6
RESULTS

This chapter presents the results obtained by the experiments. Section 6.1 describes the dataset. Section 6.2 explains the experimental procedure followed. Section 6.3 presents visual results of each segmentation algorithm, relates them to each metric and performs a visual analysis. The Section 6.4 shows the quantitative results obtained by each metric. Finally, Section 6.5 presents Precision-Recall plots, which are used as an additional criterion of evaluation.

6.1. Dataset

The dataset for the experiments was formed by a set of three remote sensing images from different locations in Brazil. The Ground truth images were created by Musci (2013). The first image - Image1 (see Figure 15), is a snip of the Duque de Caixas' Refinery (REDUC), located in Duque de Caixas City, Rio de Janeiro, Brazil. This image was captured in January, 7th, 2012 by the WorldView-2 sensor, with a resolution of 0.5m after pan-sharpening, red, green, blue and near-infrared bands with a dimension of 2148×1099 pixels (coordinates North/East −22.715420/−43.284727 South/West −22.725583/−43.262229, UTM-WGS84 projection). Its ground truth (see Figure 15), has the classes Tanks, Buildings and Soil.

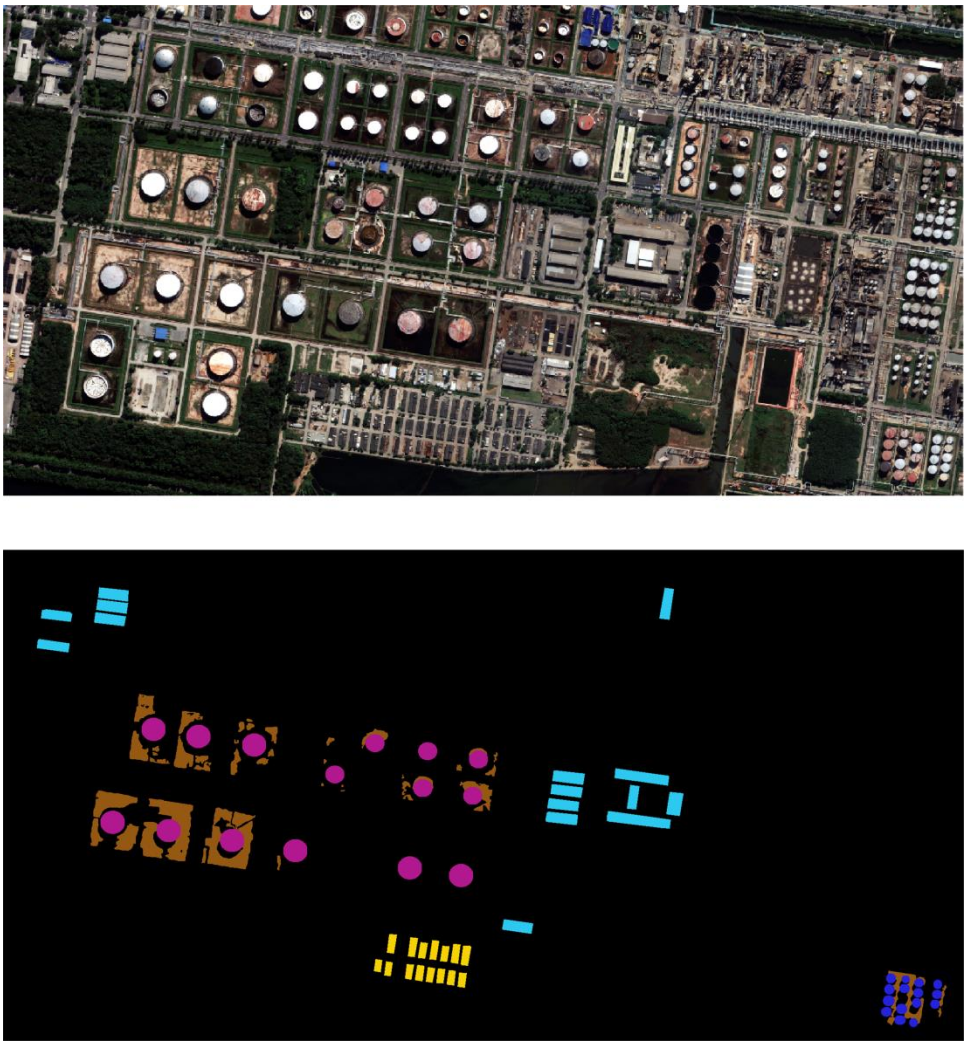

Figure 15: REDUC Image – Image 1 (top) and its ground truth (bottom).

The second image - Image2 (see Figure 16), is a snip of Maragogipe City, Bahia, Brazil. This image was captured in November, 2010 by an aero-photogrammetric lifting with a resolution of 0.6m, red, green and blue bands with a dimension of 1500×758 pixels (coordinates North/East −38.937777/−12.710833 South/West −38.935833/−12.711944, UTM-SAD69 projection). Its ground truth (see Figure 16, has the classes Boats, Roofs and Vegetation.

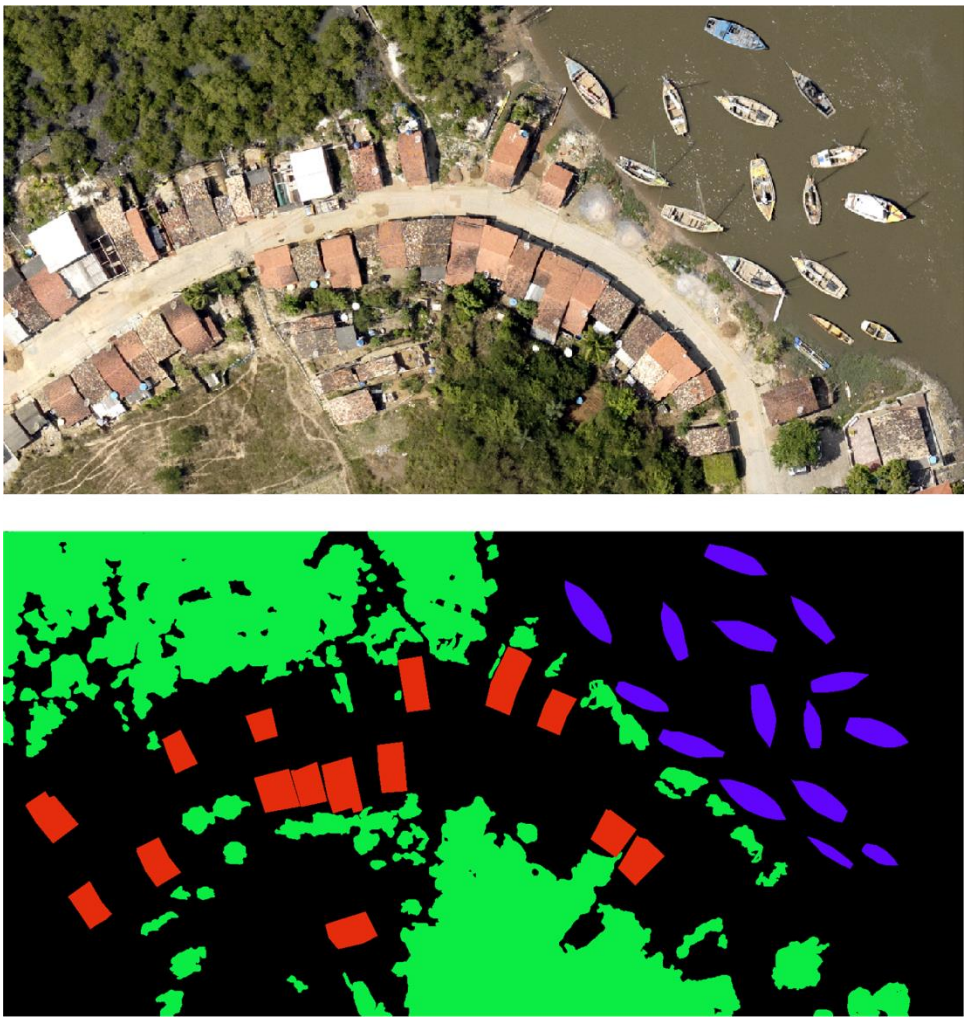

Figure 16: Image of Maragogipe City – Image 2 (top) and its ground truth (bottom).

