Published in Image Processing On Line on 2022–10–10.
Submitted on 2022–09–13, accepted on 2022–09–19.
ISSN 2105–1232 © 2022 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2022.423

A Brief Analysis of the Dense Extreme Inception Network for Edge Detection

Rafael Grompone von Gioi[1], Gregory Randall[2]

[1]Centre Borelli, Université Paris-Saclay, ENS Paris-Saclay, France CMLA, (grompone@ens-paris-saclay.fr)
[2]IIE, Fing, UdelaR, Uruguay (randall@fing.edu.uy)

Communicated by Thibaud Ehret *Demo edited by* Rafael Grompone von Gioi

Abstract

This work describes DexiNed, a Dense Extreme Inception Network for Edge Detection proposed by Xavier Soria, Edgar Riba and Angel Sappa in [IEEE Winter Conference on Applications of Computer Vision (WACV), 2020]. The network is organized in blocks that extract edges at different resolutions, which are then merged to produce a multiscale edge map. For training, the authors introduced an annotated dataset (BIPED) specifically designed for edge detection. We perform a brief analysis of the results produced by DexiNed, highlighting its quality but also indicating its limitations. Overall, DexiNed produces state-of-the-art results.

Source Code

The source code and documentation for this algorithm are available from the web page of this article[1]. Usage instructions are included in the `README.txt` file of the archive. This code is a minor modification from the original code of DexiNed available on the 'legacy' directory of the original GitHub repository[2].
This is an MLBriefs article, the source code has not been reviewed!

Keywords: image edge detection; neural network; HED; Xception

1 Introduction

Edge detection is a classic problem in computer vision and there is an extensive literature describing methods, metrics and comparisons [5]. In this work we will focus on the Dense Extreme Inception Network for Edge Detection (DexiNed) proposed by Xavier Soria, Edgar Riba and Angel Sappa [7]. DexiNed was inspired in the HED [10] and Xception [2] neural networks. The former is a multiscale edge detection method that uses the VGG16 [6] architecture as a backbone. On the other hand,

[1]https://doi.org/10.5201/ipol.2022.423
[2]https://github.com/xavysp/DexiNed, https://github.com/xavysp/DexiNed/tree/master/legacy, commit hash: 3fbbffbf4255cfcb192eb76ba9b9752c9dc99671

the fundamental idea behind Xception is to decouple the spatial processing from the inter-channel processing; this is obtained by CNN layers performing spatial or inter-channel convolutions (but not both) running in parallel, and then merged at the end of each block [2]. As HED, the DexiNed network is organized in blocks that work at different scales and the result is produced by the fusion of the edges detected at all scales. As Xception, DexiNed uses layers in parallel of spatial and inter-channel convolutions; nevertheless the inspiration is limited because, as we will see, DexiNed does not follow completely the decoupling between spatial and channel processing by using separable convolutions as proposed in Xception.

The authors emphasize the importance of training the network on a dataset annotated specifically for edge detection (bright/dark transitions on the image) as opposed to boundary/contour detection (object frontiers) or semantic segmentation. For this aim, they introduced the Barcelona Images for Perceptual Edge Detection (BIPED) [7] with careful edge annotation performed by experts. DexiNed was trained on the BIPED dataset.

DexiNed was originally introduced in 2020 [7], with the source code and trained parameters publicly available. Since then, the authors continued working on DexiNed, updating the code and presenting an updated manuscript [9]. At the time of this writing, the updated manuscript is still under review and the revised code is not yet stable. For this reason, here we describe and analyze the original version; all the experiments described here were performed using the "legacy" version included in the author's repository [8], which corresponds to the algorithm described in the WACV2020 paper [7] and which they refer to as DexiNed-v1.

2 The DexiNed Method

DexiNed is an image edge detector based on a deep neural network. Figure 1 presents the architecture. The input is a color image (three channels) of H columns and W rows. The trained parameters provided by the authors expect the input image in the BGR channel order and the pixel values at each position should have values in $[0, 255]$; these values must be normalized by subtracting the vector $(103.939, 116.779, 123.68)$ at each position before applying the image to the network. The input image dimensions must be multiples of 16.

DexiNed produces six "side output" corresponding to edge maps produced at different scales; each one is an $H \times W$ map with values between zero and one (often normalized to $[0, 255]$) indicating the confidence on the presence of an edge at each position. These side outputs are considered as intermediate results to compute the two main outputs: the "fused output" and the "average output". The fused output is produced by a convolutional layer with a 1×1 kernel taking as inputs the six side outputs. The average output is computed as the mean between all the side outputs and the fused output. Figure 2 illustrates the intermediate and main outputs of DexiNed on a sample image (Figure 3 shows the same outputs for a different set of trained parameters, as will be discussed in Section 3).

The DexiNed network is arranged into six blocks, each one composed of a repetition of sub-blocks, as illustrated in Figure 1. The first block takes as input the three channel input image. Each sub-block is composed of a convolutional layer with a 3×3 kernel (green boxes), batch normalization and ReLU activation functions (yellow boxes). The number of channels in each sub-block is indicated in the scheme. Note that there are some differences in the architecture of these six blocks; in particular the first two blocks are quite different from the other four (see in particular the presence or not of ReLU layers before the up-sampling blocks). There are also lateral connections named *Rconv* 1 to 4, *Addb_XX* and *conv_XX*. All these lateral connections are produced by convolutional layers with 1×1 kernels.

The blocks work at different scales: the first two work at half the input image full resolution; the

