Università di Catania

Department of Mathematics and Computer Science
Master of Science Degree in Computer Science

Raffaele Terracino

Removal of artifacts in a mammography dataset

Multimedia Project Report

Prof: Dario Allegra
Prof: Filippo Stanco

Academic Year 2024 - 2025

Contents

Introduction

The "Multimedia" project carried out consists of the removal of artifacts in a mammography image dataset. The dataset used is the "MIAS Mammography" dataset, consisting of 322 images of 1024x1024 gray-scale pixels, depicting mammograms taken for the purpose of diagnosing the presence of "abnormalities," presumably tumors, present in the breast region. The "MIAS Mammography" dataset is distributed under a license that allows its use for "research purposes" and requires citing the authors of the dataset. For the above project, the license requirements are met and are encapsulated in the "README.txt" file in the project folder. The images are stored in PGM format and are numbered from 1 to 322 with "mdb" prefix. Each pair of images depicts both breasts of the same subject: for example, image number 1 is the mammogram of the left breast, while the next image, number 2, is that of the right breast. In addition to the images, there are two files named "Info.txt." The first, found within the folder containing the images, is descriptive of the dataset and the license under which it is distributed, thus representing the set of metadata associated with the dataset. The second, on the other hand, is a tabular dataset that for each image describes the characteristics of the abnormality found, if any. In total, there were 119 mammograms in which abnormalities were detected. The modest size of the dataset allowed the images to be viewed for artifacts. In doing so, 221 images were found to have

text blocks identifying the acquisition device next to the breast. These text blocks consist of black characters on a white background. Beyond this, some images have white, oblique or horizontal pixel stripes due to some kind of unintentional movement of the device or subject during acquisition. The goal of the project is to remove these artifacts, using the image processing techniques seen in class, thus preparing the images for other types of analysis, e.g., classifier construction. The project is entirely developed as a jupyter notebook, using the Python language.

Methods

The artifact removal pipeline consists of 3 main steps: horizontal flipping of left sinuses, removal of text blocks, and removal of white stripes. A distinction is made between small white stripes, which span at most 100 columns, from large white stripes, which span more than 100 columns. For each section, examples are presented with two images explaining the main operations performed by the algorithms.

Horizontal flipping

To make the dataset more homogeneous and to facilitate subsequent operations, the convention is established that all images should be right-facing. Since, because of the way the dataset is constructed, all the odd-numbered images are left-facing, simply open them and use the OpenCV library's cv2.flip method passing as an argument an integer, equal to 1, representing the horizontal flip. For each flipped image, the anomaly data, if any, is also modified. Because the flip is horizontal and, as described in the metadata, the reference system used in the dataset has its origin in the lower left pixel, the x coordinate of the anomaly is replaced with the value $1024 - x$.