The third image - Image3 (see Figure 17), is a snip of an area close to the Congonhas' Airport, São Paulo, Brazil. This image was captured in November 29th, 2012 by the WorldView-2 sensor with a resolution of 0.5m after pan-sharpening, red, green and blue bands with a dimension of 1829×1444 pixels (coordinates North/East −23.623850/−46.666310 South/West −23.630420/−46.657007, UTM-WGS84 projection). Its ground truth (see Figure 17), has the classes Airplanes, Roofs and Soil.

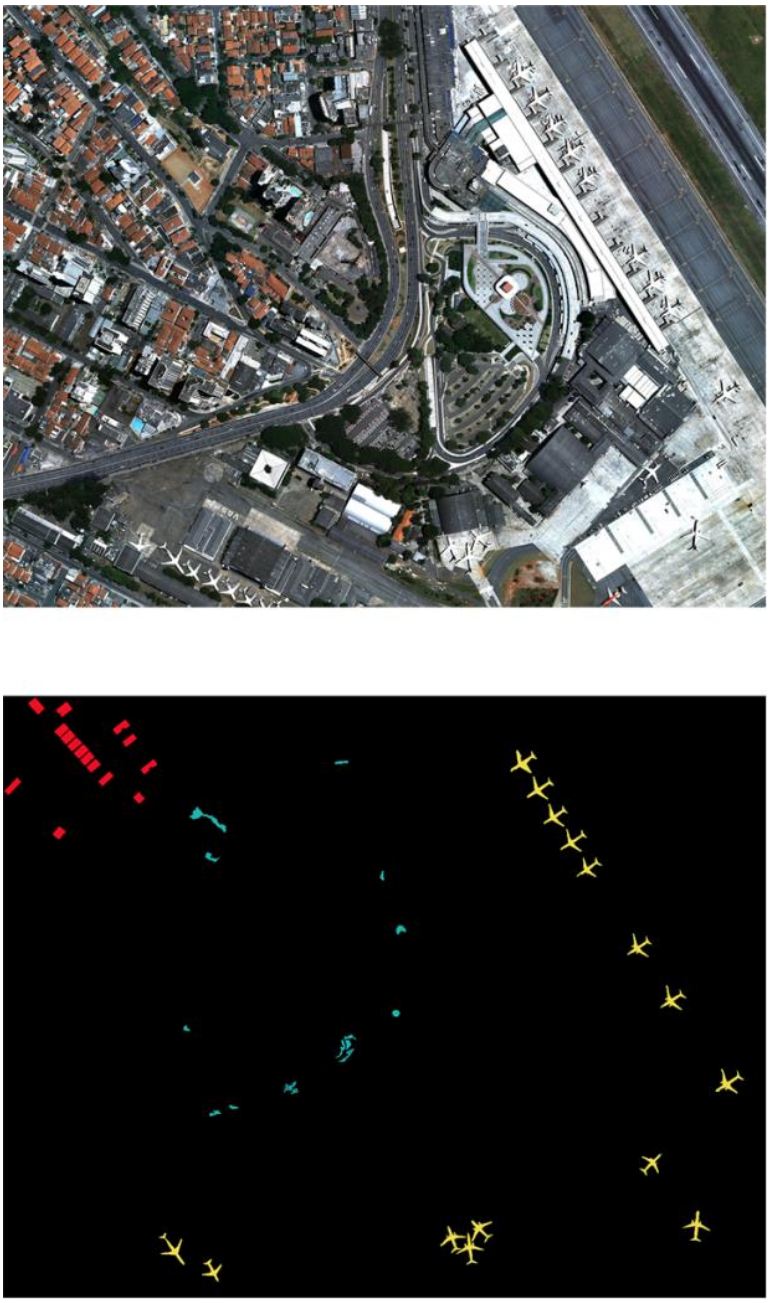

Figure 17: Image de Congonhas – Image 3 (top) and its ground truth (bottom).

6.2. Experimental Procedure

A prototype was developed to perform all the necessary experiments for this work. It was done using Qt Creator with MinGW as compiler and C++ programming language in a computer Intel Core i7-3.20 GHz with 32.0 GB of RAM memory. As it was stated before, four segmentation algorithms were tested and seven different metrics were used for evaluation. It makes a total of 28 experiments per image. As the dataset consisted in three remote sensing images,

the total number of experiments performed was 84. As Section 2.3 explained, each experiment begins with the execution of one segmentation algorithm with an image from the dataset and standard initial parameters as inputs, which gives a starting search point for the optimization algorithm. This result is evaluated according to the selected metric. If the numeric value provided by the metric is the lowest; then, an optimal segmentation have been reached. Otherwise, the optimization algorithm provides another set of parameters and the segmentation procedure is executed again. This iterative process is performed until the segmentation outcome fits the given reference. The following paragraphs describe the implementation done or taken for each segmentation algorithm as well as the difficulties related to each one of them. The times related to each experiment include the whole optimization process.

The reference images, ground truth (GT), have delineated segments corresponding to different classes such us tanks, soil, roofs for Image 1, vegetation, roofs and boats for Image 2 and airplanes, soil and roofs for Image 3. All references were used for the performed experiments.

Mean-Shift (MS) segmentation implementation was done using OpenCV libraries. The inputs for these algorithms were the image to be segmented, the name of the segmentation outcome, the spectral and spatial radius and the number of pyramid levels (cf. Section 3.1). An average experiment with MS segmentation took about 18 hours per metric.

Graph-based (Gb) segmentation implementation was taken from the author's website. The source code was available, which allow us to modify and have a better understanding of the implementation of this algorithm. An average experiment with Gb segmentation took about 15 hours per metric.

Region Merging-based (Rm) segmentation implementation was taken from the website of the Computer Vision Lab (LVC) from the Pontifical Catholic University of Rio de Janeiro (PUC-Rio). Sequential and Parallel version are included in the available package. Only the sequential version was considered for the experiments. An average experiment with Rm segmentation took about 11 hours per metric.

Conditional Random Fields-based ($CRFb$) segmentation implementation was taken from the author's website. It was developed to work with many images for training. Thus, in order to work with one image, the input image and its

ground truth were partitioned and small portions of them were used as inputs for the training algorithm. As this algorithm has an inner optimization process, it was not necessary to execute it many times for parameter tuning. Only in this case, the selected metrics were used for segmentation evaluation. An average experiment with $CRFb$ segmentation took about 8 hours.

It is important to take into consideration that in Image 3, the ground truth did not provide enough samples for training the $CRFb$ segmentation algorithm. Additionally, this ground truth does not have samples for all classes present in the image. Consequently, only in the case of Image 3, it is not recommendable to compare the $CRFb$ segmentation with the other algorithms.

As the range of Nelder – Mead (NM) algorithm is from $-\infty$ to $+\infty$, it provides values in the same range. It represented a problem because the set of parameters for a segmentation algorithm are restricted to a narrow range. Therefore, a sigmoidal function was used to limit and scale the values produced by the NM algorithm. The implementation used for the experiments was provided by Ayma (2013).

