

# Custom Dilated Edge Detection Filters

Victor BOGDAN  
West University of  
Timișoara  
Bd. V. Pârvan 4  
Timișoara, RO-300223  
Romania.  
victor.bogdan97@e-  
uvt.ro

Cosmin BONCHIȘ  
West University of  
Timișoara and  
The eAustria Research  
Institute,  
Bd. V. Pârvan 4, 045B,  
Timișoara, RO-300223  
Romania.  
cosmin.bonchis@e-uvt.ro

Ciprian ORHEI  
Politehnica University of  
Timișoara,  
Timișoara, RO-300223  
Romania.  
ciprian.orhei@cm.upt.ro

## ABSTRACT

Edge detection is a basic and fundamental function in image processing. Inspired by the new dilated convolution techniques which have impressive results in machine learning, we discuss here the idea of dilating the standard filters which are used to compute the gradient of an image. We compare the result of the proposed dilated filters with original filters and extend or custom variants of them. We also present the impact of dilatation on a complex edge detection algorithm, like Canny.

## Keywords

Dilated filters, edge detection operator, edge detection, Canny algorithm.

## 1 INTRODUCTION

The edge of an image is the most basic feature on the image and has been intensively researched. A variety of mathematical methods have been used to identify points in which the image brightness changes sharply or has discontinuities. This is a fundamental tool in image processing, image analysis, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

Standard edge detection filters are built for highlighting intensity change boundaries in the near neighborhood image regions. The most frequently used methods we can find in: [Sobel and Feldman, 1973, Roberts, 1963, Prewitt, 1970, Scharr, 2000]. Those filters are used in many computer vision algorithms which rely on edge detection for application as face recognition, target recognition, obstacle detection, image compression etc.

The extended version of the standard filters can be found in many research article in the literature. [Gupta and Mazumdar, 2013, Aybar et al., 2006] contains the mathematical proof of how to extend the Sobel operator and the performance of that extension. In [Lateef, 2008] we can find the definition of extended Sobel and Prewitt operators to a  $5 \times 5$  mask in order to obtain more continuous edges. In [Levkine, 2012] we are presented with the analytical extension of Prewitt, Sobel and Scharr operators. [Kekre and Gharge, 2010] uses the extension of Prewitt, Kirsch and Sobel operators to increase the tumor area in mammography images. In all these papers, we found that a bigger filter region is useful in order to find more accurate edges.

Merging those extension ideas with the recently dilated convolution techniques in machine learning, we propose not to extend but to simply **dilate** standard edge detection filters mentioned above. The dilated convolution methods have been proven very beneficial in many highly cited computer-vision papers: for small objects detection [Hamaguchi et al., 2018], in dense prediction tasks [Chen et al., 2018, Chen et al., 2015], on prediction without losing resolution [Yu and Koltun, 2016], for feature classifications in time series [Yazdanbakhsh and Dick, 2019] or beneficial for context aggregation [Zhao et al., 2017].

In this paper, we define an expansion of filters, by simply adding zeros in the expansion gaps and we call it dilation. This method of dilation is neither from the well known mathematical morphological sense, from [Haralick et al., 1987], nor the geometric extension of the kernels discussed above.

To evaluate the dilation benefits to an edge detection filter, we restrict our analysis only to the first order derivative gradient-based edge detection filters. The details regarding the extended versions of some edge detection filters selected by us can be found in section 2. In section 3, we present the defined dilated filters that we will use in this paper. Sections 4 and 5 highlight the results of our hypothesis regarding dilating of the filters, rather than extending them.

## 2 PRELIMINARIES

In this paper, we will present the dilated filters Prewitt [Prewitt, 1970], Scharr [Scharr, 2000]

and Sobel [Sobel and Feldman, 1973], used to compute the gradient of an image in order to find its edges. We compare the result of our dilated filters with the standard and extended filters presented in [Gupta and Mazumdar, 2013, Lateef, 2008, Levkine, 2012, Kekre and Gharge, 2010, Aybar et al., 2006], (presented in Figures 1 to 6), using different image sets. To determine the impact of the dilated filters, we modify the Canny Edge detection algorithm [Canny, 1986] in order to use the dilated or extended version of each filter for calculating gradients. For a better comparison of the results, we used the BSDS500 benchmark tool and image sets from [Arbelaez et al., 2011].