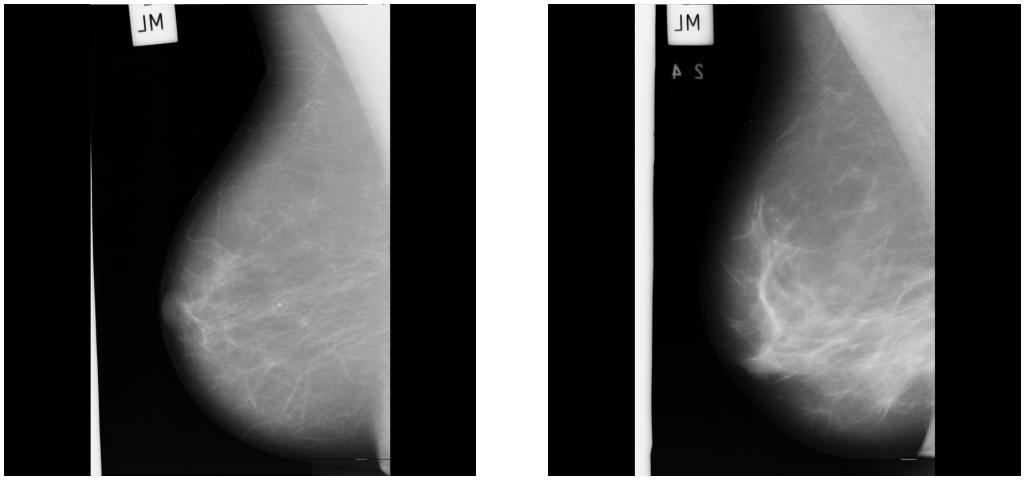

Figure 1: Input Images

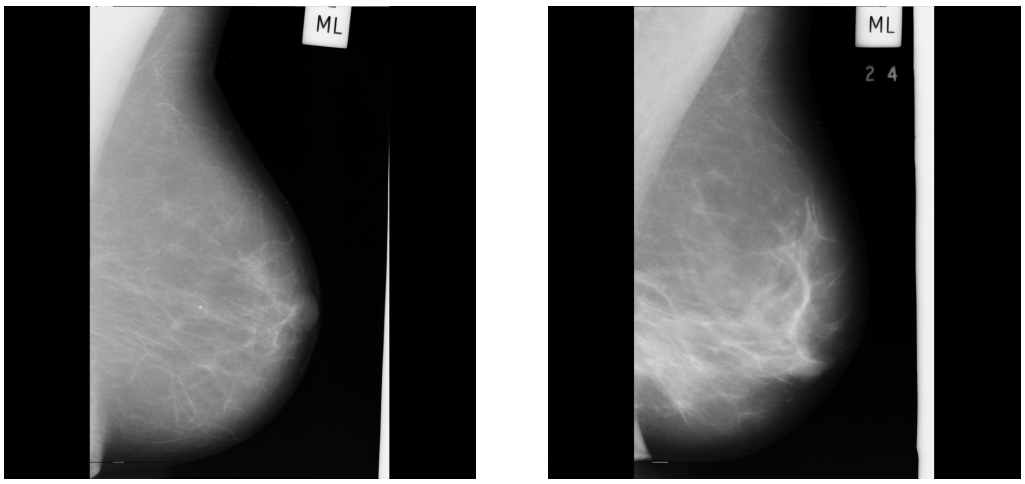

Figure 2: Flipped images

Removal of white text blocks

The text removal algorithm has several steps and makes use of a combination of spatial filtering, mathematical morphology and edge detection to identify regions to be removed and replace them with black pixels. The algorithm is implemented in the removeTexts function, which takes as input an image

represented as a two-dimensional array, passed by reference. The first step is to make a copy of the image, on which all operations will be performed. A cropping is performed of this copy, taking only the first 400 rows and columns from 400 onward. This operation is critical, as narrowing the area of artifact detection is necessary to try to avoid regions free of artifacts. The values chosen for cropping, as well as of the other algorithms used, are empirical and were chosen based on various tests performed. These values are not optimal for all 211 images, because although the artifacts are mainly present in the upper left area of the image, there may also be parts of the breast in the cropping area that are erroneously considered. The second step is to apply a Gaussian filter of size 63 to reduce noise in the cropped region. Making the difference between the cropped image and the result of filtering produces an image in which artifacts are emphasized. Next, an aperture, with elliptic kernel of size 3, is applied to that difference image to eliminate small white regions that might compromise the next steps. The next step is the application of Canny's algorithm, using a LoG kernel of size 3 and small values for thresholding. Using an elliptic kernel, the identified regions are dilated. The goal now is to create a binary mask that identifies the contours, including the inner region identified by them. To construct the inner region, the drawContours method of openCV is used, improving the result by constructing a convex envelope. At this point, the final mask is constructed by complementing the convex envelope and applying an aperture to eliminate small residual regions. The final image is obtained by making the and logical between the initial cropped region and itself, using the created binary mask. The masks created by the algorithm for the images in Figure 1 and the output of the algorithm are shown below.