The seven quality metrics were implemented. Each metric only has two inputs, which were the segmentation outcome and the ground truth.

6.3. Visual Quality Assessment

In order to perform a visual analysis of the results of the experiments, some specific regions were taken from Images 1, 2 and 3. Notice that it was done only for visualization purpose due to the high dimension of input images. From Image 1, three regions were taken for comparison, and from Images 2 and 3, only two regions were taken.

Figure 18, Figure 19 and Figure 20 show the results obtained in the experiments for three different regions from Image 1. For the first region showed in Figure 18, it can be said that Mean-Shift (MS) and Graph-based (Gb) segmentations provided a good object detection (tanks). On the other hand, Region Merging-based (Rm) segmentation provided over-segmented results. The CRF-based ($CRFb$) segmentation showed an acceptable performance in the

detection of tanks; however, other classes were not properly detected such as soil, trees, etc.

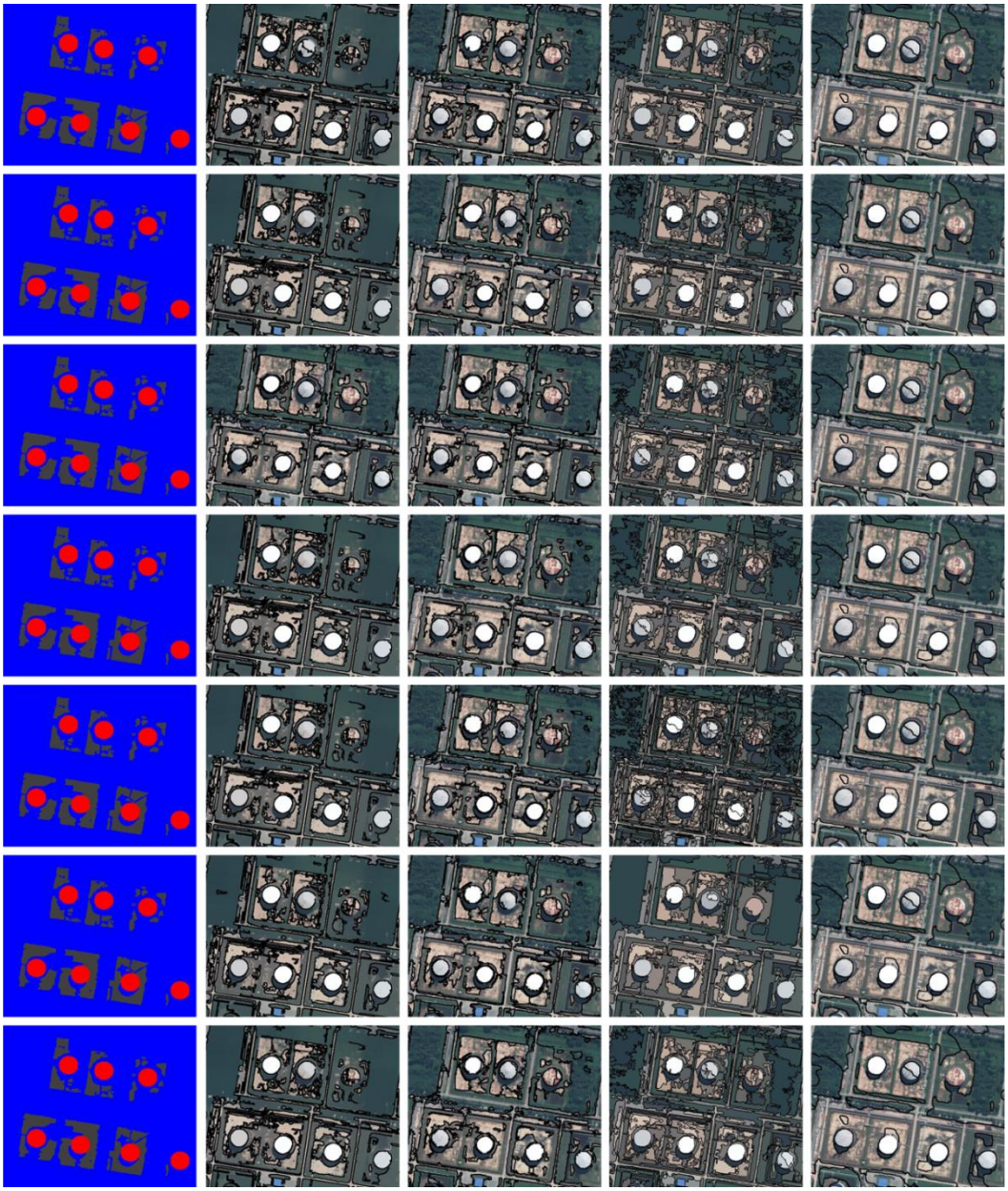

Figure 18: Results of the experiments with Image 1, each row from top to bottom represents a metric: H, AFI, SI, RI, F, $\mathcal{C}$ and $RBSB$. From left to right, each column represents: GT, MS result, Gb result, Rm result and $CRFb$ result.

For the second region (see Figure 19), not over-segmented areas are observed in Rm segmentation outcome; what's more, Rm and MS segmentations showed a good performance for roofs detection. On the other hand, Gb and $CRFb$ segmentation outcomes were not able to provide good results. The first one only

showed a good result with Hoover metric. The second one showed a poor performance due to the small amount of data for training from the ground truth.

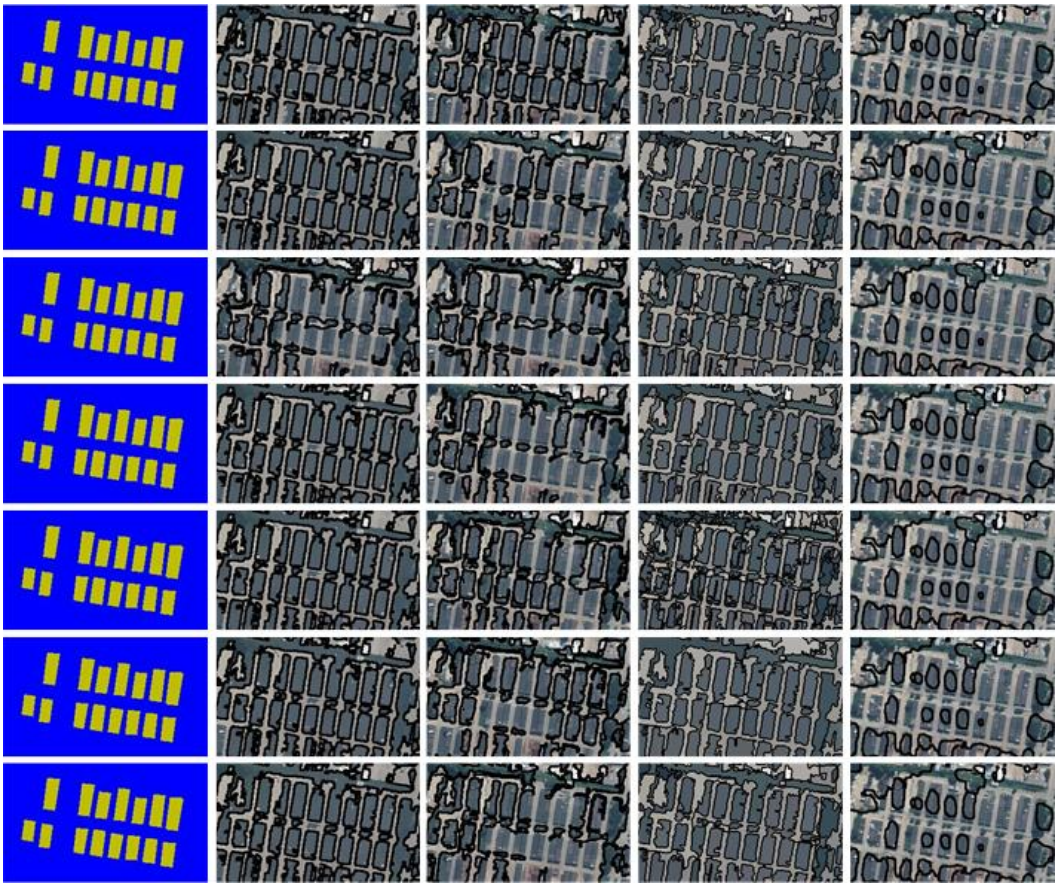

Figure 19: Results of the experiments with Image 1, each row from top to bottom represents a metric: H, AFI, SI, RI, F, $\mathcal{C}$ and $RBSB$. From left to right, each column represents: GT, MS result, Gb result, Rm result and $CRFb$ result.