### Gradient operators

We consider the standard filters presented in Figures 1, 2, 3, where  $G_x$  and  $G_y$  are the gradients masks. In [Lateef, 2008, Kekre and Gharge, 2010, Levkine, 2012, Aybar et al., 2006] we can find the extended filter we will use to compare, Figures 4 to 6.

$$\begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$$

Figure 1: Scharr  $G_x$  and  $G_y$  kernels

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Figure 2: Prewitt  $G_x$  and  $G_y$  kernels

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figure 3: Sobel  $G_x$  and  $G_y$  kernels

We will consider the following standard formula where the  $G_x$  and  $G_y$  gradient components are used to define the gradient magnitude  $|G|$  in Equation 1.

$$|G| = \sqrt{G_x^2 + G_y^2}. \quad (1)$$

We use the edge detection algorithm steps, presented in [Woods, 2011], which are generally used to convolve filters with a source image in order to obtain the edge image:

$$\begin{bmatrix} 1 & 1 & 0 & -1 & -1 \\ 2 & 2 & 0 & -2 & -2 \\ 3 & 6 & 0 & -6 & -3 \\ 2 & 2 & 0 & -2 & -2 \\ 1 & 1 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 1 & 2 & 6 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -6 & -2 & -1 \\ -1 & -2 & -3 & -2 & -1 \end{bmatrix}$$

Figure 4:  $5 \times 5$  Scharr  $G_x$  and  $G_y$  kernels

$$\begin{bmatrix} 2 & 1 & 0 & -1 & -2 \\ 2 & 4 & 0 & -4 & -2 \\ 2 & 4 & 0 & -4 & -2 \\ 2 & 4 & 0 & -4 & -2 \\ 2 & 1 & 0 & -1 & -2 \end{bmatrix} \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 1 & 4 & 4 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ -2 & -4 & -4 & -4 & -2 \end{bmatrix}$$

Figure 5:  $5 \times 5$  Prewitt  $G_x$  and  $G_y$  kernels

$$\begin{bmatrix} 5 & 4 & 0 & -4 & -5 \\ 8 & 10 & 0 & -8 & -10 \\ 10 & 20 & 0 & -20 & -10 \\ 8 & 10 & 0 & -8 & -10 \\ 5 & 4 & 0 & -4 & -5 \end{bmatrix} \begin{bmatrix} 5 & 8 & 10 & 8 & 5 \\ 4 & 10 & 20 & 10 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ -4 & 10 & -20 & -8 & -4 \\ -5 & -8 & -10 & -8 & -5 \end{bmatrix}$$

Figure 6:  $5 \times 5$  Sobel  $G_x$  and  $G_y$  kernels

Step 1 Convert the image to gray-scale, preparing it as an input for Sobel filter convolution.

Step 2 Reduce the noise in the source image by applying the Gaussian filter, in order to obtain smooth continuous values.

Step 3 Applying the filters by convolving the gray-scale image with their kernels on the  $x$  and  $y$  axes and then applying the gradient magnitude (Equation 1).

Step 4 Each pixel, which has an intensity value higher or equal to a *threshold*, will have its value set to MaxValue (e.g. 255), else to 0, therefore the edges will be represented by the white pixels.

### Canny edge operator

Furthermore, the Canny edge detection algorithm [Canny, 1986] is a widely known and one of the most used edge detection algorithms.

Step 1 Convert the image to gray-scale

Step 2 Applying the Gaussian Filter, in order to obtain smooth continuous values.

Step 3 Applying the filters by convolving the gray-scale image with their kernels on the  $x$  and  $y$  axes.

Step 4 Non-maximum suppression, for edge thinning of the obtained results.

Step 5 Edge tracking by hysteresis using double threshold.

