourhoods, even if these neighbourhoods contain edges. It tries to only smooth pixels on the same side of an edge. This results in not smoothing the image as a whole but just the supposedly homogeneous parts. Unlike a Gaussian scale-space, comparing different scales is easy because the shapes and positions do not change (Perona & Malik, 1990).

### 3.2.5 Non-Local Means Denoising (NLMD)

Instead of just taking a local average with a small kernel, the image as a whole is considered in Non-Local Means Denoising (NLMD), which makes the algorithm non-local. Then, the average is weighted. Similar pixels get a higher weight than non-similar pixels. It tends to preserve edges and other details better than local denoising algorithms (Buades, Coll, & Morel, 2005).

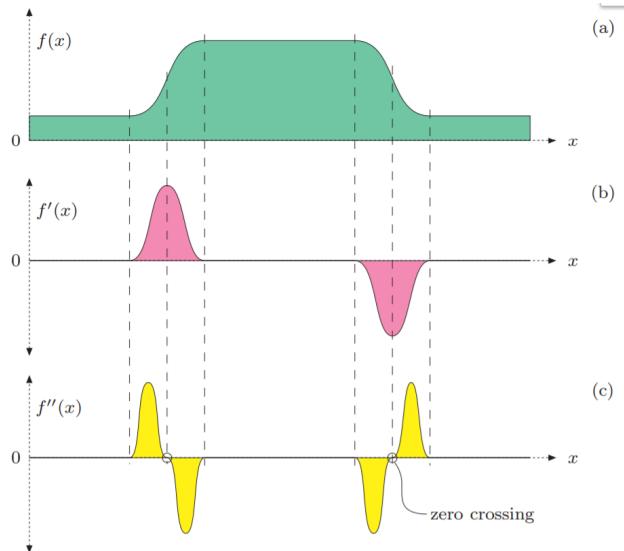
### 3.2.6 SAR Specific Denoising

Because of the multiplicative speckle that corrupts SAR images, standard denoising methods might not perform well on SAR images. Thus, denoising methods for SAR images have specifically been developed. For example, SAR expansions for NLMD, BM3D, Anisotropic Diffusion, Bilateral Filtering are presented (Zhao, Wang, Wu, Wang, & Yuan, 2014; Sica, Cozzolino, Zhu, Verdoliva, & Poggi, 2018; Gupta, Tripathi, & Bhateja, 2013; D'Hondt et al., 2013). Furthermore, deep learning methods tend to outperform classic methods (Molini, Valsesia, Fracastoro, & Magli, 2021). Nonetheless, CNNs need annotated data to train on, which is not always available for real SAR images. To tackle this problem, Molini et al. (2021) propose a self-supervised Bayesian method with similar or better performance to the supervised training approach. Recently, Cozzolino et al. (2020); Lu et al. (2021); Lattari et al. (2019) present supervised CNNs that reach or outperform the current state-of-the-art performance (Cozzolino et al., 2020; Lu et al., 2021).

## 3.3 Geometric Primitives

Geometric primitives are simple 2D or 3D shapes, like lines, squares, circles or cubes, that are able to capture complex information in a simple form. For example, a table can be described as a cube. In SAR images, monitoring tasks can be automated by detecting distinct geometric features like edges, lines and blobs. In this thesis, the focus is on the edge detection task for SAR images. Since edges form the most fundamental features of the geometric primitives, they can later be utilised to identify other shapes such as lines, corners and junctions. A frequently

Figure 3.4: Difference in first and second derivative edge detection a) Intensity b) First derivative, edges are located at peaks c) Second derivative, edges are located at the zero-crossing. Credits: Thipkham (2018)



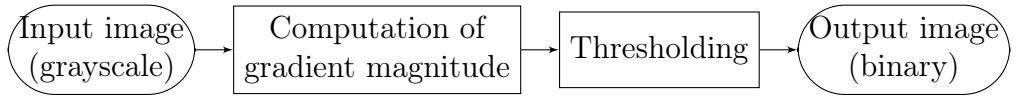
used benchmark to test performance in computer vision for segmentation and contour, and edge detection algorithms is the Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500) (Arbelaez, Maire, Fowlkes, & Malik, 2011a).

### 3.3.1 Edge and Boundary Detection

One of the oldest operations in image processing and a building block for more complex algorithms is edge detection (Spontón & Cardelino, 2015). Edges can give indications for boundaries and contours, and can help describe the (geometric) forms of objects in images. An edge reveals itself by an abrupt change in pixel intensity values. There are two main families of edge detection: first and second derivative based. The first derivative edge detection is the most commonly used (Kumar, Singh, & Shaw, 2016). With the first derivative based edge detection, an edge can be detected by a peak, whereas with the second derivative based edge detection, it can be detected by a zero-crossing, see Figure 3.4.

Edge detectors are sensitive to noise. Noise can be recognised as false edges by detectors. In other words, a pixel is identified as an edge, but in reality it is triggered by noise. The rate of false positives is preferred to be minimal. With more advanced edge detectors, the image is smoothed before calculating the edge maps, which results in fewer false positives.

Figure 3.5: First derivative edge detectors workflow. (Spontón & Cardelino, 2015)



## First Derivative

The most common edge filters are the Sobel, Roberts Cross, and Prewitt operators (Duda & Hart, 1973; Roberts, 1965; Prewitt, 1970). Sobel filters give significant weight to the centre coefficients of the kernel, which results in smoother images than Prewitt (Spontón & Cardelino, 2015). The computational cost of Sobel, Roberts and Prewitt are low compared to more complex edge operations. The edge filters calculate the derivatives in both the horizontal ( $G_x$ ) and the vertical direction ( $G_y$ ). Then, the gradient magnitude ( $G$ ) (the soft edge map) is calculated by Equation 3.3 as follows:

$$G = \sqrt{G_x^2 + G_y^2}. \quad (3.3)$$

The last step is to create a binary edge map, by thresholding, deciding from which pixels of the gradient magnitude to be considered as true edges. The standard procedure is demonstrated in Figure 3.5. Additionally, Canny is a widely used edge detection algorithm (Canny, 1986). It consists of: noise reduction, gradient calculation, non-maximum suppression, double thresholding and hysteresis. Firstly, it smooths an image with a Gaussian kernel. Secondly, it convolves a low-level edge detector (like Sobel or Roberts) on the smoothed image to obtain the gradient image. Afterwards, it applies non-maximum suppression to create thin edges by only considering pixels with high intensities, surrounded by pixels with lower intensities. Double thresholding and hysteresis are the last steps. Edges that are (partly) above both lower and upper thresholds are considered as true edges.

## Second Derivative

Second derivative edge detection is more sensitive to noise than the first derivative variants. One example is the Laplacian (Marr & Hildreth, 1980). To keep the false detection of edges to a minimum, the Marr-Hildreth algorithm applies a Gaussian filter to an image, before detecting the edges with a Laplacian convolution or uses a Laplacian of Gaussian filter directly (Spontón & Cardelino, 2015).

Deciding a threshold is a step that needs to be taken for every edge detector. Finding the optimal threshold is challenging and there are different approaches. For first derivative based edge detectors, an approach could be to sample thresholds within a given range. Another approach is the statistical approach, where a threshold is chosen based on statistics, for example, Otsu thresholding (Otsu, 1979). The

thresholding in the Marr-Hildreth algorithm is achieved by zero-crossing, which is the key feature of this algorithm (Spontón & Cardelino, 2015).

### State-of-the-art Edge and Contour Detection

Contour detection is more challenging than edge detection. Instead of finding only the change in (gray) pixel intensities, contour detection considers the texture, brightness and colour changes (Shen et al., 2015). CNN-based DeepEdge achieves state-of-the-art performance in contour detection (Bertasius et al., 2015). It is common practice to use contour (or other low-level features) for high level algorithms. Bertasius et al. (2015) use the opposite approach, by using object detection for contour detecting. Another famous contour detector is a per-pixel CNN. Moreover, Xie and Tu (2015) propose a holistically-nested edge detection (HED) that outperforms the state-of-the-art on the BSDS500, including DeepContour, DeepEdge and the per-pixel CNN proposed by Hwang and Liu (2015). Recently, Y. Liu et al. (2019) propose a boundary detector that outperforms all the aforementioned contour detectors on BSDS500. More impressively, Y. Liu et al. (2019) are able to obtain better performance than humans.

### SAR Edge Detection

In regular edge-based methods, the edges are the transitions between regions. In SAR images, they are abrupt changes in reflectivity (Ranjani, Gokila, & Thiruvengadam, 2008). For example, a road in a muddy field will have high reflectivity, while the muddy field might mostly absorb the radar signals. A number of SAR specific (often statistical) edge detectors have been proposed to deal with the speckle noise characteristics. The Ratio of Averages (ROA) is one example proposed by Airouche, Zelmat, and Kidouche (2008). Improvements on this algorithm are, for example, the Ratio of Exponentially Weighted Averages (ROEWA) and the maximum strength edge pruning ratio of averages (MSP-RoA) (Fjørtoft, Lopes, Marthon, & Cubero-Castan, 1998; Ganugapati & Moloney, 1995). ROEWA itself is improved as well and is widely used (Yu, Zhou, Jiang, Wu, & Xu, 2019; Ranjani et al., 2008). The scale invariant feature transform (SIFT) optimised for SAR images (SAR-SIFT) is based on ROEWA as well (Dellinger, Delon, Gousseau, Michel, & Tupin, 2012). Touzi et al. (1988); Schou et al. (2003) propose other statistical edge detectors based on the Constant False Alarm Rate (CFAR). Additional edge detectors for SAR are a ratio-based edge detector (RBED) and a multiscale edge detector based on Gabor filters (Q.-R. Wei & Feng, 2015; Xiang et al., 2017a). Although these edge detectors can achieve robust and effective results in their own evaluations, the comparisons tend to include only a couple of (simulated) SAR images. Therefore, our aim is to provide a comprehensive evaluation.

CNNs have found their way to SAR edge detection as well. C. Liu et al. (2020) combine Gradient by Ratio (GR) and HED for SAR into GRHED. Their CNN takes the challenging approach of edge detection on 1-look SAR images and perform better than HED and GR separately. The challenging part about SAR images is, next to the speckles, the absence of labelled ground truths for edges. To that end, C. Liu et al. (2020) provide a simulated 1-look dataset generated from the real world BSDS500 to train and evaluate their models, which we also utilize in our experiments in the upcoming sections.

## Decision Fusion Methods for Edge Detection

The idea of fusion methods is that different algorithms rarely have the exact same error, so taking the average or combining them in smart ways averages out the errors. Fusion methods are used to combine multiple results or algorithms. It is possible to fuse algorithms or their soft or binary outputs together. The latter does provide the least amount of useful information (Ruta & Gabrys, 2000). In the next part we will consider fusion for edge detecting, but fusion methods can be applied to arbitrary classifiers.

Voting methods are widely used to mask errors and combine binary outputs (Tseng, 2017). If a majority of the algorithms define a pixel as an edge, it can be considered an edge (e.g. defined by one or True). Otherwise, it will be considered a non-edge (e.g. defined by zero or False). With (speckle) noise, false positives occur in edge detection because they often are abrupt changes in pixel intensities. Using a voting method decreases the false positive rate. Moreover, due to the increasing numbers of edge detectors, the probability of multiple edges classifying the same noisy pixels as edge decreases. An extension from this majority voting algorithm can be complete agreement voting, where every algorithm has to define a pixel as an edge. Demanding all contributing algorithms to agree can significantly decrease the number of false positives. However, due to their different architectures, some algorithms are better at detecting specific details than others. Therefore, the number of details (true positives) may decrease as well. A motivation for using complete agreement voting could be having a significant amount of false positives with majority agreement. An alternative is called Byzantine voting, where  $\frac{2}{3}$  of the algorithms should agree (Melnyk, Wang, & Wattenhofer, 2018).

Before voting, there is an approach to be considered. Each method has their advantages, even methods that do not perform well. To exploit that, algorithms can be weighted before fusion. Appropriate weights can be set by basing them on the performance of each edge detector, brute-forcing or training by using supervised learning (e.g. support vector machines or artificial neural networks). Basing weights on the quantitative performance criterion (e.g. area under the curve or accuracy) is the most common approach.

### 3.3.2 Additional Features and Primitives

Blobs are regions that differ from the rest of the image. Higher level algorithms detect small circles as oil refineries and squares as houses, based on these blobs and geometric primitives. Blob detection is also used for segmentation, because clusters can be identified by blobs (Maithree, Dinushka, & Wijayasiri, 2020). A challenge in blob detection is the overlapping of adjacent blobs, especially after smoothing an image. One of the methods to isolate adjacent blobs is the Difference of Guassians (DoG) (Xu, Wu, Gao, Charlton, & Bennett, 2020). Another blob (and edge) detector is the Laplacian of Gaussian (see also Second Derivative3.3.1), which can be combined into a multiscale blob detector to isolate adjacent blobs (G. Wang, Lopez-Molina, & De Baets, 2020). A use case of blob detection is the automatic iceberg detector from Soldal, Dierking, Korosov, and Marino (2019), the detector works by detecting (small) blobs in SAR images.

Lines are, albeit simple, a form of geometric primitives. The detecting of these primitives is useful for detecting more complex primitives and line-shaped objects. Line detecting can be a challenging task. Before detecting lines, images are often preprocessed by denoising and edge detection algorithms to extract the important features. An example of line extracting in SAR images is the rapid line-extraction (RLE) method (Q.-R. Wei, Feng, Zheng, & Zheng, 2017).

Road extraction is significant for disaster response (Xin, Zhang, Zhang, & Fang, 2019). To that end, line and edge detecting can be used for road extraction (Sun, Zhang, Huang, Zhao, & Lu, 2014). Jia, Kuang, and Su (2004) detect roads with the Hough transform, a widely used line detector. Roads can be extracted pixel-, region- or knowledge-base (Chen et al., 2018). They are used for detecting the need for maintenance or blocked roads. Pixel-based methods extracts features like edges and lines which are the primary source to some of the road extraction methods (Chen et al., 2018). Recent road extraction in SAR images methods also use CNNs, some of which are based on the detection of geometric primitives (Xin et al., 2019; Y. Wei, Wang, & Xu, 2017; Q. Zhang et al., 2019; Y. Li, Zhang, & Wu, 2017; H. Li, Chen, Yang, Liu, & Zhong, 2019).

A corner is the intersection of multiple edges and is one of the most crucial steps in feature identification (Cuevas, Díaz-Cortes, & Mezura-Montes, 2019). The most widely used corner detector is Harris, but it can be used for edge detection as well (Wu, Zhou, & Ji, 2016; Kang, Han, Yang, & Tao, 2006). However, Harris is not suitable for SAR images. By combining it with a bilateral filter or phase congruency robustness can be improved and made suitable for SAR images (Wu et al., 2016; Eltanany, Amein, & Elwan, 2021). A use case of corner detection is the removal of land in images, which helped M. Zhang, Qiao, Xin, and Zhang (2021) develop a sea-land segmentation algorithm.

## Chapter 4

# Approach

The fundamental scheme for edge detection is demonstrated in Figure 4.1 (Kumar et al., 2016). The idea is to smooth images, enhance important features or details, detect edges, a threshold the smooth edge output and locate the exact edge pixels. For our approach, we use denoising algorithms that are enhancing the edges already, so we skip the enhancement step. Since the ground truth edges are not always one pixel wide either (see Section 4.1), we will also skip the localisation step (in Canny, it is part of the algorithm). For our experiments, we first expand the BSDS500 dataset by simulating SAR-like noise on the expanded BSDS500 dataset. Then, we apply four denoising algorithms and compare their performance. Afterwards, we utilize a number of edge detection methods on the best-denoised images. Finally, by iterating over a set of thresholds we evaluate the performances (see Figure 4.2).

Figure 4.1: Edge detection steps according to Kumar et al. (2016).

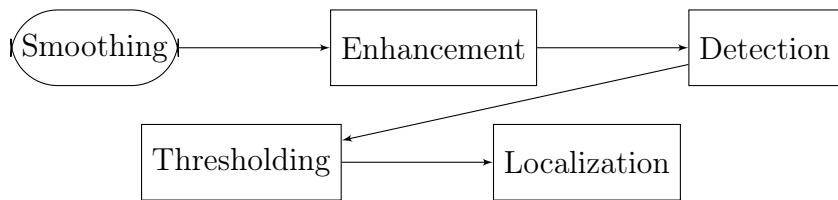
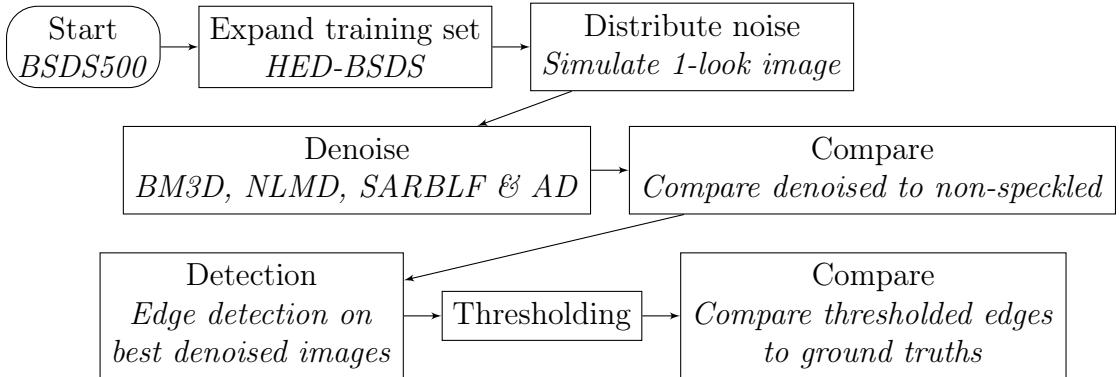


Figure 4.2: Our experimental setup.



## 4.1 Dataset

The original BSDS500 contains 500 natural images (Arbelaez, Maire, Fowlkes, & Malik, 2011b). Among those, 300 images are reserved for training and validating,<sup>1</sup> and 200 images are for testing. Ground truth edge maps are manually generated by human annotators. The number of annotations differs per image. Following the common practice, we consider any annotated pixel as ground truth for our evaluations, which results in edges that can sometimes be wider than one pixel.

