



**HAL**  
open science

# Training CNNs on speckled optical dataset for edge detection in SAR images

Chenguang Liu, Florence Tupin, Yann Gousseau

► **To cite this version:**

Chenguang Liu, Florence Tupin, Yann Gousseau. Training CNNs on speckled optical dataset for edge detection in SAR images. ISPRS Journal of Photogrammetry and Remote Sensing, 2020. hal-02424315v6

**HAL Id: hal-02424315**

**<https://hal.science/hal-02424315v6>**

Submitted on 28 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Training CNNs on speckled optical dataset for edge detection in SAR images

Chenguang Liu<sup>a,\*</sup>, Florence Tupin<sup>a</sup>, Yann Gousseau<sup>a</sup>

<sup>a</sup>*Télécom Paris, Institut Polytechnique de Paris, France*

---

## Abstract

Edge detection in SAR images is a difficult task due to the strong multiplicative noise. Many researches have been dedicated to edge detection in SAR images but very few try to address the most challenging 1-look situations. Motivated by the success of CNNs for the analysis of natural images, we develop a CNN edge detector for 1-look SAR images. We propose to simulate a SAR dataset using the optical dataset BSDS500 to avoid the tedious job of edge labeling, and we propose a framework, a hand-crafted layer followed by learnable layers, to enable the model trained on simulated SAR images to work in real SAR images. The hypothesis behind these two propositions is that both optical and SAR images can be divided into piecewise constant areas and edges are boundaries between two homogeneous areas. The hand-crafted layer, which is defined by a ratio based gradient computation method, helps to tackle the gap between training and testing images, because the gradient distribution will not be influenced by the mean intensity values of homogeneous areas. The gradient computation step is done by Gradient by Ratio (GR) and the learnable layers are identical to those in HED. The proposed edge detector, GRHED, outperforms concurrent approaches in all our simulations especially in two 1-look real SAR images.

*Keywords:*

edge detection, 1-look SAR image, optical dataset, CNNs, hand-crafted layer, GRHED

---

## 1. Introduction

Edges are important features in Synthetic Aperture Radar (SAR) images. They can be used as low level features for many applications like line segment detection [1, 2], SAR image segmentation [3, 4], coastline detection [5, 6], image

---

\*Corresponding Author.

*Email addresses:* chenguangl@whu.edu.cn (Chenguang Liu),  
florence.tupin@telecom-paris.fr (Florence Tupin), yann.gousseau@telecom-paris.fr  
(Yann Gousseau)

registration [7, 8] and even SAR image despeckling [9]. Due to the strong multiplicative speckle noise in SAR images, methods developed for optical images, which are usually based on pixel value differences, produce more false edges in brighter areas and thus are not suitable for SAR images. Many researches have been dedicated to edge detection in SAR images in the past years. In [10], the Ratio of Average (ROA) was proved to have a constant false alarm rate for SAR images (CFAR property). The ratio operator is applied along four directions and the minimum normalized ratio is used to compute the gradient magnitude. The direction corresponding to the minimum normalized ratio is regarded as the edge orientation. A threshold determined by a given probability of false alarms and a morphological operator are then applied to obtain a binary thin edge map. However, ROA is optimal only for isolated step edges. An efficient multiedge detector, ROEWA, was proposed afterwards in [11]. The Ratio of Exponentially Weighted Average was shown to be optimal in terms of minimum mean square error (MMSE) under the hypothesis of a stochastic multi-edge model. The method is based on ROA, but averages are weighted by a decreasing exponential function, allowing a better detection of multiple edges close to each other. Besides, instead of computing the ratio along four directions, the ratio is computed along the horizontal and vertical directions and the normalized ratios are considered as the horizontal and vertical components of the gradient magnitude. A modified watershed algorithm is then used to threshold the Edge Strength Map and a region merging algorithm is used to eliminate the false edge pixels. Edge detectors using different shape of window functions were introduced later in [12] and [13]. Non-maxima suppression [14] and hysteresis thresholding are applied to obtain the binary thin edge map. An edge compensation strategy was also introduced in [13] to extract weak edge pixels. To reduce the influence of isolated strong bright points in real SAR images, an Anisotropic Morphological Directional Ratio (AMDR) [15] was proposed by replacing the weighted average filter with the weighted median filter. The edge localization accuracy in the Edge Strength Map (ESM) and the Edge Direction Map (EDM) was then improved by a multiplicative spatial and directional matching filter. By introducing the improved ESM and EDM into the routine of Canny edge detector [14], the resulting edge detector is able to obtain a binary thin edge map. The connectivity of edges is finally improved by an edge remedy strategy. However, the performances of the edge detectors developed for SAR images are still not fully satisfying, especially in the very noisy 1-look situation.

On the other hand, convolutional neural networks (CNNs) have proven to be very successful for edge detection in natural images, with approaches such as Deepedge [16], DeepContour [17], HED [18, 19], CEDN [20], AMH-Net [21] or RCF [22, 23], which have permitted to improve significantly the performances of traditional edge detectors like Sobel [24], Canny [14], Statistical Edge [25], Pb [26], gPb [27] and Structured Edge [28], at least on databases similar to the training sets. Motivated by this fact, we propose to develop a CNN based edge detector for SAR images.

One crucial factor that contributes to the success of CNNs is the availability

of training datasets with ground truth, but there is still no training dataset for edge detection in SAR images. Under the hypothesis that both optical and SAR images can be divided into piecewise constant areas, we propose to simulate a SAR dataset using optical dataset. The motivation is that under this hypothesis both optical and SAR images can be divided into two kinds of regions: homogeneous areas and two homogeneous areas across boundaries (edges being boundaries between two homogeneous areas). Only if we can ensure that all homogeneous areas in the simulated SAR dataset follow similar distributions as those in real SAR images, the models trained using the simulated dataset should be applicable to real SAR images. The simulated dataset can be obtained by multiplying images in the optical dataset with speckle noise. The simulated SAR images are called speckled optical images in the following.

Even though the homogeneous areas in simulated SAR images are modified to follow similar distributions as those in real SAR images, the distributions of homogeneous areas depend on their mean intensity values. Due to the existence of some bright homogeneous areas in real SAR images which are not contained in the simulated dataset, CNN models directly trained on the images are not applicable to real SAR images as can be seen in Figure 1-(c), where we train a classical CNN edge detector HED [18, 19] using the simulated dataset and apply the trained model to the 1-look real SAR images. HED produces spurious detections since it does not learn the way to process those areas.

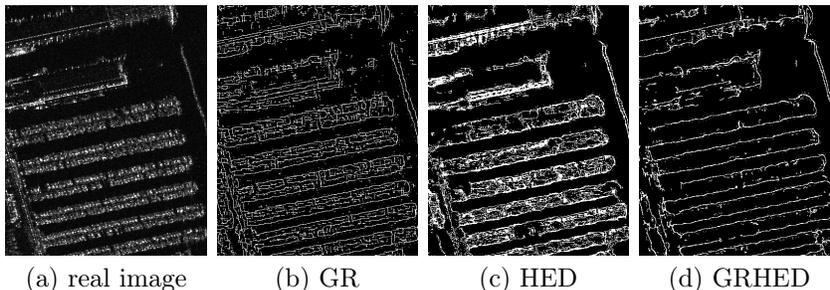


Figure 1: Edge maps produced by different methods in a 1-look real image (a subpart of the TerraSAR-X image, San Francisco). (a) 1-look real image of size  $512 \times 512$  pixels; (b) edge map computed by GR [29]; (c) edge map computed by HED; (d) edge map computed by GRHED (proposed method).

To cope with this problem, we propose in this paper to add a hand-crafted layer before the learnable layers. The hand-crafted layer is defined by a ratio based gradient computation method. Since the parameters in the hand-crafted layer are fixed, we train learnable layers on the feature maps computed by the hand-crafted layer in images, and apply the trained layers to the (gradient) feature maps of testing images. The motivation is that, with the ratio operation, the gradient distribution in each pixel depends only on the ratio of the mean intensity values of two homogeneous areas in the opposite side windows (ratio being 1 for pixels in homogeneous areas) [10]. In this paper, the hand-crafted layer is chosen as Gradient by Ratio (GR) proposed in [29]. The learnable

layers are identical to those in HED [18, 19]. We train HED layers on the gradient feature maps of training images and apply the trained model to the gradient feature maps of testing images. The proposed edge detector will thus be called GRHED. It can be seen in Figure 1-(d) that GRHED is performing efficient edge detection in 1-look real SAR images.

The pipeline of the proposed method can be found in Figure 2. In the training phase of GRHED, as shown in Figure 2-(a), training images are first fed into the hand-crafted layer to obtain the gradient feature maps. The convolutional layers of HED are trained on the feature maps computed by GR to get the edge probability map. In the testing phase, as shown in Figure 2-(b), testing images are successively processed by the hand-crafted layer (GR) and convolutional layers (HED) to obtain the edge probability map. Postprocessing steps including non-maxima suppression and thresholding are applied to obtain the binary edge map. The non-maxima suppression step is identical to that in [28], and the strategy to choose the threshold can be found in Section 4.3.1.

The contributions of this paper can be summarized as follows:

- we propose to simulate a SAR dataset using the optical dataset, since there is still no dataset for edge detection in SAR images;
- we propose a framework, which is composed of a hand-crafted layer and learnable layers, to tackle the gap between training and testing images. The hand-crafted layer defined by a ratio based method, ensures the similarity of computed feature maps in simulated SAR and real SAR images;
- the proposed edge detector GRHED, largely outperforms existing edge detectors in the most challenging situations (complex scenes, 1-look real SAR images).

This paper is organized as follows: in Section 2 we give details about the building of the speckled optical dataset and explain the gap between simulated SAR and real SAR images. In section 3, we describe the proposed framework, a hand-crafted layer defined by a ratio based gradient computation method, followed by learnable fully convolutional layers. We will also explain why the proposed strategy is able to tackle the gap between training and testing images. In section 4, we demonstrate the efficiency of the proposed edge detector, GRHED, with experiments on several 1-look simulated edge images, 200 speckled optical images simulated from BSDS500 [27], one 1-look realistically simulated SAR image, and two 1-look real SAR images.