Figure 3: Masks created by the text removal algorithm

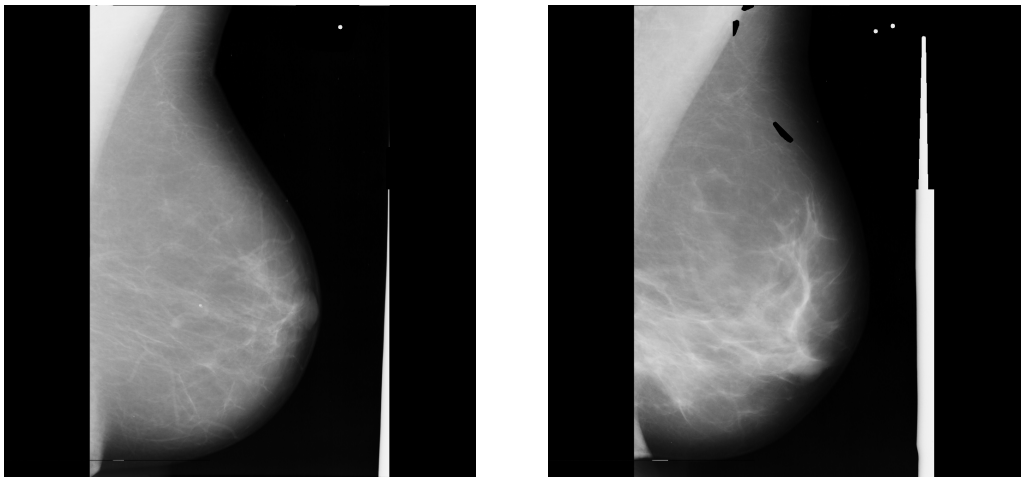

Figure 4: Output of text removal. Note the artifacts produced.

Post processing

The proposed algorithm for artifact removal was designed to be performed on all 221 images to be corrected. Therefore, all the parameters of the algorithms used within it were chosen so as to have good results on all images. However, because of this, for some images the algorithm produces small, roughly circular regions of black or white pixels, which may be due to the choice of cropping region or the thresholds set for edge detection with Canny. These issues are

resolved by a post-processing step, carried out as follows. First there is a cropping of the output image of the algorithm, taking the first 512 rows and columns from 320 onward. Next there is the application of Canny, followed by filling the contours and constructing a convex envelope, similarly to the previous algorithm. In doing so, the holes created by the previous algorithm are identified. At this point, we need to distinguish the regions in which to put 0 from those in which to replace the starting image color. 0 is set if, at the identified position, the column number is greater than 200 and in the original image that position has gray value greater than 220. These two conditions make it possible to distinguish the holes in the breast region from those in the region of the text block to be removed. The masks produced, identifying the errors made by the previous algorithm, and the output of the post-processing phase are shown below.