C. Liu et al. (2020) generate 28800 images from the training set by applying different data augmentations such as rotating and scaling to train their CNN model. We follow their procedure to distribute realistic speckle noise on the entire dataset to simulate 1-look images. The dataset consisting of 29000 simulated 1-look images will be referred to as the BSDS500-speckled dataset. The dataset and the benchmark are available at [https://github.com/readme/SAR\\_edge\\_benchmark.git](https://github.com/readme/SAR_edge_benchmark.git) and will enable us to create comprehensive and fair experimental evaluation for SAR images on edge detection.

## 4.2 Denoising

When using denoising for edge detection, there is a trade-off. The more noise gets removed, the more (micro-)edges blur or disappear. Therefore, we consider denoising algorithms that tend to preserve structures better. To that end, BM3D, bilateral filtering, Anisotropic Diffusion and NLMD are to be evaluated. We use a bilateral filtering algorithm designed for SAR specifically (SARBLF) (D'Hondt et al., 2013; D'Hondt, López-Martínez, Guillaso, & Hellwich, 2018). Both SARBLF

<sup>1</sup>From now on, the 300 images for training and validating will be referred to as the training set.

and Anisotropic Diffusion take parameter  $N$ , the number of iterations. With an higher  $N$ , more noise (and with it micro-edges) will get removed.

For quantitative evaluations, denoised images are compared against the clean images without speckle. To that end, the whole dataset of 28000 images are evaluated using mean squared error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) and normalised cross-correlation (NCC).

## 4.3 Edge Detection

Unfortunately, most of the SAR edge detectors do not make their codes publicly available. Therefore, we are not able to include many SAR based methods. Among them Gradient Ratio (GR), provided by the code of C. Liu et al. (2020) is evaluated. It is used by the authors as the baseline to compare their supervised CNN model. Apart from GR, we focus mostly on low-level edge detection methods to provide baselines for future evaluations. The edge detectors to be evaluated are Canny, Farid, Frei-Chen, Gabor, a clustering-based edge detector, LoG, Prewitt, Roberts, Scharr, Sobel and Wavelet 2D discrete transform. For the methods that have parameters, the training set of 28800 (denoised) images are utilized for tuning. Then, the best performing set of parameters are used to evaluate the test set of (denoised) 200 images. For quantitative evaluations, we use the receiver operating characteristic (ROC) curve and its area under the curve (AUC) together with the metrics derived from confusion matrices. These metrics are precision (PPV), accuracy (ACC), F1 score (F1), and markedness (MK) and the Fowlkes–Mallows index (FM). For these metrics a value of 1 means the highest possible performance, and a value of 0, or -1 for MK, means the lowest performance possible.

### 4.3.1 Canny, LoG and GR

Canny and LoG smooth the image with a Gaussian filter of width  $\sigma$  before performing edge detection. For Canny it is a separate step. For LoG the image is first smoothed with a Gaussian kernel, the edges are detected with a Laplacian. Thus, for the parameter selection of Canny and LoG, the only choice to consider is  $\sigma$ . Often  $\sigma = 1.4$  is preferred (Ma, Li, Zhang, & Yan, 2012). Values bigger than 2.7 can reduce performance (Kalbasi & Nikmehr, 2020). Saif (2017) show that a sigma between 1.4 and 2 is optimal. Indeed, the sigma value is highly dependent on use case, yet we use relatively small sigma values to also capture micro edges. To that end, we sample different  $\sigma$  values from the literature and evaluate them based on qualitative and quantitative evaluation. Finally, GR is a ratio of exponentially weighted averages (ROEWA) method realized by computing exponentially weighted local means (Dellinger et al., 2012; C. Liu et al., 2020). It

tends to be more robust to noise than the simple ratio of averages. Besides Canny and LoG, GR also uses smoothing by the exponential weight parameter  $\alpha$ .

### 4.3.2 Wavelet Transformation and Gabor Filtering

A Wavelet decomposes image signals to different scales of frequency bands (Q. Pan, Zhang, Dai, & Zhang, 1999). First, the image is transformed with a (discrete) Wavelet into different scales. Second, a threshold is estimated and applied. Finally, the inverse Wavelet transform is performed (Jain & Tyagi, 2016). Varying the specific Wavelet signal, the threshold and the scales changes the performance. For the Wavelet evaluation however, we do not tune the parameters, but use the discrete 2D Wavelet transform from Lee, Gommers, Waselewski, Wohlfahrt, and O’Leary (2019) with a wavelet from the biorthogonal families.

Gabor filters are powerful in extracting certain features and there is a endless amount of orientations and combinations. For the experiments, we consider four Gabor filter banks based on the literature. M. Wang, Gao, Huang, Jiang, and Gao (2019); Khan, Khan, Maqsood, Aadil, and Ghazanfar (2019) use GS5O8. It represents a Gabor filter bank with 5 scales (5 different frequencies) and 8 orientations. The values of the frequencies and orientations are calculated with scales (S) and orientations (O) according to Equation 4.1 and 4.2. With these equations from the literature and additional preliminary experimentation, we evaluate GS5O8, GS3O11 and GS5O11.

$$frequency(i) = \frac{0.2}{\sqrt{2}^{i-1}}, \text{ where } i = 1, 2, \dots, S \quad (4.1)$$

$$orientation(i) = \frac{((i - 1) * \pi)}{O}, \text{ where } i = 1, 2, \dots, O \quad (4.2)$$

### 4.3.3 K-means Clustering

K-means aims to separate an image in K clusters. In SAR images, the edges are often seen as boundaries between homogeneous areas. The motivation behind considering a clustering based algorithm can be demonstrated by using an example. Consider a SAR image of a road separating two fields. The ideal edge detector should only detect the road as an edge and not any noise within the two homogeneous fields. Thus, K-means (with K=2) should ideally separate the two fields in two clusters. Since in real images you often have other factors to consider that might yield more than two clusters, we also consider K=3 and K=4.

## 4.4 Fusion Methods

Additionally, we explore the effect on the performance of various fusion methods. The goal is to discover if different combinations of edge detector can perform better than any individual one. Moreover, we average the soft outputs of the edge detectors and combine them. The combination of methods is decided in different ways: a combination of the best performing algorithms and a combination of very different performing algorithms. Instead of simply averaging the soft outputs, the outputs are weighted. The weights are decided based on the AUC metric and also on an optimization scheme aiming to maximize the F1 score. In total, we evaluate 14 different fusion settings.

## 4.5 Implementation Details

### 4.5.1 BSDS500-speckled

The original BSDS500-speckled is created with the Matlab code provided by C. Liu et al. (2020)<sup>2</sup>. The images are saved in `.mat` files, and the pixels are of type `float64`. We convert them to standard `uint8 .png` images using the OpenCV Library (Bradski, 2000). It also reduces the storage space from 200GB to 16GB. Every image has a (varying) number of human annotations. In that sense, we consider any pixel annotated by a human as a ground truth edge.

### 4.5.2 Denoising

All the denoising algorithms are implemented in Python. BM3D is taken from the original repository of Mäkinen, Azzari, and Foi (2020) (Mäkinen, Azzari, & Foi, n.d.). Parrilli, Poderico, Angelino, and Verdoliva (2012) provide a SAR version of the algorithm based on a 2007 BM3D implementation. In their work, they state that the homomorphic BM3D performs quite similar. Thus, we assume the current improved version (2020) of BM3D is also sufficient for SAR images. Additionally, we use the OpenCV implementation of NLMD and the anisotropic diffusion by `medpy.filter.smoothing.anisotropic_diffusion` (n.d.) (Bradski, 2000). Moreover, SARBLF has two parameters; spatial scale parameter (`gs`) to adjust the spatial extent of the filter, similar to the window size, and radiometric scale parameter (`gr`) to control the amount of filtering for weighting the local averages of intensities. D'Hondt et al. (2013) provide their code on Github with standard parameters<sup>3</sup> of `gs=2.8` and `gr=1.4`. In their article, however, they use `gs=2.2`

---

<sup>2</sup><https://github.com/ChenguangTelecom/GRHED>.

<sup>3</sup><https://github.com/odhondt/ndsar/blob/master/ndsar.pyx>.

and  $gr=1.33$ . We will evaluate both parameter settings with different numbers of iterations. Finally, for the models assuming additive noise, the log image is taken similar to Equation 3.2.

### 4.5.3 Edge Detection

The GR images are generated with the Matlab code from C. Liu et al. (2020). Furthermore, all edge detection methods are based on Python implementation. The majority of the low level edge detectors are implemented with the scikit-image library (van der Walt et al., 2014). However, the Frei-Chen implementation is based on Umesh (n.d.), where the Frei-Chen kernels for edge detection and line detection are combined to one detector with 8 kernels. In addition, we use the discrete 2D Wavelet transform from Lee et al. (2019). The K-means based edge detection is implemented with the K-means function from OpenCV and the Sobel function from scikit-image (Bradski, 2000; van der Walt et al., 2014). In addition, our LoG implementation uses the Scipy library (Jones, Oliphant, Peterson, et al., 2001–). To be more specific, `scipy.ndimage.gaussian_laplace`. This function uses the second derivatives of a Gaussian for a multidimensional Laplace filter. For LoG we sample  $\sigma=1$ ,  $\sigma=1.4$  and  $\sigma=2.7$ , to be in compliance the literature of Canny. Preliminary experiments revealed that  $\sigma = 3$  also shows promising results. Therefore, we evaluate  $\sigma = 3$  as well. Furthermore, for combining the kernel edge detectors, like Sobel, Frei-Chen and Gabor, the convolutions are taken with the image and all those convoluted images are squared and added together. The soft output edge map is the square root of the total sum of those squared convolutions. Every edge detector takes an `uint8` denoised image as input and outputs an edge response map. The response map is normalized with min-max to the range of 0-1. After the normalization, thresholds are uniformly sampled by 0.05 from [0,1] to provide confusion matrices for the ROC curves and to derive other metrics. The same sampled thresholds are taken for both lower and upper threshold of Canny for fair evaluation, which results in over 400 combinations of thresholds for each  $\sigma$  value. We sample  $\sigma=1$ ,  $\sigma=1.4$  and  $\sigma=2.7$  from the Canny literature. Preliminary experiments revealed that all of these values might unnecessarily smooth the image. Thus, we utilize  $\sigma=0.1$  as well. Canny implementation is based on the scikit-image library (van der Walt et al., 2014).

### 4.5.4 Fusion

The fusion algorithms are based on performance or on difference. For example, for some fusion methods, we take the (weighted) average of the top 5 best performing algorithms based on the AUC, for others we pick algorithms that significantly differ in characteristics. Preliminary experiments revealed that smoothing the

image before applying the different low level edge detectors tend to improve results. Therefore, we experiment with that as well.

The simplest fusion is the average of the 5 best performing (low-level) edge detectors based on the AUC, excluding GR. The reason to exclude GR in these settings is that, since our goal is to explore the effect of fusion methods on the performance, we want to explore if fusion can improve the performance of classic non-SAR edge detectors. However, GR is, albeit very simple, an edge detector for SAR. Nonetheless, for another setting, we also include GR to explore if further achievements can be achieved. The five best (AUC) performing methods, excluding GR, are Farid, Prewitt, Sobel, Scharr and Wavelet. The first fusion setting, is simply taking the average of these five methods, for five other settings we do the same, but smooth the image with a Gaussian filter before applying the individual edge detectors. The five settings are for a  $\sigma$  value of  $\{0.4, 1, 1.4, 2.7, 3\}$ . Other settings include weighting. For two of those settings we optimise the weights to achieve the highest F1 scores with the `scipy.optimize.minimize` function, which uses the L-BFGS-B by default with boundaries (Jones et al., 2001–; Zhu, Byrd, Lu, & Nocedal, 1997). With the other settings for the five methods, the images are weighted by the AUC. The final settings combine the weighted average of LoG, Wavelet and Farid. The motivation is the differences in characteristics, which might combine the advantage of each characterises, while averaging out the disadvantages. The weights are decided upon initial experimentation.

# Chapter 5

## Evaluation

### 5.1 Denoising

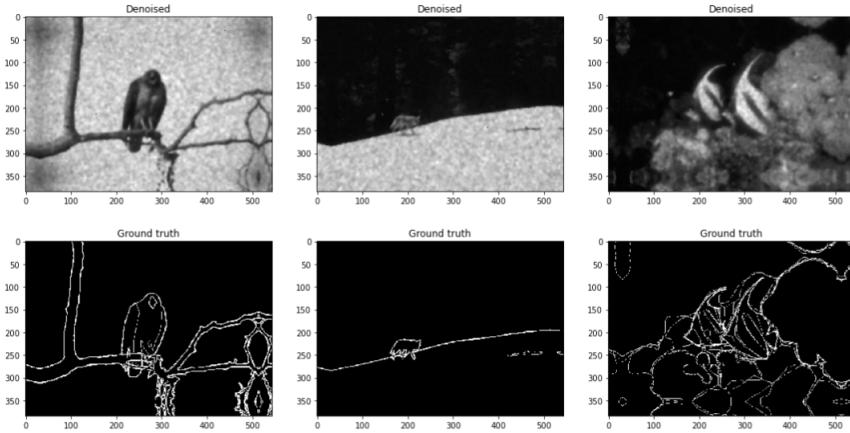
As for the denoising step, we evaluate BM3D, SARBLF, Anisotropic Diffusion and NLMD on the complete BSDS500-speckled dataset. All 29000 images are first multiplied with 1-look speckle noise as provide by C. Liu et al. (2020). Then, we apply the denoising methods and compare the outputs with the clean non-speckled images. The effect of the different denoising methods is shown in Figure A.1. In addition, Table 5.1 provides the averages of the metrics for the entire dataset, where N indicates the number of iterations.

Table 5.1: Evaluation of various denoising methods on the entire dataset.

Name	MSE ↓	PSNR ↑	SSIM ↑	NCC ↑
SARBLF (N=2, gs=2.2, gr=1.33)	0,0109	28,8672	0,7211	<b>0,9258</b>
SARBLF (N=3, gs=2.2, gr=1.33)	<b>0,0105</b>	<b>28,8904</b>	<b>0,7212</b>	0,9232
SARBLF (N=3, gs=2.8, gr=1.4)	0,0121	28,7958	0,7023	0,9148
SARBLF (N=4, gs=2.2, gr=1.33)	0,0115	28,8321	0,7076	0,9171
Anisotropic Diffusion (N=1)	0,0793	27,8338	0,4095	0,7298
Anisotropic Diffusion (N=2)	0,0596	27,9018	0,4808	0,7957
Anisotropic Diffusion (N=10)	0,0229	28,2714	0,6460	0,9026
Anisotropic Diffusion (N=15)	0,0185	28,3742	0,6667	0,9099
NLMD	0,0748	27,9419	0,5385	0,9007
BM3D	0,0550	27,9572	0,6023	0,8963

The results show that SARBLF (N=3, gs=2.2, gr=1.33) achieves the best results on MSE, PSNR and SSIM metrics, whereas SARBLF (N=2, gs=2.2, gr=1.33) achieves the best NCC results. Therefore, we use SARBLF (N=3, gs=2.2, gr=1.33)

Figure 5.1: Examples of the denoised images and their ground truths.



as our denoising algorithm and apply it to the entire dataset as a pre-processing step for our edge detection evaluation. The results are also consistent with the findings of the authors of SARBLF in terms of number of iterations and scale parameters (D'Hondt et al., 2013). In addition, anisotropic diffusion with 15 iterations appears as the second best method, whereas NLMD performs the worst.

Additionally, Figure A.1 shows that the methods are able to preserve relatively sharp edges. BM3D and NLMD seem to create better homogeneous areas, but they tend to create artefacts that might be considered as false edges, especially BM3D. Moreover, both BM3D and NLMD appear to have low contrast and remain darker, even though their pixels are in the same range as anisotropic diffusion and SARBLF. SARBLF appear significantly less noisy than the anisotropic diffusion. SARBLF ( $N=3$ ,  $gs=2.2$ ,  $gr=1.33$ ) shows stronger edges, than SARBLF ( $N=3$ ,  $gs=2.8$ ,  $gr=1.4$ ). Therefore, the qualitative evaluations are consistent with the quantitative evaluations. Additional comparisons of the are provided in Figure A.2.

## 5.2 Edge Detection

In Figure 5.1, examples of denoised speckled images and their ground truth edge maps are presented. Ideally, a ground truth map is a binary map with edges of one pixel wide. Due to the multiple annotators, the edges in the BSDS500-speckled set can be multiple pixels wide, or even annotated at slightly different places. That, and of course the strong speckle, albeit denoised, keeps the performance of any fundamental edge detector on this dataset modest.

### 5.2.1 Low Level Methods

The ROC curves for the algorithms without hyperparameters are presented in Figure B.1. It shows that they achieve similar results, while Farid slightly outperforms others with an AUC of 0.6824. In addition, Figure A.3 provides qualitative results for the thresholds achieving the best F1 scores. It shows that the methods capture many false positives due to the speckle noise. Nonetheless, Farid has the lowest false negatives while capturing a relatively accurate representation of the ground truth. Frei-Chen performs the worst by a large margin. It might be attributed to its isotropic nature that considers all directions equally, which is negatively affected by the speckle noise. In addition, the quantitative results in Figure C.1 show that Farid algorithm achieves the best performance on accuracy, F1 score, PPV, FM and MK metrics, while being on par with Prewitt on BA.

The 2D discrete Wavelet edge detector performs better than Frei-Chen in terms of AUC. However, it gets outperformed by the other low level detectors. Figure A.3 shows that the mediocre performance is again due to high false positives. Varying the default Wavelet signal, the threshold and the scales might change the performance. In our case, the threshold might be too low for the BSDS500, which results in low robustness to the noise. Another reason is that the 2D Wavelet transform by Lee et al. (2019) returns an edge map of a different size than the original images due to the filter size. This unfortunate feature forces us to resize the images which manipulates pixel values and might stretch out the noisy pixels.