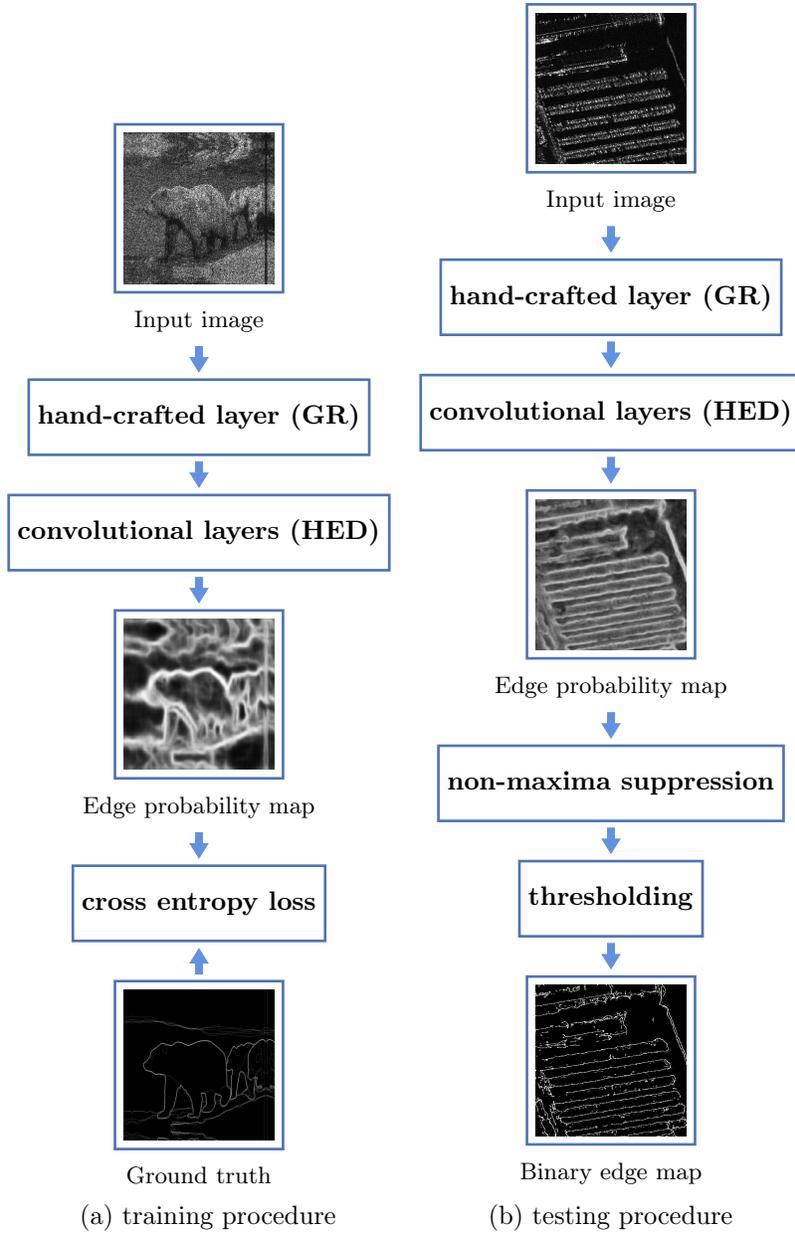


Figure 2: The pipeline of GRHED. During training, gradient feature maps of training images are precomputed, then HED layers are trained on the feature maps to produce the edge probability map. During testing, testing images are successively processed by GR, HED, non-maxima suppression (identical to [28]) and thresholding to obtain the edge map.

## 2. Simulated SAR dataset and the gap between simulated SAR and real SAR images

### 2.1. Speckled optical dataset

Since the aim of this paper is to train CNNs for edge detection in SAR images, the training dataset is of crucial importance for the performances of the edge detector. In order to avoid the tedious job of manual edge labeling, we simulated a SAR dataset using optical dataset for edge detection in natural images, under the hypothesis that both optical and SAR images are composed of piecewise constant areas and edges are boundaries between two homogeneous areas. If we can ensure that all homogeneous areas in the simulated dataset follow similar distributions as those in real SAR images, the simulated dataset should be usable for edge detection in real SAR images.

Due to the coherent imaging system, SAR data present the well known speckle phenomenon. Following Goodman model [30] of fully developed speckle, it can be shown that the amplitude of the backscattered electro-magnetic field of a homogeneous area with mean intensity  $\langle I \rangle$  follows a Nakagami distribution :

$$f(t|\langle I \rangle) = \frac{2}{\Gamma(L)} \left( \frac{L}{\langle I \rangle} \right)^L t^{2L-1} e^{-(Lt^2/\langle I \rangle)}, \quad (1)$$

$L$  being the number of looks of the image. For images with the best resolution,  $L = 1$  and the amplitude of a physically homogeneous area follows a Rayleigh distribution. Another way of modeling 1-look data is the multiplicative model :  $t = \sqrt{\langle I \rangle} s$ ,  $s$  representing the speckle noise and following the Rayleigh distribution given in eq. (1) with  $\langle I \rangle = 1$  and  $L = 1$ .

Using the multiplicative noise model it is therefore easy to generate speckled data by multiplying an image by  $s$ . In this paper, we use the Berkeley Segmentation Data Set 500 (BSDS500) [27] with the same data augmentation as in HED [19] for training and validation, to simulate a SAR dataset. Specifically, each image in the training and validation set is rotated by 16 angles, flipped horizontally, and rescaled to 50%, 100% and 150% of its original size. There is no data augmentation for the testing set. The resulting speckled optical dataset, which we will call BSDS500-speckled in the following, is obtained by multiplying the grayscale intensity channel of each color image with 1-look speckle noise. It contains  $300 \times 16 \times 2 \times 3 = 28800$  images for training (80%) and validation (20%) and 200 images for testing.

### 2.2. The differences between speckled optical images and real SAR images

In order to compare the differences between speckled optical images and real SAR images, we plot the histograms for both of them. We display the histograms of five speckled optical images which are randomly selected from the training dataset in Figure 3-(a), the histogram of the entire training dataset in Figure 3-(b) and compare them with the histograms of two 1-look real SAR images (a Sentinel-1 image of Lelystad and a TerraSAR-X image of San Francisco) as

shown in Figure 4. From Figure 3 and Figure 4 we can see that the main differences between speckled optical images and real SAR images are the range of pixel values. The values of some pixels in real SAR images are much larger than those of pixels in the speckled optical images.

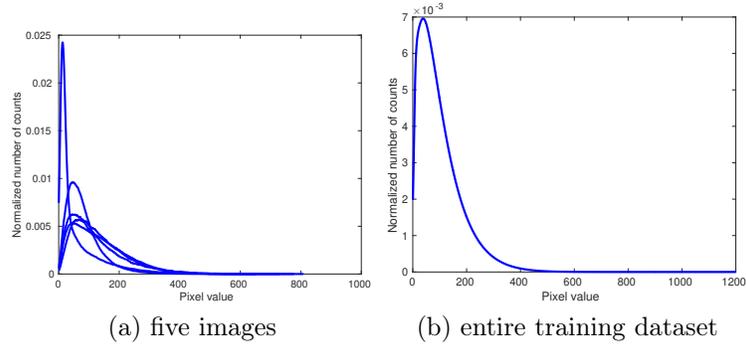


Figure 3: (a) histograms of five speckled optical images which are randomly chosen from the training dataset; (b) the histogram of the entire training dataset. The size of the bin is 1.0.

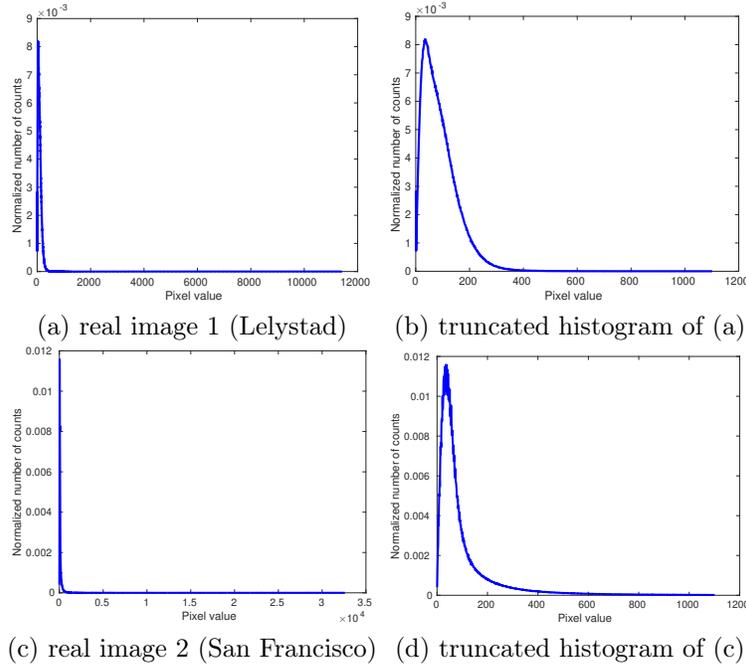


Figure 4: Histograms of two 1-look real SAR images and the truncated histogram for each image. The size of the bin is 1.0.

Formula (1) is the most usual way to describe the statistics of real SAR im-

images, which gives the way of modelling the homogeneous areas in real SAR images. In addition, most existing gradient computation methods for SAR images, such as ROA [10], ROEWA [11] and other ratio based methods, are justified under the hypothesis of piecewise constant areas so all images can be considered as a mixture of homogeneous areas (the ratio operator was shown to be more suitable for SAR images in [10] with the hypothesis that in each (of the two) windows used to compute the ratio, pixel values are independently, identically and Gamma distributed, and in [11], the exponentially weighted average was shown to be optimal to estimate the local mean of the homogeneous areas under the hypothesis of a multi-edge model). In this work, we proceed in the same way and assume that both speckled optical images and real SAR images can be divided into homogeneous areas with many different mean intensity values. The edges are defined as boundaries between any two homogeneous areas.