For the third region, presented in Figure 20, Rm segmentation provided some results with under-segmentation for tanks detection. However, detections of other classes were fine (grass, soil, etc.). Gb segmentation missed some buildings, which led to under-segmented results. $CRFb$ segmentation had a similar performance. Finally, MS segmentation showed the best results for building detection.

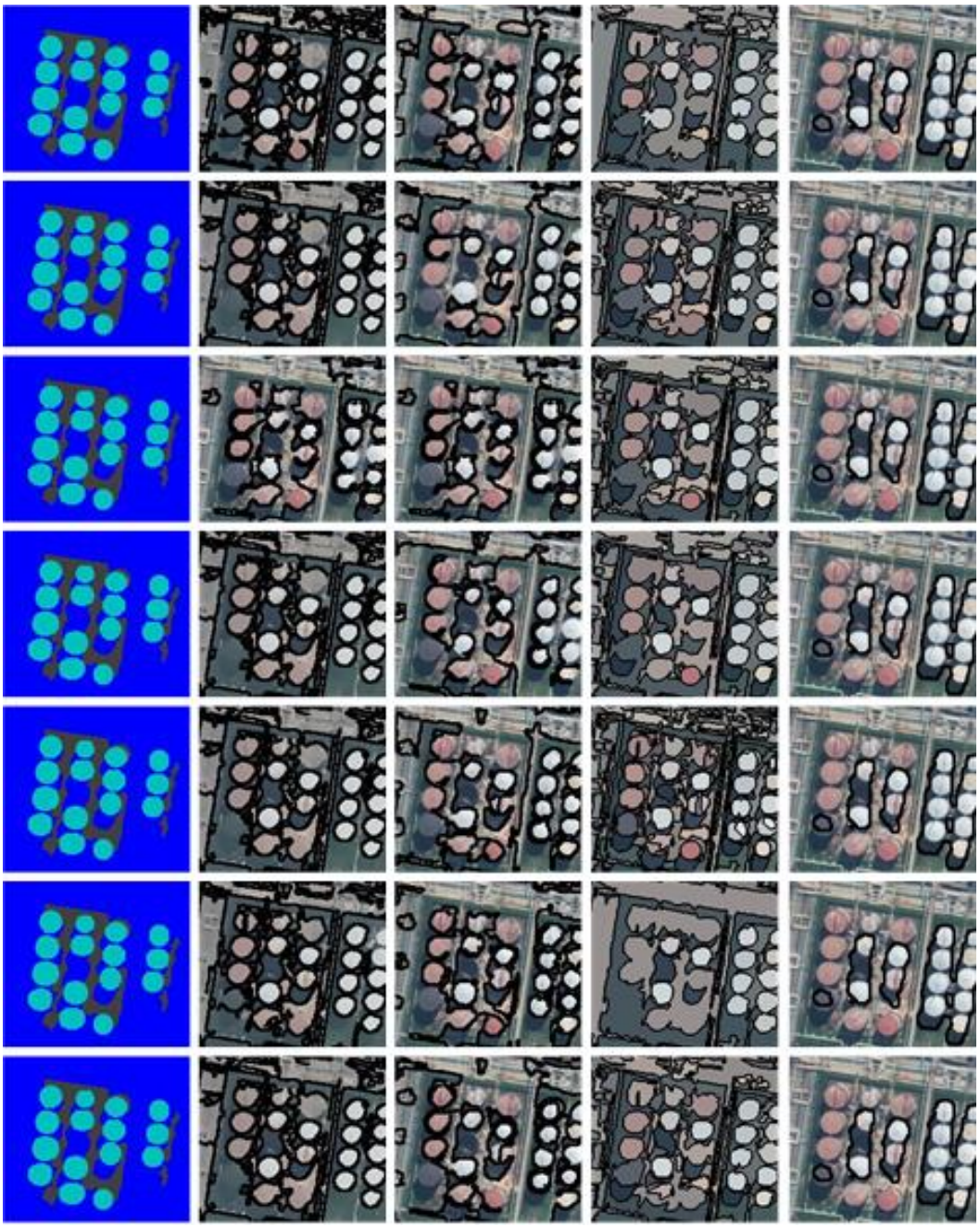

Figure 20: Results of the experiments with Image 1, each row from top to bottom represents a metric: H, AFI, SI, RI, F, C and $RBSB$. From left to right, each column represents: GT, MS result, Gb result, Rm result and $CRFb$ result.

Figure 21 and Figure 22 show the results obtained in the experiments for two different regions from Image 2. For the first region, showed in Figure 21, it can be said that Gb segmentation delivered the best performance. It successfully detected roofs, trees and soil in the given image. Moreover, Rm segmentation had a good performance for all metrics, except for a few cases of over-segmentation in vegetation areas. MS segmentation missed many classes such as roofs and soil

delivering some under-segmented results. $CRFb$ segmentation was good for roofs and vegetation detection. However, it yielded under-segmented areas for soil.