### 5.2.2 LoG

The LoG performance is low compared to the low level methods, reaching an AUC of 0.5537, see Figure B.2. This is in compliance with the findings of Bachofer et al. (2016) that LoG is not performing well on SAR images because of the low gradients due to the remaining noise negatively influencing the Gaussian filter. Based on the AUC, it achieves the best performance with  $\sigma=2.7$  on PPV, ACC, F1, MK and FM. When comparing the edge maps of LoG in Figure A.5, it is also clear that the results do not look decent. Nonetheless, the best result with the least false positives is achieved for the highest  $\sigma$ .

### 5.2.3 Canny

Canny is a powerful algorithm. However, as stated before, the edges in the BSDS500-speckled can be multiple pixels wide, or even annotated at slightly different places. One of the features of Canny is the ability to detect edges of one pixel wide, which makes it less suitable for our benchmark. The ROC curves for different sigma values and different thresholds are provided in Figures B.4, B.5

and B.6. All the settings capture (almost) all the true (and false) positives with a threshold of 0, and all the true (and false) negatives with a threshold of 1. Since it produces one pixel wide edges the ROC curves are shorter and the AUCs are lower. Thus, due to the disharmony between Canny and the ground truths, the AUC appears as an unfair evaluation criterion. Additional quantitative results are provided in Figure C.6 and C.7. It can be observed that Canny has the best PPV and MK performance for  $\sigma=1.4$ , with a low threshold of 0.55. The highest ACC is reached with  $\sigma=1$  and low threshold of 0.35. We reach the highest F1 with  $\sigma=0.1$  and low threshold of 0.1 and highest FM with  $\sigma=1$  and low threshold of 0.15. In addition, Figure A.7 provides a number of examples. It shows that  $\sigma=1.4$  appears to be the most suitable setting. Additional examples are provided in Figure A.8 and A.9. The optimal average high threshold for PPV and MK is lower than the low threshold, which makes the credibility for the measurements of PPV and MK for Canny questionable. Hence, only F1, ACC and FM should be considered. Further, the thresholds are in compliance with the literature; Canny (1986) suggests that the upper threshold should be either two or three times as high as the low threshold. Even though the thresholds differ for FM and ACC, the edge maps appear identical and show significantly less noise than the F1 images. Thus, we consider that either FM or ACC achieves the optimal results. The FM image has low threshold and does not contain more noise than the ACC image, but still show slightly more micro edges. FM is in compliance with the literature; the upper threshold is exactly two times the size of the low threshold. Thus, we conclude that the optimal results for Canny are those according to the FM, which leads to  $\sigma=1$  and a optimal low threshold of 0.15 and high threshold of 0.30.

#### 5.2.4 K-means

Based on the ROC curves provided in Figure B.2, it achieves better results than LoG (0.5942 vs. 0.5537). However, the performance is lower than all of the low level methods and also the Gabor filters. Higher number of clusters tend to improve the results. Figure C.4 shows the same outcome, it performs better for 4 clusters for ACC, F1, PPV and MK and the same for FM. Thus, 4 clusters appears as the best performing setting. Recall the example of a road separating two fields. Since real world is way more complex and diverse than a simple road separating two fields, it might be expected that the number of clusters should be larger than 2. Since the algorithm aims at grouping similar pixels, variations due to different scatters might effect the grouping due to intensity variations. Theoretically, the K-means based edge detector could perform well on images with just a couple of strong edges. Figure A.10 shows that it recognizes the main edges in the image with 2 clusters by differentiating between the background and foreground, whereas with 4 clusters the image is divided in sub-regions of similar intensities. Even

thought the performance tends to keep increasing with more clusters, we do not experiment with more than 4. The advantage of the K-means method disappears with more clusters; the idea of detecting the edges by dividing big objects from the surroundings. However, with a large number of clusters the image barely differs from the original. Figure A.11 shows that there is no (or barely) any difference between the clustering based method with 20 clusters and Sobel, while the computation time significantly increases with more clusters. Hence, K-means does not appear as a decent edge detection method for highly speckled images.

### 5.2.5 Gabor Filters

For the Gabor filters, the ROC curves provide a fair evaluation, where every curve starts close to 0 and ends close to 1, see Figure B.3. Based on the AUC, it outperforms LoG and K-means. The best performing settings, which are for GS508, outperform all low level methods, except for Farid. GS5011 is close in performance, but due to its 55 kernel operations run time is significantly longer. Moreover, the additional run time does not seem to be worth the performance trade-off, according to the ROC curves. The quantitative metrics in Figure C.5 are in compliance with these results as well for F1 and FM scores. GS5011, however, performs better on ACC, PPV and MK than GS508. Nonetheless, since the difference is in the third decimal, again it does not outweigh the run time; GS5011 takes over 500 hours to finish with our implementation. Since GS3O11 has 7 filters less than GS508 and the performance is the best for ACC, it could be considered for those with either low computation power or time constraints. All in all, we conclude that GS508 achieves the best performance, based on F1, FM and AUC (0.6774). In addition, the edge predictions in Figure A.4 show also a minor difference between the performance of GS508 and GS5011, whereas GS3O11 appears noisy. All the Gabor filters show thick edges, or even blobs, even with low thresholds. Due to the multiple ground truth annotations, the edges in the ground truth are sometimes thicker as well, which explains why the performance of Gabor is reasonable. Nonetheless, for real life applications, the edges might be too thick. Smaller scales can be preferred or edge thinning could be considered with, for example, zero-crossing thresholding to make Gabor suitable for real life applications.

### 5.2.6 GR

For the GR with  $\alpha$  values 3,4,5 and 6, the ROC curves show a higher AUC than any of the previous methods, see Figure B.7. In addition, Figure C.3 shows that the best performance is with the  $\alpha$  of 2 for ACC, PPV and MK, which is not in compliance with the ROC curves and the AUC. Nonetheless, in terms of F1 scores, it indeed performs best on average for the  $\alpha$  of 5, which also achieves the

best AUC. For FM, the highest score is achieved with  $\alpha$  of 6. In addition, Figure A.6 shows that increasing the  $\alpha$  values also increases the thickness of the edges and further removes the noise effects. The  $\alpha$  that preserves the thinnest edges, while being robust to a fair amount of noise, is the  $\alpha$  of 3. However, GR with  $\alpha$  value of 5 removes all false positives due to noise and preserves most of the micro edges, at the cost of slightly thicker edges. Thus, we conclude that GR with the  $\alpha$  value of 5 is the superior setting.

### 5.2.7 Fusion

With the basic voting methods the binary output of multiple edge detectors with their best performing threshold (based on F1) is combined. On the other hand, fusing soft outputs tends to maintain more information. Figure B.8 and B.9 show the ROC curves for all different fusion settings. As expected, many fusion methods outperform the individual methods on the AUC. The highest AUC is obtained by a single method is 0.6994, by GR with  $\alpha$  of 5. In contrast, most fusion methods obtain AUCs of .7000 and above. However, this seems valid only for fusion methods using a (wide) Gaussian filter. Therefore, the question arises: is it the fusion or only the smoothing improves the performance? To that end, GR with  $\alpha=5$  does not use smoothing itself and fusing it with Farid does improve the performance, which shows that fusion indeed is useful. According to the F1 score, the average and weighed methods have slightly lower scores than the highest performing individual fused methods. However, with the smoothing of the images before applying the fusion, the F1 score and the AUC improve significantly. The optimized weights improves the F1 score even further. Thus, future work should focus more on the optimization than weighting with metrics or simple combinations. Based on the evaluations, the best fusion methods of traditional non-SAR edge detectors, is the weighted fusion of LoG, Wavelet and Farid, where Wavelet and Farid have the highest weights.

## 5.3 BSDS500-speckled Benchmark

To provide a more comprehensive comparison and form our benchmark, we use the ground truths and metrics from the original BSDS500 benchmark. The images are first corrupted with 1-look speckles and denoised as done in the previous cases. The soft edge map outputs of every edge detector, with the parameters found for the training set, are calculated. Then these edge maps are compared to the ground truths. As suggested by the developers of the original BSDS500 dataset, three metrics are used to compare the different algorithms: ODS F1 (fixed contour threshold for 200 images), OIS F1 (best threshold for each image), and average

Table 5.2: The performances of different methods on the 200 speckled optical images in BSDS500-speckled test. The rows are sorted by AP.

Algorithm	ODS (F1)	OIS (F1)	AP
Frei-Chen	0.4932	0.4922	0.0430
Wavelet	0.1976	0.1976	0.0655
LoG	0.4111	0.4129	0.0818
Roberts	0.2265	0.2265	0.0843
Gabor	0.3802	0.4206	0.2276
Scharr	0.4723	0.4733	0.2427
Sobel	0.4994	0.5004	0.2801
Prewitt	0.5092	0.5102	0.2976
Farid	0.5297	0.5368	0.3713
K-means	0.4970	0.5242	0.3801
Fusion	<b>0.5581</b>	<b>0.5694</b>	<b>0.4304</b>
GR ( $\alpha = 5$ )	0.5359	0.5589	0.4118
GRHED	<u>0.6603</u>	<u>0.6762</u>	<u>0.7208</u>

precision (AP). The results are provided in Table 5.2. Among the methods we provide, the fusion method achieves the best ODS F1, OIS F1 and AP while overall best results are achieved by the GRHED, CNN model of C. Liu et al. (2020). The performance gap is significant, which shows the power of the supervised CNNs. Similar to the previous experiments, Frei-Chen achieves very low performance on AP together with Wavelet, LoG and Roberts. The fusion achieves better results than SAR specific GR method. Thus, it appears as a promising alternative.

## 5.4 Real SAR image

### 5.4.1 Odessa, United States

Finally, we consider a TerraSAR-X image captured with staring spotlight mode from Odessa, Texas, United States. It is first denoised with SARBLF similar to previous experiments, see Figure C.8. We provide high resolution images and all its with different detectors detected soft edge maps in the appendix, see the Figures on [https://github.com/readmees/SAR\\_edge\\_benchmark](https://github.com/readmees/SAR_edge_benchmark). Besides evaluating the performance of each edge detector, we evaluate the runtime, since SAR images are of high resolution and often require powerful machines to analyse.

The results again show strong resemblance with the BSDS500-speckled. The optimal thresholds, based on the highest average F1 scores on the training set appear to work very well. Considering that the evaluated edge detection meth-

ods are not specifically for SAR, and tradition edge detectors are not supposed to perform well, it is not strange that the detectors detect a significant amount of false positives. However, the optimal thresholds for the BSDS500-speckled data appear to make a fair trade-off in preserving edge-like structures, while keeping the false positives to a minimum. The qualitative evaluation shows similar relative performance for the different edge detectors. Figure 5.2 shows that GS508 does indeed show thicker lines than the other algorithms. However, due to the high resolution, the blobs are small compared to the whole SAR image, and take in large neighbourhoods characteristics like edges, while in the small low resolution BSDS500-speckled image total neighbourhood is to small to show these blobs are forming edge like structures. Farid an Sobel find similar edges, but just as with the BSDS500-speckled, Sobel is more sensitive to noise. Canny was broadly outperformed on the training set. Also on the real SAR image it shows very high sensitivity to noise, while not giving strong responses for the true edges. Our smoothed fusion method outperformed all the other methods on the BSDS500-speckled training set. While the performance was better, it was not a mayor difference to the other edge detectors. For the SAR image however, it is very clear how the fusion method shows less false positives and good edge-like structures, even for micro edges. Therefore we can conclude, although this might be due the pre-processing Gaussian filter, that fusion methods can improve the performance significantly.

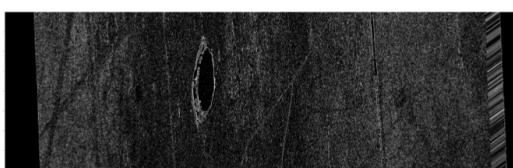
On the real SAR image, the fusion method took and Canny take around 40 seconds. Which is slightly more than Farid, Roberts, Sobel, Scharr and Prewitt, that took around 5 seconds. However, compared to the computation GS508 (11208 seconds) is a short runtime.

Figure 5.2: Odessa qualitative (cropped) comparison.

Denoised input



Farid edge map, threshold = 0.095703125



Canny, with  $\sigma=1$ , low threshold = .15 and upperthreshold=.3



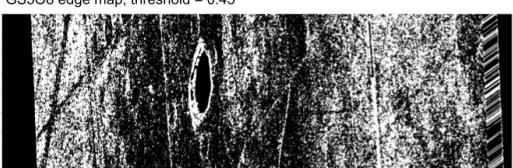
Sobel edge map, threshold = 0.115234375



Fusion Wavelet, LoG and Farid, with  $\sigma=3$  edge map, threshold = 0.2



GS5O8 edge map, threshold = 0.45



## Chapter 6

# Conclusion and Future Work

Our aim was to provide a baseline for fair evaluation of edge detection on SAR images, to answer the question: How can we provide a comprehensive and fair experimental evaluation for SAR images on edge detection? As an answer, we provided the most extensive experimental survey up to date, by providing evaluations on BSDS500-speckled (C. Liu et al., 2020). We investigated the performance of a wide variety of edge detectors and also combined them via fusion methods. The best set of parameters were selected based on the quantitative BSDS500-speckled evaluation, a benchmark was provided and two different real world remote sensing images were visually evaluated.

With the extensive experimental survey, our aim was to answer the question: How do the traditional edge detection methods perform on SAR images? The evaluations showed that the classic edge detectors do not reach high performance on simulated SAR images due to the presence of speckle noise. The qualitative evaluations on real images showed the same results. Canny appeared to not be compatible with our benchmark. However, an interesting finding is that Canny with the evaluations that were able to be evaluate on our benchmark, including the real SAR image, was mostly outperformed. The fusion methods without smoothing did not perform better than the best performing individual edge detectors. Even though further parameter optimisation and different combinations were able to improve results, we conclude that the fusion methods do not significantly improve performance, unless pre-processed with a Gaussian filter. Therefore, future work should consider extra smoothing as a fundamental step in their algorithms, and when using fusion methods weights should applied instead of simply averaging the image. As our smoothed fusion method outperformed the other evaluated method by far on the real SAR image and also, albeit with less difference, on the test and training set of BSDS500 with ROC AUC, ODS F1, OIS F1 and AP. Because of these findings future researches now know, they should consider extra smoothing

as a fundamental step in their (fusion) algorithms. Moreover, the outcome to the question: What is the effect of fusion methods on the performance? Is that it significantly improves results.

Finally, the edge detection performance on the BSDS500-speckled dataset shows a fair representation of the real SAR image as the performances of the evaluate methods were consistence with the evaluation on the real image.

## 6.1 Challenges

The Matlab scripts in the HED-BSDS dataset are able to simulate 1-look SAR images. However, for the test images, the scrips not only distribute the SAR-like noise but also resize the images. This can be dealt with by using the algorithm for the train images, by changing the `train_pair.txt`<sup>1</sup>. The train pairs of filenames for original and ground truth images should be replaced for the filenames of the test pairs. Afterwards, the produced images can be moved manually to the proper directories. Algorithms can take over 500 hours to run on a 32-core Intel(R) Xeon(R) CPU E5-2667 v2 @ 3.30GHz, 126GB RAM machine (see Section 4.5) on the BSDS500-speckled dataset. Our programs used only one core each and not the maximum around of RAM, thus multi-threading might improve computation time significantly. Therefore, we recommend taking enough time for the research.

Another challenge with the data of the BSDS500-speckled and BSDS500, is that ground truths are annotated by multiple subjects, creating ambiguity the real edges in the image and producing better comparative results for algorithms that produce thick lines than they deserve. It is however, the option for now since annotated ground truths for real SAR images are very scarce.

Most SAR based edge detection and denoising algorithms are not publicly available. Similarly, another tendency is to provide only the binary files so that it is not possible to modify the algorithm, explore different variations and adapt them. Thus, we were not able to include many SAR specific methods to our evaluations. That hinders scientific reproducibility and prevent researchers to provide comprehensive evaluations. We believe that a collaborative platform could substantially improve the field further. Since our timeline allowed mostly for exploring fundamental edge detectors we encourage researches to provide their evaluations on BSDS500-speckled test set, which we provide as a benchmark together with our evaluations at [https://github.com/readme/SAR\\_edge\\_benchmark](https://github.com/readme/SAR_edge_benchmark).

---

<sup>1</sup>This file can be found at [https://github.com/ChenguangTelecom/GRHED/blob/master/data/HED-BSDS/train\\_pair.txt](https://github.com/ChenguangTelecom/GRHED/blob/master/data/HED-BSDS/train_pair.txt).

