XL CILAMCE
IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
NOVEMBER 11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

IMAGE DENOISER ENHANCEMENT VIA GENETIC ALGORITHMS

Matheus Lacerda Bezerra
Leonardo França Bessa
Math2.0@hotmail.com
Leofranca16@gmail.com
Universidade Estadual do Maranhão
Philipe Manoel Pinheiro
Marta Barreiros
Philipepinheiro@aluno.uema.br
Marta-barreiros@hotmail.com

Abstract. . Image processing is becoming a more open and extended field, with endless possibilities and meaningful applications, specially for medical images. There, a point that brings much interest is the X-ray imaging, which are generated by sending high-frequency electromagnetic radiowaves, and then retrieving them in a metal plates that they will collide with. But those metal plates might receive electromagnetic interference or uncontinuous radiowaves, causing image noise, which are inevitable. That noise can be decreased by image filters. Although those filters have become standard and work well, their parameters values are static, even though there are many possible parameters values and each image will have different denoise results with different values. This articles descibres the use of Genetic Algorithms to find the best parameters with fewer processes. GA are heuristic search procedures based on natural selection. Only the algorithms that give the best results will generate other algs, decreasing the number of processes needed. The method uses Genetic Algorithms to find best parameters according to each image for the "Non-Local Means" denoise filter from the OpenCV Python library. The X-ray images used in were Optical Coherence Tomography of people with Pneumonia, published on the online database Mendeley by the University of San Diego California, and tested on the Python scripts implementations. By the results obtained, it was noticeable the positive correlation between the filter parameters chosen by the GA for each given image and the improvement of image denoise from the classical methods, proving that Genetic Algorithms are great for such applications, decreasing the number of processes needed to find the the best parameters for any input.

Keywords: images, X-ray, genetic algorithms, process, noise filters.

1 Introduction

Image processing is an ever evolving field due to it's great potential. The images should be in good quality to increase the efficiency of the processings. However, image noise is an unavoidable problem in image capturing, most usually due to electromagnetic interference. There are many denoise methods [1], to deal with all sorts of noises, and delivering various results.

Nowadays, these methods vary from static parameters and structural functions to even machine learning. Those methods are of general use to usually deliver satisfactory results, and are not usually targeted to better address noise causes or case-particularities. This article describes a method to better direct the filtering process using Genetic Algorithms, heuristic searches inspired by natural selection. This method improved the results of the already great digital filter "Non-Local Means", while shorting the search for better parameters.

This article aimed to filter X-ray images. Those images are generated by retrieving electromagnetic radiations emitted over the observed object. Since they are generated by radiations, common cause of noise, and generate digital images, they were an interesting choice for study. The images used on this project were of pneumonia patients, made available online by the University of Chicago [2], and put to analysis by a neural network classifier to prove the efficiency of the GA enhanced filter for image processing and it's improvement over traditional methods.

Despite having X-ray images in mind, it was noticed that the algorithm can work with any kind of image. Unlike most articles designed for image denoise, our algorithm did not make use of any reference images (original or noise-free images), since these are most often not available in practical everyday situation. Despite that, the algorithm was able to hold it's own.

2 Background

In this section, we briefly aboard noise sources and the current well received denoise methods.

Noise is a random distortion in the captured data, which in digital images randomly affects the value of the pixels. There are several sources of noise [3], the most often being electromagnetic interference. No image capture is noise-free, and since it's effects are random, the treatment is tricky. Not only find the correct value of the pixel, but also determine which pixels were affected by noise.

There are many well received methods for image denoising [4]. All of them take the source of the noise and the image format into consideration. One very approved method is Local Means [5], which determines if the pixel has noise by generating a Gaussian function based on the surrounding pixels.

With GA, some works have been done [6]. There are works which use GA alongside neural network, to accelerate the learning of the machine. Most previous works used a reference image. The algorithm would filter an image and compare it to a noise-free version to determine the quality of the filter. As in theory, real situations would not have such an image available, we decided to distinguish our implementation by not having a reference image, while still aiming to compete with these previous works.