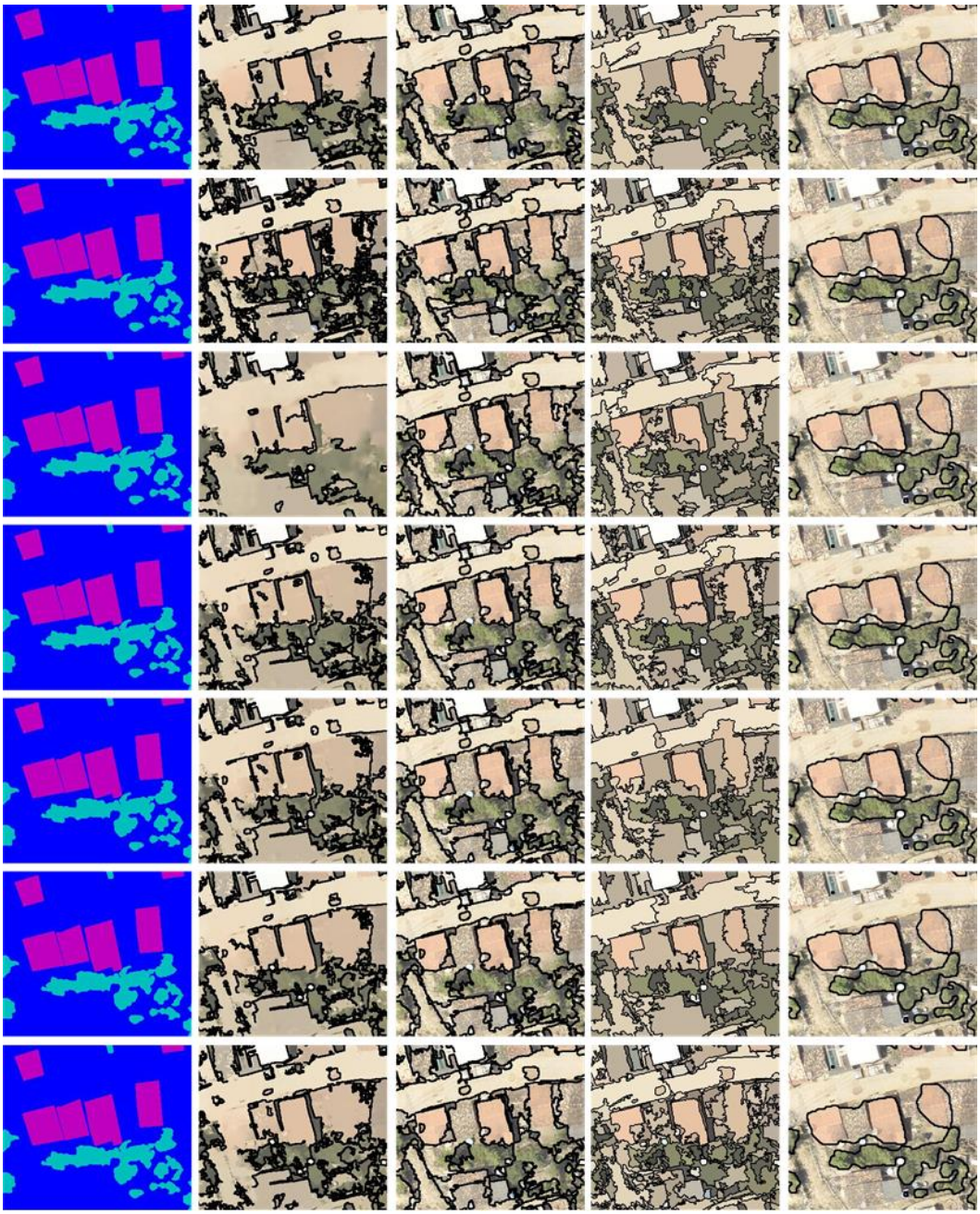

Figure 21: Results of the experiments with Image 2, each row from top to bottom represents a metric: H, AFI, SI, RI, F, $\mathcal{C}$ and $RBSB$. From left to right, each column represents:GT, MS result, Gb result, Rm result and $CRFb$ result.

In the second region, showed in Figure 22, $CRFb$ segmentation had the best performance. It successfully detected boats and river. MS and Gb segmentation

had good results too, except for a few cases of over-segmentation inside the boats. Rm segmentation provided over-segmented results outside the boats.

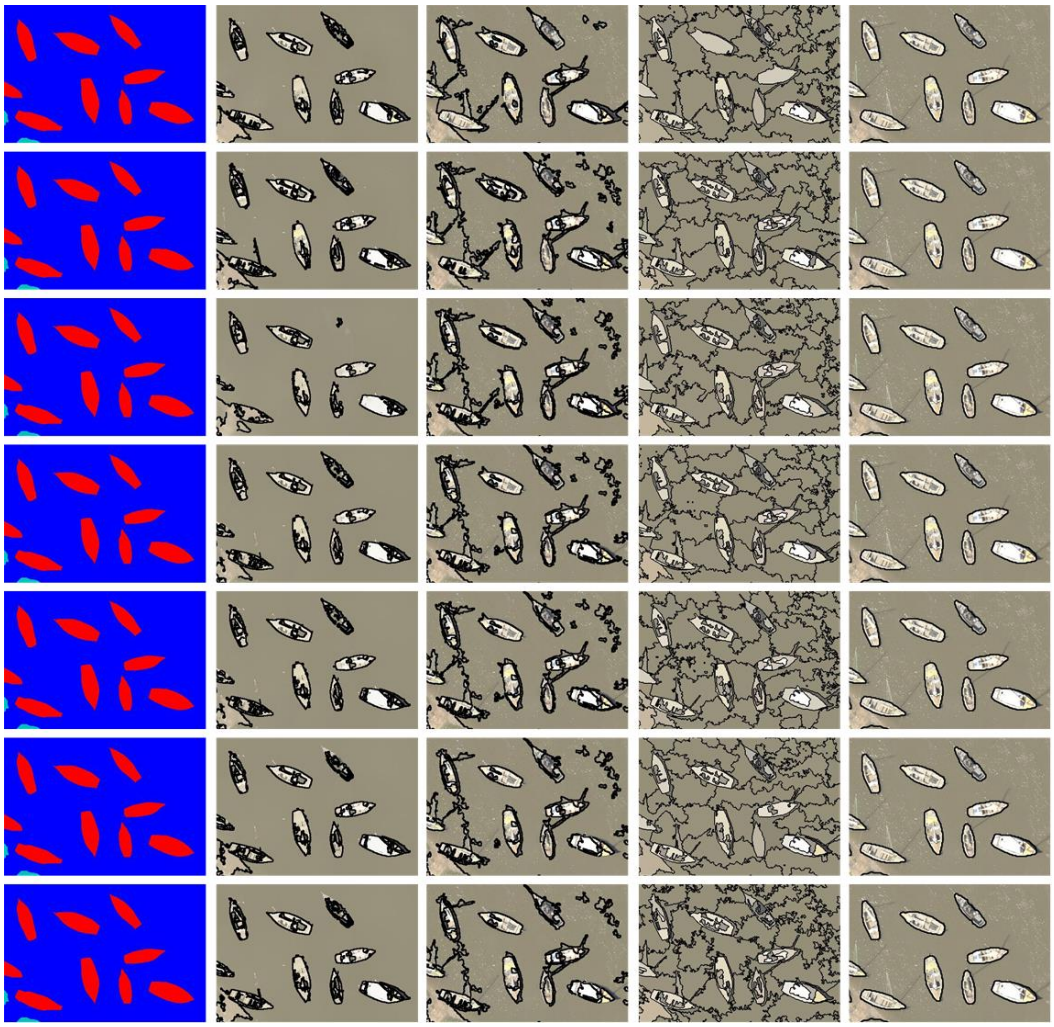

Figure 22: Results of the experiments with Image 2, each row from top to bottom represents a metric: H, AFI, SI, RI, F, $\mathcal{C}$ and $RBSB$. From left to right, each column represents:GT, MS result, Gb result, Rm result and $CRFb$ result.

Figure 23 and Figure 24 show the results obtained for two different regions of Image 3.

For the first region, presented in Figure 23, the results of MS, Gb and Rm segmentations can be regarded as good for roofs, vegetation and road. Rm presented a few cases of over-segmentation but can still be regarded as acceptable. On the other hand, $CRFb$ segmentation presented a poor performance due to the low quantity of data for training.