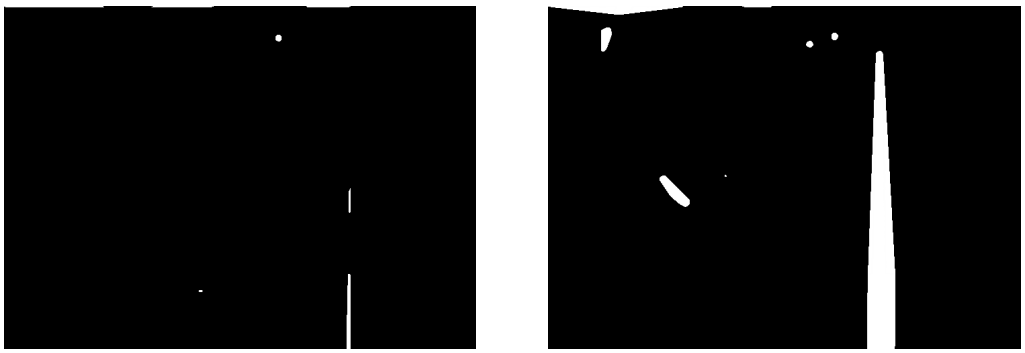

Figure 5: Masks created by the post processing algorithm

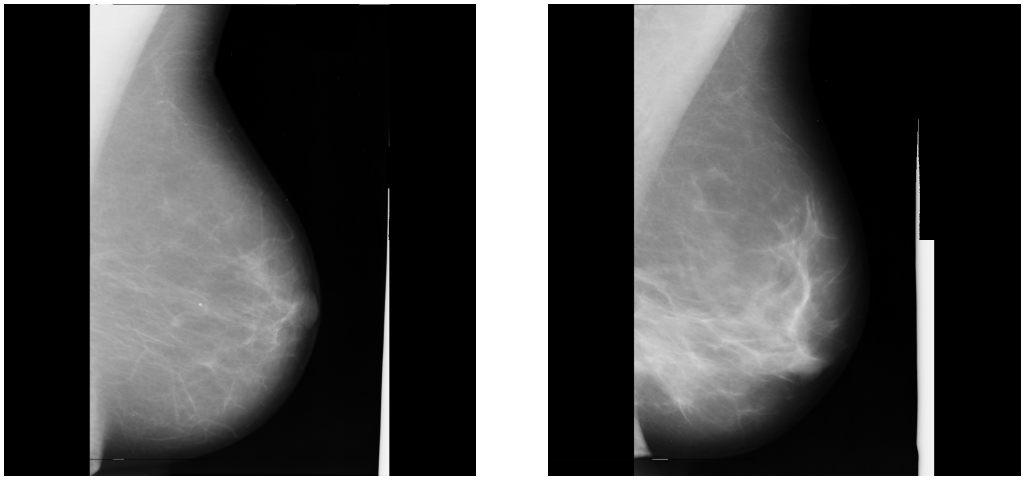

Figure 6: Image correction by post processing

Removal of large white stripes

Among the 211 images to be corrected, 49 images were identified as having large white pixel stripes. The stripes present are mostly oblique. Among these, 38 also required the removal of text blocks using the previous procedures. The small number of images allowed manual identification of regions to be corrected. The regions of interest are corrected by use of a supporting csv file, where the column number from which to set the pixel gray values to 0 is indicated.The images in which this procedure was applied are shown below, and they no longer require further correction. For the two images in Figure 1, the pipeline can be said to have come to an end.

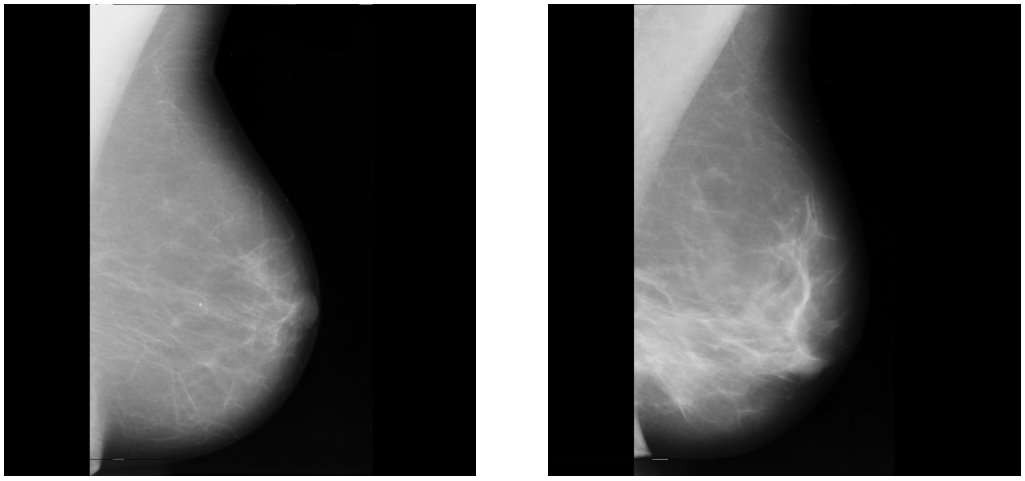

Figure 7: Removal of white stripes and subsequent final pipeline output

Removal of small white regions

To further improve the process, the following procedure of eliminating small white regions, which may be stripes or small geometric shapes, is also performed for a subset of the images. Of the part of the image where such artifacts are present, contrast enhancement is performed, followed by the application of a binary thresholding of value 170. An aperture with a small elliptical structuring element is performed on the result. The complement of the difference between the aperture and the thresholded image identifies the regions to be corrected, whose pixels take on the color black. The procedure for two images with small white regions is illustrated below, also showing the mask produced by the algorithm.