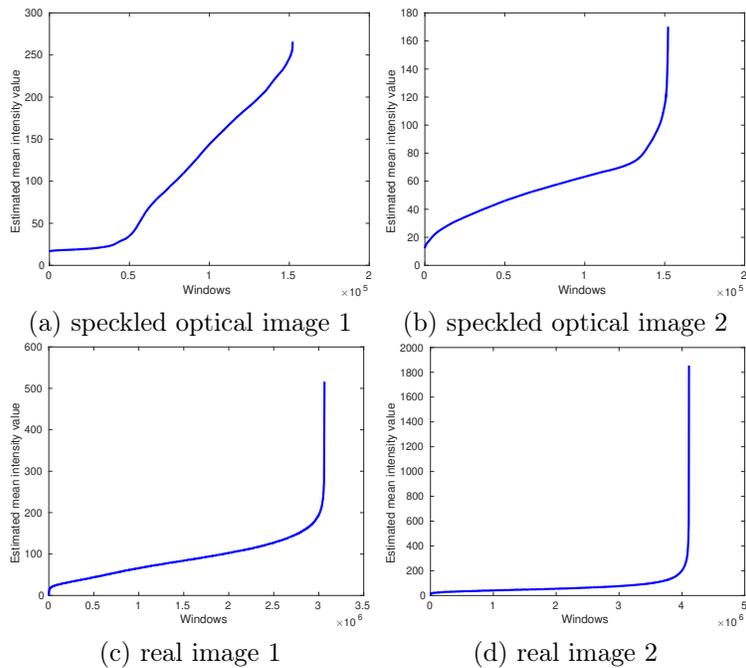


Figure 5: The curves for square root of mean intensity values estimated from two speckled optical images and two real SAR images, using windows of size  $20 \times 20$  pixels. (a) speckled optical image 1; (b) speckled optical image 2; (c) real image 1 (Leystad); (d) real image 2 (San Francisco).

Under the hypothesis that both speckled optical images and real SAR images are composed of piecewise constant areas, we assume that the differences in the range of pixel values are caused by the differences in the mean intensity values of homogeneous areas. We try to evaluate the possible underlying mean intensity values for both speckled optical images and real SAR images. We use windows of size  $20 \times 20$  pixels to estimate the underlying mean intensity values for areas

whose size is equal to the size of the windows, in both speckled optical images and real SAR images. The curves of the square root of the mean intensity values (in ascending order) estimated from two randomly selected speckled optical images and two 1-look real SAR images can be found in Figure 5. For the estimation in speckled optical images, we use the maximum likelihood estimator to estimate the square root of the mean intensity values, while for real SAR images, we use the mode (the most frequent pixel value after digitizing the pixel values) to estimate the mean intensity values in order to reduce the influence of strong bright points in real SAR images (the square root of the mean intensity value is obtained by multiplying the mode with  $\sqrt{2}$ ). It should be pointed out that both maximum likelihood estimator and mode estimator are under the hypothesis that samples in the windows follow a Rayleigh distribution.

From Figure 5 we can see that the underlying mean intensity values in homogeneous areas of real SAR images can be much larger than those in speckled optical images (it should be noted that the underlying mean intensity values for all homogeneous areas in the training dataset will not be larger than  $255^2$  because the maximum possible value in natural images before multiplying the speckle noise is 255). Therefore, the main differences between speckled optical and real SAR images are that some areas with high mean intensity values exist in real SAR images but not in speckled optical images.

After creating a simulated dataset for training and knowing the gap between simulated SAR and real SAR images, the next question is how to train CNNs using this dataset and how to enable the trained model to work in real SAR images. We will describe the way to tackle the gap between the training and testing data in the next section.

### 3. GRHED, a CNN based edge detector for SAR images

In order to deal with the differences in the mean intensity values of homogeneous areas between speckled optical images and real SAR images, we propose to add a hand-crafted layer before the learnable layers. The learnable layers are trained on the feature maps computed by the hand-crafted layer. The hand-crafted layer is defined by a ratio based gradient computation method GR [29] and it aims to tackle the gap between training and testing images, The convolutional layers are identical to those in HED [18, 19]. We therefore call GRHED the resulting architecture. In the following we will describe the details about GR, the benefits of adding the hand-crafted layer defined by GR, HED network architecture and the proposed edge detector GRHED.

#### 3.1. Gradient by Ratio (GR)

For a given pixel located at position  $(x, y)$  in the image  $u$ , the horizontal and vertical gradient components (GR) are defined as

$$G^h(x, y) = \log(R^h(x, y)),$$

$$G^v(x, y) = \log(R^v(x, y)).$$

where  $R^h(x, y)$  and  $R^v(x, y)$  is the ratio of exponentially weighted average in the opposite side windows of pixel located at  $(x, y)$ , along the horizontal and vertical directions. In the horizontal direction,  $R^h(x, y)$  can be computed as

$$R^h(x, y) = \frac{m_1^h(x, y)}{m_2^h(x, y)},$$

where

$$\begin{aligned} m_1^h(x, y) &= \sum_{x'=-W}^W \sum_{y'=1}^W u(x+x', y+y') \times e^{-\frac{|x'|+|y'|}{\alpha}}, \\ m_2^h(x, y) &= \sum_{x'=-W}^W \sum_{y'=-1}^{-1} u(x+x', y+y') \times e^{-\frac{|x'|+|y'|}{\alpha}}, \end{aligned}$$

and where  $W$  is the upper integer part of  $\log(10) \times \alpha$ .  $R^v(x, y)$  can be computed in a similar way.

The magnitude  $G_{gr}(x, y)$  and orientation  $ang_{gr}(x, y)$  of GR at position  $(x, y)$  are defined by

$$\begin{aligned} G_{gr}(x, y) &= \sqrt{G^h(x, y)^2 + G^v(x, y)^2}, \\ ang_{gr}(x, y) &= \arctan \frac{G^v(x, y)}{G^h(x, y)}. \end{aligned}$$

### 3.2. Interest of adding a hand-crafted layer before learnable layers

Under the hypothesis that both optical and SAR images are composed of piecewise constant areas, they can be divided into two kinds of regions: homogeneous areas and boundaries (boundaries exist between two homogeneous areas).

If we train CNN models directly on the images, the aim of training can be summarized as follows: 1) the model should learn to suppress all pixels in homogeneous areas; 2) the model should learn to highlight the boundaries between any two homogeneous areas. In this case, CNNs are trained to process samples drawn from many different distributions and the way of the model to process those samples depends on their corresponding distributions. In 1-look SAR images, the amplitude of all homogeneous areas follows a Rayleigh distribution depending on their mean intensity values. For two homogeneous areas across boundaries, the total distribution of them depends on the mean intensity values of both two homogeneous areas. Adding a hand-crafted layer defined by a ratio based gradient computation method before CNN models helps to deal with the gap between training and testing images because for all homogeneous areas, their gradient magnitude fields follow the same distribution, regardless of their mean intensity values, and the gradient distribution computed over two homogeneous areas across boundaries depends only on the ratio of the mean intensity values. The feature maps computed by the hand-crafted layer (GR) in simulated SAR and real SAR images are similar. Adding the hand-crafted layer

enables the network trained on speckled optical dataset to work in real SAR images. In addition, computing GR can be considered as a form of data augmentation since we have much less kinds of distributions to learn while the total amount of data remains unchanged. Besides, the ratio operation contributes to an edge detector having constant false alarm rate for SAR images.

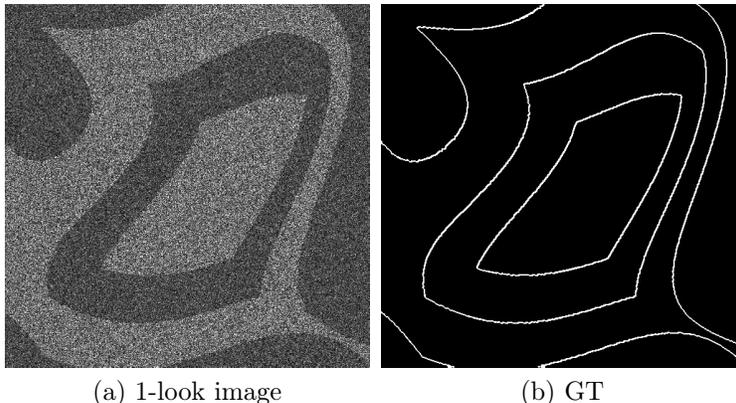


Figure 6: A 1-look edge image with ratio contrast 1.5 and the associated ground truth. The size of the image are  $512 \times 512$  pixels.

In order to give a better explanation on the benefits of computing GR, we compare the distribution of the data before and after computing the gradient for both homogeneous areas and two homogeneous areas across boundaries. We simulate eight 1-look pure noise images of size  $4096 \times 4096$  pixels and eight 1-look synthetic images of size  $512 \times 512$  pixels with amplitude ratio contrast 1.5. The square root of the mean intensity values of the homogeneous images (proportional to an amplitude value) are 60, 90, 120, 150, 180, 210, 240 and 270. For two homogeneous areas across boundaries, the amplitude ratio is chosen as 1.5 for mean values (for the smallest value along the edge) of 20, 50, 70, 90, 110, 130, 150, 200. One example of the synthetic edge image can be found in Figure 6-(a).

The histograms of the eight 1-look pure noise images and the histograms of their gradient magnitude fields computed by GR with weighting parameter  $\alpha = 4$  can be found in Figure 7. The histograms of eight 1-look synthetic edge images as well as the histograms of their gradient magnitude fields can be found in Figure 8. From Figure 7 and Figure 8 we can see that the gradient distribution computed by GR depends only on the ratio of the mean intensity values of homogeneous areas (ratio being 1 for homogeneous areas) and will not be influenced by the mean intensity values. The bright homogeneous areas in real SAR images which are not contained in the training dataset will not be a problem because: 1) the gradient distribution of those homogeneous areas will be the same as that of homogeneous areas with low mean intensity values; 2) the gradient distribution for those homogeneous areas and their neighbouring ones depends only on the ratio of the mean intensity values, which can be the same

as the gradient distribution of two homogeneous areas, whose mean intensity values are both contained in the training dataset. In addition, it can be deduced that the number of distributions that has to be learned by CNNs when trained on the gradient magnitude fields is much smaller than that of a direct training on the images. Since the amount of data is unchanged, the amount of training data for each distribution is therefore increased.