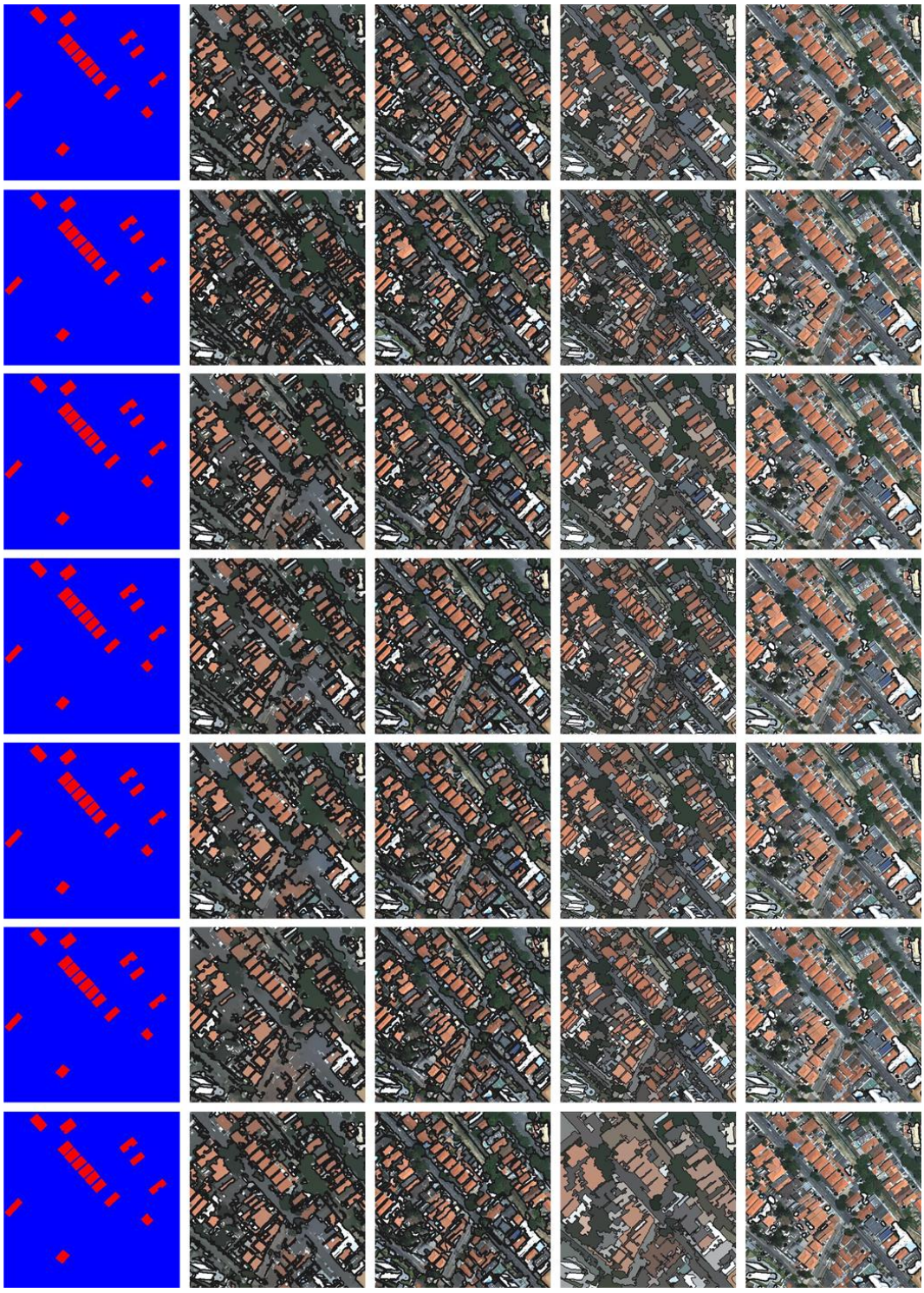

Figure 23: Results of the experiments with Image 3, each row from top to bottom represents a metric: H, AFI, SI, RI, F, $\mathcal{C}$ and $RBSB$. From left to right, each column represents:GT, MS result, Gb result, Rm result and $CRFb$ result.

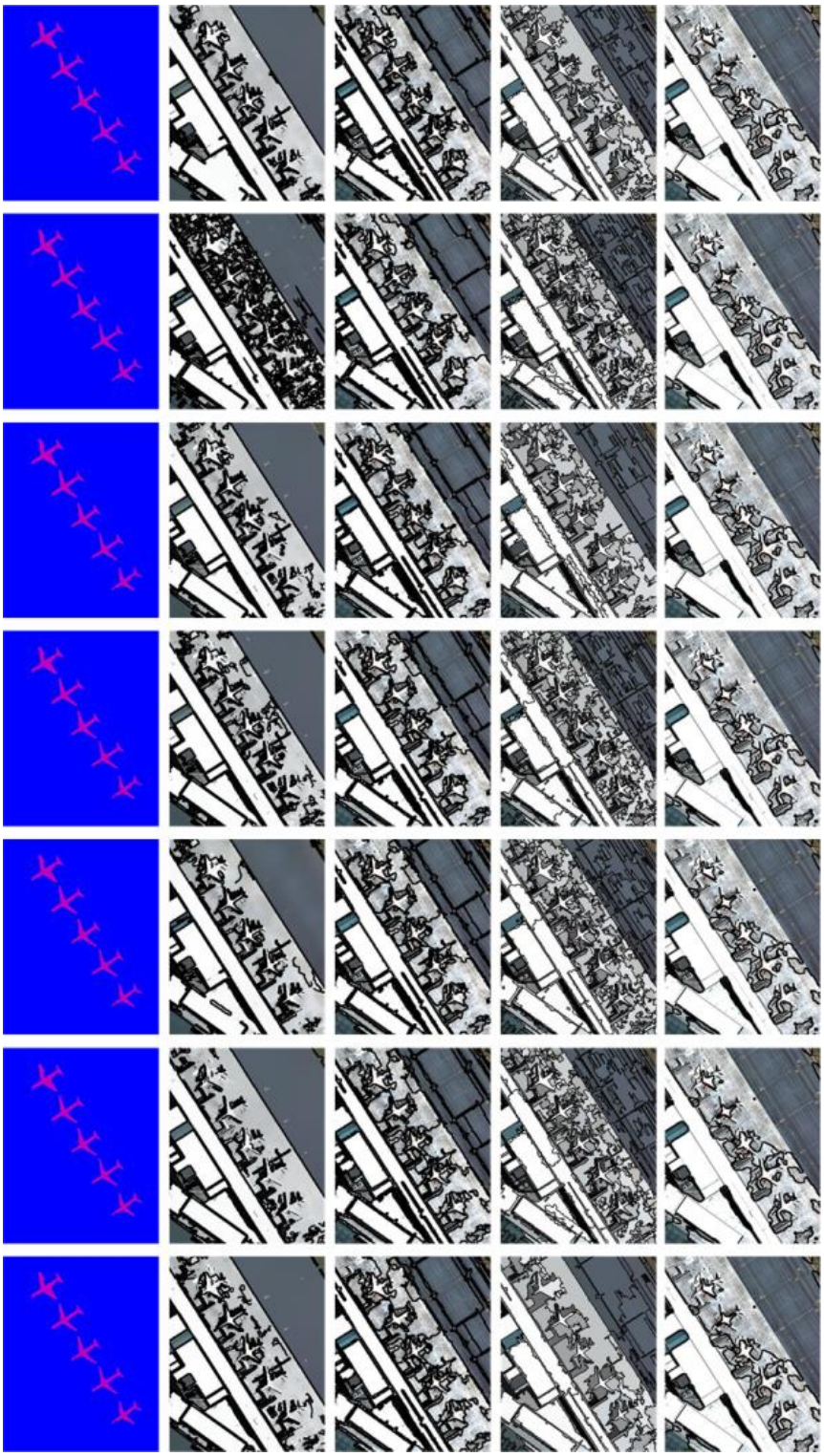

Figure 24: Results of the experiments with Image 3, each row from top to bottom represents a metric: H, AFI, SI, RI, F, $\mathcal{C}$ and $RBSB$. From left to right, each column represents:GT, MS result, Gb result, Rm result and $CRFb$ result.