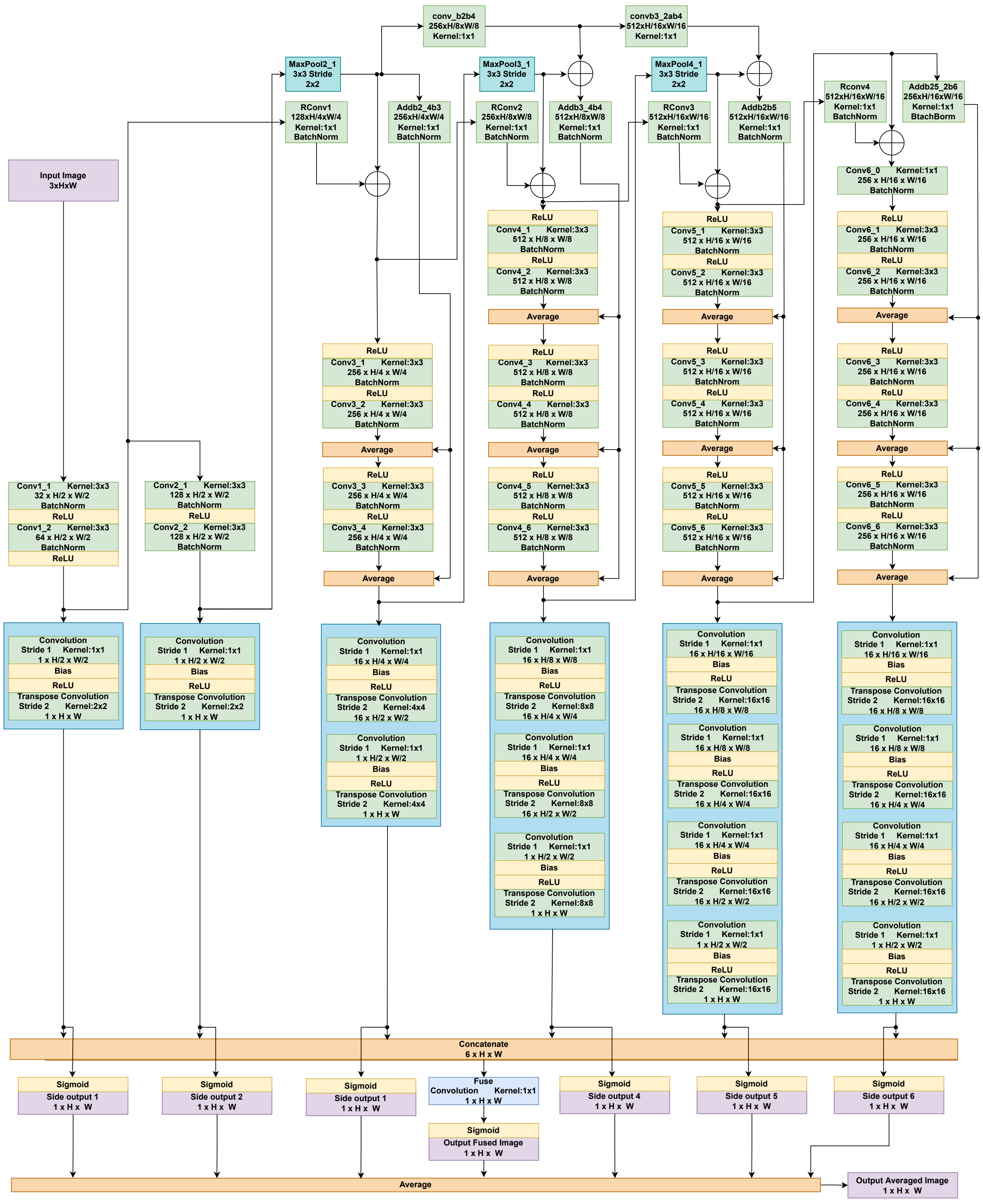


Figure 1: The DexiNed neural network architecture. In total, the DexiNed neural network has 34,744,385 parameters.

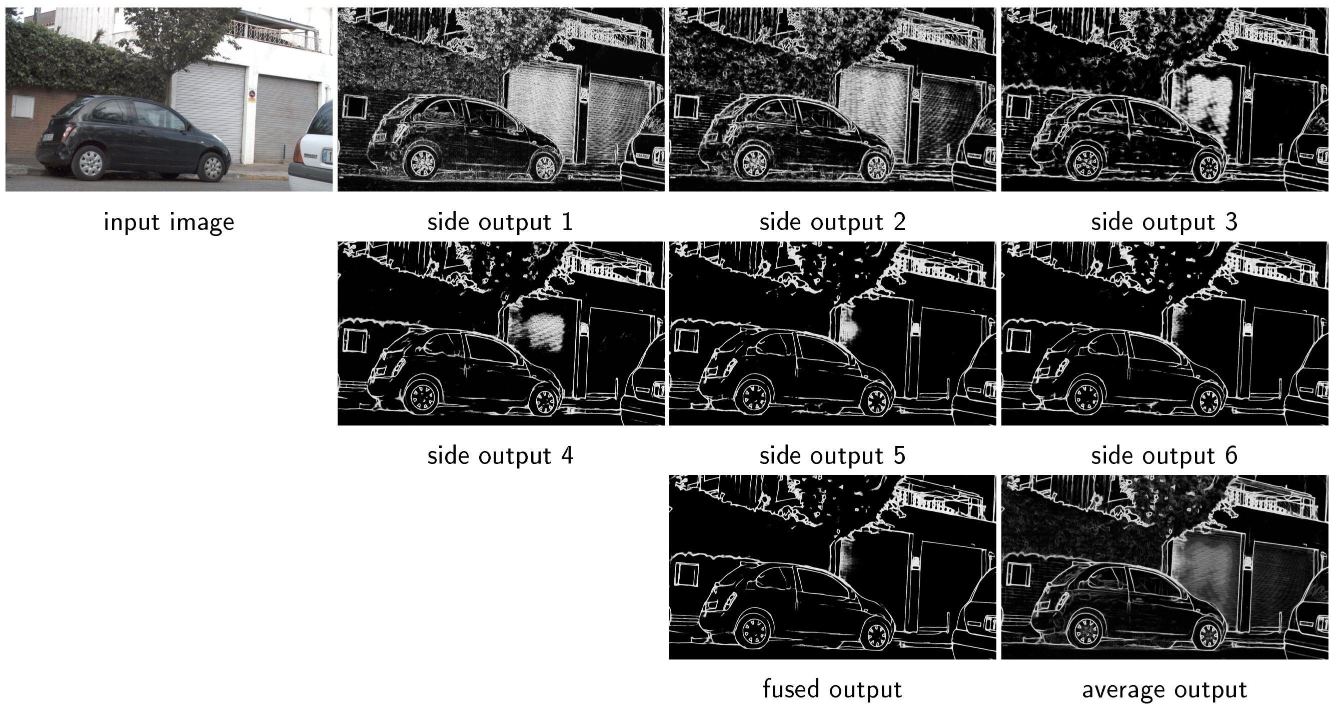


Figure 2: Examples of the eight outputs produced by DexiNed. The "side outputs" produce intermediate edge maps obtained by analyzing the input image at different scales. Those six edge maps are then fused into a multi-scale edge map called the "fused output". The "average output" is the average of the seven previous outputs, namely the six side outputs and the fused output. These results were obtained with the training set "train1" as provided by the authors' repository [8, see README.md at directory 'legacy'].