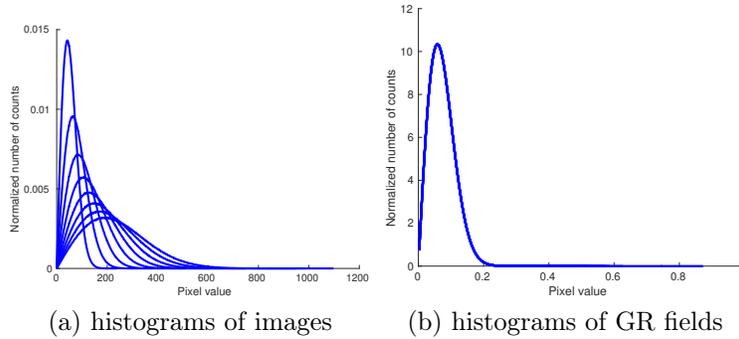


Figure 7: Histograms of the eight 1-look pure noise images and histograms of their gradient magnitude fields computed by GR with  $\alpha = 4$ .

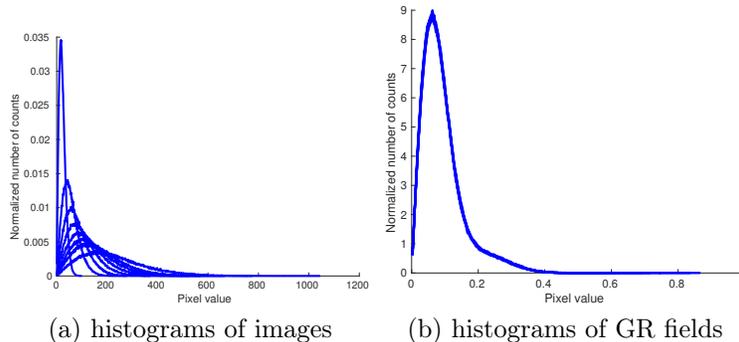


Figure 8: Histograms of the eight 1-look synthetic edge images and histograms of their gradient magnitude fields computed by GR with  $\alpha = 4$ .

A further illustration of the interest to compute GR can be done by comparing the histograms of GR fields in both speckled optical images and real SAR images. The histograms of the gradient magnitude fields computed by GR in 5 randomly selected speckled optical images and the histograms of the GR field computed in the entire training dataset can be found in Figure 9. The histograms of the gradient magnitude fields computed by GR in two 1-look real SAR images can be found in Figure 10. The weighting parameter  $\alpha$  is set to 4.0. The size of the bin is 0.01. From Figure 9 and Figure 10 we can see that even though the range of pixel values in the GR fields of those two 1-look real SAR images may be larger than that of some speckled optical images, nearly

all pixel values in the GR fields of real SAR images are within the range of GR field computed in the entire training dataset (The maximum value in the GR field of the TerraSAR-X image (San Francisco) is 3.7, the maximum pixel value in the GR field of the Sentinel-1 image (Lelystad) is 5.66, and the maximum pixel value in the GR field of the entire training dataset is 5.36). Compared to the huge differences in the range of pixel values of images shown in Figure 3 and Figure 4, this observation illustrates the benefits of computing GR.

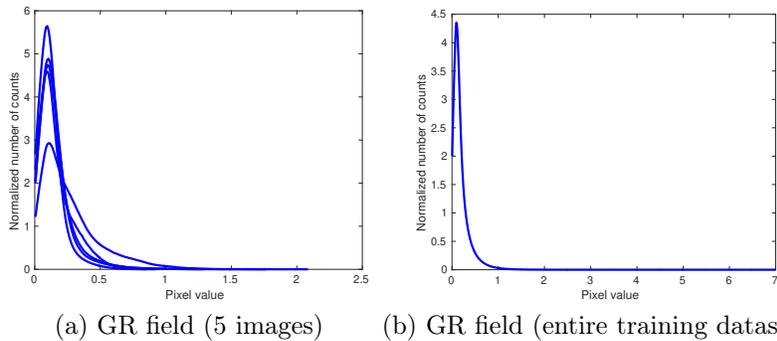


Figure 9: (a) histograms of the gradient magnitude fields computed by GR in five speckled optical images which are randomly chosen from the training dataset; (b) the histogram of the gradient magnitude field computed by GR in the entire training dataset. The weighting parameter  $\alpha$  is set to 4.0. The size of the bin is 0.01. The maximum pixel value in the GR field of the entire training dataset is 5.36. The tail of the histogram in (b) goes up to 7.0 because we set the maximum value to be 7.0 when we plot the histogram.

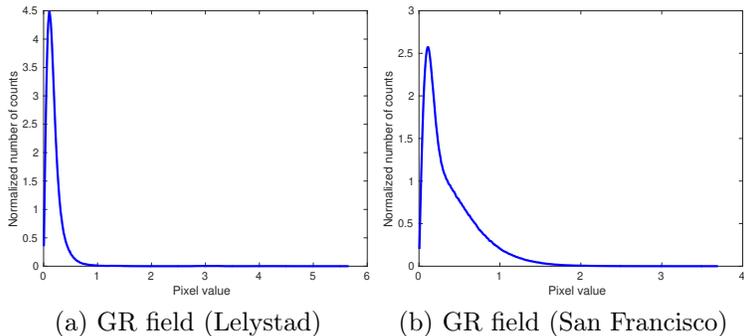


Figure 10: Histograms of the gradient magnitude fields computed by GR in two 1-look real SAR images. The weighting parameter  $\alpha$  is set to 4.0. The size of the bin is 0.01. The maximum pixel value in the GR field of the Sentinel-1 image (Lelystad) is 5.66 and the maximum pixel value in the GR field of the TerraSAR-X image (San Francisco) is 3.7.

### 3.3. HED layers

In this work, we choose to use the convolutional layers of HED, and we train HED layers on the gradient feature maps computed by GR. The HED

method, introduced in [18, 19], relies on a fully convolutional network [31, 32], which is trained end-to-end to perform image-to-image prediction. The network architecture of HED is shown in table 1. The HED network is trimmed from the VGG-16 [33] net by discarding the last max-pooling layer and the 3 fully connected layers. Motivated by the deeply-supervised nets [34], five side outputs are added to the convolutional layers just before the five max-pooling layers in the original VGG-16 net. The loss function of the HED network is composed of the loss function from the side layers and the loss function from the fused output. The final output of HED is an average of the side outputs and the fused output. In-network bilinear interpolation [31, 32] is used to upsample the side outputs so that they all have the same size as the edge ground truth.

Table 1: The network architecture of HED. The convolutional layers are denoted as "conv(receptive field size)-(number of channels)". For brevity, we do not show the ReLU activation function following the convolutional filters in each convolutional layer.

conv3-64	
conv3-64	→ side output 1
MAX-POOLING	
conv3-128	
conv3-128	→ side output 2
MAX-POOLING	
conv3-256	
conv3-256	
conv3-256	→ side output 3
MAX-POOLING	
conv3-512	
conv3-512	
conv3-512	→ side output 4
MAX-POOLING	
conv3-512	
conv3-512	
conv3-512	→ side output 5

#### 3.4. GRHED, the proposed framework for edge detection in real SAR images

Since the gradient distribution of GR is the same for homogeneous areas with all possible gradient mean intensity values, and the gradient distribution in two homogeneous areas across boundaries is influenced by the ratio and not by their mean intensity values, we assume that the possible distributions in the gradient feature space of real SAR images are included in those of the training dataset. Therefore, we propose to train HED on the gradient magnitude field computed by GR in training images and apply the trained model to the gradient features

of testing images. This can be seen as the addition of a hand-crafted layer before the usual HED layers. The resulting GRHED has the following advantages:

- the distribution of feature map values obtained by GR depends only on the ratio of the mean intensity values on the opposite side windows of each pixel (ratio being 1 for pixels located in homogeneous areas), therefore, homogeneous areas with high mean intensity values in real SAR images will not impair the performances of GRHED;
- constant false alarm rate (CFAR) is ensured for SAR images because of the ratio operation;
- by using multiple weighting parameter  $\alpha$  values in GR [29] we can combine diverse informations from the image by concatenating together gradient magnitude fields produced by GR with different  $\alpha$  values (the input of HED will have multiple channels with each channel being the gradient magnitude field computed by GR with a certain value of  $\alpha$ ). It has been studied in [35] that GR with different  $\alpha$  values can capture complementary informations.

### 3.5. Formulation of GRHED

*Training phase.* Given an image  $u$  in the training dataset, noting  $\mathbb{G}$  for its associated edge ground truth, where  $u = \{u_j, j = 1, \dots, |u|\}$  denotes the input image and  $\mathbb{G} = \{\mathbb{G}_j, j = 1, \dots, |\mathbb{G}|\}$  denotes the edge ground truth, we first precompute the feature map of  $u$  with GR (noting  $G_{gr}^u$  for the feature map of  $u$ ), and then train HED layers on  $G_{gr}^u$ . The goal of training HED layers is to produce an output approaching the edge ground truth from the feature maps of each input image. Noting  $\mathbb{W}$  for the collection of parameters in all network layers (excluding those corresponding to side output layers), and noting  $\mathbb{W}_{side} = (\mathbb{W}_{side}^{(1)}, \mathbb{W}_{side}^{(2)}, \dots, \mathbb{W}_{side}^{(5)})$  for the collection of parameters in those side output layers, the objective function for the side output layers is defined as:

$$\mathcal{L}_{side}(\mathbb{W}, \mathbb{W}_{side}) = \sum_{m=1}^5 \ell_{side}^{(m)}(\mathbb{W}, \mathbb{W}_{side}^{(m)}), \quad (2)$$

where  $\ell_{side}$  denotes the loss corresponding to the side outputs. The loss functions are computed over all pixels of the image.