3 Methodology

One method that distinguishes itself is "Non-Local Means", available in the OpenCV library. It's an improvement on Local Means, because it uses not only the surrounding pixels, but also areas that are similar to one another. By doing so, it retrieves more information to work with, improving the filtering results. It's choice for the project was very natural, not only for working with pixels, but also for being a well received method that does not take much details from the image during the filtering process. The function is defined as follows: but also other areas in the image that area similar to the current one. It works directly with the pixel and their values, comparing each one with their surroundings. It's an improvement on Local Means, because it uses not only the surrounding pixels, but other areas that are similar.

The function is defined as follows:

Void fastNlMeansDenoising (**InputArray** src, **Output** Array dst, **float** h, **int** search_window, int block_size)

Where the *src* is the source image to be treated and *dst* is the destination result image to be saved. *H* is the strength of the filter, the greater it is the most noise will be removed, but a too great strength may cause blurriness or remove details of the image. *Search_window* is the size of the search for similar areas in the image, and *block_size* is the size of the surrounding pixel area.

The quality of the results is directly related to the quality of the parameters used (Input and Output should not be considered). There are many possible combinations of parameters to be passed, and OpenCV suggests a few that generally deliver a good result. However, those are general use and each image can make better use of different values. The focus of the GA was to always find the best parameters for any image.

In the algorithm, a combination is treated as an individual. It will generate populations of these, and than use it to filter the image. It goes as:

```
def firstPopulation (Size):
    population = []
            for i in range (size):
            Population.append((randint(1, 60), randint(1, 60), randint(1,60)))
    return population
```

The individuals that delivered the 40% best results would generate other individuals that would keep on searching. The fitness of the individuals was determined by the Median Absolute Deviation. It's defined as the median of the absolute deviations from the data's median ~X = median (X). It measures the variability of the pixel values in the filtered image from the original one.

```
def fitness (h, twindows, swindows):
    img_denoise = cv2.fastNlMeansDenoising (img_noise, none, h, twindows, swindows
    s_denoise = 1 / mad (img_denoise, axis = none)
    return s_denoise
```

By measuring how much the filtered image was structurally similar to the original one, we could determine if the result was keeping the details of the image while removing the noise. That was pretty much the only way to determine the quality of the filtering, as we did not have a reference to work with.

As for the crossover, the individuals would be matched randomly (inside that 40% pool). For the mutation, it would or not affect each gene in each individual randomly, shifting it's value to higher or lower (since they were integers and a float). Since the genes wouldn't change by themselves, the mutations were essentially responsible for the evolution of the GA.

The images used were made available online by the University of Chicago. More than 600 test and 5100 images of pneumonia patients, used by them for machine learning. Likewise, we used them to train a neural network to identify the images. After analyzing more than 1000 images, it was ready to process new ones. We than proceeded to give it filtered and non-filtered images to compare it's efficiency.

4 Experimental results

Many possible changes to the algorithms implementation were tested. It was observed that the genetic variability of the initial population was a key factor to it's efficiency. Starting a population only with values close to the suggested by OpenCV would not end up with better results. Almost any manual change to the algorithm's natural processes would affect it's efficiency negatively. Thefore, manual changes to the algorithms working process would affect it negatively, so the mutation was simply random, and the mutation rate was set as high as 50% for each gene.

It was observed that 20 generations were the ideal for the GA, as more generations would not give a significative gain. 12 individuals per generation were the standard, as it was a good balance between quality and performance. The application ran on a Core i5 2.2Ghz (Dual Core, 4 Threads) with 8gb of RAM, on Windows 10 64bits. For a 1000x1000 pixels image, each generation would take about 4 seconds to process.

A point of interest of the application was how well it kept the details of the filtered images. It's known as the Non-Local Means filter nature, but the fact that the fitness was of the individuals were based on it played a certain role.