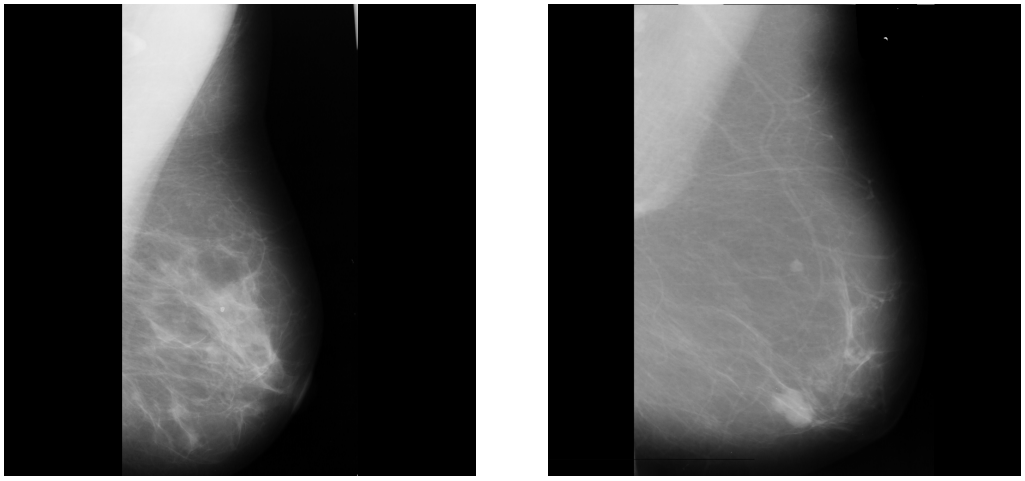

Figure 8: Input images presenting a small vertical white stripe and a small white circle, respectively

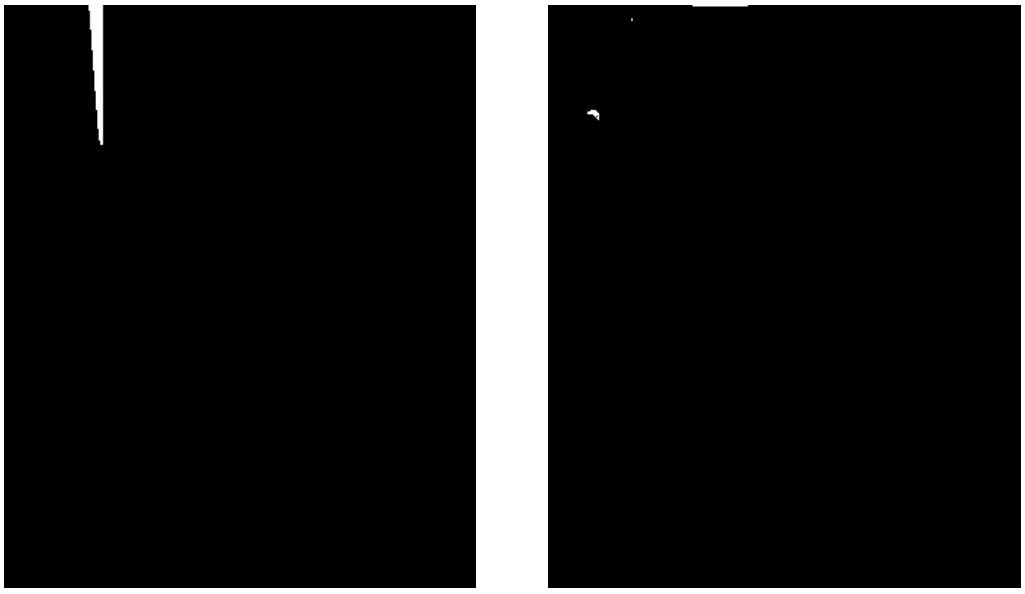

Figure 9: Masks produced by the algorithm, related to the area in which the regions are present