In order to balance the loss between positive/negative classes, a class-balanced cross-entropy loss function is used in formula (2):

$$\begin{aligned} \ell_{side}^{(m)}(\mathbb{W}, \mathbb{W}_{side}^{(m)}) &= -\lambda \sum_{j \in \mathbb{G}_+} \log \mathbb{P}(\mathbb{G}_j = 1 | G_{gr}^u : \mathbb{W}, \mathbb{W}_{side}^{(m)}) \\ &\quad - (1 - \lambda) \sum_{j \in \mathbb{G}_-} \log \mathbb{P}(\mathbb{G}_j = 0 | G_{gr}^u : \mathbb{W}, \mathbb{W}_{side}^{(m)}), \end{aligned} \quad (3)$$

where  $\lambda = \frac{|\mathbb{G}_-|}{|\mathbb{G}|}$ , and  $1 - \lambda = \frac{|\mathbb{G}_+|}{|\mathbb{G}|}$ ,  $|\mathbb{G}_-|$  and  $|\mathbb{G}_+|$  represent the non-edge and edge label sets.  $\mathbb{P}(\mathbb{G}_j = 1 | G_{gr}^u : \mathbb{W}, \mathbb{W}_{side}^{(m)}) = \sigma(a_j^{(m)}) \in [0, 1]$  with  $\sigma(\cdot)$

representing the sigmoid function and  $a_j^{(m)}$  representing the activation value at pixel  $j$ . The edge probability map produced by each side layer is thus defined as:

$$\hat{\mathbb{G}}_{side}^{(m)} = \sigma(\hat{A}_{side}^m), \quad (4)$$

where  $\hat{A}_{side}^{(m)} \equiv \{a_j^{(m)}, j = 1, 2, \dots, |\mathbb{G}|\}$  represents the activations of the output of the side layer.

The fused output is a weighted fusion of those side outputs:

$$\hat{\mathbb{G}}_{fuse} \equiv \sigma\left(\sum_{m=1}^5 h_m \hat{A}_{side}^{(m)}\right), \quad (5)$$

and the corresponding loss function for the fused output  $\mathcal{L}_{fuse}$  is defined as:

$$\mathcal{L}_{fuse}(\mathbb{W}, \mathbb{W}_{side}, h) = \text{Dist}(\mathbb{G}, \hat{\mathbb{G}}_{fuse}), \quad (6)$$

where  $h$  denotes the fusion weights which are learned during training.  $\text{Dist}(\cdot, \cdot)$  represents the distance between the fused output and the edge ground truth, which is measured by the cross entropy loss.

The objective function that needs to be minimized during training using stochastic gradient descent is thus defined as:

$$(\mathbb{W}, \mathbb{W}_{side}, h)^* = \text{argmin}(\mathcal{L}_{side}(\mathbb{W}, \mathbb{W}_{side}) + \mathcal{L}_{fuse}(\mathbb{W}, \mathbb{W}_{side}, h)). \quad (7)$$

*Testing phase.* Given a testing image  $u$ , six edge probability maps (five side outputs and one fused output) are obtained from GRHED:

$$(\hat{\mathbb{G}}_{fuse}, \hat{\mathbb{G}}_{side}^{(1)}, \dots, \hat{\mathbb{G}}_{side}^{(5)}) = \text{CNN}(\text{GR}(u), (\mathbb{W}, \mathbb{W}_{side}, h)^*), \quad (8)$$

where  $\text{GR}(u)$  computes the feature maps of  $u$  using GR.

The final output is computed as the average of all the outputs:

$$\hat{\mathbb{G}}_{prediction} = \text{Average}(\hat{\mathbb{G}}_{fuse}, \hat{\mathbb{G}}_{side}^{(1)}, \dots, \hat{\mathbb{G}}_{side}^{(5)}). \quad (9)$$

After obtaining  $\hat{\mathbb{G}}_{prediction}$ , we apply the same non-maxima suppression step as that in [28], and then plain thresholding is applied to obtain the binary edge map. The strategy of choosing thresholds can be found in Section 4.3.1.

## 4. Experiments

In this section we demonstrate the efficiency of the method we propose, GRHED, using several 1-look synthetic edge images, two hundred 1-look speckled optical images in BSDS500-speckled, one 1-look realistically simulated SAR image and two 1-look real SAR images. In order to show the efficiency of GRHED, we compare it with the original HED algorithm and to the result of training HED on the logarithm of images, as defined below, an algorithm that

we will call HED-log. Observe that for GRHED, HED and HED-log, the convolutional layers that need to be trained are the same, the difference between these methods being the input of the network: gradient magnitude fields of images for GRHED, plain images for HED and logarithm of these for HED-log. The training strategy for those convolutional layers is as follows: we use Adam optimizer to train the network from scratch on the speckled optical dataset BSDS500-speckled. The number of iterations for training is 10000 and the learning rate is 0.001. The size of the batch is 10. The preprocessing step before feeding the data into the learnable layers is global mean subtraction, as done in VGG. Since the outputs of all methods are probability edge maps, they should be processed further to obtain the binary edge maps. In order to obtain the binary edge map, we use the same Non-maxima Suppression procedure as the one in Structured Edge [28] and use a threshold to discard pixels with low values in the probability edge map.

The following methods will be used in the comparison of the next sections:

- GR with  $\alpha = 4$  (this choice of  $\alpha = 4$  being adapted to GR in 1-look situations, as discussed in [35]). For the GR magnitude field, we use the same postprocessing steps as for the magnitude field produced by HED, HED-log and GRHED;
- HED: HED is trained on the speckled optical images and applied to the testing images;
- HED-log: HED is trained on the logarithm of the images in BSDS500-speckled, and is applied to the logarithm of testing images;
- GRHED with multiple  $\alpha$  values,  $\alpha = 2, 3, 4, 5$ : HED is trained on the gradient feature maps which are obtained by concatenating the gradient magnitude fields produced by GR with different  $\alpha$  values.

#### 4.1. Comparison in 1-look synthetic edge images

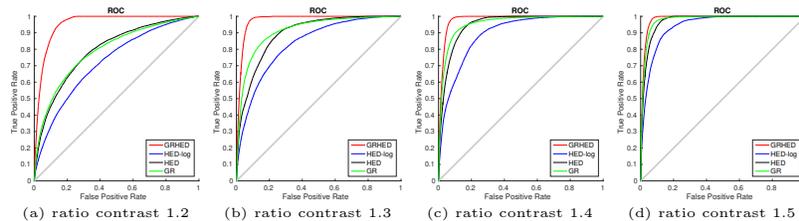


Figure 11: ROC curves computed from the magnitude field produced by GR, HED, HED-log and GRHED in 1-look simulated images with contrast 1.2, 1.3, 1.4 and 1.5. The size of the images is  $512 \times 512$  pixels.

In order to give a fair comparison between the different methods, we compare them in terms of ROC curves [36, 37] and F1-score curves in 1-look synthetic edge images with different ratio contrasts.

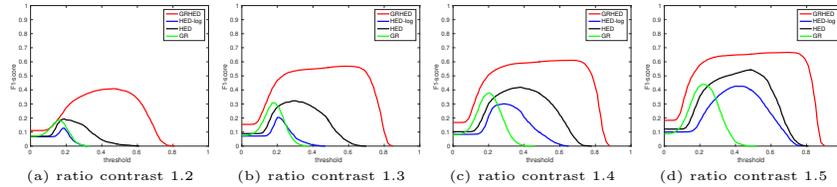


Figure 12: F1-score curves obtained by GR, HED, HED-log and GRHED in synthetic edge images with ratio contrast 1.2, 1.3, 1.4 and 1.5. The F1-score curves are obtained by varying the threshold from 0.0 to 0.9 with step 0.01.

One example of the simulated 1-look edge images with contrast 1.5 and the corresponding ground truth can be found in Figure 6. The ROC curves computed in the magnitude fields produced by GR, HED, HED-log and GRHED in 1-look simulated images with ratio contrast 1.2, 1.3, 1.4 and 1.5 can be found in Figure 11. It can be seen from Figure 11 that GRHED yields the best performances in all situations, especially in 1-look and low contrast situations. The performances of GR and HED appear comparable in terms of ROC curves, and both are better than HED-log.

In order to give a clearer comparison between the different methods, we apply the same Non-maxima suppression step for all methods and vary the threshold from 0.0 to 0.9 with step 0.01 for each method. The F1-score is computed for each threshold and the corresponding F1-score curves for each method in the 4 simulated edge images can be found in Figure 12. From Figure 12 we can see that the F1-score curves of GRHED are above the F1-score curves of the other methods. In addition, flat areas of F1-score curves of GRHED indicate greater stability regarding the threshold choice than with other approaches. Although the best F1-scores that can be obtained by GR and HED are comparable, the performances of HED are less sensitive to the choice of the threshold.

The optimal edge maps obtained by GR, HED, HED-log and GRHED in those synthetic images with contrast 1.2 and 1.4 (using the threshold which gives the best F1-score) can be found in Figure 13 and Figure 14. From Figure 13 and Figure 14 we can see that the ability of GRHED to preserve true edge pixels and suppress noise pixels is clearly greater on these images than when using the other methods; the GRHED method detects most true edge pixels while having the least number of false detections. It should be noted that although these edge maps are obtained using the optimal threshold, the edge maps obtained by HED and GRHED will remain reasonable with the threshold in a relatively large range, as can be deduced from Figure 12, especially for GRHED.

#### 4.2. Comparison of different algorithms on the speckled optical images in BSDS500-speckled

In order to give a more comprehensive comparison in more general situations between different methods, we compare GR, HED, HED-log and GRHED in the two hundred 1-look speckled optical images in BSDS500-speckled. For GR, we use  $\alpha = 2$  and  $\alpha = 4$ . For GRHED, GRHED with a single  $\alpha$  value

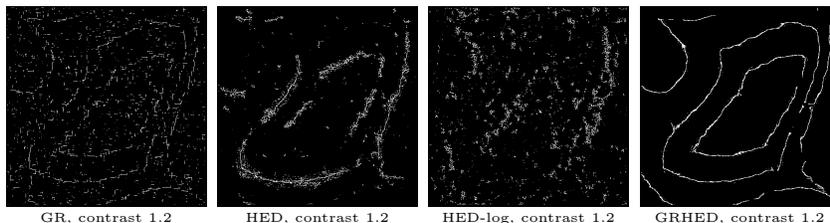


Figure 13: Optimal edge maps obtained by GR, HED, HED-log and GRHED in 1-look simulated edge images with contrast 1.2. For each method, the chosen threshold gives the best F1-score in this image. The size of the images is  $512 \times 512$  pixels.