The double threshold is found by using the maximum pixel intensity in the input image and applying the formula from Equation 2, similar with the equation defined in [Xu et al., 2011].  $T_h$  is the upper threshold,  $T_l$  is the lower threshold and  $max(input)$  is the maximum pixel intensity in the input image. From our experiments the best results were obtained with the fixed weights values  $w_h = 0.7$  and  $w_l = 0.3$ .

$$\begin{aligned} T_h &= max(input) \times w_h \\ T_l &= T_h \times w_l \end{aligned} \quad (2)$$

### Benchmarking the edge operators

For highlighting the results obtained, we use BSDS500 [Arbelaez et al., 2011] which contains a dataset of natural images that have been manually segmented. The human annotations serve as ground truth for the benchmark for comparing different segmentation and boundary detection algorithms. For evaluating the images generated from algorithms to the ground truth images the Corresponding Pixel Metric (CPM) algorithm [Prieto and Allen, 2003] is used. This metric is reliable for correlating similarities with a small localization error in the detected edges. The metric first finds an optimal matching of the pixels between the edge images and then estimate the error produced by this matching.

The benchmark [Arbelaez et al., 2011] uses 500 test images, which are split in 3 different sets, each having at least 5 human segmented boundary ground-truth images.

For each image two quantities Precision (P) and Recall (R) will be computed, as were defined in [Sasaki, 2007]. Precision, with formula 3, represents the probability that a resulting edge/boundary pixel is a true edge/boundary pixel. Recall, with formula 4, represents the probability that a true edge/boundary pixel is detected. Where TP (True Positive) represents the number of matched edge pixel, FP (False Positive) the number of edge pixels which are incorrectly highlighted and FN (False Negative) the number of pixel that have not been detected.

$$P = \frac{TP}{TP + FP}. \quad (3)$$

$$R = \frac{TP}{TP + FN}. \quad (4)$$

Those two quantities are used to compute  $F$ -measure (F1-score) by applying the formula 5, as is defined in [Arbelaez et al., 2011].

$$F - measure = \frac{2 * TP}{2 * TP + FP + FN}. \quad (5)$$

### 3 DILATED FILTERS

One of the commonly used methods in detecting the edges in images is by convolving the initial image with an edge detection operator. The filter highlights the difference between the pixel intensities. To obtain better results, we can combine kernels which will result in a higher change in pixel intensity.

**Definition 1.** A dilated filter is obtained by expanding the original filter by a dilation factor/size.

In this paper, we propose to use dilated filters for detecting the edges in images that are obtained by the Definition 1. The standard  $3 \times 3$  kernels from Sobel, Prewitt and Scharr are dilated to  $5 \times 5$ ,  $7 \times 7$  filters, for both axes. By dilating the kernels, we propose to increase the distance between the pixels, distance which influences the result of the convolution. This expansion induces the possibility of finding stronger intensity changes in the image. When we dilate the kernels, we are filling the newly added positions with 0s, as one can see the Figures 7 to 12.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -2 & 0 & -1 \end{bmatrix}$$

Figure 7: Sobel  $5 \times 5$  dilated Gx and Gy filters

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -2 & 0 & 0 & -1 \end{bmatrix}$$

Figure 8: Sobel  $7 \times 7$  dilated Gx and Gy filters

With our approach, in most of the test cases, we obtain better edge detection results than the filters presented in

$$\begin{bmatrix} 3 & 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & -10 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} 3 & 0 & 10 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & -10 & 0 & -3 \end{bmatrix}$$

Figure 9: Scharr  $5 \times 5$  dilated Gx and Gy filters

[Gupta and Mazumdar, 2013] or [Levkine, 2012]. Another advantage that we noticed was the fact that even if the filter size has increased, the number of calculation hasn't, fact that resulted in low run time compared to extended  $5 \times 5$  or  $7 \times 7$  filters. Even if the proposed filters don't respect the geometrical gradient formulas and the rules of extension, the good results in edge finding suggested that dilating is a solution to take in consideration, rather than extending.

$$\begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 & 10 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & -10 & 0 & 0 & -3 \end{bmatrix}$$

Figure 10: Scharr  $7 \times 7$  dilated Gx and Gy filters

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 \end{bmatrix}$$

Figure 11: Prewitt  $5 \times 5$  dilated Gx and Gy filters

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix}$$

Figure 12: Prewitt  $7 \times 7$  dilated Gx and Gy filters

#### 4 FILTER CONVOLUTION RESULTS

The following section consists of comparisons between the results of convolving an image with the standard, extended and dilated filters presented until now. We