As for the classifier, it would give a percentage chance of the patient had or not pneumonia. It's results were as follows:

Coefficients of certainty of the classifier

No filter	Filter	Gain
71,62	78,68	7,06
73,42	80,52	7,1
75,04	82	6,96
76,8	87,02	10,22
84,6	87,04	2,44
91,02	94,58	3,56
91,12	97,44	6,32
93,83	98,29	4,46
94,89	98,7	3,81
Median	Median	Median
83,59	89,31	5,77

*Calculation errors are due to rounding

Running the GA in the same image twice would deliver different results, since the initial population and mutations were random, but the results were nearly identical. Therefore, it's correct to assume that the GA did converge only to the best values.

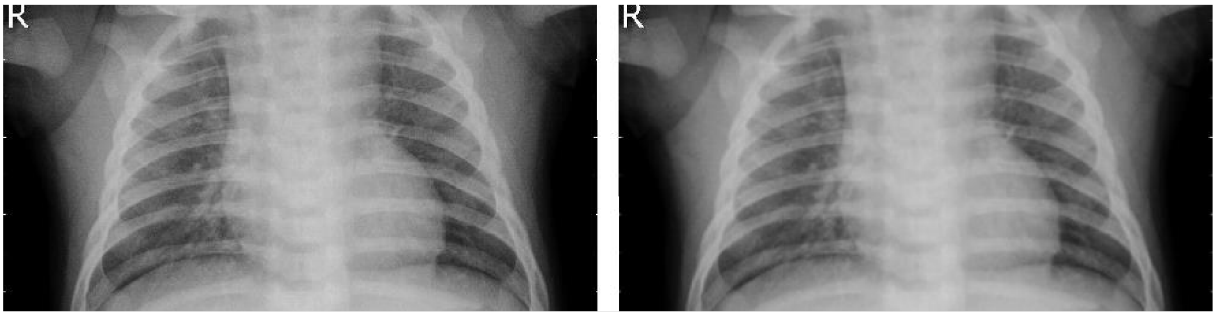
5 Conclusion

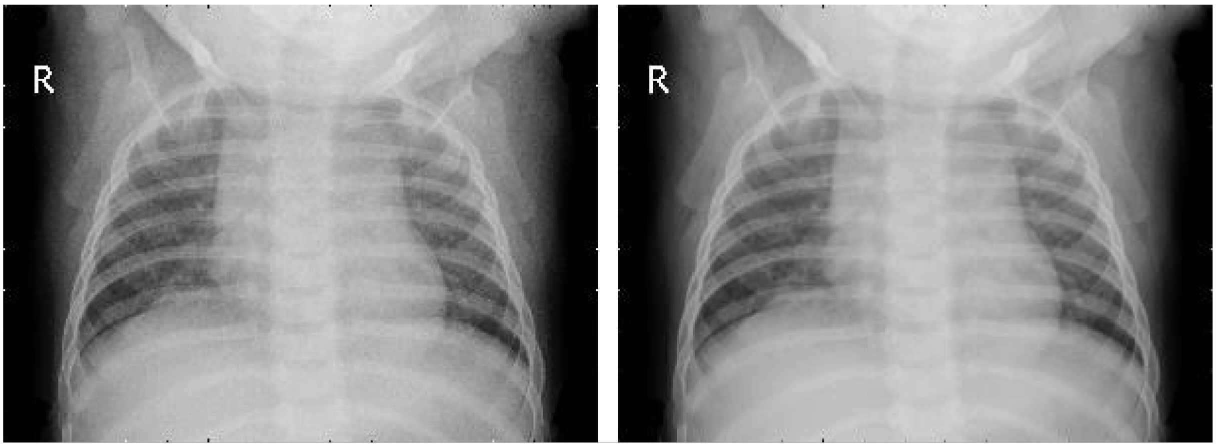
This article presented a method for enhancing the efficiency of image denoising with genetic algorithms. The project used the Non-Local Means from the OpenCV library as the denoiser to be enhanced. The results show that GA are a great way of improving the filters, redirecting it's operations for a better use of it's processes.