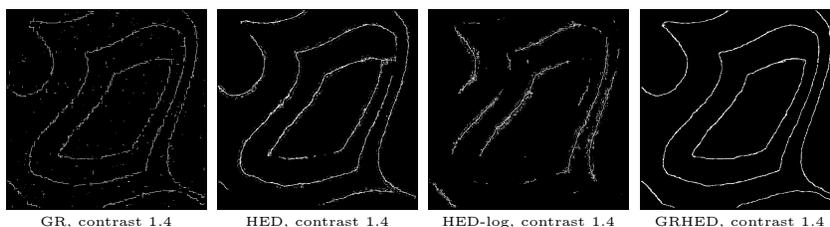


Figure 14: Optimal edge maps obtained by GR, HED, HED-log and GRHED in 1-look simulated edge images with contrast 1.4. For each method, the chosen threshold gives the best F1-score in this image. The size of the images is  $512 \times 512$  pixels.

and GRHED combining multiple  $\alpha$  values are all provided. Three criteria are used to compare different algorithms: Optimal Dataset Scale (ODS) F1 (fixed contour threshold for 200 images), Optimal Image Scale (OIS) F1 (best threshold for each image), and average precision (AP). The quantitative comparison can be found in table 2. From table 2 we can see that compared to GR, the CNN-based methods give much higher values for all three criteria. In addition, GRHED gives at least comparable or even better performances than HED and HED-log in the 200 1-look images, especially when combining multiple  $\alpha$  values. We also notice that GRHED combining multiple  $\alpha$  values yields better performances than GRHED using a single  $\alpha$  value, which is probably due to the richer information which is provided to HED convolutional layers.

The edge maps obtained by GR (with  $\alpha = 2$ ), GR (with  $\alpha = 4$ ), HED, HED-log, GRHED (with  $\alpha = 2$ ) and GRHED (combining  $\alpha = 2, 3, 4, 5$ ) on one speckled optical image can be found in Figure 15. For each method, the threshold is chosen to be the one corresponding to the ODS F1, which gives the best results in the 200 images. Specifically, the threshold used for each method can be found in table 3. It should be noted that using the threshold corresponding to the ODS F1, GRHED (combining  $\alpha = 2, 3, 4, 5$ ) will obtain near-optimal edge maps in the simulated edge images as shown in Figure 12. The ground truth of this image is the one provided by 1 labeler (there are usually five labelers for each image in BSDS500). From Figure 15 we can see that GRHED detects more true edge pixels than HED, and that both preserve

Table 2: The performances of different methods over the 200 speckled optical images in BSDS500-speckled.

methods	ODS (F1)	OIS (F1)	AP
GR ( $\alpha = 2$ )	0.5658	0.5852	0.5094
GR ( $\alpha = 4$ )	0.5894	0.6151	0.5286
HED	0.6461	0.6671	0.6981
HED-log	0.6258	0.6466	0.6838
GRHED ( $\alpha = 1$ )	0.6427	0.6523	0.6949
GRHED ( $\alpha = 2$ )	0.6603	0.6762	0.7208
GRHED ( $\alpha = 3$ )	0.6570	0.6783	0.7018
GRHED ( $\alpha = 4$ )	0.6552	0.6729	0.7050
GRHED ( $\alpha = 5$ )	0.6492	0.6687	0.6897
GRHED ( $\alpha = 6$ )	0.6463	0.6660	0.6893
GRHED ( $\alpha = 2, 3, 4, 5$ )	0.6643	0.6826	0.7109
GRHED ( $\alpha = 1, 2, 3, 4, 5, 6$ )	0.6643	0.6832	0.7070

more true edge pixels while detecting less false detections than GR. Using HED layers to postprocess the feature maps computed by GR, the GRHED is able to strengthen true edge pixels and suppress false detections.

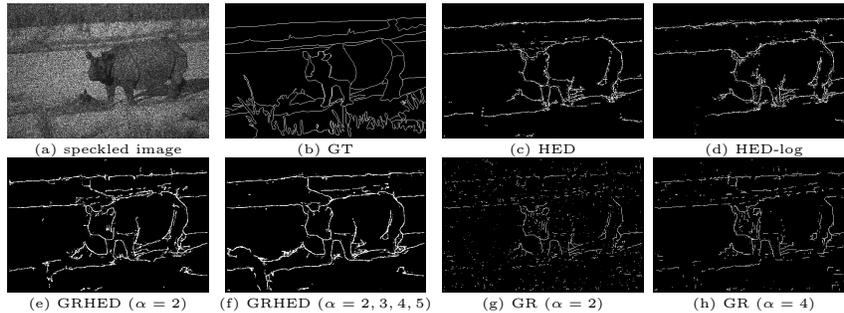


Figure 15: Comparison of different edge detectors on 1-look speckled optical images.

Table 3: The threshold corresponding to the ODS F1-score for each method.

methods	HED	HED-log	GRHED ( $\alpha = 2$ )	GRHED ( $\alpha = 2, 3, 4, 5$ )	GR ( $\alpha = 2$ )	GR ( $\alpha = 4$ )
threshold (ODS F1)	0.5666	0.5306	0.5686	0.5516	0.3137	0.2745

### 4.3. Comparison of different methods in 1-look SAR images

Though the efficiency of GRHED has been demonstrated in both simulated edge images and speckled optical images, demonstrating its ability to detect edges in real SAR images is the most important point in practice.

#### 4.3.1. Setting thresholds according to a given probability of false alarms

Usually, detection thresholds for SAR images are set according to a chosen probability of false alarms (pfa). This is only possible when the detectors have a constant false alarm rate (CFAR). Therefore, it is important to study whether HED, HED-log and GRHED have CFAR for SAR images. The CFAR property can be checked experimentally by plotting the histograms of their gradient magnitude fields computed in noise images having different mean values. A method is considered to have CFAR if the histograms of its gradient magnitude fields computed in all images overlap well. We plot the histograms of the magnitude fields produced by GR, HED, HED-log and GRHED in 160 1-look noise images of size  $1024 \times 1024$  pixels in Figure 16. The square root of the mean intensity values (proportional to the amplitude) of these noise images can take the following 8 values: 50, 80, 120, 150, 180, 200, 230 and 250. For each possible mean value, there are 20 random realizations. From Figure 16 we can see that GR and GRHED both hold CFAR, but HED and HED-log do not.

Next, we set the detection thresholds according to a given pfa (even though HED and HED-log do not have CFAR for SAR images). The threshold corresponding to a given probability of false alarms for GR, HED, HED-log and GRHED is estimated from 160 noise images as shown in table 4. Notice that for different methods, we will use the testing threshold corresponding to the pfa ( $10^{-5}$ , fifth column in table 4).

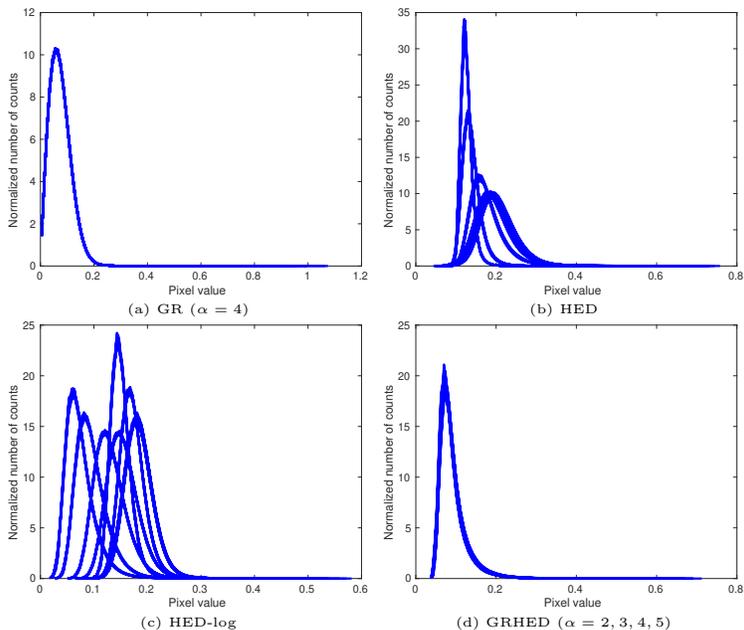


Figure 16: The histograms of the gradient magnitude fields produced by GR, HED, HED-log and GRHED in 160 1-look pure noise images of size  $1024 \times 1024$  pixels. There are 8 possibilities for the square root of the mean intensity values of these images, namely, 50, 80, 120, 150, 180, 200, 230 and 250. For noise images with each mean intensity value, there are 20 random realizations.

Table 4: The threshold corresponding to a given pfa for each method.

pfa	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$
threshold (GR)	0.18	0.22	0.26	0.29	0.31
threshold (HED)	0.3	0.38	0.48	0.57	0.63
threshold (HED-log)	0.23	0.27	0.3	0.34	0.39
threshold (GRHED)	0.17	0.26	0.37	0.45	0.52

#### 4.3.2. Comparison of different algorithms in synthetic realistic SAR images

It is usually difficult to annotate the edges in real SAR images due to very strong multiplicative noise. In order to give a quantitative evaluation on the performances of HED, HED-log, GRHED, and GR in images with targets similar to those in real SAR images, we use the ground truth which is obtained by applying HED-clean (HED trained on the grayscale images converted from clean natural images in BSDS500) to a SAR image with very little speckle noise. This SAR image with reduced speckle is obtained by averaging a large amount of well

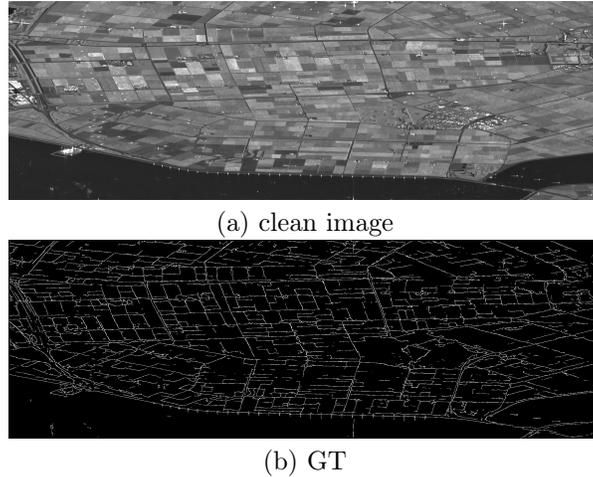


Figure 17: A denoised multi-look image (Leystad, Sentinel 1) and its ground truth. The size of the image is  $1024 \times 3072$  pixels.

registered Sentinel-1 images (equivalent to temporal multi-looking). Although not justified in changing areas, this temporal multi-looking allows a strong reduction of the speckle in stable ones like roads, urban areas, etc. To overcome the varying residual noise, a final despeckling step is applied [38]. The image is then converted to an 8-bits image using a clipping between  $[0, 255]$  with a threshold given by the mean value of the image plus three times its standard deviation. This image is a natural image, but contains targets similar to those in real SAR images. The multi-temporal despeckled SAR image and its associated ground truth can be found in Figure 17.