# References

- Airouche, M., Zelmat, M., & Kidouche, M. (2008, 01). Statistical edge detectors applied to sar images. *International Journal of Computers, Communications & Control (IJCCC)*, 3.
- Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011a, May). Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5), 898-916. doi: 10.1109/TPAMI.2010.161
- Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011b, May). Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5), 898-916. Retrieved from <http://dx.doi.org/10.1109/TPAMI.2010.161> doi: 10.1109/TPAMI.2010.161
- Bachofer, F., Quénéhervé, G., Zwiener, T., Maerker, M., & Hochschild, V. (2016). Comparative analysis of edge detection techniques for sar images. *European Journal of Remote Sensing*, 49(1), 205-224. doi: 10.5721/EuJRS20164912
- Bajwa, S. G., Rupe, J. C., & Mason, J. (2017). Soybean disease monitoring with leaf reflectance. *Remote Sensing*, 9(2). Retrieved from <https://www.mdpi.com/2072-4292/9/2/127> doi: 10.3390/rs9020127
- Barbu, T. (2013). Variational image denoising approach with diffusion porous media flow. In *Abstract and applied analysis* (Vol. 2013). doi: 10.1155/2013/856876
- Bertasius, G., Shi, J., & Torresani, L. (2015). Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proceedings of the ieee conference on computer vision and pattern recognition* (p. 4380-4389). doi: 10.1109/CVPR.2015.7299067
- Boncelet, C. (2009). Chapter 7 - image noise models. In A. Bovik (Ed.), *The essential guide to image processing* (p. 143-167). Boston: Academic Press. doi: 10.1016/B978-0-12-374457-9.00007-X
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Braun, A., & Hochschild, V. (2017). Potential and limitations of radar remote sensing for humanitarian operations. In *Gi forum* (Vol. 1, p. 1228-243). doi: 10.1553/giscience2017\_01\_s228
- Buades, A., Coll, B., & Morel, J.-M. (2005). A non-local algorithm for im-

- age denoising. In *2005 ieee computer society conference on computer vision and pattern recognition (cvpr'05)* (Vol. 2, p. 60-65 vol. 2). doi: 10.1109/CVPR.2005.38
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8*(6), 679-698. doi: 10.1109/TPAMI.1986.4767851
- Chan, R. H., Ho, C.-W., & Nikolova, M. (2005). Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on image processing, 14*(10), 1479-1485. doi: 10.1109/TIP.2005.852196
- Chen, L., Zhu, Q., Xie, X., Hu, H., & Zeng, H. (2018). Road extraction from vhr remote-sensing imagery via object segmentation constrained by gabor features. *ISPRS International Journal of Geo-Information, 7*(9). Retrieved from <https://www.mdpi.com/2220-9964/7/9/362> doi: 10.3390/ijgi7090362
- Chierchia, G., El Gheche, M., Scarpa, G., & Verdoliva, L. (2017). Multitemporal sar image despeckling based on block-matching and collaborative filtering. *IEEE Transactions on Geoscience and Remote Sensing, 55*(10), 5467-5480. doi: 10.1109/TGRS.2017.2707806
- Cozzolino, D., Verdoliva, L., Scarpa, G., & Poggi, G. (2020). Nonlocal cnn sar image despeckling. *Remote Sensing, 12*(6), 1006. doi: 10.3390/rs12061006
- Cuevas, E., Díaz-Cortes, M.-A., & Mezura-Montes, E. (2019). Corner detection of intensity images with cellular neural networks (cnn) and evolutionary techniques. *Neurocomputing, 347*, 82-93. doi: 10.1016/j.neucom.2019.03.014
- Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing, 16*(8), 2080-2095. doi: 10.1109/TIP.2007.901238
- Delgado Blasco, J. M., Fitzyk, M., Patruno, J., Ruiz-Armenteros, A. M., & Marconcini, M. (2020). Effects on the double bounce detection in urban areas based on sar polarimetric characteristics. *Remote Sensing, 12*(7), 1187. doi: 10.3390/rs12071187
- Dellinger, F., Delon, J., Gousseau, Y., Michel, J., & Tupin, F. (2012). Sar-sift: A sift-like algorithm for applications on sar images. *IEEE Transactions on Geoscience and Remote Sensing, 50*, 3478-3481. doi: 10.1109/TGRS.2012.6350671
- D'Hondt, O., Guillaso, S., & Hellwich, O. (2013). Iterative bilateral filtering of polarimetric sar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6*(3), 1628-1639. doi: 10.1109/JSTARS.2013.2256881
- Duda, R., & Hart, P. (1973). *Pattern classification and scene analysis*. New York:

- John Wiley & Sons.
- D'Hondt, O., López-Martínez, C., Guillaso, S., & Hellwich, O. (2018). Nonlocal filtering applied to 3-d reconstruction of tomographic sar data. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1), 272-285. doi: 10.1109/TGRS.2017.2746420
- El-Sayed, M. A., & Sennari, H. (2014). Convolutional neural network for edge detection in sar grayscale images. *IOSR journal of VLSI and Signal Processing*, 4, 75-83.
- Eltanany, A. S., Amein, A. S., & Elwan, M. S. (2021). A modified corner detector for sar images registration. In *International journal of engineering research in africa* (Vol. 53, pp. 123–156). doi: 10.4028/www.scientific.net/JERA.53.123
- Fjørtoft, R., Lopes, A., Marthon, P., & Cubero-Castan, E. (1998). An optimal multiedge detector for sar image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 36(3), 793-802. doi: 10.1109/36.673672
- Ganugapati, S., & Moloney, C. (1995). A ratio edge detector for speckled images based on maximum strength edge pruning. , 2, 165-168 vol.2. doi: 10.1109/ICIP.1995.537440
- Goyal, B., Dogra, A., Agrawal, S., Sohi, B. S., & Sharma, A. (2020). Image denoising review: From classical to state-of-the-art approaches. *Information fusion*, 55, 220-244. doi: 10.1016/j.inffus.2019.09.003
- Gu, S., & Timofte, R. (2019). A brief review of image denoising algorithms and beyond. *Inpainting and Denoising Challenges*, 1-21. doi: 10.1007/978-3-030-25614-2\_1
- Gupta, A., Tripathi, A., & Bhateja, V. (2013). Despeckling of sar images via an improved anisotropic diffusion algorithm. In *Proceedings of the international conference on frontiers of intelligent computing: Theory and applications (ficta)* (p. 747-754). doi: 10.1007/978-3-642-35314-7\_85
- Herndon, K., Meyer, F., Flores, A., Cherrington, E., & Kucera, L. (2020). *What is synthetic aperture radar?* NASA. Retrieved 2021-06-13, from <https://earthdata.nasa.gov/learn/backgrounders/what-is-sar>
- Hussain, E., & Shan, J. (2010). Mapping major floods with optical and sar satellite images. In *Ieee geoscience and remote sensing symposium, honolulu, hawaii, usa, july* (p. 25-30).
- Hwang, J.-J., & Liu, T.-L. (2015). Pixel-wise deep learning for contour detection. *arXiv preprint arXiv:1504.01989*.
- Insight, O. (2017). *It's always sunny at orbital insight: Our work with sar imagery.* From the Macroscopic. Retrieved 2021-06-16, from <https://medium.com/from-the-macroscope/its-always-sunny-at-orbital-insight-our-work-with-sar-imagery-9907c850fe45>
- Jain, P., & Tyagi, V. (2016). A survey of edge-preserving image denoising methods.

- Information Systems Frontiers*, 18(1), 159-170. doi: 10.1007/s10796-014-9527-0
- Jha, M. N., Levy, J., & Gao, Y. (2008). Advances in remote sensing for oil spill disaster management: State-of-the-art sensors technology for oil spill surveillance. *Sensors*, 8(1), 236-255. Retrieved from <https://www.mdpi.com/1424-8220/8/1/236> doi: 10.3390/s8010236
- Jia, C.-L., Kuang, G., & Su, Y. (2004). Road extraction from the high-resolution sar imagery using hough transform. , 26, 50-55.
- Jing, W., Jin, T., & Xiang, D. (2020). Sar image edge detection with recurrent guidance filter. *IEEE Geoscience and Remote Sensing Letters*. doi: 10.1109/LGRS.2020.2990688
- Jones, E., Oliphant, T., Peterson, P., et al. (2001–). *SciPy: Open source scientific tools for Python*. Retrieved 2021-06-23, from <http://www.scipy.org/>
- Kalbasi, M., & Nikmehr, H. (2020). Noise-robust, reconfigurable canny edge detection and its hardware realization. *IEEE Access*, 8, 39934-39945. doi: 10.1109/ACCESS.2020.2976860
- Kang, X., Han, C., Yang, Y., & Tao, T. (2006). Sar image edge detection by ratio-based harris method. In *2006 ieee international conference on acoustics speech and signal processing proceedings* (Vol. 2, p. II-II). doi: 10.1109/ICASSP.2006.1660473
- Khan, S., Khan, A., Maqsood, M., Aadil, F., & Ghazanfar, M. A. (2019). Optimized gabor feature extraction for mass classification using cuckoo search for big data e-healthcare. *Journal of Grid Computing*, 17(2), 239-254. doi: 10.1007/s10723-018-9459-x
- Kumar, S., Singh, M., & Shaw, D. (2016). Comparative analysis of various edge detection techniques in biometric application. *International Journal of Engineering and Technology (IJET)*, 8(6), 2452-2459. doi: 10.21817/ijet/2016/v8i6/160806409
- Lattari, F., Gonzalez Leon, B., Asaro, F., Rucci, A., Prati, C., & Matteucci, M. (2019). Deep learning for sar image despeckling. *Remote Sensing*, 11(13), 1532. doi: 10.3390/rs11131532
- Lee, G. R., Gommers, R., Waselewski, F., Wohlfahrt, K., & O'Leary, A. (2019). Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36), 1237. doi: 10.21105/joss.01237
- Ley, A., D'Hondt, O., & Hellwich, O. (2018). Regularization and completion of tomosar point clouds in a projected height map domain. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(6), 2104-2114. doi: 10.1109/JSTARS.2018.2814219
- Li, H. (2014). Deep learning for image denoising. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 7(3), 171-180. doi:

10.14257/ijcip.2014.7.3.14

- Li, H., Chen, Y., Yang, Y., Liu, P., & Zhong, C. (2019). Deep learning road extraction model based on similarity mapping relationship. In *Igarss 2019-2019 ieee international geoscience and remote sensing symposium* (p. 9799-9802). doi: 10.1109/IGARSS.2019.8898155
- Li, Y., Zhang, R., & Wu, Y. (2017). Road network extraction in high-resolution sar images based cnn features. In *2017 ieee international geoscience and remote sensing symposium (igarss)* (pp. 1664–1667). doi: 10.1109/IGARSS.2017.8127293
- Liu, C., Tupin, F., & Gousseau, Y. (2020). Training cnns on speckled optical dataset for edge detection in sar images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 170, 88-102.
- Liu, G., Yang, W., Xia, G.-S., & Liao, M.-S. (2013). Structure preserving sar image despeckling via l0-minimization. *Progress In Electromagnetics Research*, 1-22.
- Liu, Y., Cheng, M.-M., Hu, X., Bian, J.-W., Zhang, L., Bai, X., & Tang, J. (2019). Richer convolutional features for edge detection. In (Vol. 41, p. 1939-1946). doi: 10.1109/TPAMI.2018.2878849
- Lu, L., Zhang, G., Nie, Y., Liu, J., Zhang, G., Wu, Y., et al. (2021). Application of improved cnn in sar image noise reduction. In *Journal of physics: Conference series* (Vol. 1792, p. 012053). doi: 10.1088/1742-6596/1792/1/012053
- Luo, Y., An, D., Wang, W., & Huang, X. (2020). Improved roewa sar image edge detector based on curvilinear structures extraction. *IEEE Geoscience and Remote Sensing Letters*, 17(4), 631-635. doi: 10.1109/LGRS.2019.2926428
- Ma, X., Li, B., Zhang, Y., & Yan, M. (2012). The canny edge detection and its improvement. In *International conference on artificial intelligence and computational intelligence* (p. 50-58). doi: 10.1007/978-3-642-33478-8\_7
- Maithree, H., Dinushka, D., & Wijayasiri, A. (2020). Change detection using synthetic aperture radar videos. *arXiv preprint arXiv:2007.14001*.
- Mäkinen, Y., Azzari, L., & Foi, A. (2020). Collaborative filtering of correlated noise: Exact transform-domain variance for improved shrinkage and patch matching. *IEEE Transactions on Image Processing*, 29, 8339–8354.
- Marr, D., & Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167), 187-217. doi: 10.1098/rspb.1980.0020
- Martinis, S., & Rieke, C. (2015). Backscatter analysis using multi-temporal and multi-frequency sar data in the context of flood mapping at river saale, germany. *Remote Sensing*, 7(6), 7732-7752. doi: 10.3390/rs70607732
- medpy.filter.smoothing.anisotropic\_diffusion.* (n.d.). Retrieved 2021-06-23, from <https://loli.github.io/medpy/generated/medpy.filter.smoothing>

- .anisotropic\_diffusion.html
- Melnyk, D., Wang, Y., & Wattenhofer, R. (2018). Byzantine preferential voting. In *International conference on web and internet economics* (p. 327-340). doi: 10.1007/978-3-030-04612-5\_22
- Molini, A. B., Valsesia, D., Fracastoro, G., & Magli, E. (2021). Speckle2void: Deep self-supervised sar despeckling with blind-spot convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*. doi: 10.1109/TGRS.2021.3065461
- Moreira, A., Prats-Iraola, P., Younis, M., Krieger, G., Hajnsek, I., & Papathanassiou, K. P. (2013). A tutorial on synthetic aperture radar. *IEEE Geoscience and remote sensing magazine*, 1(1), 6-43. doi: 10.1109/MGRS.2013.2248301
- Mäkinen, Y., Azzari, L., & Foi, A. (n.d.). *bm3d*. Retrieved 2021-06-23, from <https://pypi.org/project/bm3d/>
- Nava, L., Catani, F., & Monserrat, O. (2021). Rapid mapping of landslides by deep-learning of combined optical and sar data. In *Egu general assembly conference abstracts* (p. EGU21-13181).
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.
- Ouchi, K. (1985). On the multilook images of moving targets by synthetic aperture radars. *IEEE Transactions on Antennas and Propagation*, 33(8), 823-827. doi: 10.1109/TAP.1985.1143684
- Overview. (2020). NASA. Retrieved 2021-06-14, from <https://nisar.jpl.nasa.gov/mission/get-to-know-sar/overview/>
- Pan, B. (2013). Bias error reduction of digital image correlation using gaussian pre-filtering. *Optics and Lasers in Engineering*, 51(10), 1161-1167. doi: 10.1016/j.optlaseng.2013.04.009
- Pan, Q., Zhang, L., Dai, G., & Zhang, H. (1999). Two denoising methods by wavelet transform. *IEEE transactions on signal processing*, 47(12), 3401-3406. doi: 10.1109/78.806084
- Parrilli, S., Poderico, M., Angelino, C. V., & Verdoliva, L. (2012). A non-local sar image denoising algorithm based on llmmse wavelet shrinkage. *IEEE Transactions on Geoscience and Remote Sensing*, 50(2), 606-616. doi: 10.1109/TGRS.2011.2161586
- Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), 629-639. doi: 10.1109/34.56205
- Prewitt, J. M. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1), 15-19.
- Ranjani, J., Gokila, M., & Thiruvengadam, S. (2008). Edge detection in

- speckled sar images with improved roewa. In *2008 sixth indian conference on computer vision, graphics & image processing* (p. 644-649). doi: 10.1109/ICVGIP.2008.46
- Rao, M. (2020). *Press release: Opt/net granted esa funding to tackle illegal fishing!* OPT/NET B.V. Retrieved 2021-06-16, from <https://medium.com/optoss/press-release-opt-net-granted-esa-funding-to-tackle-illegal-fishing-27245398488c>
- Rao, M. (2021). *Opt/net's tsar ai for flood detection.* OPT/NET B.V. Retrieved 2021-06-16, from <https://medium.com/optoss/opt-nets-tsar-ai-for-flood-detection-6428102603c1>
- Roberts, L. G. (1965). Machine perception of three-dimensional solids, in optical and electro-optical. *formation Processing*. Retrieved 2021-06-16, from <https://ci.nii.ac.jp/naid/10006714761/en/>
- Ruta, D., & Gabrys, B. (2000). An overview of classifier fusion methods. *Computing and Information systems*, 7(1), 1-10.
- Saevarsson, B., Sveinsson, J., & Benediktsson, J. (2004). Combined wavelet and curvelet denoising of sar images. In *Igarss 2004. 2004 ieee international geoscience and remote sensing symposium* (Vol. 6, p. 4235-4238 vol.6). doi: 10.1109/IGARSS.2004.1370070
- Saif, J. A. (2017). Tuning of canny image edge detection. *International Journal of Computer Science and Information Security (IJCSIS)*, 15(11).
- Salehi, H., Vahidi, J., Abdeljawad, T., Khan, A., & Rad, S. Y. B. (2020). A sar image despeckling method based on an extended adaptive wiener filter and extended guided filter. *Remote Sensing*, 12(15), 2371. doi: 10.3390/rs12152371
- Schou, J., Skriver, H., Nielsen, A., & Conradsen, K. (2003). Cfar edge detector for polarimetric sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 41(1), 20-32. doi: 10.1109/TGRS.2002.808063
- Sharma, R., Kumar, B., Desai, N., & Gujraty, V. (2008). Sar for disaster management. *IEEE Aerospace and Electronic Systems Magazine*, 23(6), 4-9. doi: 10.1109/MAES.2008.4558001
- Shen, W., Wang, X., Wang, Y., Bai, X., & Zhang, Z. (2015). Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of the ieee conference on computer vision and pattern recognition* (p. 3982-3991). doi: 10.1109/CVPR.2015.7299024
- Sica, F., Cozzolino, D., Zhu, X. X., Verdoliva, L., & Poggi, G. (2018). Insar-bm3d: A nonlocal filter for sar interferometric phase restoration. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6), 3456-3467. doi: 10.1109/TGRS.2018.2800087
- Simard, M., DeGrandi, G., Thomson, K., & Benie, G. (1998). Analysis of

- speckle noise contribution on wavelet decomposition of sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 36(6), 1953-1962. doi: 10.1109/36.729367
- Simonovic, S. P., et al. (2002). *Role of remote sensing in disaster management*. Department of Civil and Environmental Engineering, The University of Western Ontario.
- Singh, P., Diwakar, M., Shankar, A., Shree, R., & Kumar, M. (2021). A review on sar image and its despeckling. *Archives of Computational Methods in Engineering*, 1-21. doi: 10.1007/s11831-021-09548-z
- Singh, P., & Shree, R. (2016). Analysis and effects of speckle noise in sar images. In *2016 2nd international conference on advances in computing, communication, automation (icacca) (fall)* (p. 1-5). doi: 10.1109/ICACCAF.2016.7748978
- Soldal, I. H., Dierking, W., Korosov, A., & Marino, A. (2019). Automatic detection of small icebergs in fast ice using satellite wide-swath sar images. *Remote Sensing*, 11(7), 806. doi: 10.3390/rs11070806
- Spontón, H., & Cardelino, J. (2015). A review of classic edge detectors. *Image Processing On Line*, 5, 90-123. doi: 10.5201/ipol.2015.35
- Sun, N., Zhang, J., Huang, G., Zhao, Z., & Lu, L. (2014). Review of road extraction methods from sar image. In *Iop conference series: Earth and environmental science* (Vol. 17, p. 012245). doi: 10.1088/1755-1315/17/1/012245
- Thipkham, P. (2018). *Image processing class (egbe443) #5-edge and contour. Towards Data Science*. Retrieved 2021-06-13, from <https://towardsdatascience.com/image-processing-class-egbe443-5-edge-and-contour-d5d410f4483c>
- Thorp, K., & Tian, L. (2004). A review on remote sensing of weeds in agriculture. *Precision Agriculture*, 5(5), 477-508. doi: 10.1007/s11119-004-5321-1
- Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (ieee cat. no.98ch36271)* (p. 839-846). doi: 10.1109/ICCV.1998.710815
- Touzi, R., Lopes, A., & Bousquet, P. (1988). A statistical and geometrical edge detector for sar images. *IEEE Transactions on geoscience and remote sensing*, 26(6), 764-773.
- Tseng, L. (2017). Voting in the presence of byzantine faults. In *2017 ieee 22nd pacific rim international symposium on dependable computing (prdc)* (p. 1-10). doi: 10.1109/PRDC.2017.11
- Umesh. (n.d.). *Frei-chen edge detector*. Retrieved 2021-06-23, from <https://mail.python.org/pipermail//scikit-image/2013-April/002012.html>
- van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., ... the scikit-image contributors (2014, 6). scikit-image:

- image processing in Python. *PeerJ*, 2, e453. doi: 10.7717/peerj.453
- Wang, G., Lopez-Molina, C., & De Baets, B. (2020). Automated blob detection using iterative laplacian of gaussian filtering and unilateral second-order gaussian kernels. *Digital Signal Processing*, 96, 102592. doi: 10.1016/j.dsp.2019.102592
- Wang, M., Gao, L., Huang, X., Jiang, Y., & Gao, X. (2019). A texture classification approach based on the integrated optimization for parameters and features of gabor filter via hybrid ant lion optimizer. *Applied Sciences*, 9(11), 2173.
- Wei, Q.-R., & Feng, D.-Z. (2015). An efficient sar edge detector with a lower false positive rate. *International Journal of Remote Sensing*, 36(14), 3773-3797. doi: 10.1080/01431161.2015.1054046
- Wei, Q.-R., & Feng, D.-Z. (2018). Antistretch edge detector for sar images. *IEEE Geoscience and Remote Sensing Letters*, 15(9), 1382-1386. doi: 10.1109/LGRS.2018.2840161
- Wei, Q.-R., Feng, D.-Z., Zheng, W., & Zheng, J.-B. (2017). Rapid line-extraction method for sar images based on edge-field features. *IEEE Geoscience and Remote Sensing Letters*, 14(10), 1865-1869. doi: 10.1109/LGRS.2017.2738703
- Wei, Y., Wang, Z., & Xu, M. (2017). Road structure refined cnn for road extraction in aerial image. *IEEE Geoscience and Remote Sensing Letters*, 14(5), 709-713.
- Wójtowicz, M., Wójtowicz, A., Piekarczyk, J., et al. (2016). Application of remote sensing methods in agriculture. *Communications in Biometry and Crop Science*, 11(1), 31-50.
- Wu, B., Zhou, S., & Ji, K. (2016). A novel method of corner detector for sar images based on bilateral filter. In *2016 ieee international geoscience and remote sensing symposium (igarss)* (p. 2734-2737). doi: 10.1109/IGARSS.2016.7729706
- Xiang, Y., Wang, F., Wan, L., & You, H. (2017a). An advanced multiscale edge detector based on gabor filters for sar imagery. *IEEE Geoscience and Remote Sensing Letters*, 14(9), 1522-1526. doi: 10.1109/LGRS.2017.2720684
- Xiang, Y., Wang, F., Wan, L., & You, H. (2017b). Sar-pc: Edge detection in sar images via an advanced phase congruency model. *Remote Sensing*, 9(3), 209. doi: 10.3390/rs9030209
- Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *Proceedings of ieee international conference on computer vision*. doi: 10.1109/ICCV.2015.164
- Xin, J., Zhang, X., Zhang, Z., & Fang, W. (2019). Road extraction of high-resolution remote sensing images derived from denseunet. *Remote Sensing*, 11(21), 2499. doi: 10.3390/rs11212499
- Xu, Y., Wu, T., Gao, F., Charlton, J. R., & Bennett, K. M. (2020). Improved small blob detection in 3d images using jointly constrained deep learning and

- hessian analysis. *Scientific reports*, 10(1), 1-12.
- Yu, Q., Zhou, S., Jiang, Y., Wu, P., & Xu, Y. (2019). High-performance sar image matching using improved sift framework based on rolling guidance filter and roewa-powered feature. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(3), 920-933.
- Zhan, Y., You, H., & Fuqing, C. (2013). Bayesian edge detector for sar imagery using discontinuity-adaptive markov random field modeling. *Chinese Journal of Aeronautics*, 26(6), 1534-1543. doi: 10.1016/j.cja.2013.04.059
- Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7), 3142-3155. doi: 10.1109/TIP.2017.2662206
- Zhang, M., Qiao, B., Xin, M., & Zhang, B. (2021). Phase spectrum based automatic ship detection in synthetic aperture radar images. *Journal of Ocean Engineering and Science*, 6(2), 185-195. doi: 10.1016/j.joes.2020.09.002
- Zhang, Q., Kong, Q., Zhang, C., You, S., Wei, H., Sun, R., & Li, L. (2019). A new road extraction method using sentinel-1 sar images based on the deep fully convolutional neural network. *European Journal of Remote Sensing*, 52(1), 572-582. doi: 10.1080/22797254.2019.1694447
- Zhao, H., Wang, Q., Wu, W., Wang, Q., & Yuan, N. (2014). Sar image despeckling based on improved non-local means algorithm. In *2014 international conference on electromagnetics in advanced applications (iceaa)* (p. 844-847). doi: 10.1109/ICEAA.2014.6903978
- Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. (1997). Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4), 550–560.

# Appendices

## Appendix A

### Training - Qualitative Evaluation

Figure A.1: Qualitative evaluation of the denoising methods.

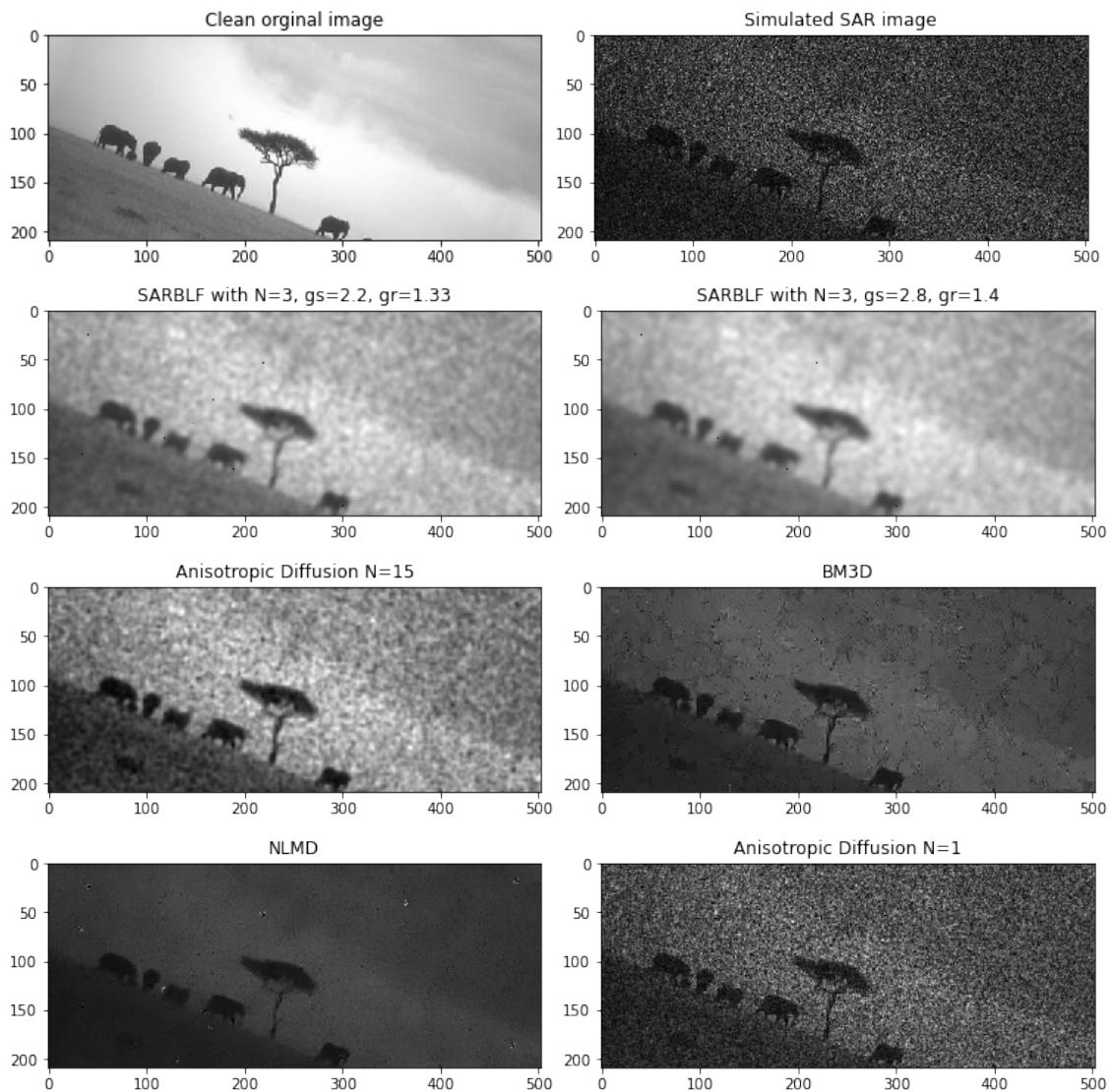


Figure A.2: Qualitative evaluation of the denoising methods (2).

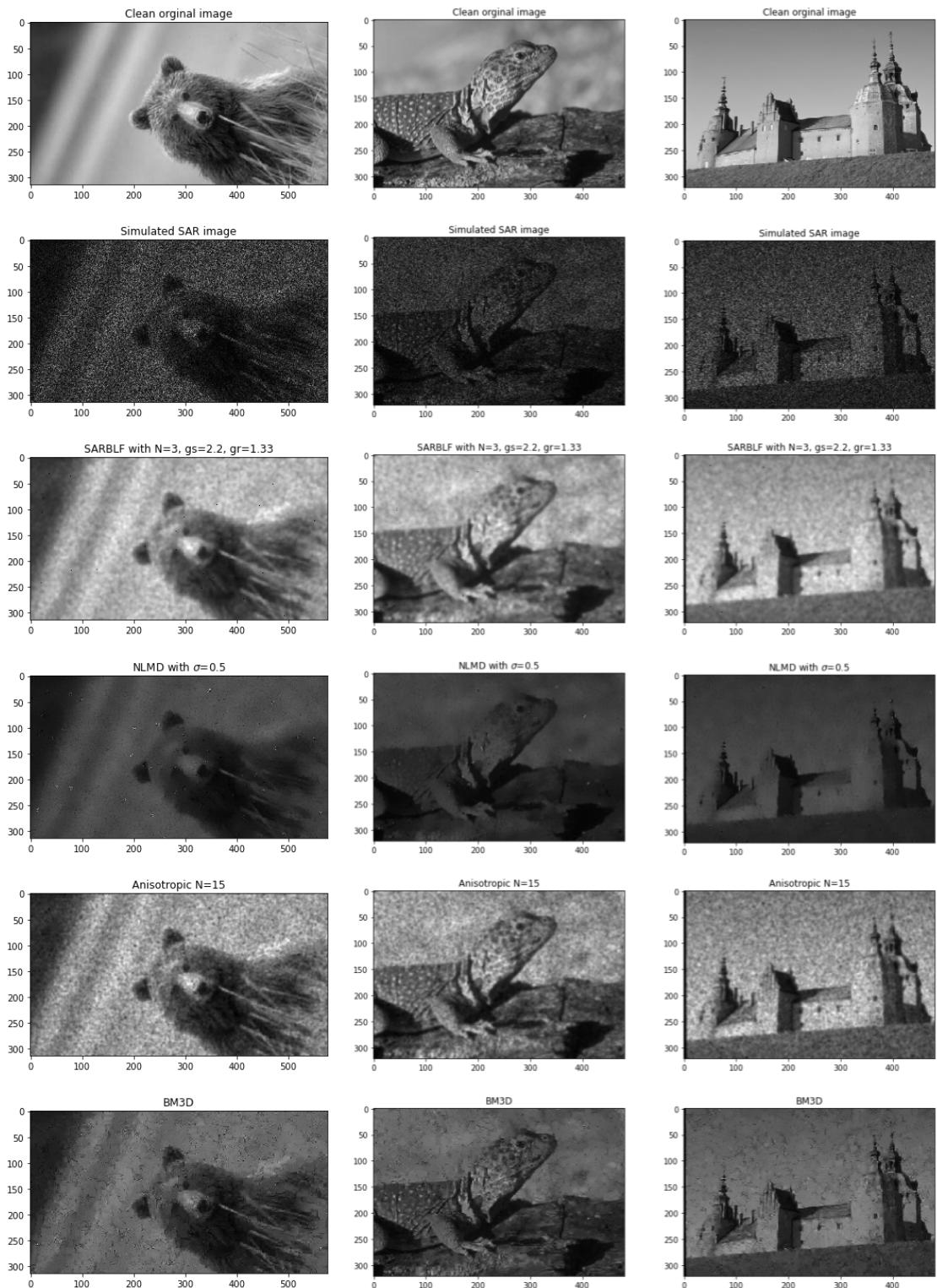


Figure A.3: Qualitative evaluation of the low level algorithms.  
Thresholds are based on the best F1 score.

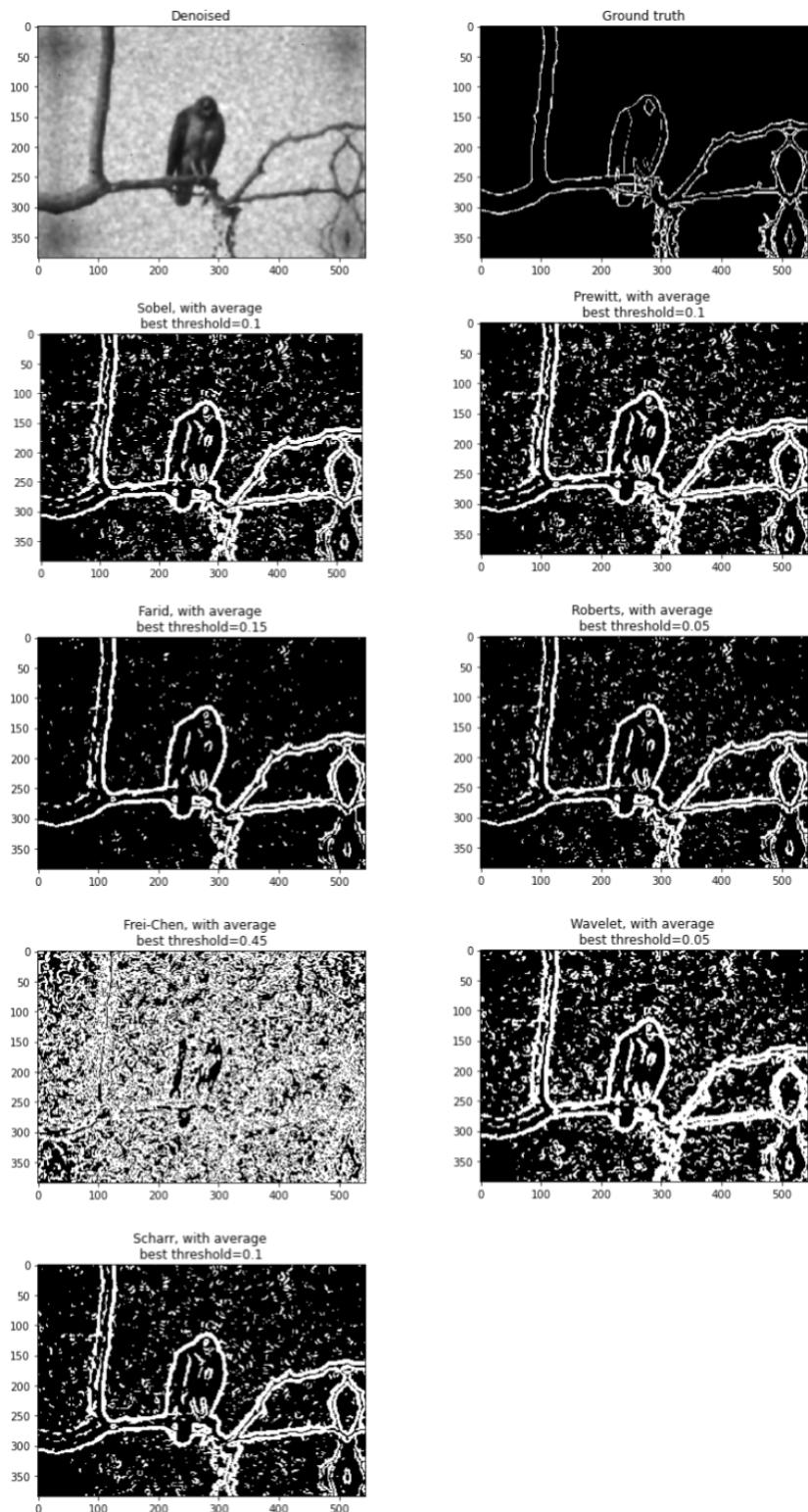


Figure A.4: Qualitative evaluation of the Gabor filterbanks.  
Thresholds are based on the best F1 score.