In the second region, showed in Figure 24, Rm provided the best performance in airplanes detection; however, there were over-segmented areas for

ground and road. Gb segmentation presented a behavior similar to Rm segmentation but without many over-segmented areas. MS segmentation was not able to detect airplanes. However, it successfully detected the other classes. Finally, $CRFb$ segmentation, as stated before, showed a poor performance due to the absence of enough data for training.

6.4. Comparison based on the selected metrics

In addition to the visual assessment, all algorithms were quantitatively evaluated by the selected metrics. Their parameters were calculated following the approach proposed in Section 2.3 (see Figure 4). These results are shown in Table 2, Table 3 and Table 4. The minimum value for each metric is highlighted in red.

Before going into the analysis itself, some aspects of each aforementioned metric are worth being recalled.

The Hoover Index (H), as defined in Section 5.1, measures the number of correct detections presented in a segmentation. It does not give any information at pixel level; it is practical but subjective due to the absence of a more extensive criterion of evaluation.

In another way, an algorithm with the lowest Area-Fit-Index (AFI) value will have the lowest quantity of pixels not-considered as part of the final result. Furthermore, the Shape Index (SI) compares geometrical features between regions. This morphological approach looks for a shape consistency between compared regions.

The Rand Index (RI) measures the ratio between the group of pixels that were correctly classified and non-classified as part of the segmentation, and the total number of groups of pixels. Similarly, the F measure quantifies a trade-off between Precision (P) and Recall (R) (cf. Section 5.5). Notice that the lowest value of F would not lead to the best segmentation result. It is necessary to contrast it with the Precision-Recall plots in order to make a good decision.

Segmentation Covering ($\mathcal{C}$) measures the number of pixels in the intersection of two regions. Low values of $\mathcal{C}$ mean a low number of pixels outside the intersection, which implies a good overlapping between them. Reference

Bounded Segments Booster ($RBSB$) is similar to Area-Fit-Index (AFI) and takes additionally into account the number of incorrectly detected pixels.

Now, let's analyze the results obtained by each metric. From Table 3 and Table 4, it could be inferred that Graph-based (Gb) segmentation gave the best results among the selected segmentation algorithms. Based on the metrics, Gb provided results close the lowest (best) values in almost all cases. However, it was not able to reach good values with the H index.

$CRFb$ segmentation reached good values only in the images where it had enough data for training. These are the cases of Image 1 and 2, where there were plenty of samples for parameter training of the CRF model. Its results were not the best ones but were close to them. Its results with Image 3 won't enter in this comparison due to the absence of enough data for a correct training of the algorithm.

The results obtained in Table 2 were not regular and a little confusing because there is not a segmentation algorithm that provided the lowest values for each metric. It is necessary to contrast these results with the visual results obtained and Precision-Recall plots. For that reason, this table will be analyzed in the following section.

		SEGMENTATION ALGORITHMS			
		MS	***Gb***	***Rm***	***CRFb***
METRIC	***H***	0,705128	0,782051	0,833333	0,970000
	AFI	0,073543	0,069082	0,062827	0,071068
	SI	0,031427	0,015005	0,010553	0,046364
	RI	0,012183	0,011596	0,021670	0,010891
	F	0,006141	0,006243	0,010200	0,005499
	C	0,110033	0,099456	0,160878	0,117899
	RBSB	0,142438	0,133450	0,193639	0,132962

Table 2: Results obtained for each metric by the optimization algorithm for Image 1.

		SEGMENTATION ALGORITHMS			
		MS	Gb	Rm	$CRFb$
METRIC	H	0,886364	0,909091	0,931818	0,769912
	AFI	0,018609	0,002393	0,018172	0,056273
	SI	0,057650	0,012467	0,044314	0,044649
	RI	0,007922	0,006302	0,007544	0,015313
	F	0,003979	0,003161	0,004671	0,007749
	$\mathcal{C}$	0,103128	0,098383	0,115261	0,141025
	$RBSB$	0,109581	0,098120	0,116189	0,164505

Table 3: Results obtained for each metric by the optimization algorithm for Image 2.

		SEGMENTATION ALGORITHMS			
		MS	Gb	Rm	$CRFb$
METRIC	H	0,909091	0,931818	0,840909	1,000000
	AFI	0,006599	0,006234	0,007181	0,264463
	SI	0,120127	0,004628	0,120818	0,300093
	RI	0,013512	0,001708	0,015074	0,121413
	F	0,005210	0,000855	0,005183	0,064630
	$\mathcal{C}$	0,131292	0,057996	0,121212	0,432836
	$RBSB$	0,135428	0,059246	0,111492	0,479339

Table 4: Results obtained for each metric by the optimization algorithm for Image 3.

6.5. Precision and Recall plots

Figure 25, Figure 26 and Figure 27 show the obtained values of Precision and Recall. Bearing in mind those considerations introduced in Section 5.6, let's look at the plots. Figure 25 shows the Precision-Recall values for the Image 1. Under this point of view, Graph-based segmentation (Gb) provided the best results among all algorithms due to the proximity of its value to the good corner. On the other hand, as the over-segmentation corner is preferable than the opposite corner, Mean-Shift (MS) and CRF-based ($CRFb$) segmentations should be rated as the second and third best algorithms. Finally, Region Merging-based (Rm) segmentation presents a higher quantity of false positives (fp) than the others, making it the poorest one.

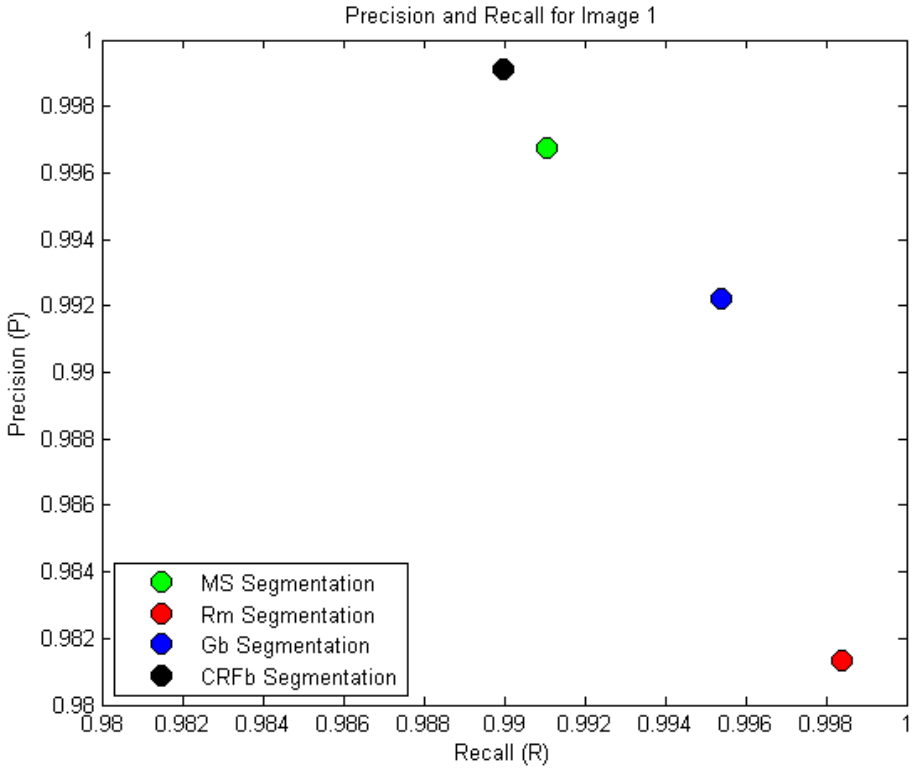

Figure 25: Precision-Recall plot for Image 1. Each point represents an iteration of the optimization algorithm. Each color represents a segmentation algorithm (see legends).