F1-score curves computed for GR, HED, HED-log and GRHED for the synthetic realistic 1-look SAR image can be found in Figure 18, where the 1-look SAR image is obtained by multiplying the clean SAR image with 1-look speckle noise. What can be seen from Figure 18 is that the F1-score curve of GRHED is above the F1-score curves of all the other methods. We observe also that the performances of GRHED are relatively robust to the choice of the threshold. In addition, both HED and HED-log are shown to be more powerful than GR to detect edges in complex situations. The edge maps obtained with the threshold corresponding to pfa ( $10^{-5}$ ) for different methods (for GR, the threshold is corresponding to pfa  $10^{-3}$  in order to preserve more true edges) can be found in Figure 19. From Figure 19 we can see that GRHED detects more true edge pixels than the other methods and the F1-scores of GRHED is higher than with other methods. In addition, HED also provides competitive edge detection results. GR detects more false detections probably because of the threshold chosen from a higher pfa, but it should be noted that the F1-score of GR is close to its optimal value, according to Figure 18. The method HED-log yields significantly more false detection, probably as a result of not having CFAR. Due

to the poor performances of HED-log, we do not use it for comparison in the following.

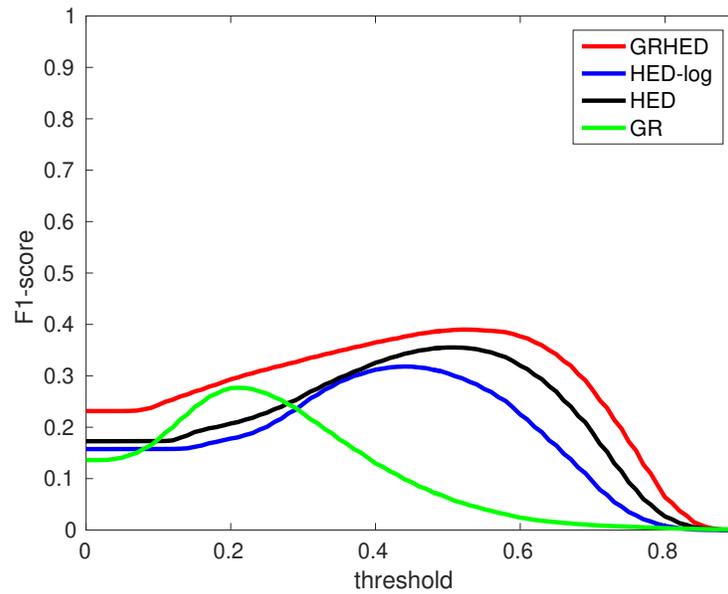
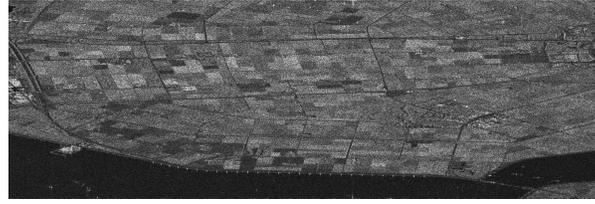
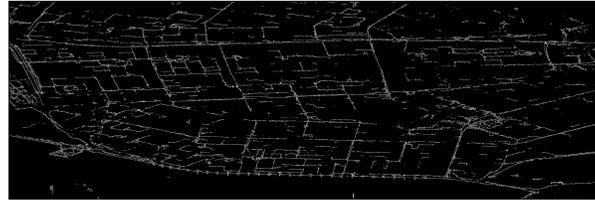


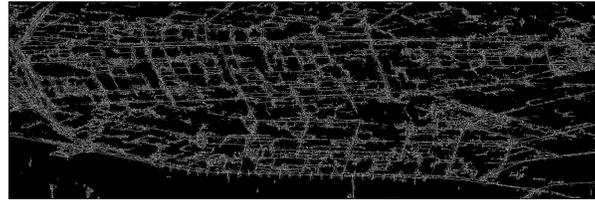
Figure 18: F1-score curves computed for GR, HED, HED-log and GRHED on a 1-look synthetic SAR image of size  $1024 \times 3072$  pixels. The 1-look synthetic SAR image is obtained by multiplying the clean SAR image in Fig. 17 with 1-look speckle noise.



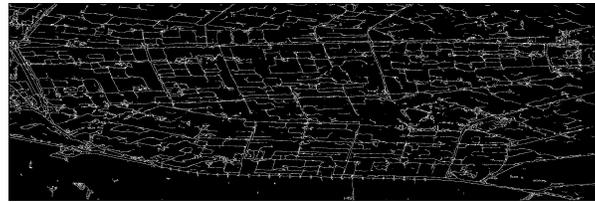
(a) 1-look image



(b) HED (F1-score 0.34)



(c) HED-log (F1-score 0.28)



(d) GRHED (F1-score 0.38)



(e) GR (F1-score 0.28)

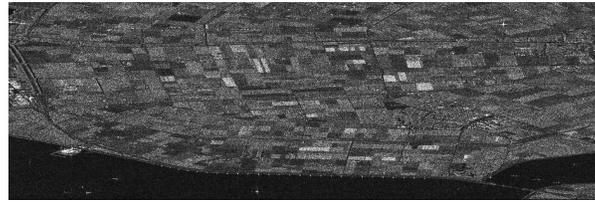
Figure 19: Edge maps computed with a threshold corresponding to  $pfa (10^{-5})$  on a synthetic realistic 1-look image (Leystad, Sentinel 1) for different methods. For GR, we use the threshold corresponding to  $pfa (10^{-3})$ , which is very close to the threshold corresponding to the best F1-score.

#### 4.3.3. Further comparisons on real SAR images

In this part, we test the efficiency of GRHED with two 1-look real SAR images. We tested the efficiency of the state-of-art edge detectors AMDR [15] in

SAR images, but we found it not suitable for such complex and noisy situations and did not investigate its performances further. We then compare HED and GRHED with GR on a 1-look real SAR image (Lelystad, Sentinel 1) of size  $1024 \times 3072$  pixels as shown in Figure 20. From Figure 20 we can see that though GR is able to detect many true edges, but it is not able to provide a good separation between true edge pixels and noise pixels. Therefore, there are also many false detections in the edge maps produced by GR. In comparison, both HED and GRHED detect many true edge pixels with clearly less false detections. However, HED produces spurious detections in bright areas (probably because of the lack of such zones in the training set). On the other hand, GRHED does not suffer from a similar problem. On this experiment for GRHED, the number of false detections is smaller and more evenly distributed, and edges appear to be better connected.

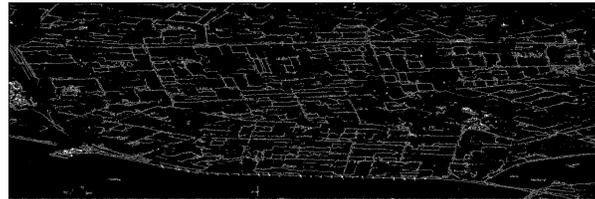
We compare GR, HED, and GRHED in another 1-look real SAR image (San Francisco, TerraSAR-X) of size  $2048 \times 2048$  pixels. The edge maps obtained by these methods can be found in Figure 21. Again, the weakness of GR is that it is not able to separate the true edge pixels and noise pixels efficiently. For HED, since many homogeneous areas with high mean intensity values exist in the image, and these kinds of areas do not exist in the training images, it is difficult for HED to perform effective edge detection on them. Therefore, the edge map outputted by HED has spurious detections in many bright areas. In comparison, those bright areas do not cause troubles to GRHED because the gradient distribution computed by GR will not be influenced by the mean intensity values of homogeneous areas. What is more important, GRHED is able to separate true edge pixels and false edge pixels efficiently.



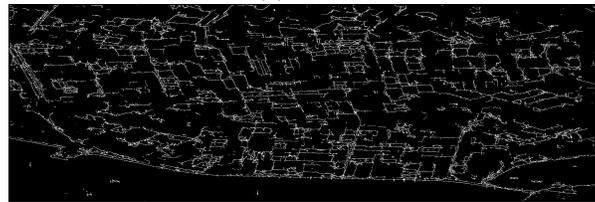
(a) 1-look image



(b) GR



(c) HED



(d) GRHED ( $\alpha = 2, 3, 4, 5$ )

Figure 20: Comparison of different methods on a 1-look real SAR image (Leystad, Sentinel 1). The size of the image is  $1024 \times 3072$  pixels.