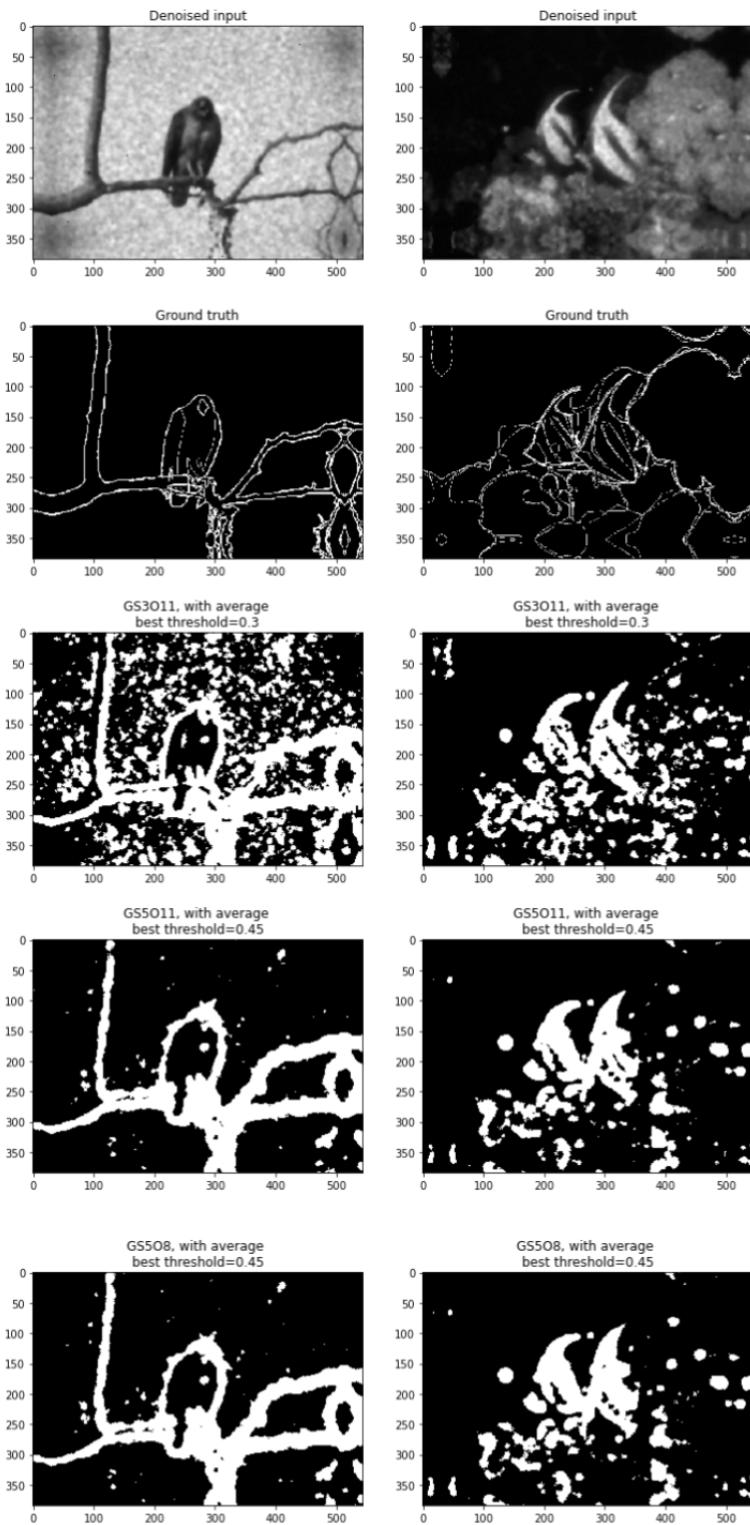


Figure A.5: Qualitative evaluation on LoG with different values of  $\sigma$ . Every edge map is the soft (non-normalised) edge map.

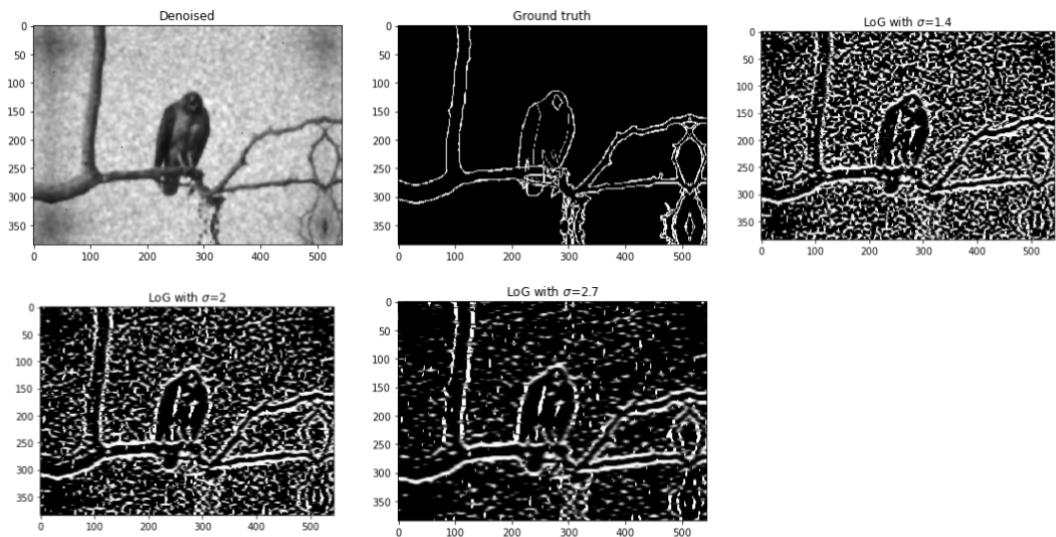


Figure A.6: Qualitative evaluation on GR with different values of  $\sigma$ .  
 Thresholds are based on the best F1 score.

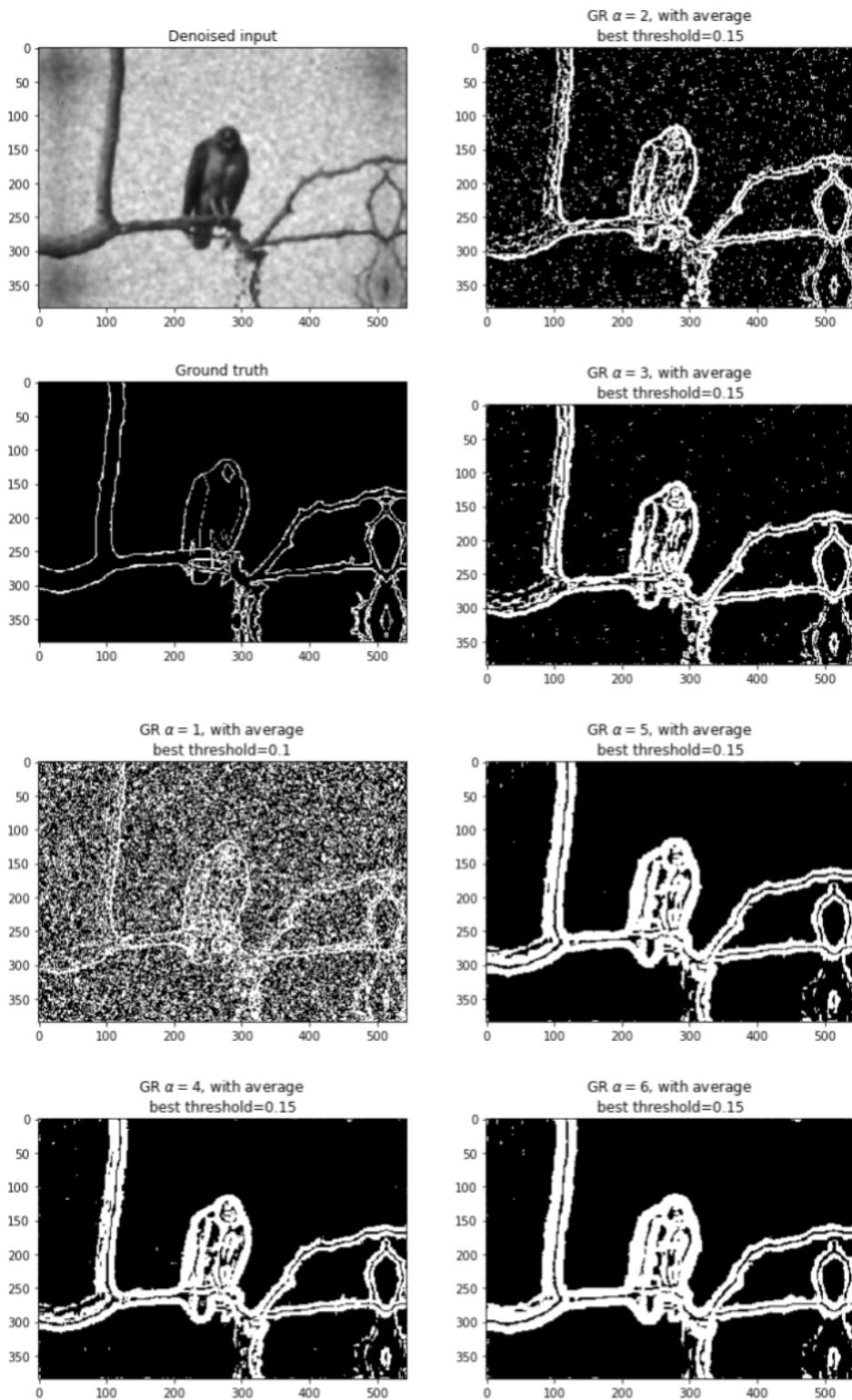


Figure A.7: Qualitative evaluation on Canny with different values of  $\sigma$ . Threshold is the same for different  $\sigma$ .

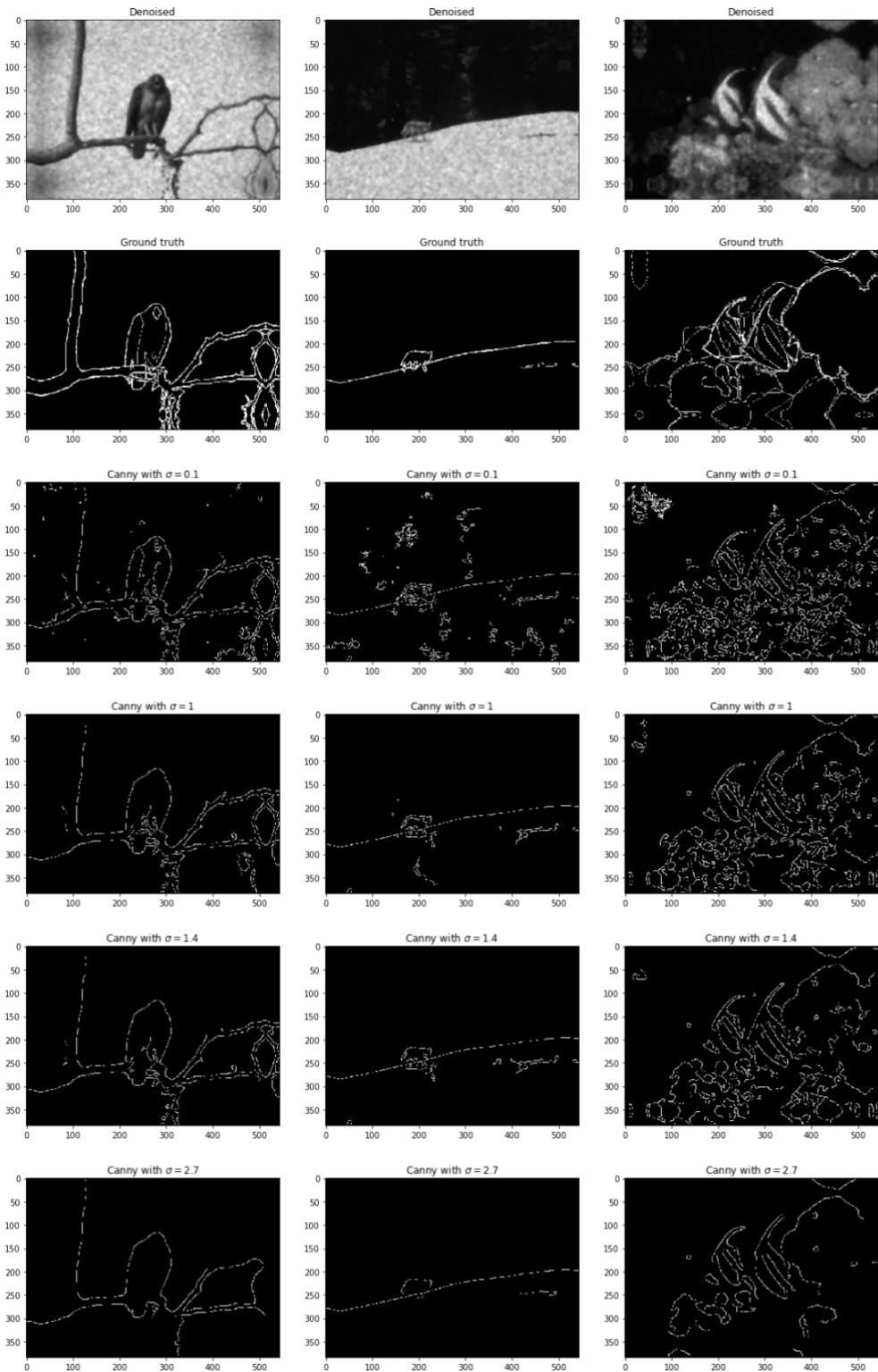


Figure A.8: Qualitative evaluation on Canny with parameters and thresholds for optimal average values for ACC, F1, PPV, FM and MK 1.

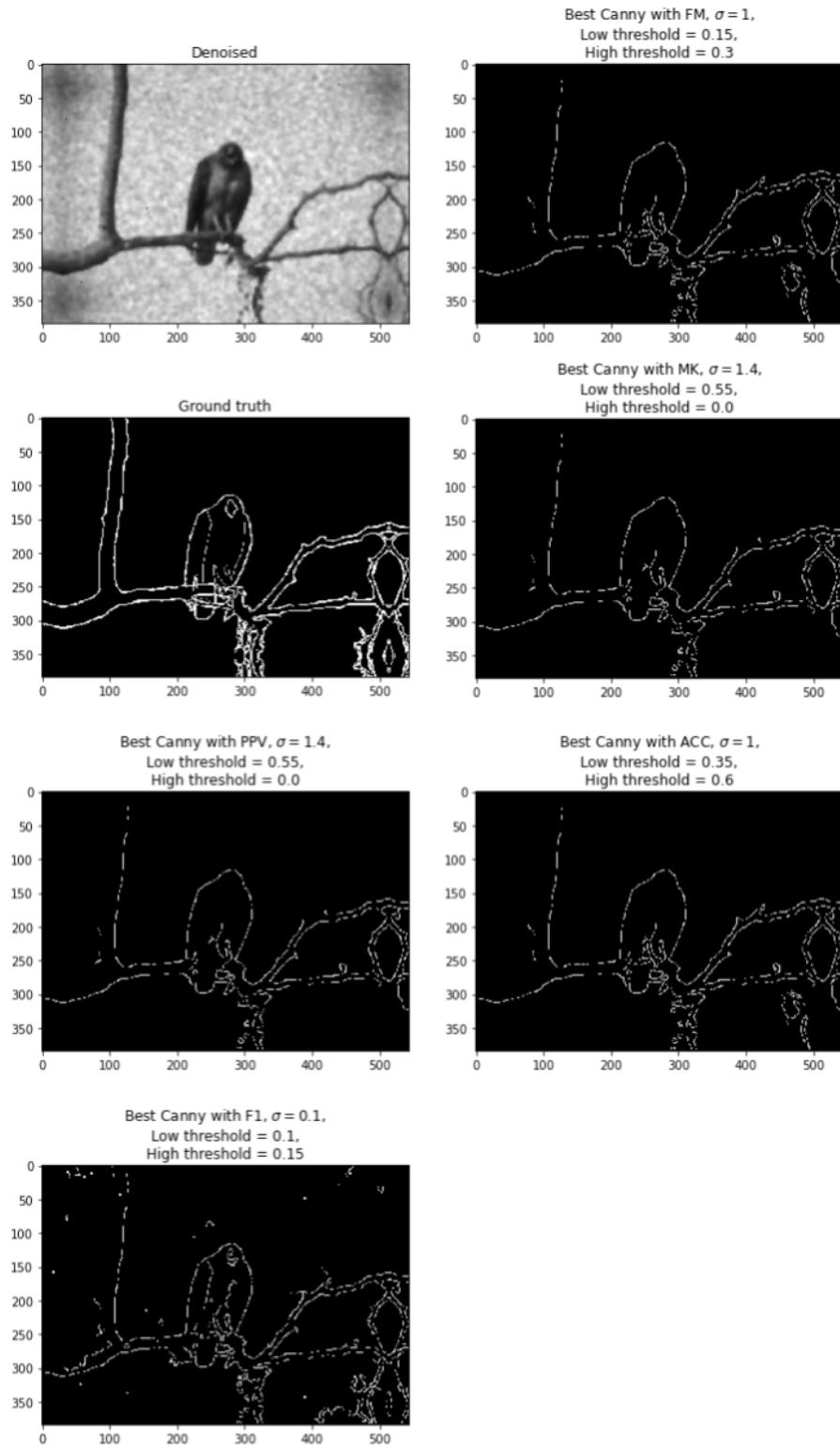


Figure A.9: Qualitative evaluation on Canny with parameters and thresholds for optimal average values for ACC, F1, PPV, FM and MK 2.