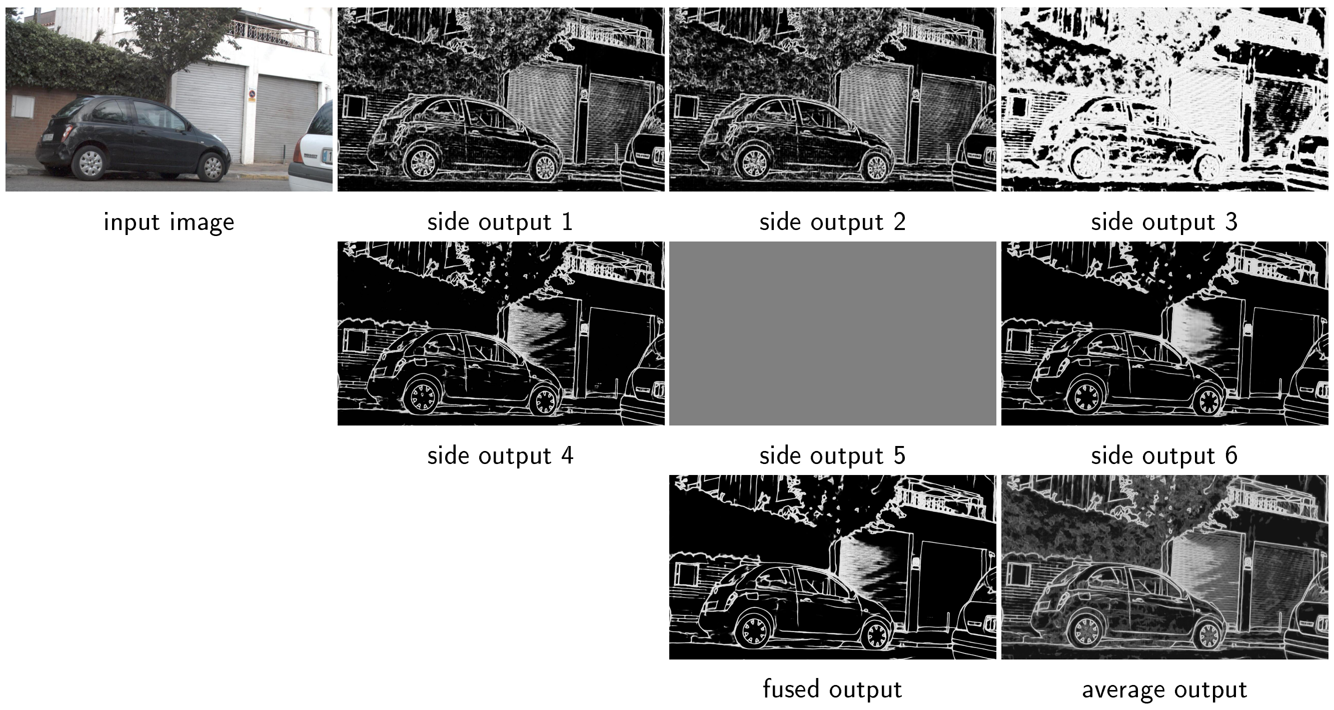


Figure 3: This experiment is identical to the one illustrated in Figure 2, but using a different set of trained parameters, namely the training set "train2" as provided by the authors' repository [8, see README.md at directory 'legacy']. Note that the side output3 produces a much stronger response, while the side output 5 always gives a null result regardless of the input image. These results suggest that the "train2" parameter set is sub-optimal. In spite of that, the fused and average results are quite good edge maps.

third one at 1/4 resolution; the fourth at 1/8 resolution; and the two last blocks at 1/16 resolution. The resolution is reduced from one block to the next by max pooling operations (light blue boxes) with 2×2 kernels and 2×2 strides. The first block is composed of two convolutional layers with 32 and 64 filters each. The second block is composed of two convolutional layers with 128 filters each. The third block is composed of four convolutional layers with 256 filters each. The fourth and fifth blocks are composed of six convolutional layers of 512 filters each. The last block has 7 convolutional layers of 256 filters each.

At different stages in each block, the processed image is added (or averaged in some cases) with the output of a lateral connection. This was inspired by the Xception network [2], where the main idea is to decouple cross-channel correlations and spatial correlations by using different layers for depthwise convolutions and pointwise convolutions. This requires using separable convolutions where the spatial convolutions are performed independently over each channel. This is not the case in DexiNed which uses standard convolutional layers; so the inspiration from Xception is limited as the main idea of cross-channel/spatial decoupling is not used.

The output of each of the six blocks (organized in columns in Figure 1) is then up-scaled to the full resolution of the input image. This is done by a set of 1×1 convolution kernels, ReLU activation function and 2×2 transpose convolution kernels (grouped by blue rectangles in the figure). The number of such stages is the one required according to the resolution of each block.

The up-scaled outputs of the block are the six side outputs (violet boxes). The six side outputs are then concatenated and fused by a convolutional layer with a 1×1 kernel to produce the so-called *Fused Output*. The so-called *Averaged Output* is obtained by the average of the six side outputs and the *Fused Output*. All outputs use sigmoid activation functions; the output are thus in $[0, 1]$. A normalization to $[0, 255]$ may be applied for visualization in standard form (the authors' original code also inverses the values, resulting in a white background with edges appearing in black; we removed this inversion for our experiments and the associated demo).

In total, DexiNed comprises 66 convolutional layers, adding up to 34,744,385 parameters.

3 Training

In this work we did not try to reproduce the training stage, focusing only on the resulting method. Thus, we will not describe the training procedure. It is indeed interesting to reproduce in detail the training, but this will be left for future work. Nevertheless, we will comment on the datasets used during the training to give some context to the obtained results. The DexiNed network was trained on a dataset especially produced by the authors for this purpose; the Barcelona Images for Perceptual Edge Detection dataset, or BIPED, was build with the specific aim of training edge detection algorithms. Figure 4 shows some sample images from the BIPED dataset and the corresponding manually annotated ground truth edges.

According to the authors, BIPED was carefully annotated by experts to avoid some problems related to the very popular BSD Berkeley dataset [4]. In [9] the authors discuss how the fact that the BSD dataset was developed for semantic segmentation tasks introduces some problems when the objective is instead edge detection. In particular it is rightly signaled that the semantic objects contours are a subset of the edges present in images. On the other hand, the human marked edges also include apparent contours and ridges. In our opinion, the authors are right in their criticism of the standard datasets, such as BSD; nevertheless, a careful study of BIPED shows that the same problems remain, even if to a lesser extent. There is still no clear and absolute definition of the notion of an edge pixel in an image, resulting in ambiguities. For example, there is a continuum between clear edges and textured zones, related in particular to the scale at which the scene is analyzed. The level of detail is arbitrarily fixed for different scenes or zones of images, mainly in relation with the content