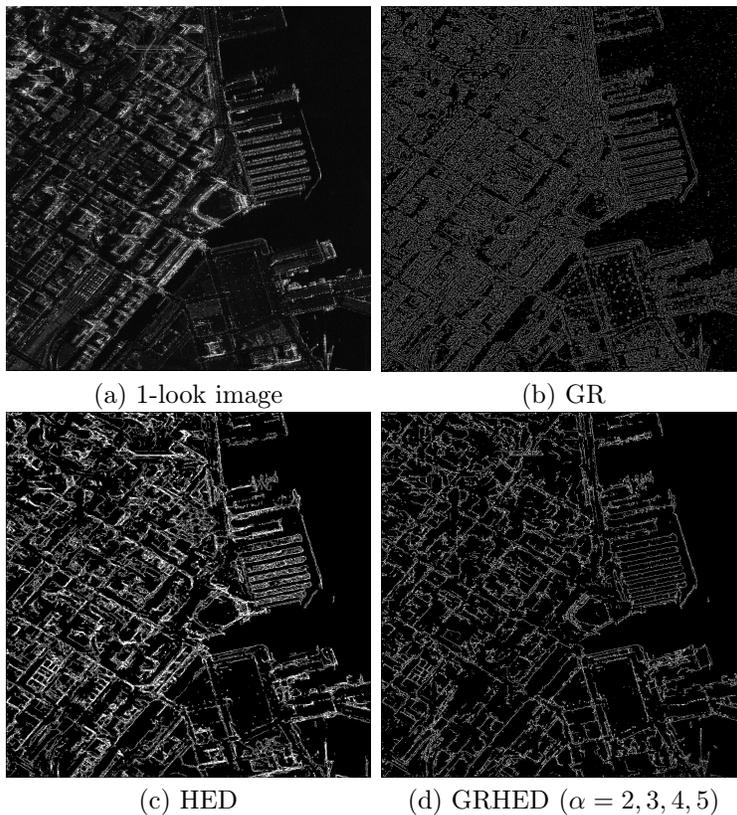


Figure 21: Comparison of different methods on a 1-look real SAR image (San Francisco, TerraSAR-X). The size of the image is  $2048 \times 2048$  pixels.

## 5. Conclusion

In this paper we addressed the challenging task of edge detection in 1-look real SAR images. Leveraging the available optical dataset, we proposed a framework enabling CNN models trained using simulated SAR dataset to work in real SAR images. By introducing the fixed hand-crafted layer (GR) instead of a learnable one, the proposed CNN edge detector GRHED is much less influenced by the gap between speckled optical and real SAR images, which can be seen from the comparison of HED and GRHED in 1-look real SAR images. The hand-crafted layer defined by a ratio based gradient computation method ensures that nearly all kinds of gradient distributions of real SAR images are included in those of speckled optical images. A straightforward observation is that nearly all the pixel values in the GR feature maps of real SAR images are within the range of pixel values of the gradient feature maps computed in the training dataset, where we assume that the range of pixel values depends on the underlying distributions. However, it should be pointed out that edges with

extremely high contrast may appear in real SAR images due to the bright homogeneous areas. The gradient distributions corresponding to these edges may not be contained in those of the training dataset. In addition, introducing the hand-crafted layer improves the performances of models composed of fully learnable layers, as can be seen from the comparison of HED and GRHED in synthetic edge images and 200 speckled optical images. From all the experiments we can see that GRHED outperforms existing edge detectors a lot, especially in the case of 1-look real SAR images.

Among the points that have not been addressed in this work and will be addressed in further works, we can mention the followings. First, the spatial correlation of the noise on real images has not been addressed and probably leads to a decreasing of the CNN performances. The method of [39] could be an interesting approach to take it into account. Secondly, the specific features of SAR images like bright points and lines due to strong backscatterings of diedral or triedral structures do not exactly correspond to edges. Therefore dedicated detectors should be developed for these specific structures to be combined with edge detectors.

### Source code

The source code of GRHED is available at <https://github.com/ChenguangTelecom/GRHED>.

### Fundings

This work is supported by the China Scholarship Council (File No. 201606270202), the National Natural Science Foundation of China (NSFC) (Grant No.61771014) and Télécom Paris, Institut Polytechnique de Paris.

### References

- [1] Q. Wei, D. Feng, Extracting line features in sar images through image edge fields, *IEEE Geoscience and Remote Sensing Letters* 13 (2016) 540–544.
- [2] Q. Wei, D. Feng, W. Zheng, J. Zheng, Rapid line-extraction method for sar images based on edge-field, *IEEE Geoscience and Remote Sensing Letters* 14 (2017) 1865–1869.
- [3] P. Yu, A. Qin, D. A. Clausi, Unsupervised polarimetric sar image segmentation and classification using region growing with edge penalty, *IEEE Transactions on Geoscience and Remote Sensing* 50 (2012) 1302–1317.
- [4] H. Song, B. Huang, K. Zhang, A globally statistical active contour model for segmentation of oil slick in sar imagery, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 6 (2013) 2402–2409.

- [5] J.-S. Lee, I. Jurkevich, Coastline detection and tracing in sar images, *IEEE Transactions on Geoscience and Remote Sensing* 28 (1990) 662–668.
- [6] C. Liu, Y. Xiao, J. Yang, A coastline detection method in polarimetric sar images mixing the region-based and edge-based active contour models, *IEEE Transactions on Geoscience and Remote Sensing* 55 (2017) 3735–3747.
- [7] T. Chen, L. Chen, Y. Su, A sar image registration method based on pixel migration of edge-point features, *IEEE Geoscience and Remote Sensing Letters* 11 (2014) 906–910.
- [8] H. Zhang, W. Ni, W. Yan, J. Wu, S. Li, Robust sar image registration based on edge matching and refined coherent point drift, *IEEE Geoscience and Remote Sensing Letters* 12 (2015) 2115–2119.
- [9] M. Dai, C. Peng, A. Chan, D. Loguinov, Bayesian wavelet shrinkage with edge detection for sar image despeckling, *IEEE Transactions on Geoscience and Remote Sensing* 42 (2004) 1642–1648.
- [10] R. Touzi, A. Lopes, P. Bousquet, A statistical and geometrical edge detection for sar images, *IEEE Transactions on Geoscience and Remote Sensing* 26 (1988) 764–773. doi:10.1109/36.7708.
- [11] R. Fjørtoft, A. Lopes, P. Marthon, E. Cubero-Castan, An optimal multi-edge detector for sar image segmentation, *IEEE Transactions on Geoscience and Remote Sensing* 36 (1998) 793–802. doi:10.1109/36.673672.
- [12] P.-L. Shui, D. Cheng, Edge detector of sar images using gaussian-gamma-shaped bi-windows, *IEEE Geoscience and Remote Sensing Letters* 9 (2012) 846–850. doi:10.1109/LGRS.2012.2184521.
- [13] Q.-R. Wei, D.-Z. Feng, H. Xie, Edge detector of sar images using crater-shaped window with edge compensation strategy, *IEEE Geoscience and Remote Sensing Letters* 13 (2016) 38–42.
- [14] J. Canny, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8* (1986) 679–698.
- [15] P. Shui, S. Fan, Sar image edge detection robust to isolated strong scatterers using anisotropic morphological directional ratio test, *IEEE Access* 6 (2018) 37272–37285.
- [16] G. Bertasius, J. Shi, L. Torresani, Deepedge: A multi-scale bifurcated deep network for top-down contour detection, in: *2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4380–4389.
- [17] W. Shen, X. Wang, Y. Wang, X. Bai, Z. Zhang, Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection, in: *2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3982–3991.

- [18] S. Xie, Z. Tu, Holistically nested edge detection, in: 2015 IEEE International Conference on Computer Vision, 2015, pp. 1395–1403.
- [19] S. Xie, Z. Tu, Holistically-nested edge detection, *International Journal of Computer Vision* 125 (2017) 3–18.
- [20] J. Yang, B. Price, S. Cohen, H. Lee, M.-H. Yang, Object contour detection with a fully convolutional encoder-decoder network, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 193–202.
- [21] D. Xu, W. Ouyang, X. Mameda-Pineda, E. Ricci, X. Wang, N. Sebe, Learning deep structured multi-scale features using attention-gated crfs for contour prediction, in: 2017 Conference on Neural Information Processing Systems (NIPS 2017), 2017.
- [22] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, X. Bai, Richer convolutional features for edge detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1939–1946.
- [23] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bien, L. Zhang, X. Bai, J. Tang, Richer convolutional features for edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2019) 1939–1946.
- [24] J. Kittler, On the accuracy of the sobel edge detector, *Image and Vision Computing* 1 (1983) 37–42.
- [25] S. Konishi, A. L. Yuille, J. M. Coughlan, S. Zhu, Statistical edge detection: Learning and evaluating edge cues, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003) 57–74.
- [26] D. R. Martin, C. C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004) 530–549.
- [27] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (2011) 898–916.
- [28] P. Dollar, C. L. Zitnick, Fast edge detection using structured forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (2015) 1558–1570.
- [29] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, F. Tupin, Sar-sift: A sift-like algorithm for sar images, *IEEE Transactions on Geoscience and Remote Sensing* 53 (2015) 453–466. doi:10.1109/TGRS.2014.2323552.
- [30] J. Goodman, Statistical properties of laser speckle patterns, Vol. ch. 2, *Laser Speckle and Related Phenomena*, 1975.

- [31] J. Long, E. Shelhamer, T. Darrel, Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
- [32] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2017) 640–651.
- [33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.
- [34] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: International Conference on Artificial Intelligence and Statistics (AISTATS), 2015.
- [35] C. Liu, R. Abergel, Y. Gousseau, F. Tupin, Ldsar, a markovian a contrario framework for line segment detection in sar images, *Pattern Recognition* 98 (2020). doi:<https://doi.org/10.1016/j.patcog.2019.107034>.
- [36] S. Dougherty, K. W. Bozyer, C. Kranenburg, Roc curves evaluation of edge detector performance, in: Proceedings 1998 International Conference on Image Processing, Vol. 2, 1998, pp. 525–529.
- [37] K. Bowyer, C. Kranenburg, S. Dougherty, Edge detector evaluation using empirical roc curves, in: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 1999, p. 359.
- [38] C. Deledalle, L. Denis, S. Tabti, F. Tupin, MuLoG, or how to apply gaussian denoisers to multi-channel sar speckle reduction?, *IEEE Transactions on Image Processing* 26 (9) (2017) 4389–4403. doi:[10.1109/TIP.2017.2713946](https://doi.org/10.1109/TIP.2017.2713946).
- [39] A. Lapini, T. Bianchi, F. Argenti, L. Alparone, Blind speckle decorrelation for SAR image despeckling, *IEEE Transactions on Geoscience and Remote Sensing* 52 (2) (2014).