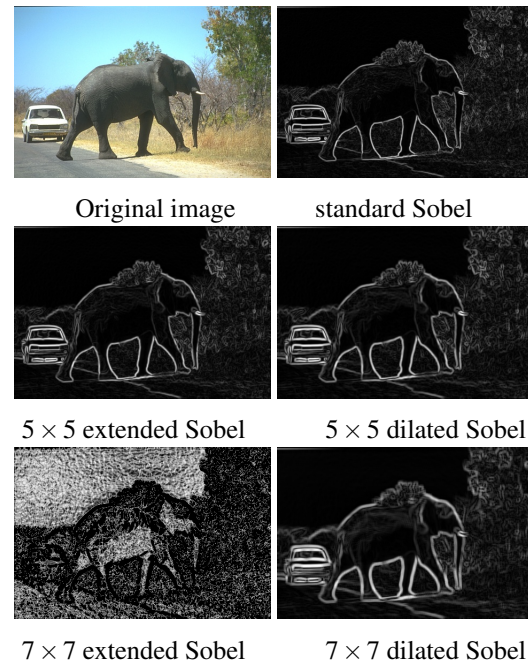


Figure 13: Standard, Extended and Dilated Sobel Results

present first some visual comparison results of the dilated filters, in Figures 13, 14, 15. By using the benchmarking tool from BSDS500 we show the statistical results by comparing the test images with ground truth, as presented in [Arbelaez et al., 2011].

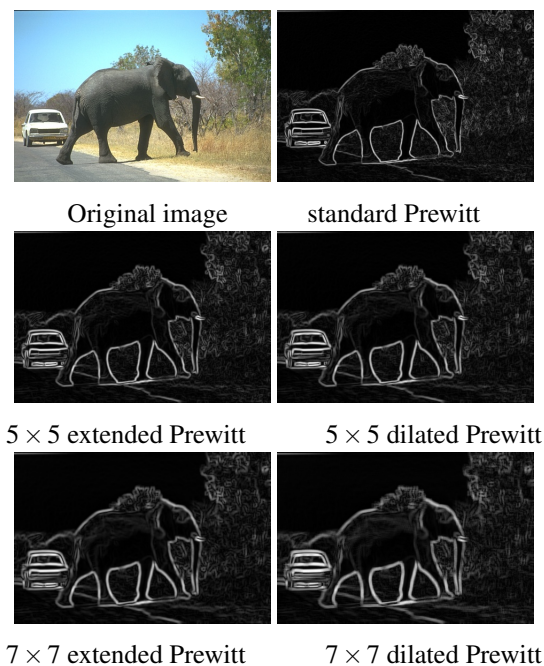


Figure 14: Standard, Extended and Dilated Prewitt Results

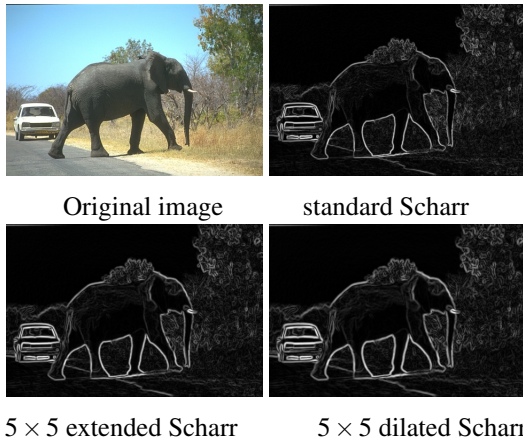


Figure 15: Standard, Extended and Dilated Scharr Results

### First results

From Figures 13 to 15 we can observe that when we convolve the original source image with the dilated filters, we obtain more pixels with a high gradient magnitude than the extended ones. Also, the more the dilation factor is increased, the more the edge images seem to lose the details while the edges seem to get blurry. Visually, there seems to be a compromise between details and the newly detected edges, see other detailed examples in Appendix in Figure 21.

We can observe some differences between the standard or extended filters and our dilated filters, however for a rigorous comparison we use the BSDS500 benchmark [Arbelaez et al., 2011], in order to observe the impact of the dilatation on a data base of ground truth images.

### Benchmark results

Our dilated filter results have been compared to the ground-truth images. The benchmark chooses a number of threshold values between 0 and 255 for creating a binary map of the edge algorithm output. The best overall F1-score is chosen from the variants of the output for each threshold. That gives us the rule for choosing the best possible result for each filter for each image in the set. For each threshold the two quantities Precision and Recall will be computed.