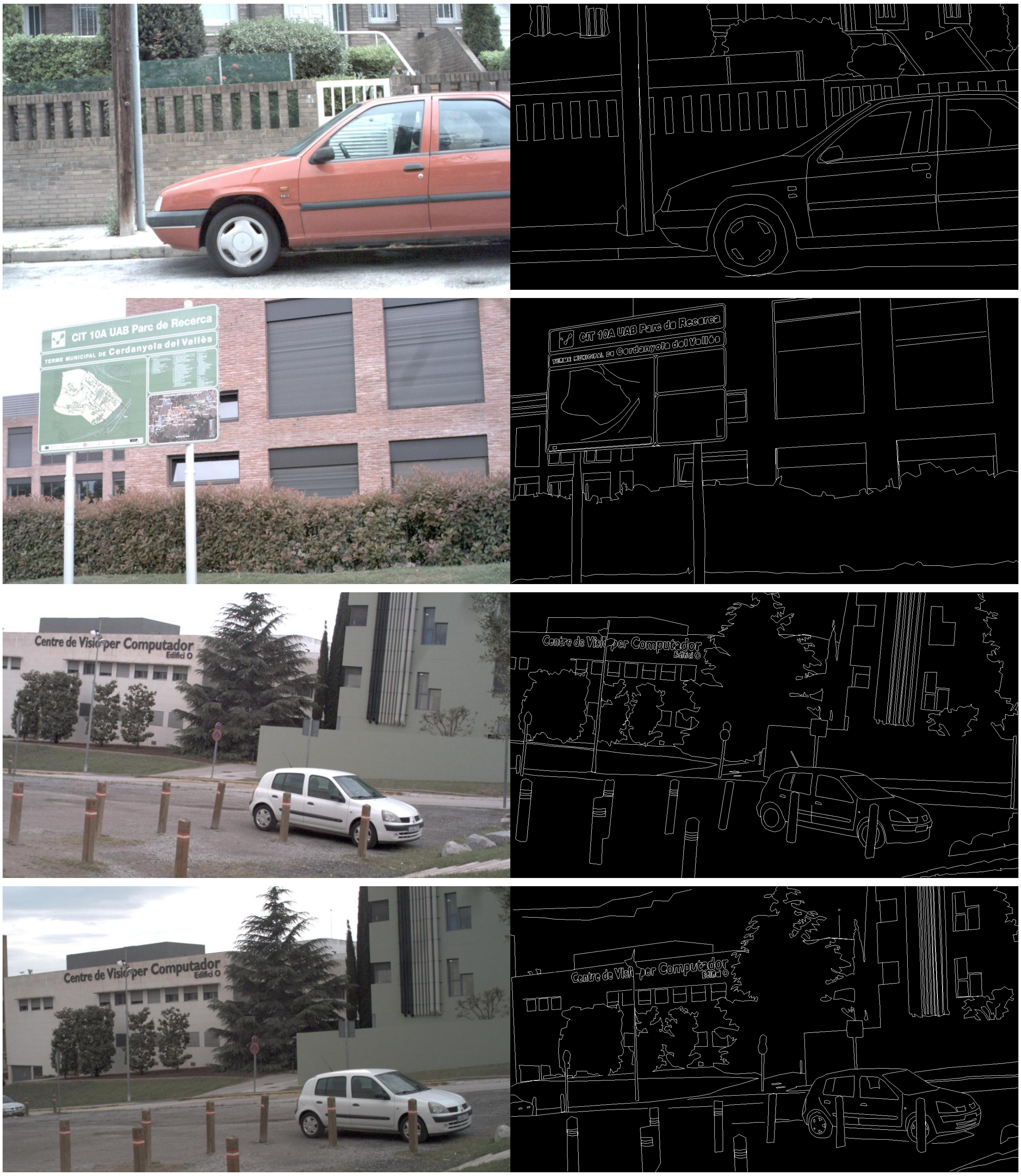

CiT 10A UAB Parc de Recerca
TERME MUNICIPAL DE Cerdanyola del Vallès
Centre de Visió per Computador
Edifici O

Figure 4: Examples from the BIPED dataset used to train DexiNed. Left: input image. Right: annotated edges. The last two images are two pictures of the same scene with slightly different viewpoints and exposition; notice that there are some differences between the annotated edges of corresponding parts. This is true in almost any region; look for instance at the detail on the car.

of the image itself. The last two rows of Figure 4 illustrate the difficulties of edge labeling: both images show the same scene from slightly different viewpoints, but the annotated edges are different for the same parts; look for example at the car or at the windows in the background building.

BIPED contains 250 images of size 1280 × 720 pixels. For the training the authors randomly chose 50 for testing and the other 200 for training and validation. The training set was augmented by performing four transformations on each image: (a) the original image was cut into two equal parts, keeping the original and the two resulting images (thus a factor 3) for further transformations; (b) 15 rotations were performed, resulting in a factor 16 when counting the non-rotated versions; (c) images were horizontally flipped, which gave a factor 2; and (d) two gamma corrections were applied, which in addition to the non-gamma corrected resulted in a factor 3. This gave 288 new images for each one of the original 200 training and validation images[3]. In contrast to the BSD dataset, each image in the BIPED dataset was annotated by a single person. The authors provide a justification for this:

> these images have been carefully annotated by experts on the computer vision field, hence no redundancy has been considered. In spite of that, all results have been cross-checked in order to correct possible mistakes or wrong edges [7].

The provided ground truth images are binary, with edge labeled pixels with value 255 and non edge pixels with value 0.

The authors made publicly available two sets of trained parameters for DexiNed, labeled “train1” and “train2”. According to the note accompanying the release, train1 corresponds to the parameter set used for the experiments reported in the WACV2020 paper [7]; on the other hand, train2 “contains our last checkpoint trained with the updated BIPED”. It is not clear for us what the latter description implies; nevertheless, when observing the side outputs produced using train2, see Figure 3, it seems to correspond just to the last training experiment before moving on to a new implementation of DexiNed. Notice how the different side outputs produce very different results, while they should correspond to similar edge maps obtained at consecutive scales; note in particular how the side output 3 shows a far stronger response and, even worse, the side output 5 is always zero at all pixels independently of the input image. We conclude that train1 is probably the result of a more careful training step. For this reason, and the fact that it shows a better fit to the published experiments, the following experiments were conducted using the train1 set of trained parameters. Nevertheless, Section 4 includes an experiment comparing the results with both training sets; this is done for the sole purpose of giving an idea of the impact of the training set on the results. In addition, the online demo associated with this work computes the result of DexiNed using both training sets, allowing the reader to perform further evaluations.

4 Experiments

Figures 5 to 11 show sample images and the corresponding results produced by DexiNed. The edge maps produced by DexiNed have a single channel and the same size as the input image. Each output pixel has a value between 0 and 1, but they are usually normalized to values between 0 and 255. The values near zero appear as black, while values near 255 appear as white. The brighter the pixel, the more confident DexiNed is about the presence of an edge at the corresponding position. As explained in Section 2, DexiNed produces two main results: the fused and average outputs.

Figure 5 shows the result of DexiNed on an image from the BSDS500 dataset [1]. This is a classic dataset for semantic segmentation, widely used for edge detection evaluation. Notice that

[3]The training dataset as well as the data augmentation procedure is available at https://github.com/xavysp/MBIPED

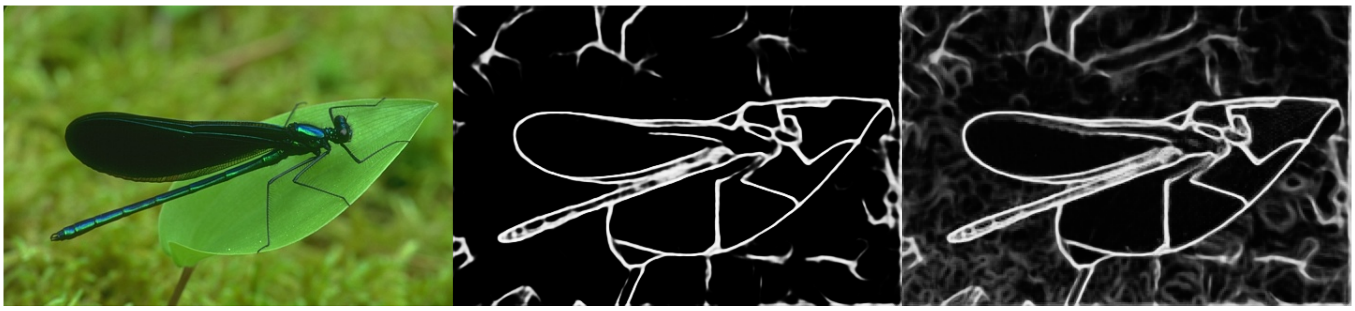


Figure 5: Input and result of DexiNed on image number 35070 of BSDS500 dataset [1] of size 481×321. DexiNed is able to detect edges (sharp brightness or color transitions as in the colorful green grass or the limits of the leaf) as well as ridges (thin structures over a background as the legs of the dragon-fly); when two edges gradually approach each other (e.g. the wing gradually approaches the abdomen of the dragon-fly), the detected edges gradually merge into one detected ridge. These results were obtained with the training set "train1" as provided by the authors' repository [8, see README.md at directory 'legacy'].