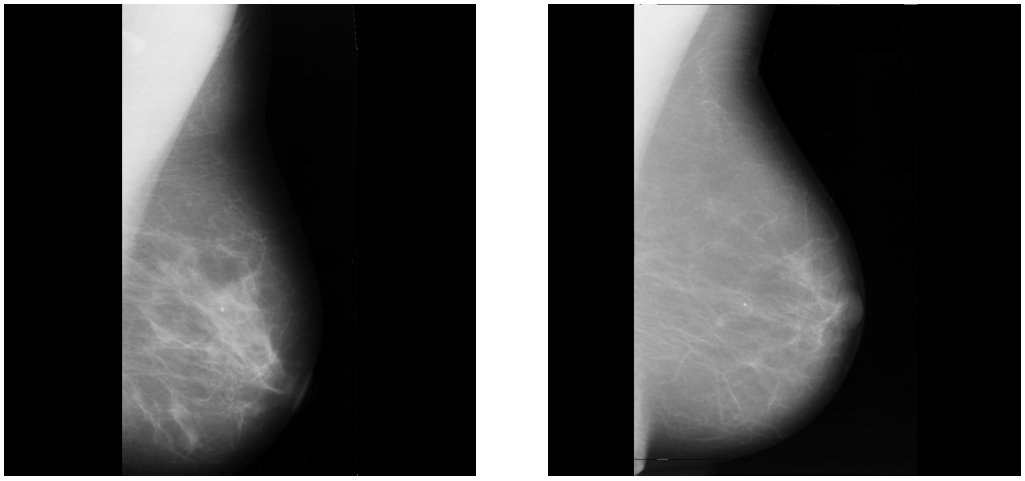

Figure 10: Final output. Regions of interest are almost completely removed.

Special cases

Three images represent special cases, where one has to operate along both axes and for which the described algorithms do not produce good results, so they are treated outside the described pipeline. The three images are illustrated below.

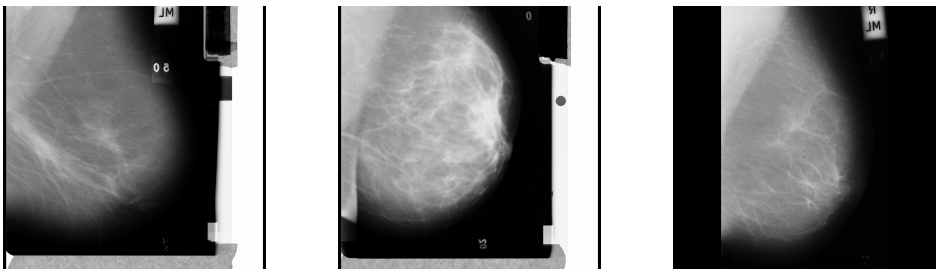

Figure 11: Special cases

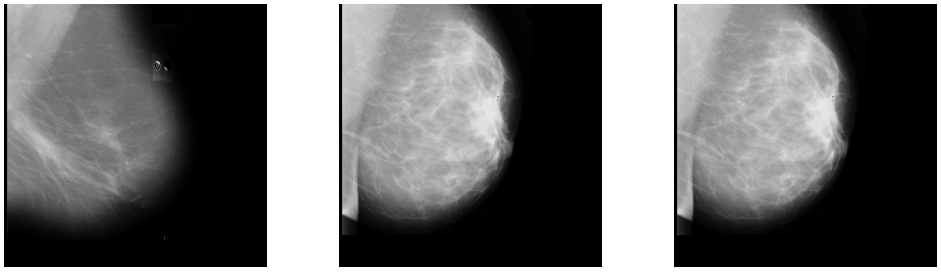

Figure 12: Output images

This last operation ends the pipeline, the final results of which are described in the next section.

Results

The described pipeline was implemented within the jupyter main.ipynb notebook, following the sections described in using the Numpy, Pandas, and OpenCV libraries. Execution of the entire pipeline took 26 seconds on a configuration equipped with 32 GB of ram, i7-10700 CPU and Windows 11 operating system, using the Visual Studio Code environment and IPython to run the notebook cells. To visualize the results of the algorithms, the images are saved in PNG format. With the described pipeline, the 211 images result in the top right-hand corner free of the blocks of text with white background and other scattered text or single letters. Small and large regions of white pixels were almost completely eliminated. Therefore, with the application of the described algorithms, the images are more compliant and ready for other types of analysis. The application of the pipeline to some of the images in the dataset is shown below. As shown below, the pipeline does not completely remove some artifacts.artefatti.