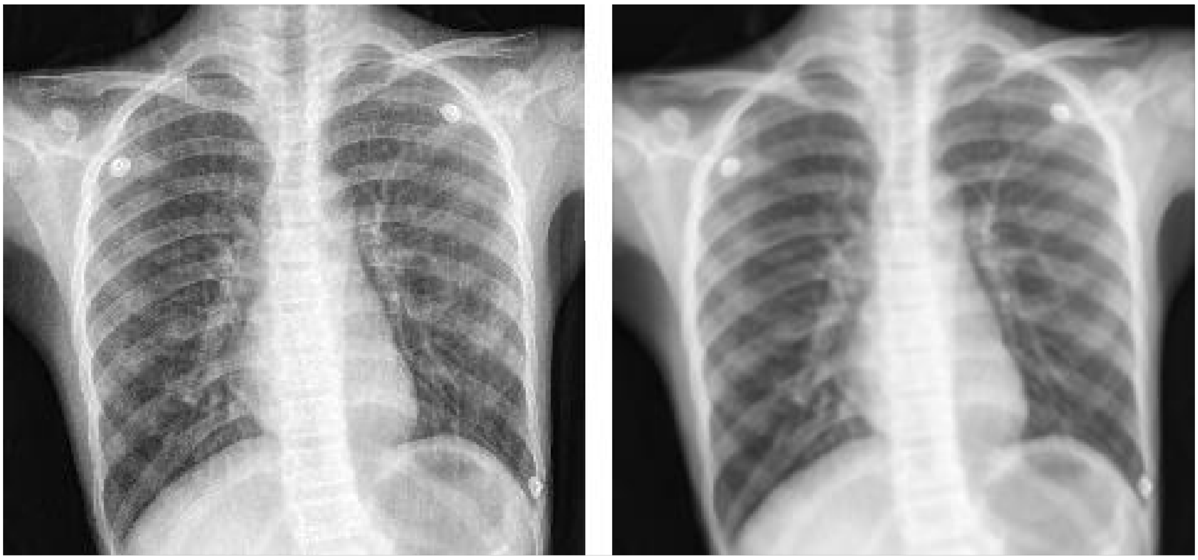
A key point of the project was the absence of reference image: while other articles use it, our team decided to shift from this approach, making it viable to most common situations, where these images are not available, while still giving robust and competitive results with very well received methods. The fact that the classifier increased the certainty of it's results only comes to reinforce the quality of this project.

Another strength of the application is the fact that it does work with any kind of image, despite initially only having X-rays in mind, due to the trust factor of the methodology implemented in the project.

As for performance, the application was satisfactory considering the overheads put over it. Running the GA in the GPU and/or parallel would put these overheads to a minimum.







References

[1] Rohankar, Jayant (Nov 2013). "SURVEY ON VARIOUS NOISES AND TECHNIQUES FOR DENOISING THE COLOR IMAGE" (PDF). International Journal of Application or Innovation in Engineering & Management. 2 (11). Retrieved 15 May 2015.

[2] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, v2. Disponível em: <http://dx.doi.org/10.17632/rscbkbr9sj.2>. Acesso em: 30 de Jul de 2019.

[3] Mehdi Mafi, Harol Martin, Jean Andrian, Armando Barreto, Mercedes Cabrerizo, Malek Adjouadi, "A Comprehensive Survey on Impulse and Gaussian Denoising Filters for Digital Images", Signal Processing, vol. 157, pp. 246-260, 2019

[4] U. Farooq, S. Ting-Zhi, Z. San-Yuan, and M. Imran, "Image Restoration by Using New AGA Optimized BPNN," Procedia Engineering , vol. 29, pp. 3028–3032, 2012.
[5] Image Denoising – OpenCV. OpenCV Docs, 2019. Disponível em: <https://docs.opencv.org/trunk/d5/d69/tutorial_py_non_local_means.html/>. Acesso em: 19 de Jul. de 2019
[6] *Leslie Stroebel; Richard D. Zakia (1995)*. The Focal encyclopedia of photography. *Focal Press. p. 507*. ISBN 978-0-240-51417-8.