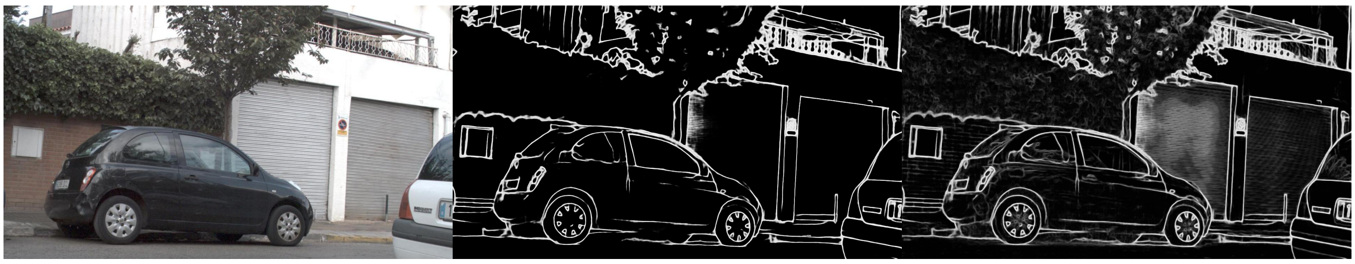


Figure 6: Input and result of DexiNed on an image belonging to the BIPED training set, size 1280×720. All the detected edges in the average output are already present in the fused output; the average output adds mainly clutter detection, note for example the garage doors. These results were obtained with the training set "train1" as provided by the authors' repository [8, see README.md at directory 'legacy'].

the method is able to detect edges (sharp brightness or color transitions as in the colorful green grass or the limits of the leaf) as well as ridges (thin structures over a background as the legs of the dragon-fly). These two kinds of structure are not differentiated; when two edges gradually approach each other (e.g. the wing gradually approaches the abdomen of the dragon-fly), the detected edges gradually merge into one detected ridge. Also, edges are detected as a bright region of a certain width, regardless of how sharp the edge is on the image; the region of bright pixels corresponding to an edge are usually about two pixels wide (in some cases wider), with slightly higher values near the center of the region.

Figure 6 shows the result of DexiNed on one of the images of the training set. This illustrates the behavior of the method under the most favorable possible conditions. The original paper [7] indicates that the average output obtains a better score on the quantitative evaluations, suggesting that it should be the preferred output. Nevertheless, after a visual inspection we do not agree with that suggestion: All the detected edges observed in the average output are already present in the fused output; moreover, these edges are detected more clearly in the fused output, probably because its intensity is diminished by the average operation. On the other hand, the average output adds mainly clutter detection, note for example the garage doors in Figure 6 and the examples on Figure 9. The fact that the average output obtains a better score on the evaluation seems to indicate a limitation of the evaluation procedure itself.

Figure 7 shows the result of DexiNed on a classic test image for edge detection extracted from [3]. The experiment is repeated after adding Gaussian white noise of increasing intensity to the image.

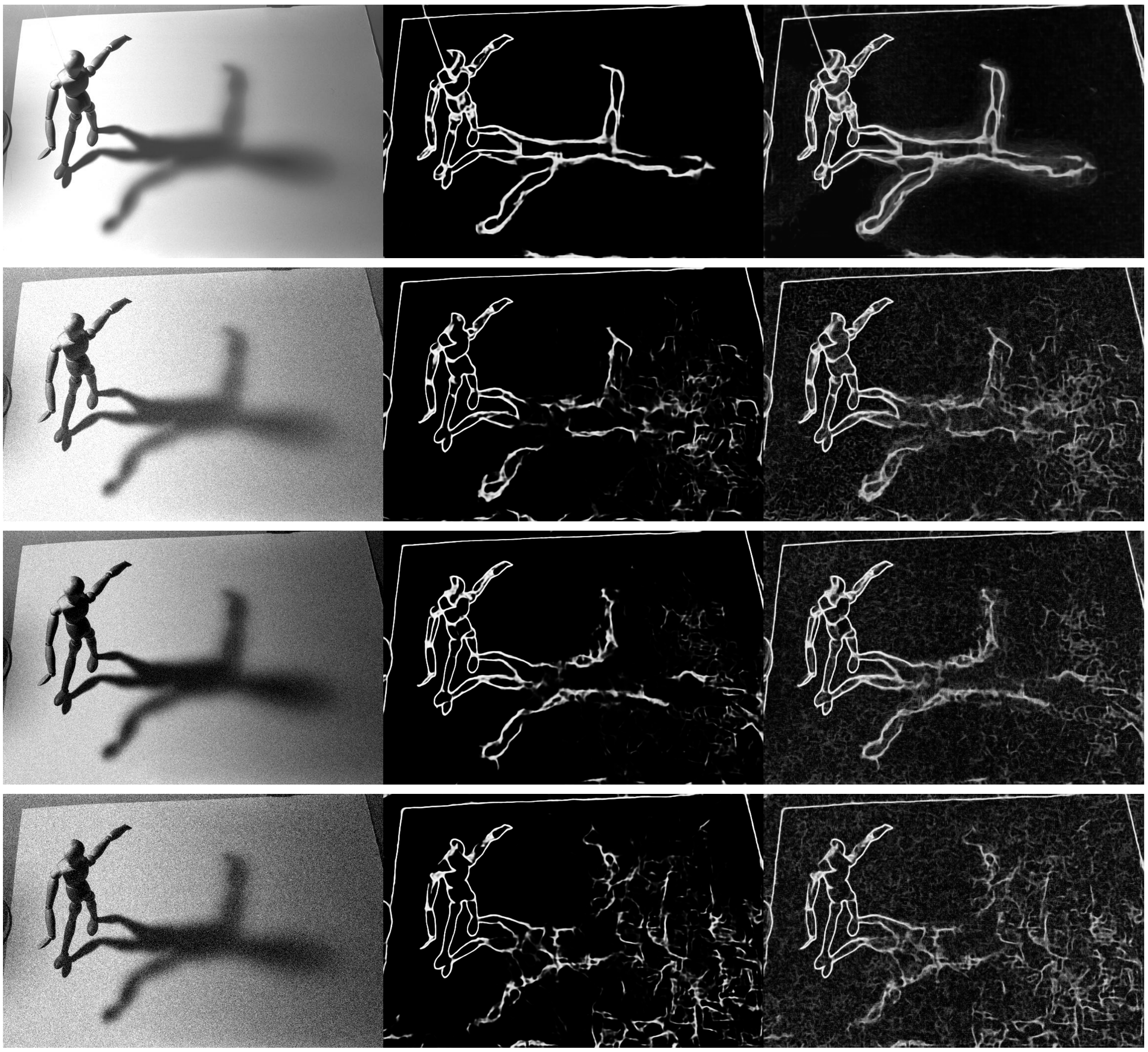


Figure 7: Effects of scale change and noise on DexiNed. Input and result of DexiNed on an image extracted from [3] of size 768×512. First row: original image and result of DexiNed. Second row: Gaussian noise of $\sigma = 20$ was added to the original image. Third row: Gaussian noise of $\sigma = 50$ was added to the original image. Fourth row: Gaussian noise of $\sigma = 100$ was added to the original image. On blurred regions the detection does not correspond well to the geometry in the image, and as noise is added the detection degrades. These results were obtained with the training set "train1" as provided by the authors' repository [8, see README.md at directory 'legacy'].