Tables 1 to 3 contain the results of the standard, extended and dilated filters on the BSDS500 train set. The overall recall, precision and F1-score represents the average for all samples.

We highlight that every **dilated filter**, with one exception the Prewitt extended  $7 \times 7$ , **obtained a better overall F1-score** than the extended filters. This can be explained by the sparsity of the dilated filter that induces more edges and less noise.

Another advantage of using the dilated filters instead of the standard or extended ones is given by runtime simulation results.

Filter Size	Overall Recall	Overall Precision	Overall F1-score
Standard $3 \times 3$	0.48723	0.54077	0.51261
Extended $5 \times 5$	0.79035	0.42298	0.55105
Dilated $5 \times 5$	0.78942	0.43077	<b>0.55738</b>
Extended $7 \times 7$	0.77057	0.46317	<b>0.57857</b>
Dilated $7 \times 7$	0.79008	0.44572	0.56992

Table 1: Prewitt Comparison on the Test Set

Filter Size	Overall Recall	Overall Precision	Overall F1-score
Standard $3 \times 3$	0.48714	0.53631	0.51054
Extended $5 \times 5$	0.79783	0.40453	0.53685
Dilated $5 \times 5$	0.78892	0.42243	<b>0.55023</b>

Table 2: Scharr Comparison on the Test Set

Filter Size	Overall Recall	Overall Precision	Overall F1-score
Standard $3 \times 3$	0.48863	0.53765	0.51197
Extended $5 \times 5$	0.79724	0.40666	0.53860
Dilated $5 \times 5$	0.79021	0.42658	<b>0.55407</b>
Extended $7 \times 7$	0.85249	0.17964	0.29675
Dilated $7 \times 7$	0.78627	0.44633	<b>0.56942</b>

Table 3: Sobel Comparison on the Test Set

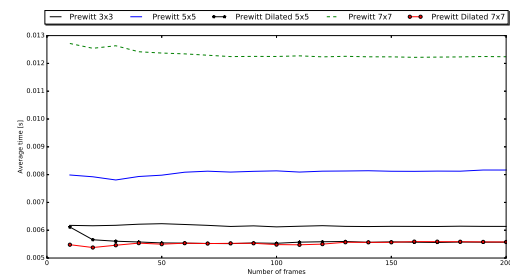


Figure 16: Prewitt average runtime

### Runtime results

We applied all the filters presented above by using our framework using Python and the results are presented in the graphics from Figures 16, 17, 18.

From the plots one can see that the runtime doesn't increase proportional with the size of the dilated filter. This stability can be explained by the fact that dilation

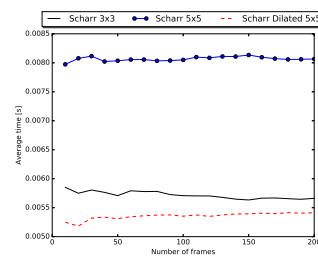


Figure 17: Scharr average runtime

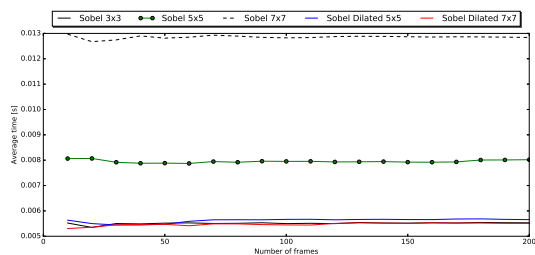


Figure 18: Sobel average runtime

doesn't increase the number of necessary operations to be executed.

In this section, we set forth that dilating the filters is a good compromise for detecting edges instead of extending them, discovering benefits regarding the edge versus noise output and runtime results.

## 5 CANNY SIMULATION RESULTS

In this section, we use the Canny edge detection algorithm to compare the results of convolving an image with the standard, extended and dilated Sobel filters. Similar with the previous analysis, we present a visual comparison, in Figure 19 and the statistic results using BSDS500 [Arbelaez et al., 2011], see the Tables 4 and 5.

### First results

From Figure 19, we can observe that applying the dilated Sobel filter in a more complex edge detection algorithm like Canny [Canny, 1986] produces visible enhancements of its results. We can also observe that by using the dilated filters the results contain less noise than the extended ones.