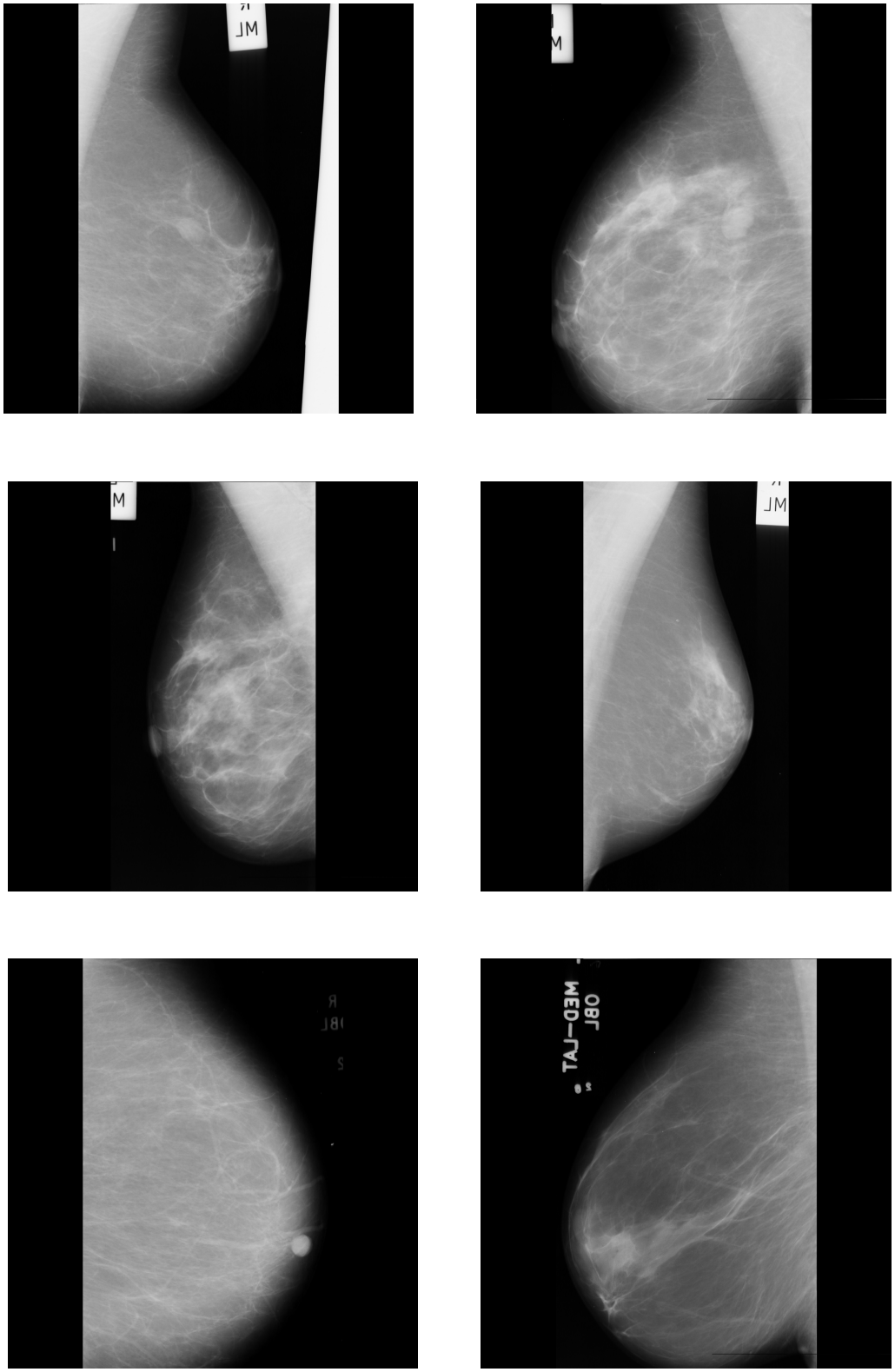

Figure 13: Input images

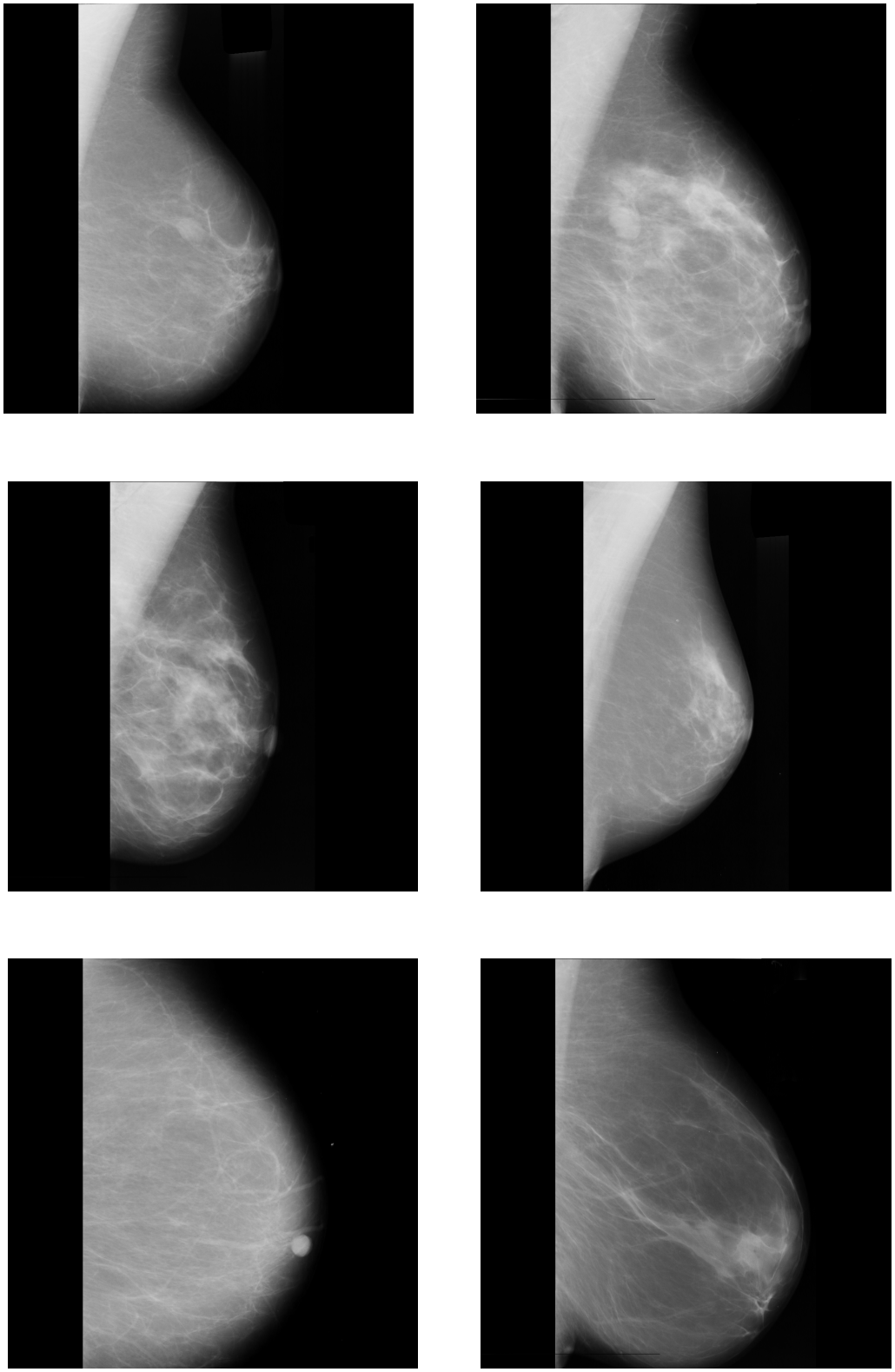

Figure 14: Output images. In the first one, the present white stripe was also removed. The removal of small white regions is negligible.

Conclusions

The algorithms described do not solve all the problems present in mammograms. There are some special cases of artifacts,including: small letters present in the lower right frame, horizontal or vertical lines in the center, uneven illumination, and impulsive noise to the right of the breast. Given the small number of these special cases, and further considering that each of them requires different image processing strategies, the best removal strategy might be to treat these special cases separately, similar to what has been done for the removal of large white stripes.