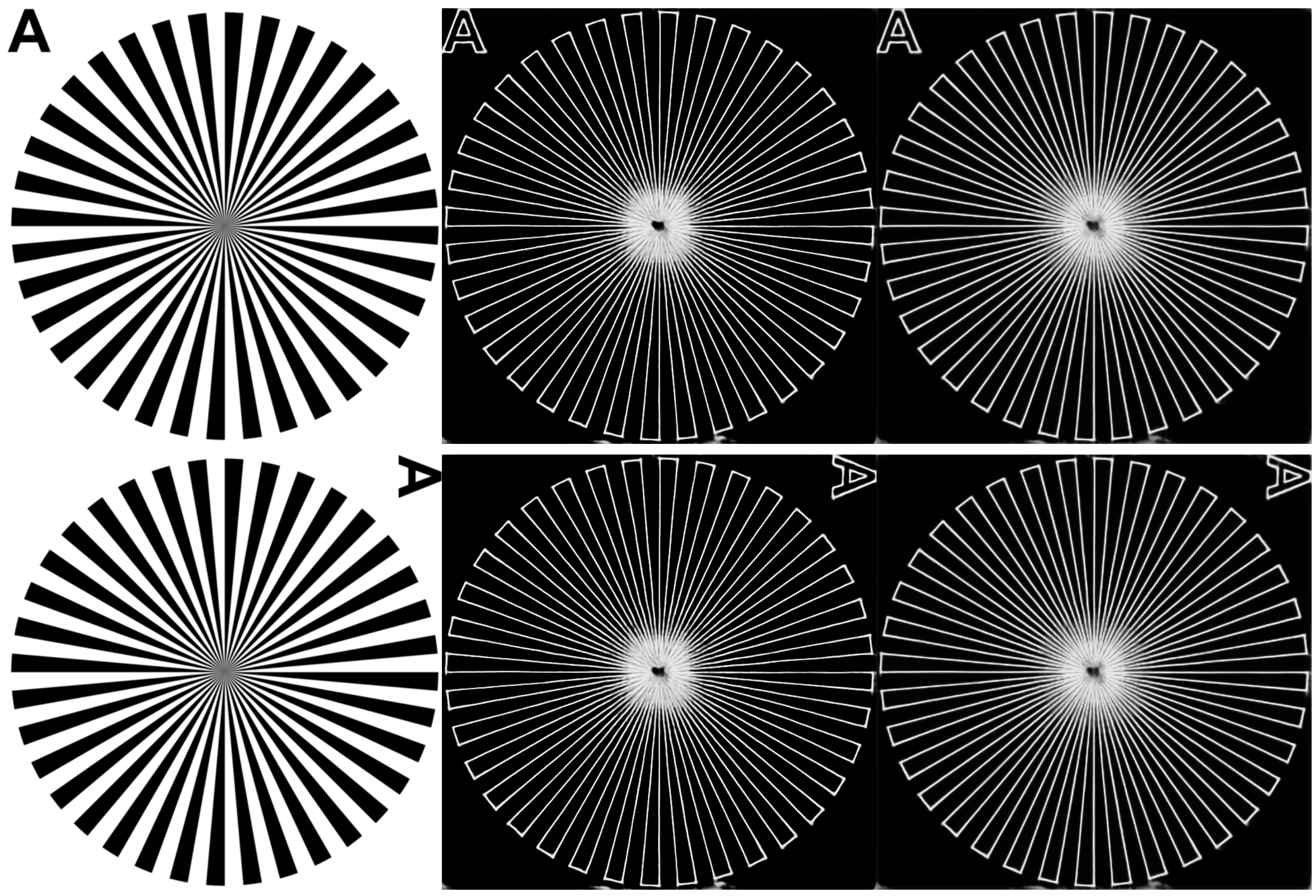


Figure 8: Experiments with a synthetic Siemens Star (700 × 700 pixels). The same experiment is repeated on the image rotated by 90 degree. Note that the result does not depend significantly on the orientation of the input image. DexiNed shows a good isotropic response. These results were obtained with the training set "train1" as provided by the authors' repository [8, see README.md at directory 'legacy'].

In the first case (no added noise) the edge map produced by DexiNed includes most of the edges observed on the image, even those that appear at coarse scales as in the shadow of the mannequin. Most of these edges are not fully sharp, appearing as white regions about four pixels wide. Some of the regions produce a block of solid detection enclosed by correct edges; see for example the right ankle of the mannequin. Some edges are double detected, as in the shadow of the right hand and on the right side of the head of the mannequin. In the shadow one can see some strong detections which appear to be inside the shadow zone, not in a clear transition. Finally, the detail of some of the detected structures does not seems to correspond to the observed geometry in the image; see the shadow of the right armpit or the shadow of the right knee. As before, the average output seems to add mainly clutter. When noise is added, as expected, the detection degrades. The details gradually disappear as more noise is present. Also, false detections are created on the background by noise, especially on flat zones.

The next experiment applies DexiNed to an image of the Siemens Star synthetic pattern, see Figure 8. This allows us to appreciate that the method has a quite isotropic response. The resulting pattern does not depends strongly on the orientation. The response to high frequency in the center of the star appears as a slight square shape, whereas a circular shape would be expected; the central part also shows a characteristic pattern which remains unchanged after turning the image by 90 degree. Nonetheless, DexiNed shows a good isotropic response.