### Benchmark results

In Table 4, we show the results of the Canny edge detection algorithm using the standard, extended and dilated Sobel filters. We can notice that the custom dilated filters have better F1-scores than the standard or extended Sobel filters when they are used together with the Canny edge detection algorithm.

The results illustrated in Table 4 show that, by increasing the dilated factor, we can obtain a better F1-score, see the  $7 \times 7$  dilated results.

Filter Size	Overall Recall	Overall Precision	Overall F1-score
Standard $3 \times 3$	0.47400	0.54822	0.50842
Extended $5 \times 5$	0.99408	0.20711	0.34280
Dilated $5 \times 5$	0.59956	0.55094	<b>0.57422</b>
Extended $7 \times 7$	0.89121	0.17991	0.29938
Dilated $7 \times 7$	0.60456	0.56940	<b>0.58645</b>

Table 4: Canny using Sobel comparison on the Test Set

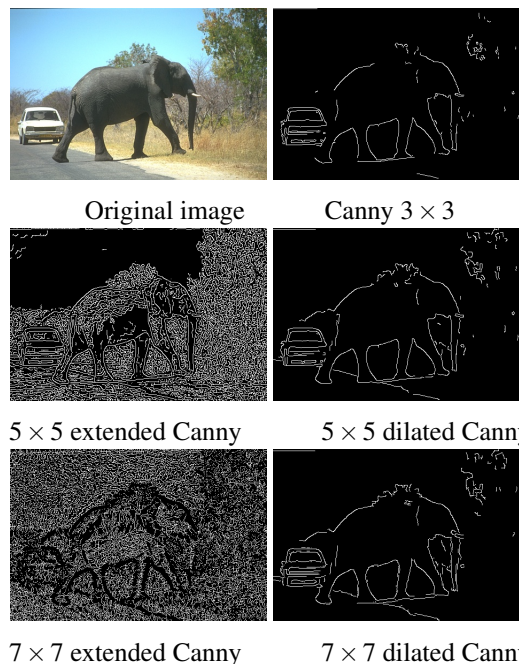


Figure 19: Standard, Extended and Dilated Canny Results

Filter Size	Overall Recall	Overall Precision	Overall F1-score
Standard $3 \times 3$	0.47400	0.54822	0.50842
Dilated $5 \times 5$	0.59956	0.55094	0.57422
<b>Dilated <math>7 \times 7</math></b>	<b>0.60456</b>	<b>0.56940</b>	<b>0.58645</b>
Dilated $9 \times 9$	0.6010	0.57216	0.58622
Dilated $11 \times 11$	0.6011	0.56266	0.58125
Dilated $13 \times 13$	0.5964	0.54197	0.56787
Dilated $15 \times 15$	0.5901	0.51935	0.55247

Table 5: Canny using Sobel dilated filters on the Test Set

By continuing to dilate the Sobel filter, we can also notice that the F1-score starts to decrease from a certain point. Therefore, in Table 5 one can see that  $9 \times 9$ ,  $11 \times 11$ ,  $13 \times 13$  and  $15 \times 15$  dilated Sobel filters obtained a significantly lower F1-score than the  $7 \times 7$  dilated filter. This is also the case for the overall precision, whereas the overall recall alternates but there are no significant differences.

### Run time results

From the simulation results, illustrated in Figure 20, we observe that neither in Canny simulation the runtime was not impacted by the size of the dilated filter.

The Canny approach validates our hypothesis that dilating the filters is better than extending them. We achieve here the good results which we expected from dilated filters regarding edge discovering and runtime.



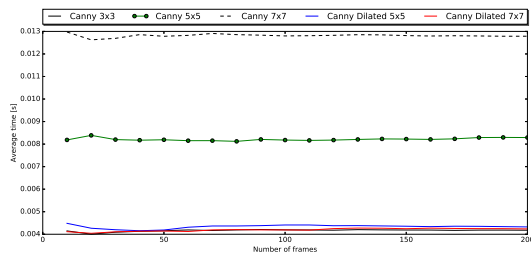


Figure 20: Canny average runtime

## 6 CONCLUSIONS AND FUTURE WORK

In both comparisons, statistical and visual, we can observe that by dilating the filters, rather than extending them, helps to find more edge pixels than the standard filters. By dilating the kernels of the Sobel or Scharr filter, we obtained a better recall and precision, which can be observed in Tables 2 and 3, and thus a better F1-score.