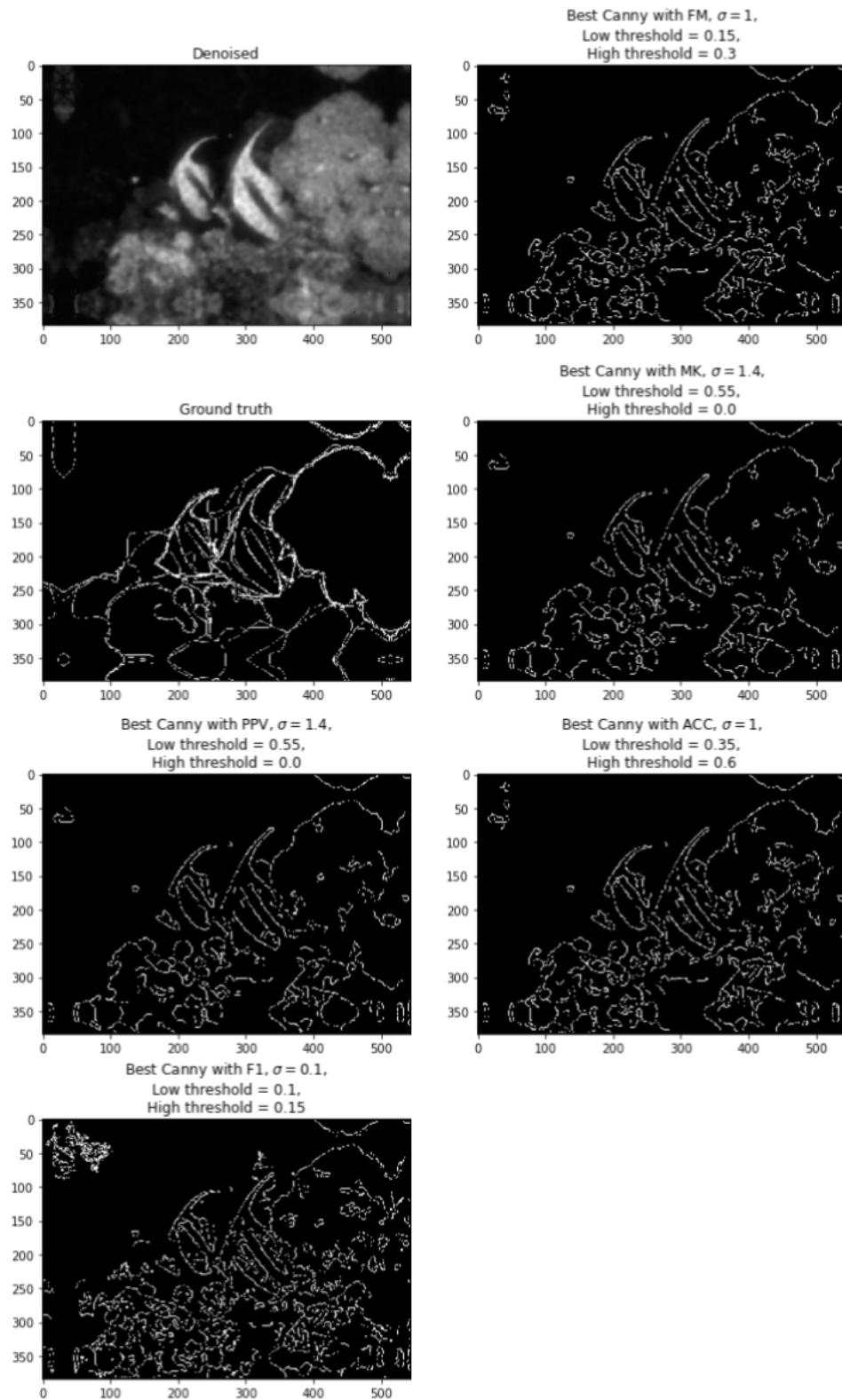


Figure A.10: Qualitative evaluation on K-means with different numbers of clusters. Every edge map is the soft (non-normalised) edge map.

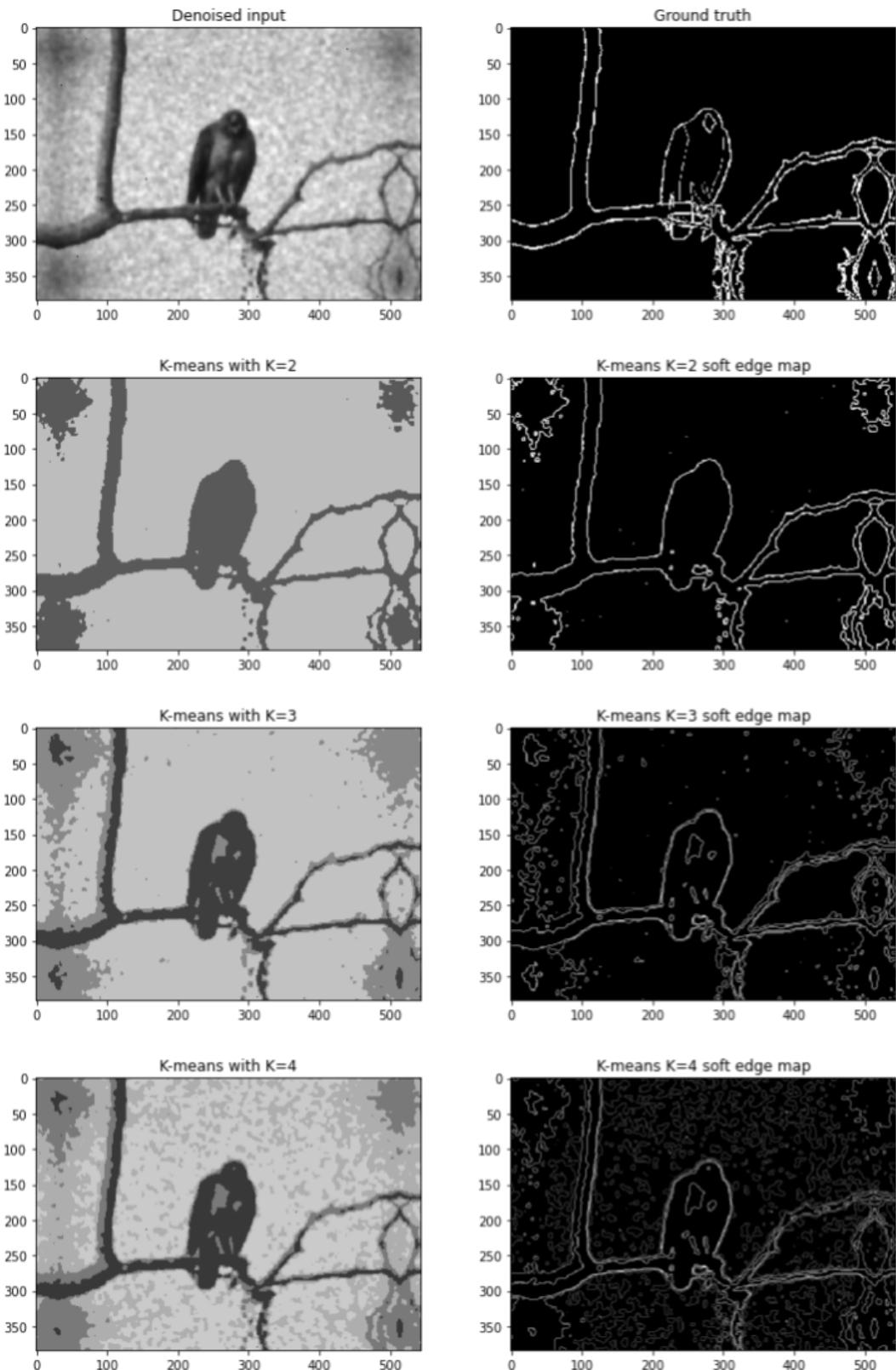
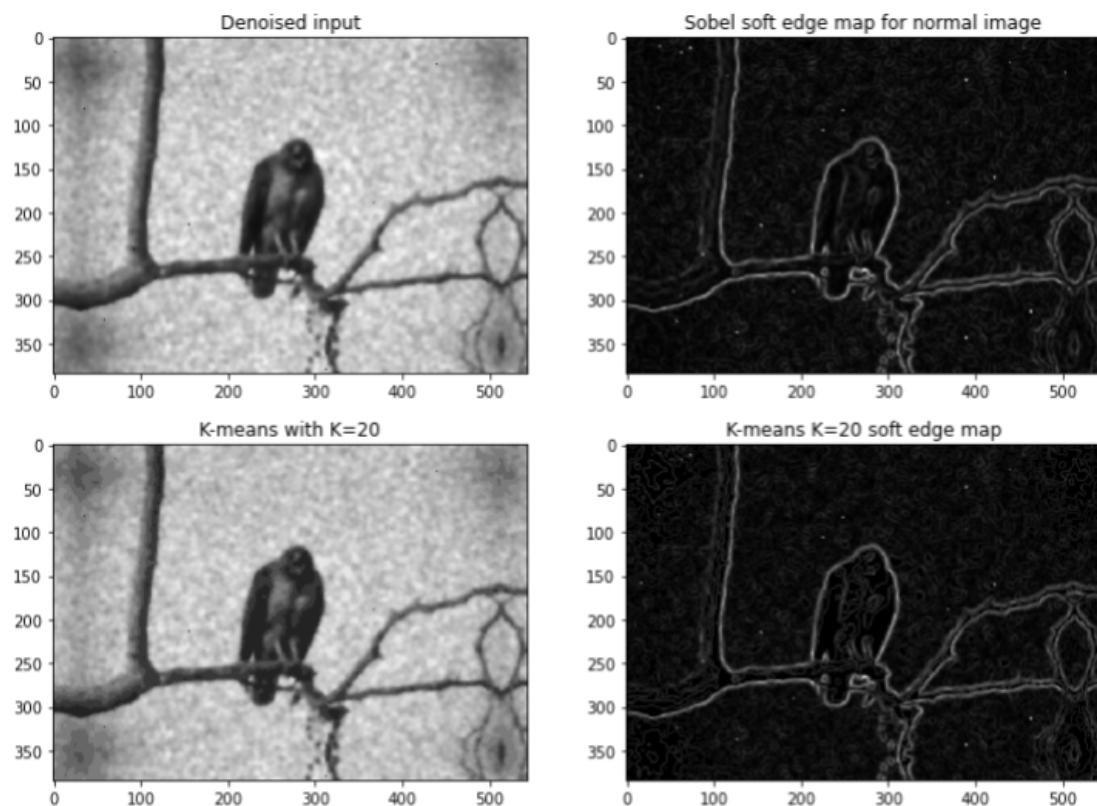


Figure A.11: Qualitative evaluation on K-means with high numbers of clusters. Every edge map is the soft (non-normalised) edge map.



# Appendix B

## Training - ROC curves

Figure B.1: ROC curves of the low level methods.

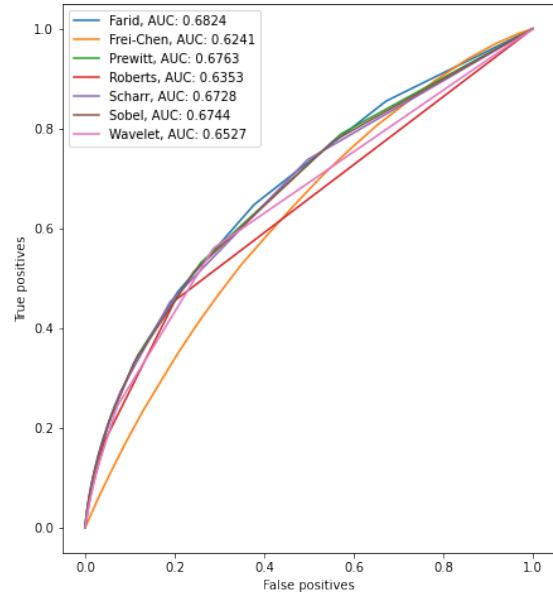


Figure B.2: ROC curves of LoG and K-means.

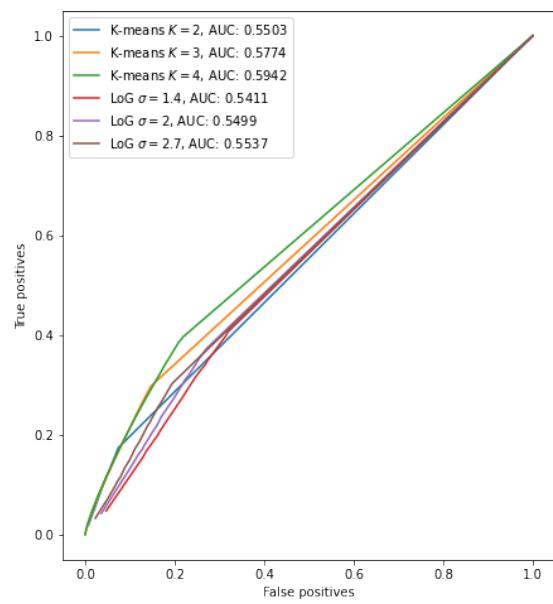


Figure B.3: ROC curves of Gabor filters.

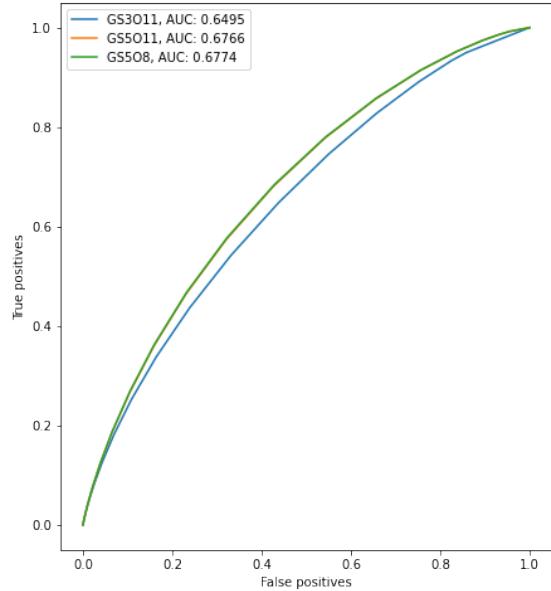


Figure B.4: ROC curves of Canny with  $\sigma=1$ .

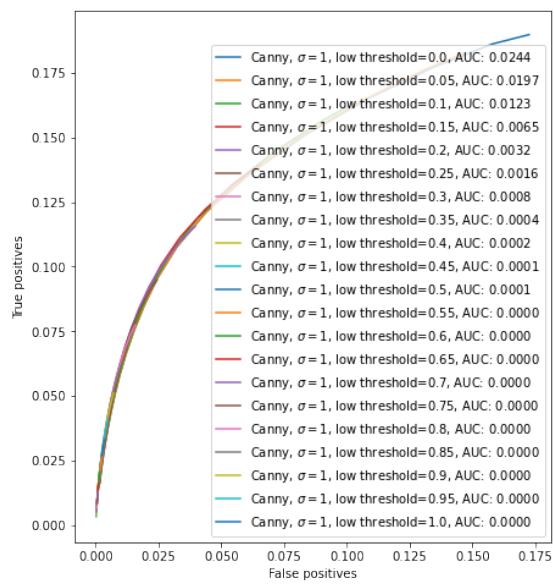


Figure B.5: ROC curves of Canny with  $\sigma=1.4$ .

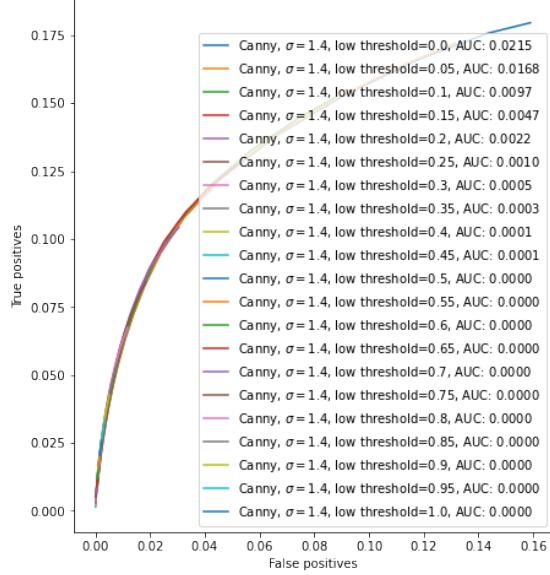


Figure B.6: ROC curves of Canny with  $\sigma=2.7$ .

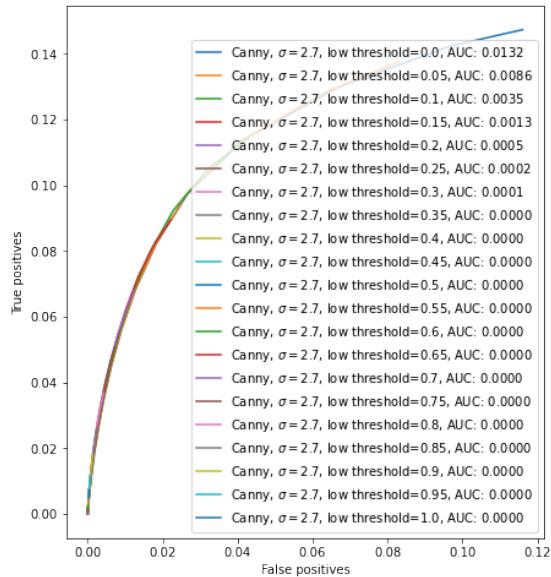


Figure B.7: ROC curves of GR.

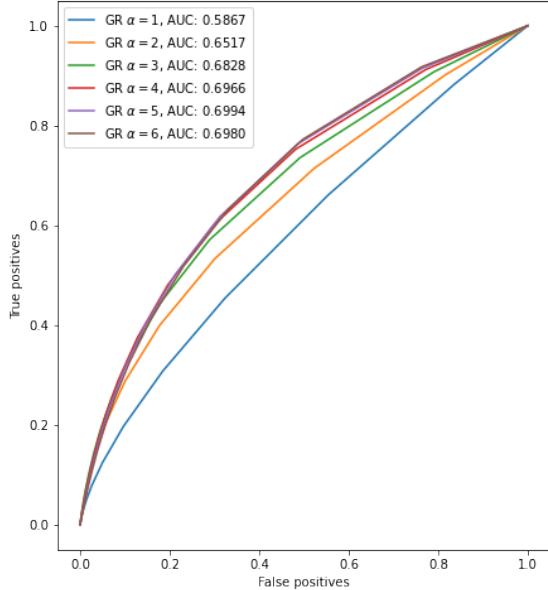


Figure B.8: ROC curves of averaged fusion methods.