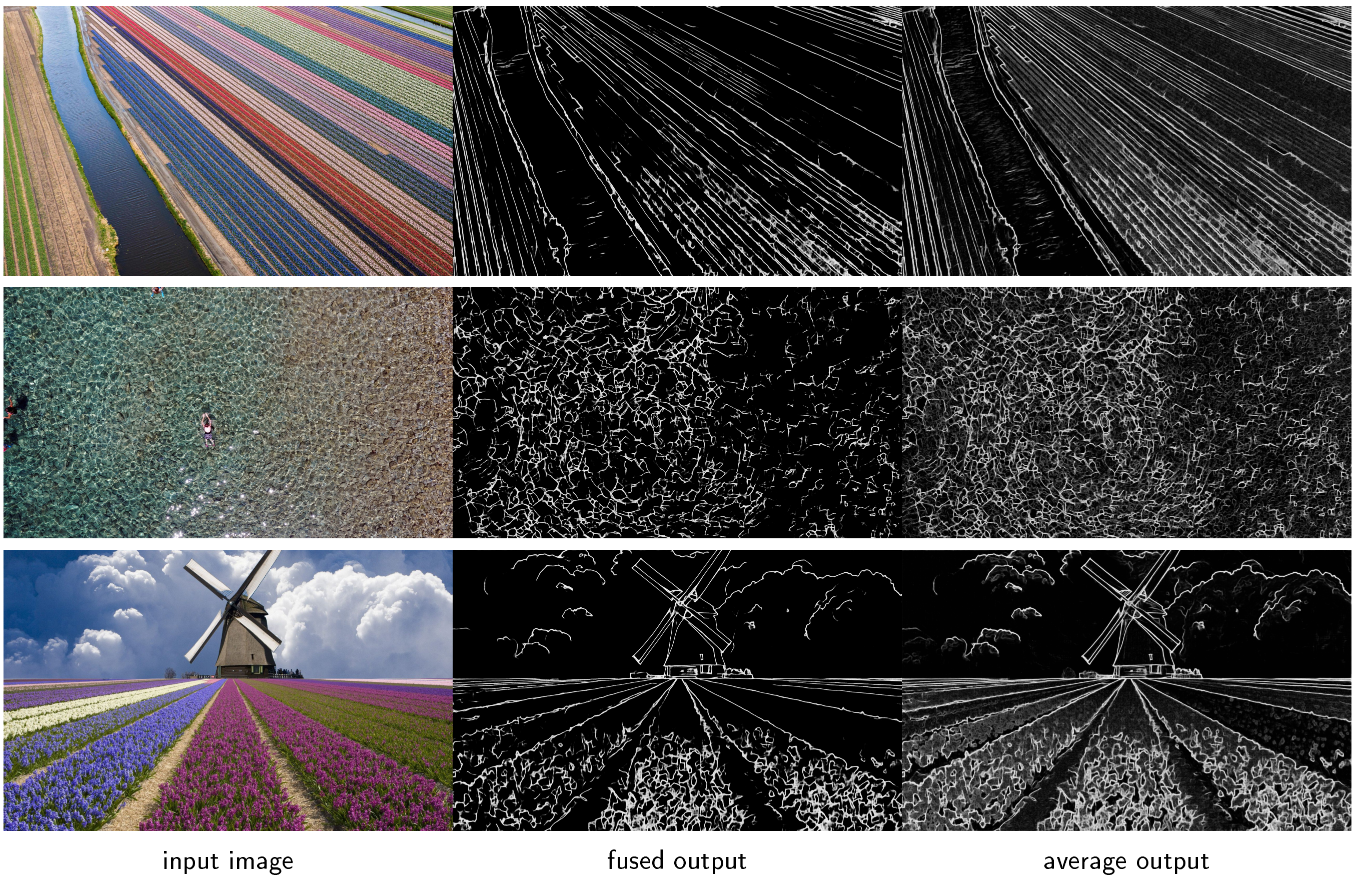


Figure 9: Edge maps produced by DexiNed illustrating some of its limitations. In the first row some of the colored stripes produce detections, while other do not, even if the appearance of those stripes is very similar. There are also some spurious detections between the stripes. The second row image contains a complex pattern of light and shadow on the water. This leads to irregular detections and again, there are parts that look very similar but produce different responses. The last row shows a transition between these two stages. The tulip field is seen in perspective; some individual tulips lead to detections on the foreground, while only the general structure of the field produces detections on the background. These results were obtained with the training set "train1" as provided by the authors' repository [8, see README.md at directory 'legacy'].

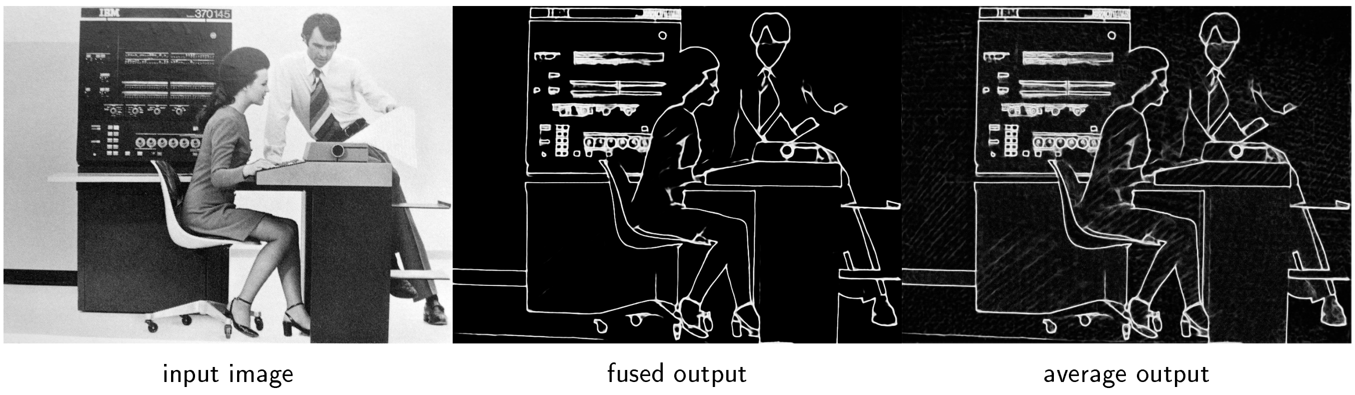


Figure 10: Experiment on an image produced by dithering (size 1024×768). Note that DexiNed correctly ignored the dithering. This image contains some subjective contours, such as in the shirt of the man or the frontier between the table and the man's legs. These subjective contours did not produce detections. These results were obtained with the training set "train1" as provided by the authors' repository [8, see README.md at directory 'legacy'].

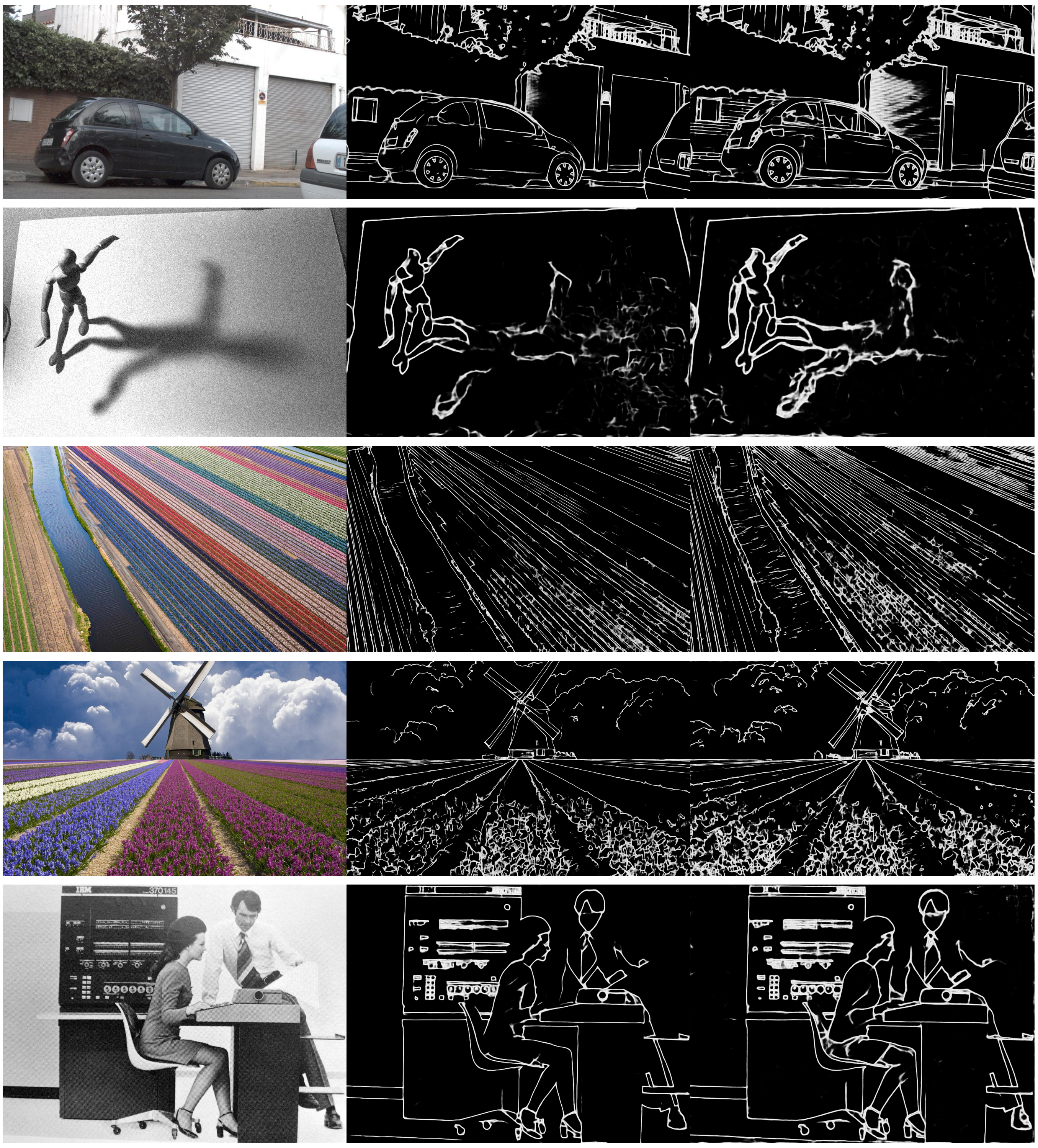


Figure 11: Fused results of DexiNed using the two different sets of trained parameters, "train1" and "train2" as provided by the authors [8, see README.md at directory 'legacy']. Note the differences between both outputs, for example in the second row.