The biggest improvement of the dilated filters can be seen in the comparison between the extended and dilated versions of the Sobel filter in the Canny algorithm, see Table 4.

From the evolution of the metrics in Table 5 we can observe that we benefit from dilating the filters up to a certain point. Someone can say that  $9 \times 9$  is better than  $7 \times 7$  if we look at the Precision and F1-score. Therefore, we would say that the best dilation depends on the input images and the nature of application in which we desire edge detection.

By visually and statistically comparing the dilated filters with the standard and extended filters, we can conclude that the dilated filters can achieve better results, in most cases.

Because of the simple structure of the custom dilated filters, they are also a good choice when the runtime matters. The other filters from [Gupta and Mazumdar, 2013, Lateef, 2008, Levkine, 2012, Kekre and Gharge, 2010, Aybar et al., 2006] require a larger number of operations in order to return the resulting edge pixels, whereas the custom dilated filters have always the same number of operations for any extension.

From the experiments that have been done we saw that dilated filters might not be very efficient in images which contain many details because the dilation of the kernels can cause loss of details in images. Those images that have high differences of pixel intensities in a small range in different regions are the hard scenario for dilated filter and could expect more investigation.

As an open topic for a further investigation, it could consist in defining the similar dilated filters on second order derivative edge detection operators or more complex filters, with focus on the performance evaluation.

## 7 ACKNOWLEDGEMENTS

The authors would like to thank Oana Rotaru-Orhei for her comments and the three anonymous reviewers for their insightful suggestions. This work was partially supported by a grant of Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P4-ID-PCE-2016-0842, within PNCDI III.

## 8 REFERENCES

- [Arbelaez et al., 2011] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916.
- [Aybar et al., 2006] Aybar, E. et al. (2006). Sobel edge detection method for matlab. *Anadolu University, Porsuk Vocational School*, 26410.
- [Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.
- [Chen et al., 2015] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected crfs. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- [Chen et al., 2018] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848.
- [Gupta and Mazumdar, 2013] Gupta, S. and Mazumdar, S. G. (2013). Sobel edge detection algorithm. *International journal of computer science and management Research*, 2(2):1578–1583.
- [Hamaguchi et al., 2018] Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T., and Hikosaka, S. (2018). Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, pages 1442–1450. IEEE Computer Society.
- [Haralick et al., 1987] Haralick, R. M., Shapiro, L., and Lee, J. (1987). Morphological edge detection. *IEEE J. Robotics Automat.*, 3:142–155.
- [Kekre and Gharge, 2010] Kekre, H. and Gharge, S. (2010). Image segmentation using extended edge operator for mammographic images. *International journal on computer science and Engineering*, 2(4):1086–1091.

- [Lateef, 2008] Lateef, R. A. R. (2008). Expansion and implementation of a  $3 \times 3$  sobel and prewitt edge detection filter to a  $5 \times 5$  dimension filter. *Journal of Baghdad College of Economic sciences University*, (18):336–348.
- [Levkine, 2012] Levkine, G. (2012). Prewitt, sobel and scharr gradient  $5 \times 5$  convolution matrices. *Image Process. Articles*.
- [Prewitt, 1970] Prewitt, J. M. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19.
- [Prieto and Allen, 2003] Prieto, M. and Allen, A. (2003). A similarity metric for edge images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:1265–1273.
- [Roberts, 1963] Roberts, L. G. (1963). *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology.
- [Sasaki, 2007] Sasaki, Y. (2007). The truth of the f-measure. Technical report, School of Computer Science, University of Manchester.
- [Scharr, 2000] Scharr, H. (2000). *Optimal operators in digital image processing*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, Germany.
- [Sobel and Feldman, 1973] Sobel, I. and Feldman, G. (1973). *Pattern Classification and Scene Analysis*, pages 271–272.
- [Woods, 2011] Woods, J. W. (2011). *Multidimensional Signal, Image, and Video Processing and Coding, Second Edition*. Academic Press, Inc., Orlando, FL, USA, 2nd edition.
- [Xu et al., 2011] Xu, Q., Chakrabarti, C., and Karam, L. J. (2011). A distributed canny edge detector and its implementation on fpga. In *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pages 500–505.
- [Yazdanbakhsh and Dick, 2019] Yazdanbakhsh, O. and Dick, S. (2019). Multivariate time series classification using dilated convolutional neural network. *CoRR*, abs/1905.01697.
- [Yu and Koltun, 2016] Yu, F. and Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- [Zhao et al., 2017] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239. IEEE Computer Society.