Figure 26 shows the Precision-Recall values for the Image 2. In this figure, the results lie very close to each other. Nevertheless, Gb segmentation is still the best option. Then, MS and Rm segmentations are preferred rather than $CRFb$ due to its proximity to the good corner. Notice that the Rm algorithm tends to the under-segmentation corner as mentioned before.

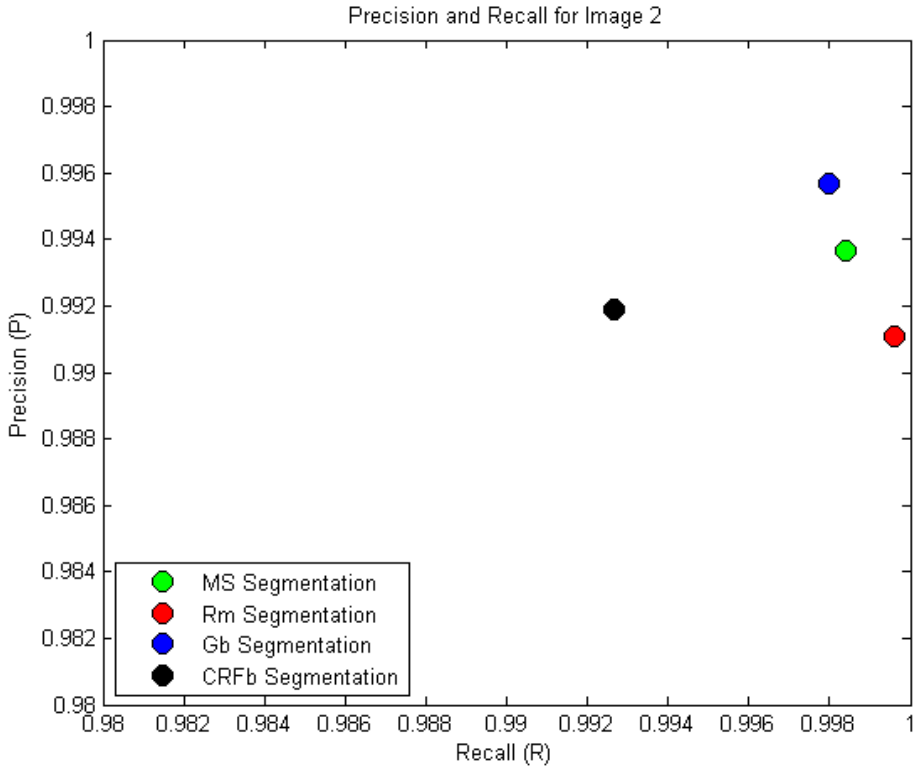

Figure 26: Precision-Recall plot for Image 2. Each point represents an iteration of the optimization algorithm. Each color represents a segmentation algorithm (see legends).

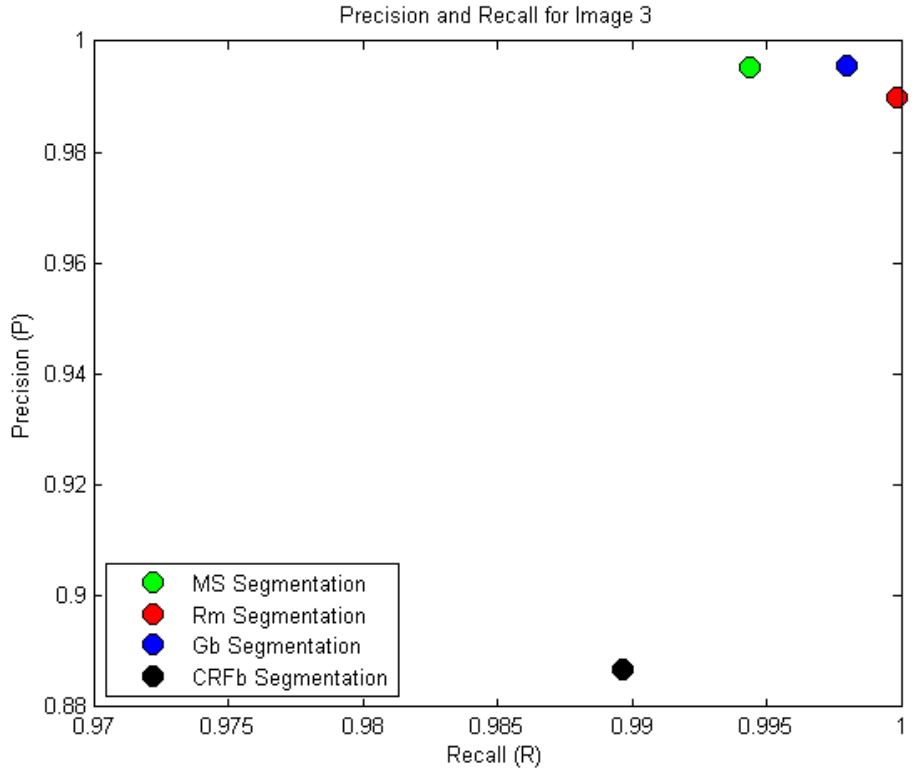

Figure 27: Precision-Recall plot for Image 3. Each point represents an iteration of the optimization algorithm. Each color represents a segmentation algorithm (see legends).

Figure 27 shows the Precision-Recall values for the Image 3. In this particular case, notice the low values of Precision and Recall for the $CRFb$ segmentation. It was a direct consequence of the absence of enough samples in the ground truth for training. Gb segmentation is still the best option. Rm and MS segmentations are almost at the same distance of the good corner. However, as the top-left corner is preferable to the bottom-right, MS segmentation will be preferred rather than the Rm segmentation.

Gb segmentation presented a more regular performance with Image 1 (see Table 2), it provided the second lowest values for each metric. As for the F measure there are available more information with the values of Precision (P) and Recall (R), let's contrast the results in Table 2 with the plot in Figure 25. $CRFb$ segmentation had the lowest value of the measure F in Table 2. However, according to Figure 25, it is in the over-segmentation area. Thus, it would not be preferred as the first option. On the other hand, Gb segmentation has a better position in the plot, so it would be preferred rather than the others.

		NUMBER OF ITERATIONS									**Mean per metric**
		Image 1			**Image 2**			**Image 3**			
		MS	***Gb***	***Rm***	***MS***	***Gb***	***Rm***	***MS***	***Gb***	***Rm***	
METRIC	***H***	108	61	175	119	67	88	77	53	89	93,0
	AFI	42	61	118	50	51	76	54	57	105	68,2
	SI	49	63	83	41	39	115	53	33	86	62,4
	RI	58	57	114	33	55	64	40	31	108	62,2
	F	58	83	98	33	55	120	108	31	100	76,2
	C	37	83	145	40	77	104	39	39	109	74,8
	RBSB	35	63	88	53	75	79	96	39	98	69,6
	Mean per Seg. Alg.	55,3	67,3	117,3	52,7	59,9	92,3	66,7	40,4	99,3	

Table 5: Number of iterations needed by the optimization algorithm to find the best parameters for each segmentation algorithm.

6.6. Optimization algorithm

The Nelder-Mead algorithm provided the different configuration of parameters for each segmentation algorithm. The average number of iterations needed for each experiment is presented in Table 5.

Rm and MS segmentation had the same number of parameters to be tuned. However, Rm needed more iterations than MS. It was mainly because the range of possible values of Rm segmentation was wider than the ones for MS. It was expected that Gb needed less iterations than the others due to the less number of parameters to be tuned. Moreover, the Hoover metric (H), was the metric that needed more iterations.