Figure 9 illustrates some limitations of DexiNed. In the first example some of the colored stripes produce detections, while others do not, even if the appearance of those stripes is very similar. There are also some spurious detections between the stripes. The second image contains a complex pattern of light and shadow on the water. This leads to irregular detections and again, there are parts that look very similar but produce different responses. This suggests that this image contains structures at a limiting scale for being considered as edges or as texture. The last image shows a transition between these two stages. The tulip field is seen in perspective; some individual tulips lead to detections on the foreground, while only the general structure of the field produces detections on the background. In all the three cases, relative to the fused output, the average output only adds an almost uniform detection level on a large part of the images.

The image in Figure 10 presents some characteristics that differ from ones on the training set. Even if it is a natural image, when observed in detail one can see a dithering pattern used to generate the gray levels. DexiNed correctly ignored the dithering. The images on the training set do not include dithering; nevertheless, the images of the training set include textured zones where the annotation indicates to ignore those patterns. The dithering may be thus treated as a texture and thus ignored. A second interesting observation is that this image includes some subjective contours, such as in the shirt of the man, a part of the hair of the woman, or the frontier between the table and the man's legs. These subjective contours did not produce detections, confirming that DexiNed is an edge detector (as responding to image intensity transitions) and not a contour detector (as responding to the contours of present objects). This point was indeed emphasized by the authors.

A final experiment compares the results of the DexiNed architecture using two different sets of trained parameters: "train1" and "train2" as made publicly available by the authors, see Figure 11. Note how the training is critical for the quality of the results. At first sight both results seems similar; observing in detail, however, there are substantial differences between the two results.

5 Discussion

The Dense Extreme Inception Network CNN Model for Edge Detection(DexiNed) [7] is a state-of-the-art neural network for edge detection which produces very good results in general. DexiNed provides a value per pixel, indicating the confidence on the presence of an edge at each position. The method does not provide a binary edge map indicating the presence or not of edges; this is in contrast to the annotation provided by humans, consisting of binary maps. The score maps produced by DexiNed provide a very useful information in a variety of applications; in some applications it can be even better than a binary edge map. For other applications that require a binary edge map, some kind of threshold process is needed on the output of DexiNed.

DexiNed provides two main outputs, namely the fused output and the average output. Even if the authors claim that the latter gives better performance, our impression is that the fused output gives the best results as an edge detector.

DexiNed does not differentiate edges from ridges and the result is not completely sharp (edge results are usually about 2 pixel wide and often more). The presence of even moderate noise degrades the detection. Also, high-frequency patterns fail to produce detections.

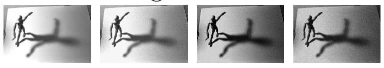
DexiNed shares a common aspect with all supervised learning methods: the results are strongly dependent on the training dataset [9]. Indeed, there is a high level of subjectivity in the annotation of edges or contours by human subjects. At what point do certain strokes cease to be edges and become a texture? Only object contours should be annotated, or should edges observed on objects be annotated too? For example, should the stripes of a zebra be annotated or not? To which degree of detail should objects be classified as different? Should the clothes of a person be considered as separate objects from the person? The authors argue that these problems may be essentially solved using a carefully annotated dataset using coherent criteria. To this aim they built and proposed the BIPED dataset. This new dataset is a good improvement, namely because it has been built specifically for edge detection purposes. But it does not solve completely the mentioned problems. In addition to the intrinsic difficulties of such a task, the amount of time required to annotate each image imposes a limit to the dataset size that can be produced by such careful work done by experts.

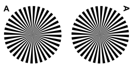
Taking into account all these considerations, the results of DexiNed are quite impressive. All in all, DexiNed is an excellent edge detector.

Image Credits

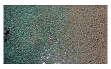
From BIPED dataset [7].

Image number 35070 from BSDS500 dataset [1].

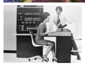
 Extracted from [3] without and with added noise.

 Synthetic images generated by the authors.

Jeffrey Groeneweg/ANP/AFP/Getty Images[4]

 Anadolu Agency/Anadolu Agency via Getty Images[5]

CC0[6]

 John Keogh, IBM System/370 Model 145, CC BY-NC 2.0[7]

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, *Contour detection and hierarchical image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2010), pp. 898–916. https://doi.org/10.1109/TPAMI.2010.161.

[2] F. Chollet, *Xception: Deep learning with depthwise separable convolutions*, 2016. https://doi.org/10.48550/arXiv.1610.02357.

[3] J.H. Elder and S.W. Zucker, *Local scale control for edge detection and blur estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 (1998), pp. 699–716. https://doi.org/10.1109/34.689301.

[4] D. Martin, C. Fowlkes, D. Tal, and J. Malik, *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*, in IEEE International Conference on Computer Vision (ICCV), vol. 2, July 2001, pp. 416–423. https://doi.org/10.1109/ICCV.2001.937655.

[4] https://i.guim.co.uk/img/media/f961b352bfa77b39fc602190ac753fb4ae73a60b/0_8_4064_2440/master/4064.jpg?width=720&quality=85&auto=format&fit=max&s=c7128a7d17e2af1c0c85ef66b682b21b

[5] https://i.guim.co.uk/img/media/9c23db8b0ba1ef232061a161ccee391d10e122b1/0_0_5472_3078/master/5472.jpg?width=720&quality=85&auto=format&fit=max&s=37ab9ff6191e4c548bf0ba3b0220d320

[6] https://pxhere.com/fr/photo/1434887

[7] https://www.flickr.com/photos/jvk/10227059/

[5] G. Papari and N. Petkov, *Edge and line oriented contour detection: State of the art*, Image and Vision Computing, 29 (2011), pp. 79–103. https://doi.org/10.1016/j.imavis.2010.08.009.

[6] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, in International Conference on Learning Representations (ICLR), 2015. https://doi.org/10.48550/arXiv.1409.1556.

[7] X. Soria, E. Riba, and A. Sappa, *Dense Extreme Inception Network: Towards a Robust CNN Model for Edge Detection*, in IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 1912–1921. https://doi.ieeecomputersociety.org/10.1109/WACV45572.2020.9093290.

[8] ——, *DexiNed: Dense Extreme Inception Network for Edge Detection (Extended version)*, 2020. https://github.com/xavysp/DexiNed.

[9] X. Soria Poma, A. Sappa, P. Humanante, and A. Arbarinia, *Dense extreme Inception network for edge detection*, (2021). https://doi.org/10.48550/arXiv.2112.02250.

[10] S. Xie and Z. Tu, *Holistically-nested edge detection*, in IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1395–1403. https://doi.org/10.1109/ICCV.2015.164.