## APPENDIX

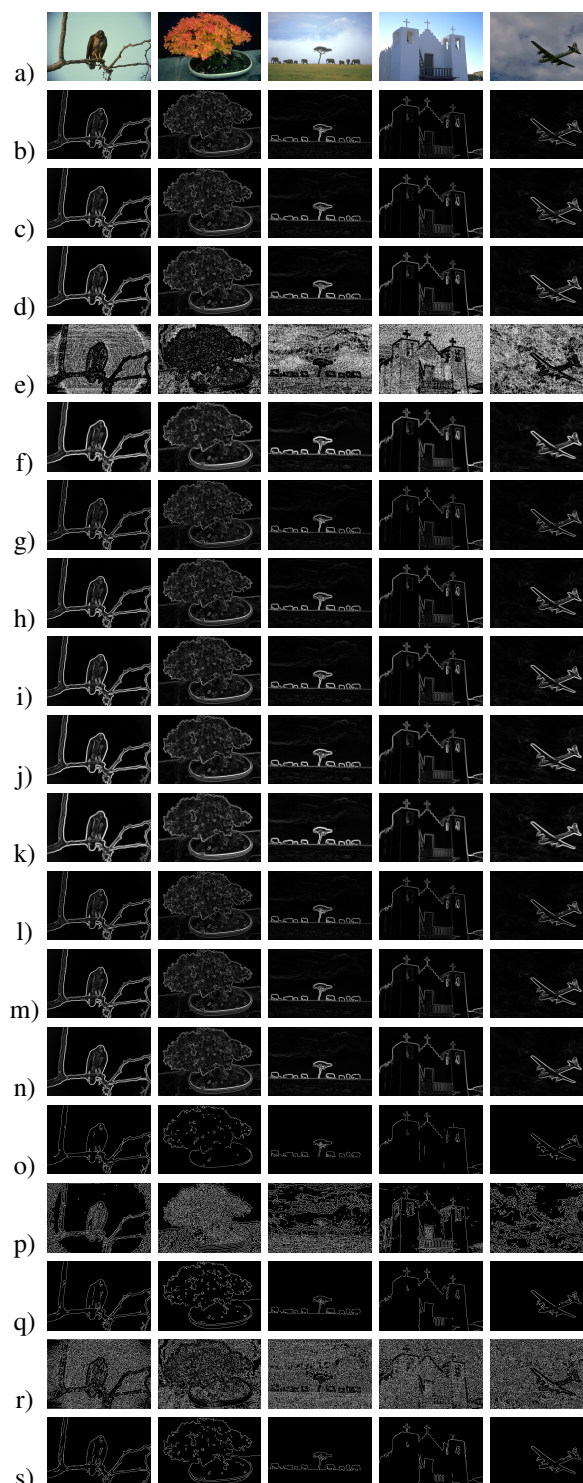


Figure 21: a) Original image, b)  $3 \times 3$  Sobel, c)  $5 \times 5$  extended Sobel, d)  $5 \times 5$  dilated Sobel, e)  $7 \times 7$  extended Sobel, f)  $7 \times 7$  dilated Sobel, g)  $3 \times 3$  Prewitt, h)  $5 \times 5$  extended Prewitt, i)  $5 \times 5$  dilated Prewitt, j)  $7 \times 7$  extended Prewitt, k)  $7 \times 7$  dilated Prewitt, l)  $3 \times 3$  Scharr, m)  $5 \times 5$  extended Scharr, n)  $5 \times 5$  dilated Scharr, o)  $3 \times 3$  Canny, p)  $5 \times 5$  extended Canny, q)  $5 \times 5$  dilated Canny, r)  $7 \times 7$  extended Canny, s)  $7 \times 7$  dilated Canny.