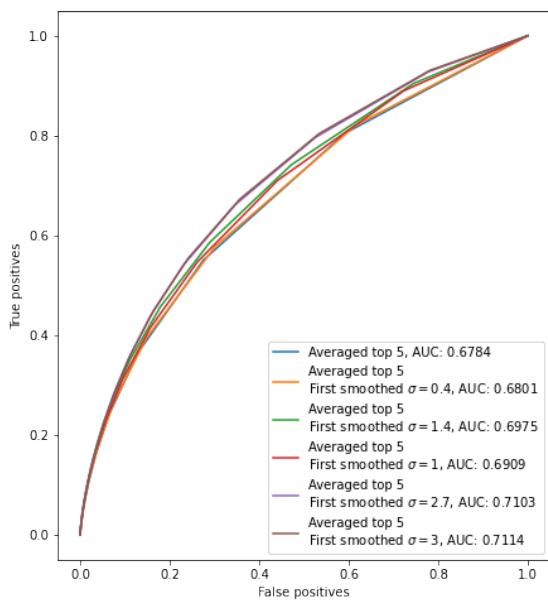
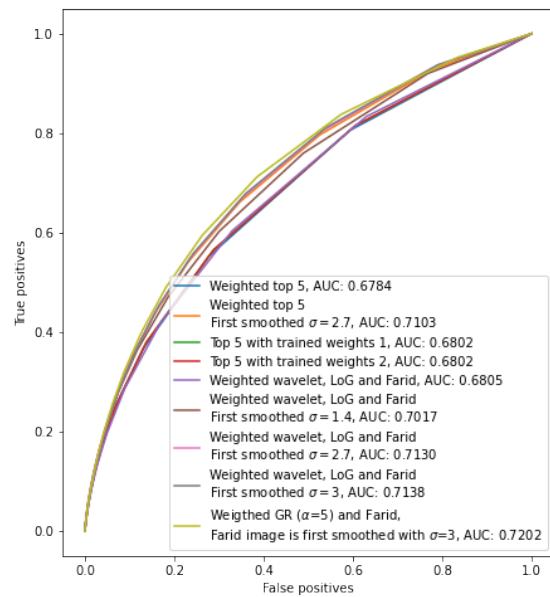


Figure B.9: ROC curves of weighted fusion methods.



## Appendix C

# Training - Quantitative Evaluation

Figure C.1: Best average metrics on simple algorithms.

	Name pretty	ACC	F1	PPV	BA	FM	MK
0	Farid	<b>0.8428354037989888</b>	<b>0.3603283034659599</b>	<b>0.6208960724781466</b>	0.6349877909666316	<b>0.4002212283937145</b>	<b>0.46126361114025394</b>
1	Prewitt	0.8426755021803201	0.35921593019815307	0.5344735517769945	<b>0.6362767983838477</b>	0.39793090365439365	0.3808044280796363
2	Sobel	0.8426224742035678	0.35890366319326233	0.5387052450843923	0.6351455039780569	0.3961143873021146	0.3851576967144903
3	Scharr	0.8425840796173899	0.3579935186016016	0.5250714062180332	0.6312769526521892	0.3958498023364447	0.3720903242027992
4	Wavelet	0.840826804298933	0.3571518805857528	0.45691750747014714	0.6368125936458656	0.3886282250498543	0.3095046505691799
5	Roberts	0.8412362527301205	0.3511639059302506	0.4778162269116179	0.6273962950905567	0.3886282250498543	0.326013886630318
6	Frei-Chen	0.8391572988878883	0.3062783135644804	0.2929558633516943	0.5989024903038247	0.3925817741667174	0.13282665262959029

Figure C.2: Best average metrics on LoG.

Name	ACC ↑	F1 ↑	PPV ↑	FM ↑	MK ↑
0 LoG $\sigma=1.4$	0.8078309076753093	<b>0.2692176366684656</b>	0.19945892684793395	<b>0.3886282250498543</b>	0.05315824031802559
1 LoG $\sigma=2$	0.8163332266309845	<b>0.2692176366684656</b>	0.21366175716004965	<b>0.3886282250498543</b>	0.06913660997100024
2 LoG $\sigma=2.7$	<b>0.8262667040000482</b>	<b>0.2692176366684656</b>	<b>0.23514396152985695</b>	<b>0.3886282250498543</b>	<b>0.09051238952964613</b>

Figure C.3: Best average metrics on GR.

Name	ACC ↑	F1 ↑	PPV ↑	FM ↑	MK ↑
0 GR $\alpha=1$	0.8398676222928334	0.2809169417112826	0.46577081277692994	0.3886282250498543	0.30627343817421104
1 GR $\alpha=2$	<b>0.8414398329135325</b>	0.33122536530252716	<b>0.5158914769020888</b>	0.3886282250498543	<b>0.35803871382914315</b>
2 GR $\alpha=3$	0.8414097407270357	0.3588558433080043	0.5111994632169153	0.3989079341474659	0.3537554824406285
3 GR $\alpha=4$	0.8403666045943006	0.369094878812759	0.4876823485269804	0.41048293083452025	0.3301306065013818
4 GR $\alpha=5$	0.839722413626837	<b>0.36919139928254885</b>	0.47241607142857134	0.4141513115457588	0.3121349596506833
5 GR $\alpha=6$	0.8397117874364499	0.3657853822355783	0.45902011054421765	<b>0.4145883680614629</b>	0.29873901989836943

Figure C.4: Best average metrics on K-means.

Name	ACC ↑	F1 ↑	PPV ↑	FM ↑	MK ↑
0 K-means K=2	0.8368863591011028	0.2692176366684656	0.36564956086858047	<b>0.3886282250498543</b>	0.20772260491546649
1 K-means K=3	0.839657636587736	0.2758236745965756	0.5860801674086524	<b>0.3886282250498543</b>	0.426049060297894
2 K-means K=4	<b>0.8397194562783902</b>	<b>0.2977682233613229</b>	<b>0.5951367964733562</b>	<b>0.3886282250498543</b>	<b>0.43516734884683006</b>

Figure C.5: Best average metrics on Gabor.

Name	ACC ↑	F1 ↑	PPV ↑	FM ↑	MK ↑
0 GS5O8*	0.8397050526196636	0.31176849513999094	0.3297404825895382	0.392811811221533	0.17164606778830518
1 GS3O11	<b>0.8397681303387531</b>	0.3224068468326724	0.4766964285714286	0.3972747193700072	0.3164144526526576
2 GS5O11	0.8397353398163107	0.340577244871359	<b>0.4888214285714286</b>	0.4052048302763412	<b>0.3285391661556656</b>
4 GS5O8	0.8397345446959911	<b>0.3416074243178092</b>	0.4825892857142857	<b>0.40559113600879826</b>	0.32230691612030293

Figure C.6: Best average metrics on Canny 1.

	Name	ACC ↑	F1 ↑	PPV ↑	FM ↑	MK ↑
<b>0</b>	Canny, $\sigma=0.1$ , low threshold=0.0	0,840201	0,183186	0,449634	0,196533	0,29442
<b>1</b>	Canny, $\sigma=0.1$ , low threshold=0.05	0,840238	0,184872	0,451781	0,198138	0,296654
<b>2</b>	Canny, $\sigma=0.1$ , low threshold=0.1	0,840607	<b>0,186417</b>	0,471281	0,201201	0,316683
<b>3</b>	Canny, $\sigma=0.1$ , low threshold=0.15	0,841074	0,185263	0,496116	0,203094	0,341474
<b>4</b>	Canny, $\sigma=0.1$ , low threshold=0.2	0,841466	0,177439	0,514943	0,20193	0,360149
<b>5</b>	Canny, $\sigma=0.1$ , low threshold=0.25	0,841744	0,16348	0,527181	0,197809	0,372135
<b>6</b>	Canny, $\sigma=0.1$ , low threshold=0.3	0,841911	0,145871	0,534875	0,1916	0,37951
<b>7</b>	Canny, $\sigma=0.1$ , low threshold=0.35	0,841955	0,127425	0,538596	0,182524	0,382995
<b>8</b>	Canny, $\sigma=0.1$ , low threshold=0.4	0,841924	0,109791	0,539654	0,170893	0,383891
<b>9</b>	Canny, $\sigma=0.1$ , low threshold=0.45	0,841841	0,093843	0,539037	0,158114	0,382858
<b>10</b>	Canny, $\sigma=0.1$ , low threshold=0.5	0,841739	0,07958	0,53759	0,1449	0,381292
<b>11</b>	Canny, $\sigma=0.1$ , low threshold=0.55	0,841638	0,067171	0,537476	0,131943	0,381012
<b>12</b>	Canny, $\sigma=0.1$ , low threshold=0.6	0,841516	0,056455	0,537108	0,119441	0,380205
<b>13</b>	Canny, $\sigma=0.1$ , low threshold=0.65	0,841365	0,047229	0,535021	0,107516	0,377661
<b>14</b>	Canny, $\sigma=0.1$ , low threshold=0.7	0,841187	0,039497	0,530845	0,096497	0,372996
<b>15</b>	Canny, $\sigma=0.1$ , low threshold=0.75	0,841005	0,033037	0,524251	0,086371	0,365997
<b>16</b>	Canny, $\sigma=0.1$ , low threshold=0.8	0,840831	0,027663	0,515904	0,07717	0,357312
<b>17</b>	Canny, $\sigma=0.1$ , low threshold=0.85	0,84067	0,023159	0,506372	0,068786	0,347498
<b>18</b>	Canny, $\sigma=0.1$ , low threshold=0.9	0,840529	0,019427	0,496312	0,061275	0,337205
<b>19</b>	Canny, $\sigma=0.1$ , low threshold=0.95	0,840407	0,016343	0,484779	0,054554	0,325481
<b>20</b>	Canny, $\sigma=0.1$ , low threshold=1.0	0,840303	0,013794	0,472849	0,048589	0,313393
<b>21</b>	Canny, $\sigma=1$ , low threshold=0.0	0,841779	0,180307	0,535171	0,200352	0,380484
<b>22</b>	Canny, $\sigma=1$ , low threshold=0.05	0,841786	0,180778	0,535743	0,200543	0,381056
<b>23</b>	Canny, $\sigma=1$ , low threshold=0.1	0,841864	0,182137	0,542319	0,202505	0,387621
<b>24</b>	Canny, $\sigma=1$ , low threshold=0.15	0,842064	0,178883	0,555413	<b>0,203923</b>	0,400643
<b>25</b>	Canny, $\sigma=1$ , low threshold=0.2	0,842265	0,167107	0,570097	0,201711	0,415152
<b>26</b>	Canny, $\sigma=1$ , low threshold=0.25	0,842448	0,148982	0,584534	0,19596	0,429307
<b>27</b>	Canny, $\sigma=1$ , low threshold=0.3	0,842537	0,128447	0,597402	0,187399	0,442194
<b>28</b>	Canny, $\sigma=1$ , low threshold=0.35	<b>0,842552</b>	0,10843	0,609123	0,17512	0,45362
<b>29</b>	Canny, $\sigma=1$ , low threshold=0.4	0,842493	0,090273	0,619446	0,160887	0,463457
<b>30</b>	Canny, $\sigma=1$ , low threshold=0.45	0,84237	0,074263	0,628006	0,145837	0,471538
<b>31</b>	Canny, $\sigma=1$ , low threshold=0.5	0,842201	0,060396	0,635336	0,13062	0,478413
<b>32</b>	Canny, $\sigma=1$ , low threshold=0.55	0,84196	0,04863	0,641893	0,115673	0,484528
<b>33</b>	Canny, $\sigma=1$ , low threshold=0.6	0,841657	0,038794	0,649835	0,101289	0,492
<b>34</b>	Canny, $\sigma=1$ , low threshold=0.65	0,84135	0,030745	0,649186	0,087827	0,49084
<b>35</b>	Canny, $\sigma=1$ , low threshold=0.7	0,841059	0,024203	0,640637	0,075361	0,481875
<b>36</b>	Canny, $\sigma=1$ , low threshold=0.75	0,840798	0,018906	0,61981	0,063908	0,46071
<b>37</b>	Canny, $\sigma=1$ , low threshold=0.8	0,840573	0,014687	0,587943	0,053636	0,428575
<b>38</b>	Canny, $\sigma=1$ , low threshold=0.85	0,840386	0,011372	0,145786	0,044619	0,38693
<b>39</b>	Canny, $\sigma=1$ , low threshold=0.9	0,840236	0,0088	0,156571	0,036826	0,336959
<b>40</b>	Canny, $\sigma=1$ , low threshold=0.95	0,840116	0,006804	0,161893	0,030215	0,286155
<b>41</b>	Canny, $\sigma=1$ , low threshold=1.0	0,840021	0,005253	0,164214	0,024649	0,236279

Figure C.7: Best average metrics on Canny 2.

<b>42</b>	Canny, $\sigma=1.4$ , low threshold=0.0	0,841827	0,177846	0,539366	0,201071	0,384216
<b>43</b>	Canny, $\sigma=1.4$ , low threshold=0.05	0,841836	0,178221	0,540058	0,201305	0,384908
<b>44</b>	Canny, $\sigma=1.4$ , low threshold=0.1	0,841931	0,178949	0,546751	0,203031	0,392074
<b>45</b>	Canny, $\sigma=1.4$ , low threshold=0.15	0,842126	0,173664	0,560538	0,202991	0,406082
<b>46</b>	Canny, $\sigma=1.4$ , low threshold=0.2	0,842317	0,158395	0,575821	0,199748	0,421124
<b>47</b>	Canny, $\sigma=1.4$ , low threshold=0.25	0,84246	0,137318	0,590563	0,192462	0,435506
<b>48</b>	Canny, $\sigma=1.4$ , low threshold=0.3	0,842524	0,11529	0,604313	0,180645	0,44881
<b>49</b>	Canny, $\sigma=1.4$ , low threshold=0.35	0,842508	0,094888	0,616195	0,165531	0,460202
<b>50</b>	Canny, $\sigma=1.4$ , low threshold=0.4	0,8424	0,076887	0,626407	0,149127	0,469915
<b>51</b>	Canny, $\sigma=1.4$ , low threshold=0.45	0,84223	0,061307	0,635406	0,132285	0,478433
<b>52</b>	Canny, $\sigma=1.4$ , low threshold=0.5	0,841958	0,048116	0,645357	0,115505	0,488102
<b>53</b>	Canny, $\sigma=1.4$ , low threshold=0.55	0,841616	0,037227	<b>0,652321</b>	0,099326	<b>0,494377</b>
<b>54</b>	Canny, $\sigma=1.4$ , low threshold=0.6	0,841268	0,028432	0,648333	0,084123	0,489834
<b>55</b>	Canny, $\sigma=1.4$ , low threshold=0.65	0,840942	0,021397	0,631022	0,069988	0,472078
<b>56</b>	Canny, $\sigma=1.4$ , low threshold=0.7	0,840657	0,015859	0,597575	0,057153	0,43828
<b>57</b>	Canny, $\sigma=1.4$ , low threshold=0.75	0,840422	0,011616	0,139571	0,045865	0,390862
<b>58</b>	Canny, $\sigma=1.4$ , low threshold=0.8	0,840236	0,008428	0,1565	0,036178	0,332135
<b>59</b>	Canny, $\sigma=1.4$ , low threshold=0.85	0,840091	0,006056	0,1635	0,028054	0,268411
<b>60</b>	Canny, $\sigma=1.4$ , low threshold=0.9	0,839982	0,004315	0,162679	0,021407	0,201833
<b>61</b>	Canny, $\sigma=1.4$ , low threshold=0.95	0,839901	0,00304	0,149536	0,01606	0,1355
<b>62</b>	Canny, $\sigma=1.4$ , low threshold=1.0	0,839843	0,002117	0,129643	0,011907	0,076363
<b>63</b>	Canny, $\sigma=2.7$ , low threshold=0.0	0,841719	0,166624	0,54262	0,19732	0,387272
<b>64</b>	Canny, $\sigma=2.7$ , low threshold=0.05	0,841734	0,167233	0,544037	0,197786	0,388688
<b>65</b>	Canny, $\sigma=2.7$ , low threshold=0.1	0,841828	0,163837	0,553646	0,198095	0,39825
<b>66</b>	Canny, $\sigma=2.7$ , low threshold=0.15	0,842029	0,143731	0,568428	0,193864	0,412831
<b>67</b>	Canny, $\sigma=2.7$ , low threshold=0.2	0,842149	0,116273	0,584309	0,180771	0,429058
<b>68</b>	Canny, $\sigma=2.7$ , low threshold=0.25	0,842208	0,09014	0,602521	0,161242	0,446749
<b>69</b>	Canny, $\sigma=2.7$ , low threshold=0.3	0,842102	0,067528	0,618889	0,139457	0,462395
<b>70</b>	Canny, $\sigma=2.7$ , low threshold=0.35	0,841849	0,048694	0,636776	0,116675	0,479488
<b>71</b>	Canny, $\sigma=2.7$ , low threshold=0.4	0,841417	0,033764	0,64176	0,094132	0,483552
<b>72</b>	Canny, $\sigma=2.7$ , low threshold=0.45	0,840956	0,02243	0,621208	0,072808	0,462299
<b>73</b>	Canny, $\sigma=2.7$ , low threshold=0.5	0,840544	0,014099	0,567557	0,053199	0,408132
<b>74</b>	Canny, $\sigma=2.7$ , low threshold=0.55	0,840225	0,008346	0,137393	0,036391	0,320961
<b>75</b>	Canny, $\sigma=2.7$ , low threshold=0.6	0,839998	0,004611	0,147821	0,022806	0,205267
<b>76</b>	Canny, $\sigma=2.7$ , low threshold=0.65	0,839855	0,002365	0,118071	0,012958	0,081483
<b>77</b>	Canny, $\sigma=2.7$ , low threshold=0.7	0,839775	0,001087	0,077107	0,006631	-0,02103
<b>78</b>	Canny, $\sigma=2.7$ , low threshold=0.75	0,839736	0,000417	0,045	0,002988	-0,08974
<b>79</b>	Canny, $\sigma=2.7$ , low threshold=0.8	0,839719	0,000124	0,021929	0,001107	-0,13075
<b>80</b>	Canny, $\sigma=2.7$ , low threshold=0.85	0,839713	2,71E-05	0,00725	0,000299	-0,15106
<b>81</b>	Canny, $\sigma=2.7$ , low threshold=0.9	0,839712	4,7E-06	0,001643	6,05E-05	-0,15843
<b>82</b>	Canny, $\sigma=2.7$ , low threshold=0.95	0,839712	7,16E-07	0,000393	1,14E-05	-0,15988
<b>83</b>	Canny, $\sigma=2.7$ , low threshold=1.0	0,839712	7,61E-08	3,57E-05	1,17E-06	-0,16025

Figure C.8: Odessa real SAR image, denoised with SARBLF.

